

Internet Research Task Force
Internet-Draft
Intended status: Informational
Expires: October 28, 2017

H. Baba
The University of Tokyo
Y. Ishida
Japan Network Enabler Corporation
T. Amatsu
Tokyo Electric Power Company, Inc.
K. Kunitake
BroadBand Tower, Inc.
K. Maeda
Individual Contributor
April 26, 2017

Problems in and among industries for the prompt realization of IoT and
safety considerations
draft-baba-iot-problems-03

Abstract

This document contains opinions gathered from enterprises engaging in the IoT business as stated in the preceding version hereof, and also examines the possibilities of new social problems in the IoT era. Recognition of the importance of information security has grown in step with the rising use of the Internet. Closer examination reveals that the IoT era may see a new direct physical threat to users. For instance, the situation at a smart house may lead it to judge that the owner has only temporarily stepped out, causing it to unlock the front door, which in turn makes it easier for thieves to enter. These kinds of scenarios may occur without identity fraud, hacking, and other means of compromising information security. Therefore, for the purpose of this document, this issue shall be referred to as "IoT Safety" to distinguish it from Information Security.

We believe that it is necessary to deepen our understanding of these new IoT-related threats through discussion and ensure there are measures to address these threats in the future. At the same time, we must also coordinate these measures with the solutions to the problems described in the previous version of this document.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on October 28, 2017.

Copyright Notice

Copyright (c) 2017 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	3
2. Technical Challenges	4
2.1. Safety, Security and Privacy	4
2.1.1. Challenges in protecting lives and property from IoT-related threats (IoT Safety)	4
2.1.1.1. Safety of body/life	5
2.1.1.2. Safety of equipment	5
2.1.1.3. Proper performance of equipment	5
2.1.2. Information Security	5
2.1.3. Privacy in acquiring data	6
2.2. Challenges posed by data acquisition, data distribution, data management and data quantity	7
2.2.1. Traffic patterns	7
2.2.2. Acquired mass data	7
2.2.3. Explosive increase and diversity of data	7
2.3. Mapping of the physical world and the virtual world	8
2.3.1. Physically handling acquired data	8
2.3.2. Data calibration	8
2.4. Product lifetime, generation management, and the cost of equipment updates	8
2.4.1. Product lifetime	8
2.4.2. Introducing IoT equipment into commodity equipment	9
2.5. Too many related standards and the speed of standardization	9

2.5.1.	Too many related standards	9
2.5.2.	Speed of standardization	10
2.6.	Interoperability, fault isolation, and total quality assurance	10
2.6.1.	Interoperability	10
2.6.2.	Fault isolation	10
2.6.3.	Quality assurance	11
2.7.	Product design policy	11
2.7.1.	Changes in design policy	11
2.8.	Various technology restrictions within actual usage	11
2.8.1.	Using radio waves	11
2.8.2.	Batteries	12
2.8.3.	Wiring	12
2.8.4.	Being open	12
3.	Non-technical Challenges	13
3.1.	Changing the product paradigm	13
3.1.1.	Ecosystems	13
3.1.2.	Coordination and significant changes in strategy	13
3.1.3.	Competition with existing industries	13
3.2.	Benefits	13
3.2.1.	Rising costs and monetization	13
3.3.	Information security and privacy of social systems	14
3.3.1.	Classification of ownership, location, and the usage of data	14
3.4.	Disclosure of data	14
3.4.1.	Side effects and malicious use potentially caused by the disclosure of data	14
3.5.	Preparing social support	14
3.5.1.	Regulations	14
3.5.2.	Corporate social responsibility	14
3.5.3.	Customization for individual customers	15
3.5.4.	IoT literacy of the users	15
3.5.5.	Individual vs. family	15
4.	Information Security Considerations	15
5.	Privacy Considerations	15
6.	Acknowledgments	16
	Authors' Addresses	16

1. Introduction

Many activities are progressing in various fields, such as the proposal of standards for creating an IoT world. There are also many reports that analyze and predict the benefits that IoT can bring to the economy and society. These developments remind us of the end of the 20th century, when the effect and impact of the Internet was actively debated.

The authors tried using the following approach to clarify the issues for the prompt realization of IoT. First, the players were conveniently divided into two groups: ICT industry players and Things industry players. Next, we met major players in the ICT industry and Things industry and asked about the challenges they faced and the challenges the other side faced in creating IoT.

The ICT industry players mentioned here include communication carriers, ICT equipment vendors, the Internet service providers, application vendors, and software houses. The Things industry players include home and housing equipment manufacturers, infrastructure providers such as railways companies and power companies, and manufacturers of home appliances such as air conditioners and refrigerators, which are also the ICT users.

This paper is principally a summary of the meetings results, and a presentation of the micro case studies about the challenges for realizing IoT services. It is not an overview of the IoT world or a macro-proposal intended to promote the benefits of IoT.

In addition, the revised version includes an examination of the possibilities of new direct physical threats in the IoT era that have not yet been seen. These threats should affect the safety of our bodies, lives, and "things," which includes property. For this reason, this issue shall be referred to as "IoT Safety" to distinguish it from Information Security for the purpose of this document.

2. Technical Challenges

2.1. Safety, Security and Privacy

2.1.1. Challenges in protecting lives and property from IoT-related threats (IoT Safety)

The introduction of IoT may generate threats to "Safety" through the actual operation of mechanical devices, in addition to the Information Security problems discussed in Section 2.1.2 below. For example, the spread of applications for visualizing electric power consumption allows for mischief in device operation without the use of identity fraud or hacking. In addition, there is the potential for problems to arise in the normal operation of individual devices that are not caused by abnormal current or voltage, another troubling aspect of the introduction of IoT. These issues cannot be resolved with ordinary information security measures for Network Layer 4 or lower. In another case, a command to an IoT device is proper by itself, but it may conflict with the other commands or its environmental status. Therefore, the authors consider it necessary

to have a system for interpreting the details of operations of many appliances and preventing operations according to the necessity in Layer 7 (what the authors tentatively call "Sekisyo".)

These threats are categorized into three types: threat to physical safety; the threat of the failure or destruction of equipment and property; and the threat of impeding the proper performance of equipment. The following section introduces examples of the different threats.

2.1.1.1. Safety of body/life

Information on things such as the use of faucets and housing equipment, the locking of the front doors and windows, and the state of electric power consumption based on the smart meter is used by smart houses to regulate homes. This information is used to determine whether anyone is at home, and the electronic lock of the front door and windows is unlocked and a notice of absence is issued to a thief.

2.1.1.2. Safety of equipment

Air conditioners and other equipment that normally are not expected to be frequently started or stopped each a day can be caused to break down by repeatedly turning them on and off as many as hundreds of times a day.

2.1.1.3. Proper performance of equipment

Water heaters containing a hot well can be caused to operate erratically. This is done by frequently transmitting signals from the mischief application instead of operation panel to tell the water heater that only 10% of the normal amount of hot water is needed, leaving the water heater perpetually low on water.

2.1.2. Information Security

We have confirmed two viewpoints regarding the information security of services using IoT equipment and devices. The first is tangible information security involving the critical infrastructure. The second concerns the information security of individuals and homes.

In regards to information security involving the critical infrastructure, the basic policy in the past was to stay physically disconnected from an external network, such as the Internet, to ensure information security. However, because of the advance in the systems from proprietary communication protocols to open IP protocols to detect symptoms of problems and to remotely maintain a large

number of facilities spread over a wide area, connecting to an external network will become unavoidable to achieve various goals. In addition, it is clear that isolated networks are also subject to the same kind of risks, even though it is not directly connected to the outside. There is no major difference in the information security risks because isolated networks are already the target of international cyber terrorism, with internal crimes and targeted attacks occurring more frequently. Based on these reasons, the ICT security of the social infrastructure requires an extremely high level of information security.

Looking at the information security of micro units, such as individuals and homes, the improved convenience provided by the introduction of IoT will lead to greater risks. For example, there is a product available for connecting the entrance door to the network. In ICT security technology, increasing the key length of the encryption makes it much harder to break. But even if the latest information security technology is used when it is installed, the information security technology will become obsolete and even pose a risk about halfway through the twenty- to thirty-year lifetime of the entrance door. As has been explained in other items, the ICT sense of time is completely different from that of Things.

2.1.1.3. Privacy in acquiring data

The problem of privacy in handling acquired data is a huge challenge for companies promoting IoT. In addition, the ownership of this data poses yet another challenge.

For example, railway companies have installed many cameras for station security and for marketing beverage vending machines. This creates problems for personal identification and privacy. At the present time, the companies are processing the images in real time and do not store the images to avoid the problems.

Another huge challenge is the ownership of data. Up until now, there has been a divided debate on whether data belonged to the company or to the users. Likewise, the relationship inside a small user group is also extremely diverse and complicated. One specific example is of a company that had obtained permission from the head of the household to use the data when it carried out an HEMS trial. Later on, the spouse of the head of the household disagreed and as a result permission to use the data was withdrawn.

2.2. Challenges posed by data acquisition, data distribution, data management and data quantity

2.2.1. Traffic patterns

The manner in which data is acquired from and distributed to IoT equipment/devices differs immensely from the traffic patterns of the present Internet. The present form of the Internet focuses on distributing information, and its systems focus on effectively delivering contents to the users. On the other hand, routinely or temporarily sending or receiving data through a huge number of various sensors and devices presents a very different kind of Internet traffic. However, questions such as how much traffic will come from what kind of Things, and how will they superimpose each other have not been sufficiently studied. There is no concrete explanation about the backbone design and operation of traffic, and there have been many cases in which the unclear specifications for IoT traffic made the design difficult on the communication company side. There are many challenges related to the set up and management of IoT equipment. We have heard from the construction companies that the configuration of IoT equipment with a large number of sensors involves a lot of hard work.

2.2.2. Acquired mass data

It is necessary to develop a management method to reuse acquired data safely and effectively. Even now, there are occasional instances of the theft and leakage of social data (such as IDs) that can be used to identify individuals. In the IoT era, there will be mass data that can lead to Things, and the Things in turn will lead to individuals. There are IoT industry players who do not invest as much in ICT systems as government agencies and large companies do, and thus a management system to safely and effectively reuse the acquired data needs to be developed. The laws and regulations related to ID management differ vastly by country and region. These issues related to society and individuals are largely affected by differences in common sense, and therefore need to be localized.

2.2.3. Explosive increase and diversity of data

In the future IoT era, there are concerns about the explosive increase in data quantity and the diversity of data sent from sensors and IoT equipment. On the other hand, M2M communication does not require mass data like images, and an extraordinary increase in traffic will be unlikely despite the increase in the number of sensors.

If data is sent from all Things, there will be an infinite number of different kinds of data. In addition, with the present form of Internet traffic, data is received by people, and most of it consists of video or image downloads. The download traffic is several times greater than that of the upload traffic. If there is a tremendous increase in the use of IoT, such as M2M communication, the difference between upload and download traffic will probably not be that much. It might be necessary to fundamentally review the network and in particular the last mile characteristics. The importance of this issue is not yet widely recognized.

2.3. Mapping of the physical world and the virtual world

2.3.1. Physically handling acquired data

The acquired data simply represents certain kinds of digital value, and it is important to uncover the meaning of this data. As described previously, configuration of IoT equipment, such as the large number of installed sensors, requires a lot of hard work. An even greater amount of effort will be needed to determine the meaning of the data and connect it to the physical world.

In energy management experiments, data is mapped manually. This is a time consuming process, and one that is prone to human error. Cases that rely on the use of human hands require the configuration of automated setting systems to reduce labor, costs, and human errors to introduce IoT

2.3.2. Data calibration

Another important thing is calibration. This involves properly linking the data sent from Things to the Things concerned, and correctly indicating the operating conditions.

It may be necessary to have a tool to treat this problem concerning continuation of operation and the one pertaining to introduction of IoT described previously as a package.

2.4. Product lifetime, generation management, and the cost of equipment updates

2.4.1. Product lifetime

The life of most ICT equipment is about 5 years or less, while the life of IoT equipment and devices is at least 10 years. There is a clear gap between these two types of equipment.

In the example of the entrance door connected to the network mentioned earlier, the door is often used for about twenty to thirty years after installed. If is connected to a network, the communication technology and communication service will most likely have undergone numerous generation changes in that twenty- to thirty-year time span. This presents a large gap between the ICT industry and the Things industry.

A solution to this problem that was reached during the meeting with the housing equipment manufacturers is that with the automatic control of multiple shutters in a building, the portion between the controller and the multiple shutters, the so-called mature technology, can be placed under the control of the shutter manufacturers, while the controller connected to the network will deal with the generation changes of the communication service.

2.4.2. Introducing IoT equipment into commodity equipment

It costs a lot to make the many different types of commodity equipment popular around the world usable as IoT equipment and devices. There are two ways to change commodity equipment into IoT equipment. One way is to convert it to IoT compatible equipment. The other way involves adding devices to commodity equipment. There are costs in both cases, and it will take a long time to introduce IoT unless different incentives are offered to help to overcome the burden of cost.

2.5. Too many related standards and the speed of standardization

2.5.1. Too many related standards

There are many standards related to IoT equipment and devices. There are multiple standards, technologies and services for communication technology, such as Bluetooth, Wi-Fi, NFC, and LTE, and it is difficult to choose which to apply.

The Things industry players do not always have the communication technology professionals needed for IoT. In the meeting, we learned that many companies were uncertain and hesitant about fields outside their own area of expertise. On the other hand, technological competition will improve quality as well as the level of completion, and thus will be beneficial for users.

In the future, a consulting business for clarifying ICT technology for the Things industry players may emerge. If there is a system that can interconnect multiple standards, it will accelerate the Things industry to enter IoT

2.5.2. Speed of standardization

The concept of product life in ICT industry is completely different from that of the Things industry, and as a result the concept of standardization also varies greatly. Before standardization occurs in the ICT industry, many different proposals are made, from which the best are selected. The final decision often changes, and products have to be updated in order to follow the changes in standards. But in the Things industry, the standards have to remain unchanged for as long as possible because of the long product lifetimes. Therefore, it takes a long time to determine when a particular standard has become mature. When the Things industry goes to implement a standard from the ICT industry, it feels that the standard is incredibly fluid and seemingly undecided. Furthermore, the standardization process of the two industries is very different, and making it difficult to work on the other side when trying to determine a standard.

2.6. Interoperability, fault isolation, and total quality assurance

2.6.1. Interoperability

The verification of interoperability poses a major challenge because of the configuration used by multi-vendors. In addition to interoperability between equipment, the ability to ensure backward compatibility is also important for bringing about the IoT world.

If these capabilities cannot be provided, it will be very difficult to create an IoT world in which past products can function.

2.6.2. Fault isolation

The method for fault isolation that may occur presents another challenge.

Many PC users have experienced various kinds of problems. When their PC experiences a problem, they have to isolate the faults by themselves, with no one available to lend a helping hand.

In the IoT world, these issues become more difficult and complicated. For example, a smart home is equipped with air conditioners, kitchen supplies, and doors connected to the Internet. A problem that occurs in the smart home poses a much more serious problem to end users than an e-mail failure or problem with a PC.

If users are left to isolate the fault on their own, they may not know which manufacturer they contact for repairs if they are unable to isolate the fault on their own, or the manufacturer may refuse to

perform repairs because they fall outside the scope of their responsibility. As can be seen, the issue is an important challenge that will determine whether the B2C specific IoT world can be established.

2.6.3. Quality assurance

The quality assurance of individual pieces of IoT equipment does not guarantee the total quality of IoT. Since IoT involves connecting multiple Things and communication, it is natural to assume that the total service quality will depend on the quality of the IoT equipment and devices, which can sometimes become bottleneck. However, users are not aware of this.

As was mentioned previously in Section 2.6 issues that are not directly related to the quality of an individual component can be important factors in determining the quality of the service. In this way, the quality of IoT is not decided by each individual Thing, but needs to be considered as a service spread across the network.

2.7. Product design policy

2.7.1. Changes in design policy

The design policy has to be changed from placing emphasis on the high functionality of a single product to stressing the singular function of individual products as well as how they work in coordination with other products. For many years, the Things industry has focused on producing high functionality products with added value. But in the IoT era, the implicit assumption is to confine Things to their basic function and enhance the level of coordination between Things, rather than focusing on the added value. Simplified Things must be able to be controlled with an external application that can also be used by the Things of cross manufacturers.

Given this situation, the Things industry faces the challenge of adopting a completely different policy. During the meeting with the manufacturing industries, we could sense their difficulty in understanding and recognizing the need to change the policy.

2.8. Various technology restrictions within actual usage

2.8.1. Using radio waves

There are many cases that have provided us with insight about issues related to the use of radio waves in IoT (such as the wave traveling range and whether or not it travels further than stated in assumptions available). The suppliers or providers who configure IoT

are not always wave communication technology experts. People who are unfamiliar with radio waves seem to think that waves travel from antenna to antenna in a straight line, and that they can be blocked by obstacles. As a result, they often ask questions about how many meters radio waves can travel or whether radio waves can actually travel. Few people understand the fact that the emitted radio waves are reflected from various locations and are superimposed at the reception point where they are received, or that depending on how waves are reflected a change in the reception signal intensity, called fading, may occur. The lack of engineers who can advise on specialized matters such as these poses a major obstacle.

2.8.2. Batteries

The power capacity and lifetime of batteries represent another set of challenges similar in nature to the issue of radio waves traveling distance. There are questions such as the difference between the real and catalog specifications, as well as factors that affect the battery power capacity. The IoT providers, who are also users of IoT, have to solve these issues, while these are difficult problems even for experts.

2.8.3. Wiring

The incredible amount of wiring and its complexity (power lines and communication lines) pose major challenges. The complexity of wiring- such as the large number of sensors and equipment, the power lines that drive them, and the communication lines that connect them to the network for acquiring information-is to the point that people doing IoT installation work will start wishing for a wire harness. In addition, the installation of cables and electric work are often done by different engineers. This makes the issue even more complicated.

2.8.4. Being open

A single company alone cannot make all the commodities for IoT. The IoT world needs to be open, and this can only be achieved with the cooperation of many different industries. Up until now, companies in the Things industry have developed products in a closed loop process, seeking to capture users with their company's own products. For this reason, they lack an open design concept of interoperability. Today, an entirely new design concept is needed to design products that can interconnect with the products of other companies.

3. Non-technical Challenges

3.1. Changing the product paradigm

3.1.1. Ecosystems

While the goal of setting up IoT is to generate new value, it may actually lead to the destruction of the ecosystems in which industries operate. In the IoT era, the traditional vertically integrated way of producing Things in manufacturing industries will consume too much time and cost. This approach also makes it difficult to incorporate the ideas of other cultures. The need for paradigm shift is easy to understand, but difficult to implement. Promoting this shift will pose a management challenge that requires a considerable amount of skill and effort to overcome.

3.1.2. Coordination and significant changes in strategy

It will become necessary to run businesses jointly with new partners, as well as cooperate and work in coordination with other industries and competitors. This issue-even when it is fully understood-will be very difficult to address and put into practice.

We have seen instances in which only a limited amount of information was given when parties exchanged opinions. There have also been instances in which communication was difficult because of differences in terminology and culture.

3.1.3. Competition with existing industries

The issue of competition with existing industries often arises when attempts are made to change or reform a business model change or reform. This issue can also be viewed as the reorganization of industries, rather than competition between existing industries. However, this realignment of industries is difficult to move forward in the absence of supervisors.

3.2. Benefits

3.2.1. Rising costs and monetization

Introducing IoT within products will cause costs to go up, and yet the benefits it provides are unclear. There is no specific killer application available, and the number of users will not rise immediately. Therefore, finding a way to make the business profitable will be very difficult. This issue is especially difficult for businesses and products that rely on cost reductions to deliver low prices that make them competitive.

3.3. Information security and privacy of social systems

3.3.1. Classification of ownership, location, and the usage of data

There are many questions regarding the wide variety of data gathered from IoT equipment, including questions related to ownership, storage location, and the authorization to grant a license to use data. These need to be addressed so that the system and equipment can be accepted by society.

For example, if a company installs a door in a house that gathers data on the opening and closing of the door, questions about the data will arise. Does it belong to the users or the company? Can another company use this data?

3.4. Disclosure of data

3.4.1. Side effects and malicious use potentially caused by the disclosure of data

For example, it has been shown that the electricity smart meter can lead to burglary because it shows when electricity is used and not used, providing an indication of the time when no one is home. This particular example demonstrates the importance of ensuring information security and privacy.

3.5. Preparing social support

3.5.1. Regulations

Systems of laws and regulations are important for ensuring the safety of the conventional products, but they can also be a barrier for innovation.

IoT can be affected by laws and regulations at home and abroad, and can also be influenced by regulations that extend across multiple countries. Regulatory authorities need to monitor IoT carefully and adjust the regulations and laws they oversee in a way that does not negatively impact the global competition environment.

3.5.2. Corporate social responsibility

In addition to pursuing profit, companies that promote IoT also need to improve the benefits offered to users and society

3.5.3. Customization for individual customers

There is an ongoing shift in demand away from general products to customized products for individual customers. This could also be viewed as a shift away from manufacturing businesses to service businesses. IoT will play an important role in this shift.

Instead of manufacturing Things through mass production, it will be easier to customize a product by moving some of the functions to an application. Likewise, the manufacturing business also needs to move forward with the previously mentioned paradigm shift in order to achieve customization

3.5.4. IoT literacy of the users

Because Things are connected to the network, apps will need to be created. Some of these will serve as the interface with which people interact with IoT.

In the IoT era of the future, users will need to possess a certain amount of knowledge about IoT apps

3.5.5. Individual vs. family

The issue of whether the data of Things in the house belongs to the family or the individual will largely affect data analysis and the handling of privacy.

As was mentioned in Section 2.1.2, the spouse could later object to the head of the household granting authorization to use data.

4. Information Security Considerations

Meetings with the players in various IoT fields provided insight into information security issues. These issues are described in the following sections.

- o Section 2.1.2 Physical damper of devices
- o Section 2.1.2 Product lifetime and encryption strength

For details, please see the corresponding text.

5. Privacy Considerations

Similarly, issues regarding privacy are described in the following sections.

- o Section 2.1.2, Section 3.3.1 Ownership of the data
- o Section 3.4.1 Data disclosure and malicious use
- o Section 3.5.5 Individual vs. family

For details, please see the corresponding text.

6. Acknowledgments

We would like to thank the foundation the promotion of industrial science and its RC-88 member companies for their cooperation.

And we also appreciate Ministry of Internal Affairs and Communications.

Authors' Addresses

Hiroyuki Baba
The University of Tokyo
Institute of Industrial Science
4-6-1 Komaba
Meguro-ku, Tokyo 153-8505
Japan

Email: hbaba@iis.u-tokyo.ac.jp

Yoshiki Ishida
Japan Network Enabler Corporation
21F KDDI Otemachi Bldg.
1-8-1 Otemachi
Chiyoda-ku, Tokyo 100-0004
Japan

Email: ishida@jpne.co.jp

Takayuki Amatsu
Tokyo Electric Power Company, Inc.
1-1-3 Uchisaiwai-cho
Chiyoda-ku, Tokyo 100-8560
Japan

Email: amatsu.t@tepcoco.jp

Koichi Kunitake
BroadBand Tower, Inc.
Uchisaiwaicho Tokyu Bldg.
1-3-2 Uchisaiwai-cho
Chiyoda-ku, Tokyo 100-0011
Japan

Email: kokunitake@bbtower.co.jp

Kaoru Maeda
Individual Contributor

Email: kaorumaeda.ml@gmail.com

Internet Research Task Force
Internet-Draft
Intended status: Informational
Expires: March 12, 2021

H. Baba
The University of Tokyo
Y. Ishida
Japan Network Enabler Corporation
T. Amatsu
Tokyo Electric Power Company, Inc.
K. Kunitake
BroadBand Tower, Inc.
K. Maeda
Individual Contributor
September 8, 2020

Problems in and among industries for the prompt realization of IoT and
safety considerations
draft-baba-iot-problems-09

Abstract

This document contains opinions gathered from enterprises engaging in the IoT business as stated in the preceding version hereof, and also examines the possibilities of new social problems in the IoT era. Recognition of the importance of information security has grown in step with the rising use of the Internet. Closer examination reveals that the IoT era may see a new direct physical threat to users. For instance, the situation at a smart house may lead it to judge that the owner has only temporarily stepped out, causing it to unlock the front door, which in turn makes it easier for thieves to enter. These kinds of scenarios may occur without identity fraud, hacking, and other means of compromising information security. Therefore, for the purpose of this document, this issue shall be referred to as "IoT Safety" to distinguish it from Information Security.

We believe that it is necessary to deepen our understanding of these new IoT-related threats through discussion and ensure there are measures to address these threats in the future. At the same time, we must also coordinate these measures with the solutions to the problems described in the previous version of this document.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on March 12, 2021.

Copyright Notice

Copyright (c) 2020 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	3
2. Technical Challenges	4
2.1. Safety, Security and Privacy	4
2.1.1. Challenges in protecting lives and property from IoT-related threats (IoT Safety)	4
2.1.1.1. Safety of body/life	5
2.1.1.2. Safety of equipment	5
2.1.1.3. Proper performance of equipment	5
2.1.2. Information Security	5
2.1.3. Privacy in acquiring data	6
2.2. Challenges posed by data acquisition, data distribution, data management and data quantity	7
2.2.1. Traffic patterns	7
2.2.2. Acquired mass data	7
2.2.3. Explosive increase and diversity of data	7
2.3. Mapping of the physical world and the virtual world	8
2.3.1. Physically handling acquired data	8
2.3.2. Data calibration	8
2.4. Product lifetime, generation management, and the cost of equipment updates	8
2.4.1. Product lifetime	8
2.4.2. Introducing IoT equipment into commodity equipment	9
2.5. Too many related standards and the speed of standardization	9

2.5.1.	Too many related standards	9
2.5.2.	Speed of standardization	10
2.6.	Interoperability, fault isolation, and total quality assurance	10
2.6.1.	Interoperability	10
2.6.2.	Fault isolation	10
2.6.3.	Quality assurance	11
2.7.	Product design policy	11
2.7.1.	Changes in design policy	11
2.8.	Various technology restrictions within actual usage . . .	11
2.8.1.	Using radio waves	11
2.8.2.	Batteries	12
2.8.3.	Wiring	12
2.8.4.	Being open	12
3.	Non-technical Challenges	13
3.1.	Changing the product paradigm	13
3.1.1.	Ecosystems	13
3.1.2.	Coordination and significant changes in strategy . .	13
3.1.3.	Competition with existing industries	13
3.2.	Benefits	13
3.2.1.	Rising costs and monetization	13
3.3.	Information security and privacy of social systems . . .	14
3.3.1.	Classification of ownership, location, and the usage of data	14
3.4.	Disclosure of data	14
3.4.1.	Side effects and malicious use potentially caused by the disclosure of data	14
3.5.	Preparing social support	14
3.5.1.	Regulations	14
3.5.2.	Corporate social responsibility	14
3.5.3.	Customization for individual customers	15
3.5.4.	IoT literacy of the users	15
3.5.5.	Individual vs. family	15
4.	Security Considerations	15
5.	Privacy Considerations	15
6.	IANA Considerations	16
7.	Acknowledgments	16
	Authors' Addresses	16

1. Introduction

Many activities are progressing in various fields, such as the proposal of standards for creating an IoT world. There are also many reports that analyze and predict the benefits that IoT can bring to the economy and society. These developments remind us of the end of the 20th century, when the effect and impact of the Internet was actively debated.

The authors tried using the following approach to clarify the issues for the prompt realization of IoT. First, the players were conveniently divided into two groups: ICT industry players and Things industry players. Next, we met major players in the ICT industry and Things industry and asked about the challenges they faced and the challenges the other side faced in creating IoT.

The ICT industry players mentioned here include communication carriers, ICT equipment vendors, the Internet service providers, application vendors, and software houses. The Things industry players include home and housing equipment manufacturers, infrastructure providers such as railways companies and power companies, and manufacturers of home appliances such as air conditioners and refrigerators, which are also the ICT users.

This paper is principally a summary of the meetings results, and a presentation of the micro case studies about the challenges for realizing IoT services. It is not an overview of the IoT world or a macro-proposal intended to promote the benefits of IoT.

In addition, the revised version includes an examination of the possibilities of new direct physical threats in the IoT era that have not yet been seen. These threats should affect the safety of our bodies, lives, and "things," which includes property. For this reason, this issue shall be referred to as "IoT Safety" to distinguish it from Information Security for the purpose of this document.

For the past few years, we got new findings through COMMA House, the experimental smart house owned by The University of Tokyo. Therefore, we will add new topics to the next version.

2. Technical Challenges

2.1. Safety, Security and Privacy

2.1.1. Challenges in protecting lives and property from IoT-related threats (IoT Safety)

The introduction of IoT may generate threats to "Safety" through the actual operation of mechanical devices, in addition to the Information Security problems discussed in Section 2.1.2 below. For example, the spread of applications for visualizing electric power consumption allows for mischief in device operation without the use of identity fraud or hacking. In addition, there is the potential for problems to arise in the normal operation of individual devices that are not caused by abnormal current or voltage, another troubling aspect of the introduction of IoT. These issues cannot be resolved

with ordinary information security measures for Network Layer 4 or lower. In another case, a command to an IoT device is proper by itself, but it may conflict with the other commands or its environmental status. Therefore, the authors consider it necessary to have a system for interpreting the details of operations of many appliances and preventing operations according to the necessity in Layer 7 (what the authors tentatively call "Sekisyo".)

These threats are categorized into three types: threat to physical safety; the threat of the failure or destruction of equipment and property; and the threat of impeding the proper performance of equipment. The following section introduces examples of the different threats.

2.1.1.1. Safety of body/life

Information on things such as the use of faucets and housing equipment, the locking of the front doors and windows, and the state of electric power consumption based on the smart meter is used by smart houses to regulate homes. This information is used to determine whether anyone is at home, and the electronic lock of the front door and windows is unlocked and a notice of absence is issued to a thief.

2.1.1.2. Safety of equipment

Air conditioners and other equipment that normally are not expected to be frequently started or stopped each a day can be caused to break down by repeatedly turning them on and off as many as hundreds of times a day.

2.1.1.3. Proper performance of equipment

Water heaters containing a hot well can be caused to operate erratically. This is done by frequently transmitting signals from the mischief application instead of operation panel to tell the water heater that only 10% of the normal amount of hot water is needed, leaving the water heater perpetually low on water.

2.1.2. Information Security

We have confirmed two viewpoints regarding the information security of services using IoT equipment and devices. The first is tangible information security involving the critical infrastructure. The second concerns the information security of individuals and homes.

In regards to information security involving the critical infrastructure, the basic policy in the past was to stay physically

disconnected from an external network, such as the Internet, to ensure information security. However, because of the advance in the systems from proprietary communication protocols to open IP protocols to detect symptoms of problems and to remotely maintain a large number of facilities spread over a wide area, connecting to an external network will become unavoidable to achieve various goals. In addition, it is clear that isolated networks are also subject to the same kind of risks, even though it is not directly connected to the outside. There is no major difference in the information security risks because isolated networks are already the target of international cyber terrorism, with internal crimes and targeted attacks occurring more frequently. Based on these reasons, the ICT security of the social infrastructure requires an extremely high level of information security.

Looking at the information security of micro units, such as individuals and homes, the improved convenience provided by the introduction of IoT will lead to greater risks. For example, there is a product available for connecting the entrance door to the network. In ICT security technology, increasing the key length of the encryption makes it much harder to break. But even if the latest information security technology is used when it is installed, the information security technology will become obsolete and even pose a risk about halfway through the twenty- to thirty-year lifetime of the entrance door. As has been explained in other items, the ICT sense of time is completely different from that of Things.

2.1.3. Privacy in acquiring data

The problem of privacy in handling acquired data is a huge challenge for companies promoting IoT. In addition, the ownership of this data poses yet another challenge.

For example, railway companies have installed many cameras for station security and for marketing beverage vending machines. This creates problems for personal identification and privacy. At the present time, the companies are processing the images in real time and do not store the images to avoid the problems.

Another huge challenge is the ownership of data. Up until now, there has been a divided debate on whether data belonged to the company or to the users. Likewise, the relationship inside a small user group is also extremely diverse and complicated. One specific example is of a company that had obtained permission from the head of the household to use the data when it carried out an HEMS trial. Later on, the spouse of the head of the household disagreed and as a result permission to use the data was withdrawn.

2.2. Challenges posed by data acquisition, data distribution, data management and data quantity

2.2.1. Traffic patterns

The manner in which data is acquired from and distributed to IoT equipment/devices differs immensely from the traffic patterns of the present Internet. The present form of the Internet focuses on distributing information, and its systems focus on effectively delivering contents to the users. On the other hand, routinely or temporarily sending or receiving data through a huge number of various sensors and devices presents a very different kind of Internet traffic. However, questions such as how much traffic will come from what kind of Things, and how will they superimpose each other have not been sufficiently studied. There is no concrete explanation about the backbone design and operation of traffic, and there have been many cases in which the unclear specifications for IoT traffic made the design difficult on the communication company side. There are many challenges related to the set up and management of IoT equipment. We have heard from the construction companies that the configuration of IoT equipment with a large number of sensors involves a lot of hard work.

2.2.2. Acquired mass data

It is necessary to develop a management method to reuse acquired data safely and effectively. Even now, there are occasional instances of the theft and leakage of social data (such as IDs) that can be used to identify individuals. In the IoT era, there will be mass data that can lead to Things, and the Things in turn will lead to individuals. There are IoT industry players who do not invest as much in ICT systems as government agencies and large companies do, and thus a management system to safely and effectively reuse the acquired data needs to be developed. The laws and regulations related to ID management differ vastly by country and region. These issues related to society and individuals are largely affected by differences in common sense, and therefore need to be localized.

2.2.3. Explosive increase and diversity of data

In the future IoT era, there are concerns about the explosive increase in data quantity and the diversity of data sent from sensors and IoT equipment. On the other hand, M2M communication does not require mass data like images, and an extraordinary increase in traffic will be unlikely despite the increase in the number of sensors.

If data is sent from all Things, there will be an infinite number of different kinds of data. In addition, with the present form of Internet traffic, data is received by people, and most of it consists of video or image downloads. The download traffic is several times greater than that of the upload traffic. If there is a tremendous increase in the use of IoT, such as M2M communication, the difference between upload and download traffic will probably not be that much. It might be necessary to fundamentally review the network and in particular the last mile characteristics. The importance of this issue is not yet widely recognized.

2.3. Mapping of the physical world and the virtual world

2.3.1. Physically handling acquired data

The acquired data simply represents certain kinds of digital value, and it is important to uncover the meaning of this data. As described previously, configuration of IoT equipment, such as the large number of installed sensors, requires a lot of hard work. An even greater amount of effort will be needed to determine the meaning of the data and connect it to the physical world.

In energy management experiments, data is mapped manually. This is a time consuming process, and one that is prone to human error. Cases that rely on the use of human hands require the configuration of automated setting systems to reduce labor, costs, and human errors to introduce IoT

2.3.2. Data calibration

Another important thing is calibration. This involves properly linking the data sent from Things to the Things concerned, and correctly indicating the operating conditions.

It may be necessary to have a tool to treat this problem concerning continuation of operation and the one pertaining to introduction of IoT described previously as a package.

2.4. Product lifetime, generation management, and the cost of equipment updates

2.4.1. Product lifetime

The life of most ICT equipment is about 5 years or less, while the life of IoT equipment and devices is at least 10 years. There is a clear gap between these two types of equipment.

In the example of the entrance door connected to the network mentioned earlier, the door is often used for about twenty to thirty years after installed. If is connected to a network, the communication technology and communication service will most likely have undergone numerous generation changes in that twenty- to thirty-year time span. This presents a large gap between the ICT industry and the Things industry.

A solution to this problem that was reached during the meeting with the housing equipment manufacturers is that with the automatic control of multiple shutters in a building, the portion between the controller and the multiple shutters, the so-called mature technology, can be placed under the control of the shutter manufacturers, while the controller connected to the network will deal with the generation changes of the communication service.

2.4.2. Introducing IoT equipment into commodity equipment

It costs a lot to make the many different types of commodity equipment popular around the world usable as IoT equipment and devices. There are two ways to change commodity equipment into IoT equipment. One way is to convert it to IoT compatible equipment. The other way involves adding devices to commodity equipment. There are costs in both cases, and it will take a long time to introduce IoT unless different incentives are offered to help to overcome the burden of cost.

2.5. Too many related standards and the speed of standardization

2.5.1. Too many related standards

There are many standards related to IoT equipment and devices. There are multiple standards, technologies and services for communication technology, such as Bluetooth, Wi-Fi, NFC, and LTE, and it is difficult to choose which to apply.

The Things industry players do not always have the communication technology professionals needed for IoT. In the meeting, we learned that many companies were uncertain and hesitant about fields outside their own area of expertise. On the other hand, technological competition will improve quality as well as the level of completion, and thus will be beneficial for users.

In the future, a consulting business for clarifying ICT technology for the Things industry players may emerge. If there is a system that can interconnect multiple standards, it will accelerate the Things industry to enter IoT

2.5.2. Speed of standardization

The concept of product life in ICT industry is completely different from that of the Things industry, and as a result the concept of standardization also varies greatly. Before standardization occurs in the ICT industry, many different proposals are made, from which the best is selected. The final decision often changes, and products have to be updated in order to follow the changes in standards. But in the Things industry, the standards have to remain unchanged for as long as possible because of the long product lifetimes. Therefore, it takes a long time to determine when a particular standard has become mature. When the Things industry goes to implement a standard from the ICT industry, it feels that the standard is incredibly fluid and seemingly undecided. Furthermore, the standardization process of the two industries is very different, and making it difficult to work on the other side when trying to determine a standard.

2.6. Interoperability, fault isolation, and total quality assurance

2.6.1. Interoperability

The verification of interoperability poses a major challenge because of the configuration used by multi-vendors. In addition to interoperability between equipment, the ability to ensure backward compatibility is also important for bringing about the IoT world.

If these capabilities cannot be provided, it will be very difficult to create an IoT world in which past products can function.

2.6.2. Fault isolation

The method for fault isolation that may occur presents another challenge.

Many PC users have experienced various kinds of problems. When their PC experiences a problem, they have to isolate the faults by themselves, with no one available to lend a helping hand.

In the IoT world, these issues become more difficult and complicated. For example, a smart home is equipped with air conditioners, kitchen supplies, and doors connected to the Internet. A problem that occurs in the smart home poses a much more serious problem to end users than an e-mail failure or problem with a PC.

If users are left to isolate the fault on their own, they may not know which manufacturer they contact for repairs if they are unable to isolate the fault on their own, or the manufacturer may refuse to perform repairs because they fall outside the scope of their

responsibility. As can be seen, the issue is an important challenge that will determine whether the B2C specific IoT world can be established.

2.6.3. Quality assurance

The quality assurance of individual pieces of IoT equipment does not guarantee the total quality of IoT. Since IoT involves connecting multiple Things and communication, it is natural to assume that the total service quality will depend on the quality of the IoT equipment and devices, which can sometimes become bottleneck. However, users are not aware of this.

As was mentioned previously in Section 2.6 issues that are not directly related to the quality of an individual component can be important factors in determining the quality of the service. In this way, the quality of IoT is not decided by each individual Thing, but needs to be considered as a service spread across the network.

2.7. Product design policy

2.7.1. Changes in design policy

The design policy has to be changed from placing emphasis on the high functionality of a single product to stressing the singular function of individual products as well as how they work in coordination with other products. For many years, the Things industry has focused on producing high functionality products with added value. But in the IoT era, the implicit assumption is to confine Things to their basic function and enhance the level of coordination between Things, rather than focusing on the added value. Simplified Things must be able to be controlled with an external application that can also be used by the Things of cross manufacturers.

Given this situation, the Things industry faces the challenge of adopting a completely different policy. During the meeting with the manufacturing industries, we could sense their difficulty in understanding and recognizing the need to change the policy.

2.8. Various technology restrictions within actual usage

2.8.1. Using radio waves

There are many cases that have provided us with insight about issues related to the use of radio waves in IoT (such as the wave traveling range and whether or not it travels further than stated in assumptions available). The suppliers or providers who configure IoT are not always wave communication technology experts. People who are

unfamiliar with radio waves seem to think that waves travel from antenna to antenna in a straight line, and that they can be blocked by obstacles. As a result, they often ask questions about how many meters radio waves can travel or whether radio waves can actually travel. Few people understand the fact that the emitted radio waves are reflected from various locations and are superimposed at the reception point where they are received, or that depending on how waves are reflected a change in the reception signal intensity, called fading, may occur. The lack of engineers who can advise on specialized matters such as these poses a major obstacle.

2.8.2. Batteries

The power capacity and lifetime of batteries represent another set of challenges similar in nature to the issue of radio waves traveling distance. There are questions such as the difference between the real and catalog specifications, as well as factors that affect the battery power capacity. The IoT providers, who are also users of IoT, have to solve these issues, while these are difficult problems even for experts.

2.8.3. Wiring

The incredible amount of wiring and its complexity (power lines and communication lines) pose major challenges. The complexity of wiring—such as the large number of sensors and equipment, the power lines that drive them, and the communication lines that connect them to the network for acquiring information—is to the point that people doing IoT installation work will start wishing for a wire harness. In addition, the installation of cables and electric work are often done by different engineers. This makes the issue even more complicated.

2.8.4. Being open

A single company alone cannot make all the commodities for IoT. The IoT world needs to be open, and this can only be achieved with the cooperation of many different industries. Up until now, companies in the Things industry have developed products in a closed loop process, seeking to capture users with their company's own products. For this reason, they lack an open design concept of interoperability. Today, an entirely new design concept is needed to design products that can interconnect with the products of other companies.

3. Non-technical Challenges

3.1. Changing the product paradigm

3.1.1. Ecosystems

While the goal of setting up IoT is to generate new value, it may actually lead to the destruction of the ecosystems in which industries operate. In the IoT era, the traditional vertically integrated way of producing Things in manufacturing industries will consume too much time and cost. This approach also makes it difficult to incorporate the ideas of other cultures. The need for paradigm shift is easy to understand, but difficult to implement. Promoting this shift will pose a management challenge that requires a considerable amount of skill and effort to overcome.

3.1.2. Coordination and significant changes in strategy

It will become necessary to run businesses jointly with new partners, as well as cooperate and work in coordination with other industries and competitors. This issue—even when it is fully understood—will be very difficult to address and put into practice.

We have seen instances in which only a limited amount of information was given when parties exchanged opinions. There have also been instances in which communication was difficult because of differences in terminology and culture.

3.1.3. Competition with existing industries

The issue of competition with existing industries often arises when attempts are made to change or reform a business model change or reform. This issue can also be viewed as the reorganization of industries, rather than competition between existing industries. However, this realignment of industries is difficult to move forward in the absence of supervisors.

3.2. Benefits

3.2.1. Rising costs and monetization

Introducing IoT within products will cause costs to go up, and yet the benefits it provides are unclear. There is no specific killer application available, and the number of users will not rise immediately. Therefore, finding a way to make the business profitable will be very difficult. This issue is especially difficult for businesses and products that rely on cost reductions to deliver low prices that make them competitive.

3.3. Information security and privacy of social systems

3.3.1. Classification of ownership, location, and the usage of data

There are many questions regarding the wide variety of data gathered from IoT equipment, including questions related to ownership, storage location, and the authorization to grant a license to use data. These need to be addressed so that the system and equipment can be accepted by society.

For example, if a company installs a door in a house that gathers data on the opening and closing of the door, questions about the data will arise. Does it belong to the users or the company? Can another company use this data?

3.4. Disclosure of data

3.4.1. Side effects and malicious use potentially caused by the disclosure of data

For example, it has been shown that the electricity smart meter can lead to burglary because it shows when electricity is used and not used, providing an indication of the time when no one is home. This particular example demonstrates the importance of ensuring information security and privacy.

3.5. Preparing social support

3.5.1. Regulations

Systems of laws and regulations are important for ensuring the safety of the conventional products, but they can also be a barrier for innovation.

IoT can be affected by laws and regulations at home and abroad, and can also be influenced by regulations that extend across multiple countries. Regulatory authorities need to monitor IoT carefully and adjust the regulations and laws they oversee in a way that does not negatively impact the global competition environment.

3.5.2. Corporate social responsibility

In addition to pursuing profit, companies that promote IoT also need to improve the benefits offered to users and society

3.5.3. Customization for individual customers

There is an ongoing shift in demand away from general products to customized products for individual customers. This could also be viewed as a shift away from manufacturing businesses to service businesses. IoT will play an important role in this shift.

Instead of manufacturing Things through mass production, it will be easier to customize a product by moving some of the functions to an application. Likewise, the manufacturing business also needs to move forward with the previously mentioned paradigm shift in order to achieve customization

3.5.4. IoT literacy of the users

Because Things are connected to the network, apps will need to be created. Some of these will serve as the interface with which people interact with IoT.

In the IoT era of the future, users will need to possess a certain amount of knowledge about IoT apps

3.5.5. Individual vs. family

The issue of whether the data of Things in the house belongs to the family or the individual will largely affect data analysis and the handling of privacy.

As was mentioned in Section 2.1.2, the spouse could later object to the head of the household granting authorization to use data.

4. Security Considerations

Meetings with the players in various IoT fields provided insight into information security issues. These issues are described in the following sections.

- o Section 2.1.2 Physical damper of devices
- o Section 2.1.2 Product lifetime and encryption strength

For details, please see the corresponding text.

5. Privacy Considerations

Similarly, issues regarding privacy are described in the following sections.

- o Section 2.1.2, Section 3.3.1 Ownership of the data
- o Section 3.4.1 Data disclosure and malicious use
- o Section 3.5.5 Individual vs. family

For details, please see the corresponding text.

6. IANA Considerations

This document has no actions for IANA.

7. Acknowledgments

We would like to thank the foundation the promotion of industrial science and its RC-88 member companies for their cooperation.

And we also appreciate Ministry of Internal Affairs and Communications.

Authors' Addresses

Hiroyuki Baba
The University of Tokyo
Institute of Industrial Science
4-6-1 Komaba
Meguro-ku, Tokyo 153-8505
Japan

Email: hbaba@iis.u-tokyo.ac.jp

Yoshiki Ishida
Japan Network Enabler Corporation
7F S-GATE Akasaka-Sanno.
1-8-1 Akasaka
Minato-ku, Tokyo 107-0052
Japan

Email: ishida@jpne.co.jp

Takayuki Amatsu
Tokyo Electric Power Company, Inc.
1-1-3 Uchisaiwai-cho
Chiyoda-ku, Tokyo 100-8560
Japan

Email: amatsu.t@tepcoco.jp

Koichi Kunitake
BroadBand Tower, Inc.
Hibiya Parkfront.
2-1-6, Uchisaiwai-cho
Chiyoda-ku, Tokyo 100-0011
Japan

Email: kokunitake@bbtower.co.jp

Kaoru Maeda
Individual Contributor
Japan

Email: kaorumaeda.ml@gmail.com

TSVWG
Internet-Draft
Intended status: Informational
Expires: March 3, 2018

G. Fairhurst
University of Aberdeen
August 30, 2017

The Impact of Transport Header Encryption on Operation and Evolution of
the Internet
draft-fairhurst-tsvwg-transport-encrypt-03

Abstract

This document describes implications of applying end-to-end encryption at the transport layer. It identifies some in-network uses of transport layer header information that can be used with a transport header integrity check. It reviews the implication of developing encrypted end-to-end transport protocols and examines the implication of developing and deploying encrypted end-to-end transport protocols. Since transport measurement and analysis of the impact of network characteristics have been important to the design of current transport protocols, it also considers some anticipated implications on transport and application evolution.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on March 3, 2018.

Copyright Notice

Copyright (c) 2017 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of

publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	3
2. Internet Transports and Pervasive Encryption	5
2.1. Authenticating the Transport Protocol Header	6
2.2. Encrypting the Transport Payload	6
2.3. Encrypting the Transport Header	6
2.4. Authenticating Transport Information and Selectively Encrypting the Transport Header	7
2.5. Adding Transport Information to Network-Layer Protocol Headers	7
3. Use of Transport Headers in the Network	8
3.1. Use to Identify Flows and Packet Formats	9
3.2. Measurements derived from Transport Header Information	9
3.2.1. Use to Characterise Traffic Rate and Volume	10
3.2.2. Measuring Loss Rate and Loss Pattern	10
3.2.3. Measuring Throughput and Goodput	11
3.2.4. Measuring Latency (Network Transit Delay and Jitter)	11
3.2.5. Measuring Flow Reordering	12
3.3. Measurements derived from Network-Transport Information	12
3.3.1. Use of IPv6 Network-Layer Flow Label	12
3.3.2. Use Network-Layer Differentiated Services Code Point Point	13
3.3.3. Use of Explicit Congestion Marking	13
4. Transport Measurement	14
4.1. Point of Measurement	14
4.2. Use by Operators to Plan and Provision Networks	15
4.3. Service Performance Measurement	15
4.4. Use for Network Diagnostics and Troubleshooting	15
4.5. Acceptable Response to Congestion	16
4.5.1. Measuring Compliance of UDP Traffic	16
4.5.2. Measuring Transport to Support Network Operations	17
5. Observing Transport Flows with Encrypted Transport Header Fields	17
5.1. Transport Information at the Network Layer	17
5.2. An Observable Transport Flow Identifier	18
5.2.1. A Method to Determine Header Format	18
5.2.2. Use of a Transport as a Substrate	18
5.2.3. Support for Mobility and Flow Migration	19
5.2.4. Flow Start and Stop	19
5.3. Observable Transport Sequence Number	20

5.4.	Observable Transport Reception	20
5.5.	Observable Transport Timestamps	21
5.6.	Observable ECN Transport Feedback Information	21
5.7.	Other Observable Transport Fields	21
5.8.	Interpretation of Transport Header Fields	21
5.9.	Requirements for Transport Measurement	22
6.	The Effect of Encrypting Transport Header Fields	23
6.1.	Independent Measurement	23
6.2.	Characterising "Unknown" Network Traffic	24
6.3.	Accountability and Internet Transport Portocols	24
7.	Implications on Evolution of the Internet Transport	25
8.	Acknowledgements	28
9.	IANA Considerations	28
10.	Security Considerations	28
11.	References	29
11.1.	Normative References	29
11.2.	Informative References	29
Appendix A.	Revision information	33
Author's Address	34

1. Introduction

This document discusses the implications of end-to-end encryption applied at the transport layer, and examines the impact on transport protocol design, transport use, and network operations and management. It also considers some anticipated implications on transport and application evolution.

The transport layer is the first end-to-end layer in the network stack. Despite headers having end-to-end meaning, some transport headers have come to be used in various ways within the Internet. In response to pervasive monitoring [RFC7624] revelations and the IETF consensus that "Pervasive Monitoring is an Attack" [RFC7258], efforts are underway to increase encryption of Internet traffic, which would prevent visibility of transport headers. This has implications on how network protocols are designed and used [I-D.mm-wg-effect-encrypt].

Transport information that is sent without end-to-end integrity check could be modified by "middleboxes" - defined as any intermediary box performing functions apart from normal, standard functions of an IP router on the data path between a source host and destination host [RFC3234]. When transport headers are modified by network devices on the path, this can change the end-to-end protocol transport protocol behaviour in a way that may have benefits (e.g., to user performance/cost) or may hinder (e.g., disrupting application experience). Whatever the outcome, modification of packets by a middlebox was not

usually intended when the protocol was specified and is usually not known by the sending or receiving endpoints.

Middleboxes have been deployed for a variety of reasons [RFC3234], including protocol enhancement, proxies such as Protocol Enhancing Proxies (PEPs) [RFC3135], TCP acknowledgement (ACK) enhancement [RFC3449], use by application protocol caches [I-D.mm-wg-effect-encrypt], application layer gateways [I-D.mm-wg-effect-encrypt], etc. [I-D.dolson-plus-middlebox-benefits] summarizes some of the functions provided by such middleboxes, and benefits that may arise when used in specific deployment scenarios. Such methods, which involve in-network modification of transport headers, are not further discussed.

Transport protocols can be designed to encrypt or authenticate transport header fields. Authentication methods at the transport layer can detect any changes to an immutable header field that were made by a network device along a path. These methods do not require encryption of the header fields, and hence authenticated fields may remain visible to network devices. A receiving transport endpoint can use an integrity check to avoid accepting modified protocol headers. This document therefore does not consider the case where there is undetected modification of the transport header fields as a packet traverses the network path. The intentional modification of transport headers by middleboxes (such as Network Address Translation with Protocol Translation, NAT-P) is not considered.

Authentication methods (that provide integrity checks of protocols fields) have also been specified at the network layer, and this also protects transport header fields. The network layer itself carries protocol header fields that are increasingly used to help forwarding decisions reflect the need of transport protocols, such the IPv6 Flow Label [RFC6437], the Differentiated Services Code Point (DSCP) [RFC2474] and Explicit Congestion Notification (ECN) [RFC3168].

Encryption methods can hide information from an eavesdropper in the network. Encryption can also help protect the privacy of a user, by hiding data relating to user/device identity or location. Neither an integrity check nor encryption methods prevent traffic analysis, and usage needs to reflect that profiling of users and fingerprinting of behaviour can take place even on encrypted traffic flows.

This document seeks to identify the implications of various approaches to transport protocol authentication and encryption.

2. Internet Transports and Pervasive Encryption

End-to-end encryption can be applied at various protocol layers. It can be applied above the transport to encrypt the transport payload. One motive to use encryption is a response to perceptions that the network has become ossified by over-reliance on middleboxes that prevent new protocols and mechanisms from being deployed. This has led to a common perception that there is too much "manipulation" of protocol headers within the network, and that designing to deploy in such networks is preventing transport evolution. In the light of this, a method that authenticates transport headers may help improve the pace of transport development, by eliminating the need to always consider deployed middleboxes [I-D.trammell-plus-abstract-mech], or potentially to only explicitly enable middlebox use for particular paths with particular middleboxes that are deliberately deployed to realise a useful function for the network and/or users[RFC3135].

Another perspective stems from increased concerns about privacy and surveillance. Some Internet users have valued the ability to protect identity and defend against traffic analysis, and have used methods such as IPsec ESP and Tor [Tor]. Revelations about the use of pervasive surveillance [RFC7624] have, to some extent, eroded trust in the service offered by network operators, and following the Snowden revelation in the USA in 2013 has led to an increased desire for people to employ encryption to avoid unwanted "eavesdropping" on their communications. Whatever the reasons, there are now activities in the IETF to design new protocols that may include some form of transport header encryption (e.g., QUIC [I-D.ietf-quic-transport]).

The use of transport layer authentication and encryption exposes a tussle between middlebox vendors, operators, applications developers and users.

- o On the one hand, future Internet protocols that enable large-scale encryption assist in the restoration of the end-to-end nature of the Internet by returning complex processing to the endpoints, since middleboxes cannot modify what they cannot see.
- o On the other hand, encryption of transport layer header information has implications for people who are responsible for operating networks and researchers and analysts seeking to understand the dynamics of protocols and traffic patterns.

Whatever the motives, a decision to use pervasive of transport header encryption will have implications on the way in which design and evaluation is performed, and which can in turn impact the direction of evolution of the TCP/IP stack.

The next subsections briefly review some security design options for transport protocols.

2.1. Authenticating the Transport Protocol Header

Transport layer header information can be authenticated. An integrity check that protects the immutable transport header fields, but can still expose the transport protocol header information in the clear, allowing in-network devices to observe these fields. An integrity check can not prevent in-network modification, but can avoid accepting changes and avoid impact on the transport protocol operation.

An example transport authentication mechanism is TCP-Authentication (TCP-AO) [RFC5925]. This TCP option authenticates TCP segments, including the IP pseudo header, TCP header, and TCP data. TCP-AO protects the transport layer, preventing attacks from disabling the TCP connection itself. TCP-AO may interact with middleboxes, depending on their behavior [RFC3234].

The IPSec Authentication Header (AH) [RFC4302] works at the network layer and authenticates the IP payload. This therefore also authenticates all transport headers, and verifies their integrity at the receiver, preventing in-network modification.

2.2. Encrypting the Transport Payload

The transport layer payload can be encrypted to protect the content of transport segments. This leaves transport protocol header information in the clear. The integrity of immutable transport header fields could be protected by combining this with an integrity check (Section 2.1).

Examples of encrypting the payload include Transport Layer Security (TLS) over TCP [RFC5246] [RFC7525] or Datagram TLS (DTLS) over UDP [RFC6347] [RFC7525].

2.3. Encrypting the Transport Header

The network layer payload could be encrypted (including the entire transport header and payload). This method does not expose any transport information to devices in the network, which also prevents modification along the network path.

The IPSec Encapsulating Security Payload (ESP) [RFC4303] is an example of encryption at the network layer, it encrypts and authenticates all transport headers, preventing visibility of the

headers by in-network devices. Some Virtual Private Network (VPN) methods also encrypt these headers.

2.4. Authenticating Transport Information and Selectively Encrypting the Transport Header

A transport protocol design can encrypt selected header fields, while also choosing to authenticate fields in the transport header. This allows specific transport header fields to be made observable by network devices. End-to-end integrity checks can prevent an endpoint from undetected modification of the immutable transport headers.

The choice of which fields to expose and which to encrypt is a design choice for the transport protocol. Any selective encryption method requires trading two conflicting goals for a transport protocol designer to decide which header fields to encrypt. On the one hand, security work typically employs a design technique that seeks to expose only what is needed. On the other hand, there may be performance and operational benefits in exposing selected information to network tools.

Mutable fields in the transport header provide opportunities for middleboxes to modify the transport behaviour (e.g., the extended headers described in [I-D.trammell-plus-abstract-mech]). This considers only immutable fields in the transport headers, that is, fields that may be authenticated end-to-end across a path.

An example of a method that encrypts some, but not all, transport information is GRE-in-UDP [RFC8086] when used with GRE encryption.

2.5. Adding Transport Information to Network-Layer Protocol Headers

The transport information can be made visible in a network-layer header. This has the advantage that this information can then be observed by in-network devices. This has the advantage that a single header can support all transport protocols, but there may also be less desirable implications of separating the operation of the transport protocol from the measurement framework.

Some measurements may be made by adding additional protocol headers carrying operations, administration and management (OAM) information to packets at the ingress to a maintenance domain (e.g., an Ethernet protocol header with timestamps and sequence number information using a method such as 802.1lag) and removing the additional header at the egress of the maintenance domain. This approach enables some types of measurements, but does not cover the entire range of measurements described in this document.

Another example of a network-layer approach is the IPv6 Performance and Diagnostic Metrics (PDM) Destination Option [I-D.ietf-ippm-6man-pdm-option]. This allows a sender to optionally include a destination option that carries header fields that can be used to observe timestamps and packet sequence numbers. This information could be authenticated by receiving transport endpoints when the information is added at the sender and visible at the receiving endpoint, although methods to do this have not currently been proposed. This method needs to be explicitly enabled at the sender.

A drawback of using extension headers is that IPv4 network options are often not supported (or are carried on a slower processing path) and some IPv6 networks are also known to drop packets that set an IPv6 header extension. Another disadvantage is that protocols that separately expose header information do not necessarily have an advantage to expose the information that is utilised by the protocol itself, and could manipulate this header information to gain an advantage from the network.

3. Use of Transport Headers in the Network

This section identifies ways that actors can benefit by observing (non-encrypted) transport header fields at devices in the network. The list of actors who perform measurements include:

- o Protocol developers and implementors of TCP/IP stacks;
- o Researchers working on new mechanisms;
- o Use of new applications using existing applications;
- o Analysis researching the impact of mechanisms on network equipment or specific network topologies;
- o Staff supporting operation of a network.

One approach is to use active measurement using dedicated tools to generate and measure test traffic. To test a transport path, such active tools need to be run from an endpoint, and most operators do not have access to user equipment. There may also be costs associated with running such tests (e.g., the implications of bandwidth tests in a mobile network are obvious). Some active measurements (e.g., response under load or particular workloads) may perturb other traffic, and could require dedicated access to the network segment. An alternative approach is to use in-network techniques that observe transport packet headers in operational networks to make the measurements.

Transport layer information can help identify whether the link/network tuning is effective and alert to potential problems that can be hard to derive from link or device measurements alone. The design

trade offs for radio networks are often very different to those of wired networks. A radio-based network (e.g., cellular mobile, enterprise WiFi, satellite access/backhaul, point-to-point radio) has the complexity of a subsystem that performs radio resource management - with direct impact on the available capacity, and potentially loss/reordering of packets. The impact of the pattern of loss and congestion, differs for different traffic types, correlation with propagation and interference can all have significant impact on the cost and performance of a provided service. The need for this type of information is expected to increase as operators bring together heterogeneous types of network equipment and seek to deploy opportunistic methods to access radio spectrum.

In-network observation of transport protocol headers requires knowledge of the format of the transport header:

- o Flows, need to be identified at the level required for monitoring;
- o The protocol and version of the header that is being used. As protocols evolve over time and there may be a need to introduce new transport headers. This may require interpretation of protocol version information or connection setup information;
- o The position and syntax of any transport headers that need to be observed. IETF transport protocols specify this information.

The following subsections describe various ways that observable transport information may be utilised.

3.1. Use to Identify Flows and Packet Formats

Transport protocol header information can identify a flow and the connection state of the flow, together with the protocol options being used. In some usages, a low-numbered (well-known) port that can identify a protocol (although port information alone is not sufficient to guarantee identification of a protocol). Transport protocols, such as TCP and SCTP specify a standard base header that includes sequence number information and other data, with the possibility to negotiate additional headers at connection setup and identified by an option number in the transport header. UDP-based protocols sometimes do not use well-known ports but also can instead be identified by signalling protocols or through the use of magic numbers placed in the first byte(s) of the datagram payload.

3.2. Measurements derived from Transport Header Information

Some actors have a need to characterise the performance of link/network segments. Passive monitoring uses observed traffic to make inferences from transport headers to derive measurements. A variety of open source and commercial tools can utilise this information.

Transport fields in the Real Time Protocol (RTP) header[RFC3550] [RFC4585] can be observed to derive traffic volume measurements and provide information on the progress and quality of a session using RTP. Key performance indicators are retransmission rate, packet drop rate, sector utilization level, a measure of reordering, peak rate, the CE-marking rate, etc. Metadata is often important to understand the context under which the data was collected, including the time, observation point, and way in which metrics were accumulated.

Some Internet transports report summary performance data that is observable in the network (e.g., RTCP feedback[RFC3550]). A user of summary measurement data needs to trust the source of this data and the method used to generate the summary information.

When encryption conceals information in packet headers, measurements need to rely on pattern inferences and other heuristics grows, and accuracy suffers [I-D.mmm-wg-effect-encrypt].

3.2.1. Use to Characterise Traffic Rate and Volume

Transport headers may be observed to derive volume measures per-application, to characterise the traffic using a network segment and pattern of network usage. This may be measured per endpoint or aggregate of endpoint (e.g., by an operator to assess subscriber usage). This can also be used to trigger measurement-based traffic shaping and to implement QoS support within the network and lower layers. Volume measures can be valuable for capacity planning (providing detail of trends rather than the volume per subscriber).

3.2.2. Measuring Loss Rate and Loss Pattern

Flow loss rate is often used as a metric for performance assessment and to characterise the transport behaviour. Understanding the root cause of loss can help an operator determine whether this requires corrective action.

There are various cause of loss, including: corruption on a link (e.g., interference on a radio link), buffer overflow (e.g., due to congestion), policing (traffic management), buffer management (e.g., Active Queue Management (AQM)). Loss can be monitored at the interface level by devices in the network. It is often important to understand the conditions under which packet loss occurs, which usually means relating loss to the traffic flowing on the network segment at the time of loss. Understanding flow loss rate requires either maintaining per flow packet counters or by observing sequence numbers in transport headers.

Observation of transport feedback information (observing loss reports, e.g., RTCP, TCP SACK) can increase understanding of the impact of loss and help identify cases where loss may have been wrongly identified, or the transport did not require the lost packet. It is sometimes more important to understand the pattern of loss, than the loss rate - since losses can often occur as bursts, rather than randomly timed events.

3.2.3. Measuring Throughput and Goodput

The throughput observed by a flow can be determined even when a flow is encrypted, providing the individual flow can be identified. Goodput [RFC7928] is a measure of useful data exchanged (the ratio of useful/total volume of traffic sent by a flow), which requires ability to differentiate loss and retransmission of packets (e.g., by observing packet sequence numbers in TCP or RTP).

3.2.4. Measuring Latency (Network Transit Delay and Jitter)

Latency is a key performance metric that impacts application response time and user-perceived response time. It also often indirectly impacts throughput and flow completion time. Latency determines the reaction time of the transport protocol itself, impacting flow setup, congestion control, loss recovery, and other transport mechanisms. The observed latency can have many components [Latency]. Of these, unnecessary/unwanted queuing in network buffers has often been observed as a significant factor. Once the cause of unwanted latency has been identified, this can often be eliminated, and determining latency metrics is a key driver in the deployment of AQM [RFC7567], DiffServ [RFC2474], and ECN [RFC3168] [RFC8087].

To measure latency across a part of the path, an observation point can measure the experienced round trip time (RTT) using packet sequence numbers, and acknowledgements, or by observing header timestamp information. Such information allows an observation point in the network to determine not only the path RTT, but also to measure the upstream and downstream contribution to the RTT. This may be used to locate a source of latency, e.g., by observing cases where the ratio of median to minimum RTT is large for a part of a path.

An example usage of this method could be to identify excessive buffers and to deploy or configure Active Queue Management (AQM) [RFC7567] [RFC7928]. Operators deploying such tools can effectively eliminate unnecessary queuing in routers and other devices. AQM methods need to be deployed at the capacity bottleneck, but are often deployed in combination with other techniques, such as scheduling [RFC7567] [I-D.ietf-aqm-fq-codel] and although parameter-less methods

are desired [RFC7567], current methods [I-D.ietf-aqm-fq-codel] [I-D.ietf-aqm-codel] [I-D.ietf-aqm-pie] often cannot scale across all possible deployment scenarios. The service offered by operators can therefore benefit from latency information to understand the impact of deployment and tune deployed services.

Some network applications are sensitive to packet jitter, and it can be necessary to measure the jitter observed along a portion of the path. The requirements to measure jitter resemble those for the measurement of latency.

3.2.5. Measuring Flow Reordering

Significant flow reordering can impact time-critical applications and can be interpreted as loss by reliable transports. Many transport protocol techniques are impacted by reordering (e.g., triggering TCP retransmission, or rebuffering of real-time applications). Packet reordering can occur for many reasons (from equipment design to misconfiguration of forwarding rules).

As in the drive to reduce network latency, there is a need for operational tools to be able to detect misordered packet flows and quantify the degree of reordering. Techniques for measuring reordering typically observe packet sequence numbers. Metrics have been defined that evaluate whether a network has maintained packet order on a packet-by-packet basis [RFC4737] and [RFC5236].

There has been initiatives in the IETF transport area to reduce the impact of reordering within a transport flow, possibly leading to reduced the requirements for ordering. These have promise to simplify network equipment design as well as the potential to improve robustness of the transport service. Measurements of reordering can help understand the level of reordering within deployed infrastructure, and inform decisions about how to progress such mechanisms.

3.3. Measurements derived from Network-Transport Information

This section describes transport information that is already observable in network-layer header fields.

3.3.1. Use of IPv6 Network-Layer Flow Label

Endpoints should expose flow information in the IPv6 Flow Label field of the network-layer header (e.g. [RFC8085]). This can be used to inform network-layer queuing, forwarding (e.g., for equal cost multi-path (ECMP) routing, and Link Aggregation (LAG)). This can provide useful information to assign packets to flows in the data collected

by measurement campaigns. Although important to characterising a path, it does not directly provide any performance data.

3.3.2. Use Network-Layer Differentiated Services Code Point Point

Application can expose their delivery expectations to the network, by setting the Differentiated Services Code Point (DSCP) field of IPv4 and IPv6 packets. This can be used to inform network-layer queuing and forwarding, and can also provide information on the relative importance of packet information collected by measurement campaigns, but does not directly provide any performance data.

This field provides explicit information that can be used in place of inferring traffic requirements (e.g., by inferring QoS requirements from port information via a multi-field classifier). The DSCP value can therefore impact the quality of experience for a flow. Observations of service performance need to consider this field when a network path has support for differentiated service treatment.

3.3.3. Use of Explicit Congestion Marking

Explicit Congestion Notification (ECN)[RFC3168] uses a codepoint in the network-layer header. Use of ECN can offer gains in terms of increased throughput, reduced delay, and other benefits when used over a path that includes equipment that supports an AQM method that performs Congestion Experienced (CE) marking of IP packets [RFC8087].

This exposes the presence of congestion on a network path to the transport and network layer. The reception of Congestion Experienced (CE) marked packets can therefore be used to monitor the presence and estimate the level of incipient congestion on the upstream portion of the path from the point of observation (Section 2.5 of [RFC8087]). Because ECN marks carried in the IP protocol header, measuring ECN can be much easier than metering packet loss. However, interpreting the marking behaviour (i.e., assessing congestion and diagnosing faults) requires context from the transport layer (path RTT, visibility of loss - that could be due to queue overflow, congestion response, etc)[RFC7567].

Some ECN-capable network devices can provide richer (more frequent and fine-grained) indication of their congestion state. Setting congestion marks proportional to the level of congestion (e.g., DCTP [I-D.ietf-tcpm-dctcp], and L4S [I-D.ietf-tsvwg-l4s-arch]).

AQM and ECN offer a range of algorithms and configuration options, it is therefore important for tools to be available to network operators and researchers to understand the implication of configuration choices and transport behaviour as use of ECN increases and new

methods emerge [RFC7567] [RFC8087]. ECN-monitoring is expected to become important as AQM is deployed that supports ECN [RFC8087]

Section 5.6 describes the transport layer feedback information that accompanies the use of ECN.

4. Transport Measurement

The common language between network operators and application/content providers/users is packet transfer performance at a layer that all can view and analyze. For most packets, this has been transport layer, until the emergence of QUIC, with the obvious exception of VPNs and IPsec. When encryption conceals more layers in a packet, people seeking understanding of the network operation need to rely more on pattern inferences and other heuristics. The accuracy of measurements therefore suffers, as does the ability to investigate and troubleshoot interactions between different anomalies. For example, the traffic patterns between server and browser are dependent on browser supplier and version, even when the sessions use the same server application (e.g., web e-mail access). Even when measurement datasets are made available (e.g., from endpoints) additional metadata, such as the state of the network, is often required to interpret the data. Collecting and coordinating such metadata is more difficult when the observation point is at a different location to the bottleneck/device under evaluation.

Packet sampling techniques can be used to scale processing involved in observing packets on high rate links. This only exports the packet header information of (randomly) selected packets. The utility of these measurements depends on the type of bearer and number of mechanisms used by network devices. Simple routers are relatively easy to manage, a device with more complexity demands understanding of the choice of many system parameters. This level of complexity exists when several network methods are combined.

This section discusses topics concerning transport measurement.

4.1. Point of Measurement

Often measurements can only be understood in the context of the other flows that share a bottleneck. A simple example is the monitoring of AQM. For example, FQ-CODEL [I-D.ietf-aqm-fq-codel], combines sub queues (statistically assigned per flow), management of the queue length (CODEL), flow-scheduling, and a starvation prevention mechanism. Usually such algorithms are designed to be self-tuning, but current methods typically employ heuristics that can result in more loss under certain path conditions (e.g., large RTT, effects of multiple bottlenecks [RFC7567]).

In-network measurements that can distinguish between upstream and downstream metrics with respect to the measurement point. They are particularly useful for locating the source of problems or to assess the performance of a network segment or a particular device configuration.

4.2. Use by Operators to Plan and Provision Networks

Traffic measurements (e.g. Traffic volume, loss, latency) is used by operators to help plan deployment of new equipment and configurations in their networks. Data is also important to equipment vendors who need to understand traffic trends traffic and patterns of usage as inputs to decisions about planning products and provisioning for new deployments. This measurement information can also be correlated with billing information when this is also collected by an operator.

A network operator supporting traffic that uses transport header encryption may not have access to per-flow measurement data. Trends in aggregate traffic can be observed and can be related this to the endpoint addresses being used, but it may not be possible to correlate patterns in measurements with changes in transport protocols (e.g., the impact of changes in introducing a new transport protocol mechanism). This increases the dependency on other indirect sources of information to inform planning and provisioning.

4.3. Service Performance Measurement

Traffic measurements (e.g., traffic volume, loss, latency) can be used by various actors to help understand the performance available to users of a network segment. While active measurements may be used in-network passive measurements can have advantages in terms of eliminating unproductive traffic, reducing the influence of test traffic on the overall traffic mix, and the ability to choose the point of measurement Section 4.1.

4.4. Use for Network Diagnostics and Troubleshooting

Transport header information is useful for a variety of operational tasks [I-D.mm-wg-effect-encrypt]: to diagnose network problems, assess performance, capacity planning, management of denial of service threats, and responding to user performance questions. These tasks seldom involve the need to determine the contents of the transport payload, or other application details.

A network operator supporting traffic that uses transport header encryption can see only encrypted transport headers. This prevents deployment of performance measurement tools that rely on transport protocol information. Choosing to encrypt all information may be

expected to reduce the ability for networks to "help" (e.g., in response to tracing issues, making appropriate Quality of Service, QoS, decisions). For some this will be blessing, for others it may be a curse. For example, operational performance data about encrypted flows needs to be determined by traffic pattern analysis, rather than relying on traditional tools. This can impact the ability of the operator to respond to faults, it could require reliance on endpoint diagnostic tools or user involvement in diagnosing and troubleshooting unusual use cases or non-trivial problems. Although many network operators utilise transport information as a part of their operational practice, the network will not break because transport headers are encrypted.

4.5. Acceptable Response to Congestion

Congestion control is a key transport function. Many network operators implicitly accept that TCP traffic to comply with a behaviour that is acceptable for use in the shared Internet. TCP algorithms have been continuously improved over decades, and they have reached a level of efficiency and correctness that custom application-layer mechanisms will struggle to easily duplicate [RFC8085]. A standards-compliant TCP stack provides congestion control that is therefore judged safe for use across the Internet. Applications developed on top of well-designed transports can be expected to appropriately control their network usage, reacting when the network experiences congestion, by back-off and reduce the load placed on the network. This is the normal expected behaviour for TCP and other IETF-defined transports.

Tools exist that can interpret the transport protocol header information to help understand the impact of specific transport protocols (or protocol mechanisms) on other traffic that shares their network. An observation in the network can gain understanding of the dynamics of a flow and its congestion control behaviour. Analysing observed packet sequence numbers can be used to help build confidence that an application flow backs-off its share of the network load in the face of persistent congestion, and hence to understand whether the behaviour is appropriate for sharing limited network capacity. For example, it is common to visualise plots of TCP sequence numbers versus time for a flow to understand how a flow shares available capacity, deduce its dynamics in response to congestion, etc.

4.5.1. Measuring Compliance of UDP Traffic

UDP provides a minimal message-passing transport that has no inherent congestion control mechanisms. Because congestion control is critical to the stable operation of the Internet, applications and other protocols that choose to use UDP as an Internet transport must

employ mechanisms to prevent congestion collapse, avoid unacceptable contributions to jitter/latency, and to establish an acceptable share of capacity with concurrent traffic [RFC8085].

A network operator has no way of knowing the specific methods used by a UDP application, unless the header format can be determined. Tools are needed to understand if UDP flows comply with congestion control expectations and therefore whether there is a need to deploy methods such as rate-limiters, transport circuit breakers or other methods to enforce acceptable usage. UDP flows that expose a well-known header by specifying the format of header fields can allow information to be observed that gains understanding of the dynamics of a flow and its congestion control behaviour. For example, tools exist to monitor various aspects of the RTP and RTCP header information of real-time flows (see Section 3.2).

4.5.2. Measuring Transport to Support Network Operations

By correlating observations at multiple points along the path (e.g., at the ingress and egress of a network segment), an observer can determine the contribution of a portion of the path to an observed metric (to locate a source of delay, jitter, loss, reordering, congestion marking, etc).

Information provided by tools can help determine whether mechanisms are needed in the network to prevent flows from acquiring excessive network capacity. Operators can manage traffic flows (e.g., to prevent flows from acquiring excessive network capacity under severe congestion) by deploying rate-limiters, traffic shaping or network transport circuit breakers [RFC8084].

5. Observing Transport Flows with Encrypted Transport Header Fields

This section examines implications of encrypting specific transport header information.

5.1. Transport Information at the Network Layer

Some transport information is made visible in the network-layer protocol header. These header fields are not encrypted and can be used to make flow observations. Endpoints should expose flow information in the IPv6 Flow Label Section 3.3.1 in the network-layer header. This can be used to inform network-layer queuing, forwarding (e.g., for equal cost multi-path (ECMP) routing, and Link Aggregation (LAG)). For transport measurement, this can provide useful information to assign packets to flows in the data collected by measurement campaigns, but does not directly provide any performance data. Similarly the Differentiated Services CodePoint (DSCP)

indicates expected forwarding treatment Section 3.3.2. The ECN field provides observable congestion data and can help inform measurement of flow congestion Section 3.3.3.

5.2. An Observable Transport Flow Identifier

To measure and analyse a transport protocol, a measurement tool needs to be able to identify traffic flows. Aggregation of sessions, and persistent use of established transport flows by multiple sessions means that a flow at the transport layer is not necessarily the same as a flow seen at the application layer. This is usually not a consequence. Data is measured for the aggregate transport flow.

Some measurement methods sample traffic, rather than collecting all packets passing through a measurement point. These methods still require a way to determine the presence, size and position of any observable header fields - but may need to do this without observing a protocol exchange for a connection setup.

5.2.1. A Method to Determine Header Format

If flow information is observed from transport headers, then there needs to be a way to identify the format of the header Section 3.1. Some IETF transport protocols are identified by an IP protocol number (e.g., TCP, SCTP, UDP). All IETF-defined transport protocols include a transport port field in their transport header. Higher layer protocols (e.g., HTTP) can be sometimes be observed by a well-known port value, which can be indicative of the protocol being encapsulated, but there is no way to enforce this usage. This can be used to configure decapsulation, alternatives include a "magic" number placed at the start of each UDP datagram.

Once the protocol has been determined, the transport header can be determined from a published specification. If multiple formats are permitted, this may also require observing the protocol version being used and possibly parameter negotiation at connection setup.

5.2.2. Use of a Transport as a Substrate

When a transport is used as a substrate, the transport provides an encapsulation that allows another transport flow to be within the payload of a transport flow. The transported protocol header may provide additional information for multiplexing multiple flows over the same 5-tuple. The UDP Guidelines [RFC8085] provides some guidance on using UDP as a substrate protocol. If there is no additional information about the protocol transported by the substrate, this may be viewed as an opaque traffic aggregate, and prevents transport measurement in the network. Examples include GRE-

in-UDP [RFC8086], SCTP-in-UDP. The GRE-in-UDP encapsulation may encrypt the payload, but does not encrypt the GRE protocol header.

5.2.3. Support for Mobility and Flow Migration

With the proliferation of mobile connected devices, there is a stated need for connection-oriented protocols to maintain connections after a network migration by an endpoint. The ability and desirability of in-network devices to track such migration depends on the context. On the one hand, a load-balancer device in front of server may find it useful to map a migrated connection to the same server endpoint. On the other hand, a user performing migration to avoid detection may prefer the network not to be able to correlate the different parts of a migrating session. Care must then be exercised to make sure that the information encoded by the endpoints is not sufficient to identify unique flows and facilitate a persistent surveillance attack vector [I-D.mm-wg-effect-encrypt].

The impact of flow migration on measurement activities depends on the data being measured, rate of migration and level of encryption that is employed. Requirements for load balancing and mobility can lead to complex protocol interactions.

5.2.4. Flow Start and Stop

Transports can expose that start and end of flows in a transport header field (e.g., TCP SYN, FIN, RST). This can also help measurement devices identify the start of flows, or to remove stale flow information. This information is supplemental - flows can start and end at any time, the Internet network layer provides only a best effort service that allows alternate routing, reordering, loss, etc, so a network measurement tool can not rely upon observing these indicators. The time to complete a protocol connection and/or session setup can be reported.

Flow information can provide in-network devices to manage their forwarding state [I-D.trammell-plus-statefulness]. It can assist a firewall in deciding which flows are permitted through a security gateway, or to help maintain the network address translation (NAT) bindings in a NAT or application layer gateway. This information may also find use in load balancers, where visibility of the 5-tuple could assist in selecting a server [I-D.mm-wg-effect-encrypt].

Access to flow information and an observable start/stop indication [I-D.trammell-plus-statefulness] can avoid stateful middleboxes relying on timeouts to remove old state. Without this, middleboxes are unaware when a particular flow ceases to be used by an

application[RFC8085]. This can lead to the state table entries keeping state for less time for flows that are not identifiable.

5.3. Observable Transport Sequence Number

The TCP or RTP sequence number can be observed in one direction (the direction that carries data segments). An authenticated header prevents this field being modified or terminated/split [RFC3135] by a network device, but allows this still to be used to observe progress of the network flow.

An incrementing sequence number enables detection of loss (either by correlating ingress and egress value, or when assuming that all packets follow a single path), duplication and reordering (with understanding that not necessarily all packets of a flow follow the same path, and reordering can complicate processing of observations). Tools are widely available to interpret RTP and TCP sequence numbers, ranging from open source tools to dedicated commercial packages. As for TCP, use by in-network measurement devices needs to account for the impact of load-balancing of flows, changes in forwarding behaviour, measurement loss (rather than observed packet loss), etc.

5.4. Observable Transport Reception

Acknowledgement (ACK) data provides information about the path from the network device to the remote endpoint. The information can help identify packet loss (or the point of loss), RTT, and other network-related performance parameters (e.g., throughput, jitter, reordering). Unless this information is correlated with other data there is no way to disambiguate the cause of impairments (congestion loss, link transmission loss, equipment failure).

An in-network device must not modify the flow of end-to-end ACK data when using an authenticated protocol. That is, must not use the in-network methods described in [RFC3449]. This can impact the performance and/or efficiency (e.g., cost) of using paths where the return capacity is limited or has implications on the overall design (e.g., using TCP with cellular mobile uplinks, DOCSIS uplinks).

The TCP stream can be observed by correlating the stream of TCP ACKs that flow from a receiver in the return direction. Although these ACKs are cumulative, and are not necessarily sent on the same path as the forward data, when visible, their sequence can confirm successful transmission and the path RTT. In the case of TCP they may also indicate packet loss (duplicate ACKs).

An RTP session can provide reception information [RFC3550] [RFC4585] feedback using the RTCP framework. This reception information and

can be observed by in-network measurement devices and can be interpreted to provide a variety of quality of experience information for the related RTP flow, as well as basic network performance data (RTT, loss, jitter, etc).

5.5. Observable Transport Timestamps

The use of timestamps for latency and jitter measurements Section 3.2.4 is discussed in other sections of the current version of the document.

5.6. Observable ECN Transport Feedback Information

Transport protocols that use ECN Section 3.3.3 need to provide ECN feedback information in the transport header to inform the sender whether packets have been received with an ECN CE-mark [RFC3819]. This information can be in the form of feedback once each RTT [RFC3819] or more frequent. The latter may involve sending a detailed list of all ECN-marked packets (e.g., [I-D.ietf-tcpm-accurate-ecn] and [RFC6679]). The detailed information can provide detail about the pattern and rate of marking. The information provided in these protocol headers can help a network operator to understand the congestion status of the forward path and the impact of marking algorithms on the traffic that is carried [RFC8087].

IETF specifications for Congestion Exposure (CONVEX) [RFC7713] is an example of a framework that monitors reception reports for CE-marked packets to support network operations.

5.7. Other Observable Transport Fields

This section is not complete - later revision may determine other fields or remove this section.

5.8. Interpretation of Transport Header Fields

Understanding and analysing transport protocol behaviour typically demands tracking changes to the protocol state at the transport endpoints. Although protocols communicate state information in their protocol headers, a protocol implementation typically also contains internal state that is not directly visible from observing transport protocol headers. Effective measurement tools need to consider that not all packets may be observed (due to drops at the capture tap or because packets take an alternate route that does not pass the tap). Some flows of packets may also be encapsulated within a maintenance domain in other protocols, which further complicates analysis.

Some examples of using network measurements of transport headers to infer internal TCP transport state information include:

- o The TCP congestion window (cwnd) and slow start threshold (ssthresh). Tools for analysing in-network performance of TCP may observe sequence number to infer the current congestion controller state.
- o The TCP RTT estimator and TCP Retransmission Time Out (RTO) value. This can be estimated by correlating sequence and acknowledgement numbers, or possibly by observing TCP timestamp options.
- o Use of pacing (and pacing rate) and use of methods such as Proportional Rate Reduction (PRR) and Congestion Window validation (CWV). This may be estimated from observing timing of segments with TCP sequence numbers. This is important to some congestion control mechanisms and can be important for applications that are rate limited or send traffic bursts.
- o Receiver window and flow control state. This may be inferred from information in TCP ACK segments. It is important to applications where the remote endpoint is resource constrained, or the path exhibits a large RTT.
- o Retransmission state and receiver buffer. This may be inferred from information in TCP ACK segments (especially when SACK blocks are provided), this can be important to the performance of applications that send traffic bursts.
- o Use of ACK delay and Nagle algorithm. This may be estimated from observing timing of segments with TCP sequence numbers, and is important to the performance of thin application flows.

5.9. Requirements for Transport Measurement

Transport measurement and analysis of the impact of network characteristics have been important to the design of current transport protocols. Transport measurement introduces the following requirements to identify the observable information:

- o Observable protocol type and version information is needed to identify the protocol being used when characterising the traffic, and to enable further observation of the flow.
- o Observable format information is needed to allow an observer to determine the presence of any observable header fields.
- o A published specification is needed to allow an observer to determine the size and position of any observable header fields so that these fields may be decoded by a measurement tool.
- o Observable flow start/stop information can assist some forms of measurement and has utility for middleboxes that track state.

The need for in-network transport measurement introduces the following requirements for observable information in transport header fields:

- o Observable transport information to determine the progress of flows for each direction of communication. This requires observable packet numbers.
- o Observable transport information to determine loss, and understand the response to congestion for a network segment. This requires observable reception information (e.g., packet acknowledgment information).
- o Observable transport information is needed for more advanced measurement of latency, jitter, etc. This requires an observable field and a method to correlating return information with the observed field. This could utilise a packet number and/or transmission timestamp information. This information needs to be available in both directions of transmission.
- o Exposure of Transport ECN feedback provides a powerful tool to understand ECN-enabled AQM-based networks. (Forward ECN information is already observable in the network header).

6. The Effect of Encrypting Transport Header Fields

This section explores key implications of working with encrypted transport protocols.

6.1. Independent Measurement

Independent observation by multiple actors is important for scientific analysis. Encrypting transport header encryption changes the ability for other actors to collect and independently analyse data. Internet transport protocols employ a set of mechanisms. Some of these need to work in cooperation with the network layer - loss detection and recovery, congestion detection and congestion control, some of these need to work only end-to-end (e.g., parameter negotiation, flow-control).

When encryption conceals information in the transport header, it could be possible for an applications to provide summary data on performance and usage of the network. This data could be made available to other actors. However, this data needs to contain sufficient detail to understand (and possibly reconstruct the network traffic pattern for further testing) and to be correlated with the configuration of the network paths being measured. Sharing information between actors needs also to consider the privacy of the user and the incentives for providing accurate and detailed information. Protocols that expose the state information used by the transport protocol in their header information (e.g., timestamps used

to calculate RTT, packet numbers used to assess congestion and requests for retransmission) provide an incentive for the sending endpoint to provide correct information, increasing confidence that the observer understands the transport interaction with the network. This becomes important when considering changes to transport protocols, changes in network infrastructure, or the emergence of new traffic patterns.

6.2. Characterising "Unknown" Network Traffic

If "unknown" or "uncharacterised" traffic patterns form a small part of the traffic aggregate passing through a network device or segment of the network the path, the dynamics of the uncharacterised traffic may not have a significant collateral impact on the performance of other traffic that shares this network segment. Once the proportion of this traffic increases, the need to monitor the traffic and determine if appropriate safety measures need to be put in place.

Tracking the impact of new mechanisms and protocols requires traffic volume to be measured and new transport behaviours to be identified. This is especially true of protocols operating over a UDP substrate. The level and style of encryption needs to be considered in determining how this activity is performed. On a shorter timescale, information may also need to be collected to manage denial of service attacks against the infrastructure.

6.3. Accountability and Internet Transport Portocols

Attention therefore needs to be paid to the expected scale of deployment of new protocols and protocol mechanisms. Whatever the mechanism, experience has shown that it is often difficult to correctly implement combination of mechanisms [RFC8085]. These mechanisms therefore typically evolve as a protocol matures, or in response to changes in network conditions, changes in network traffic or changes to application usage.

The growth and diversity of applications and protocols using the Internet continues to expand - and there has been recent interest in a wide range of new transport methods, e.g., Larger Initial Window, Proportional Rate Reduction (PRR), congestion control methods based on measuring bottleneck bandwidth and round-trip propagation time, the introduction of AQM techniques and new forms of ECN response (e.g., Data Centre TCP, DCTP [I-D.ietf-tcpm-dctcp], and methods proposed for Low Latency Low Loss Scalable throughput, L4S). For each new method it is desirable to build a body of data reflecting its behaviour under a wide range of deployment scenarios, traffic load, and interactions with other deployed/candidate methods.

Measurement therefore has a critical role in the design of transport protocol mechanisms and their acceptance by the wider community (e.g., as a method to judge the safety for Internet deployment. Open standards suggest that such evaluation needs to include independent observation and evaluation of performance data.

7. Implications on Evolution of the Internet Transport

The transport layer provides the first end-to-end interactions across the Internet. Transport protocols are layered directly over the network service and are sent in the payload of network-layer packets. However, this simple architectural view hides one of the core functions of the transport - to discover and adapt to the properties of the Internet path that is currently being used. The design of Internet transport protocols is as much about trying to avoid the unwanted side effects of congestion on a flow and other capacity-sharing flows, avoiding congestion collapse, adapting to changes in the path characteristics, etc., as it is about end-to-end feature negotiation, flow control and optimising for performance of a specific application.

To achieve stable Internet operations the IETF transport community, has to date, relied heavily on measurement and insight provided from the wider community to understand the trade-offs and to inform selection of select appropriate mechanisms to ensure a safe, reliable and robust Internet since the 1990's.

There are many motivations for deploying encrypted transports, and encryption of transport payloads. The increasing public concerns about the interference with Internet traffic have led to a rapidly expanding deployment of encryption to protect end-user privacy, in protocols like QUIC. At the same time, network operators and access providers, especially in mobile networks, have come to rely on the in-network functionality provided by middleboxes both to enhance performance and support network operations.

This document has expanded upon the expected implications on operational practices when working with encrypted transport protocols, and offers insight into the potential benefit of authentication, encryption and techniques that require in-network devices to interpret specific protocol header fields. It presents a need for architectural changes and consideration of approaches to the way network transport protocols are designed when using encryption[Measure].

The use of encryption at the transport layer comes with implications that need to be considered:

Troubleshooting and diagnostics: Encrypting all transport information eliminates the incentive for operators to troubleshoot what they cannot interpret: one flow experiencing packet loss looks like any other. When transport header encryption prevents decoding the transport header (if sequence numbers and flow ID are obscured), and hence understanding the impact on a particular flow or flows that share a common network segment. Encrypted traffic therefore implies "don't touch", and a likely first response will be "can't help, no trouble found", or the need to add complexity that comes with an additional operational cost [I-D.mm-wg-effect-encrypt].

Open verifiable data: The use of transport header encryption may reduce the range of actors who can capture useful measurement data. This may in future restrict the information sources available to the Internet community to understand the operation of the network and transport protocols, necessary to inform standardisation and design decisions for new protocols, equipment and operational practices. There are dangers in a model where transport information is only observable at endpoints: i.e., at user devices and within service platforms and a need for independently captured data to develop open standards and stimulate research into new methods.

Operational practice: Published transport specifications allow operators to check compliance. This can bring assurance to those operating networks, often avoiding the need to deploy complex techniques that routinely monitor and manage TCP/IP traffic flows (e.g. Avoiding the capital and operational costs of deploying flow rate-limiting and network circuit-breaker methods). This should continue when encrypted transport headers are used, but methods need to confirm that the traffic produced conforms to the expectations of the operator or developer.

Traffic analysis: The use of encryption could make it harder to determine which transport methods are being used across a network segment and the trends in usage. This could impact the ability for an operator to anticipate the need for network upgrades and roll-out. It can also impact on-going traffic engineering activities. Although the impact in many case may be small, there are cases where operators directly support services (e.g., in radio links, or to troubleshoot QoS-related issues). The more complex the underlying infrastructure the more important this impact.

Interactions between mechanisms: An appropriate vantage point, coupled with timing information for the flow (fine-grained timestamps) is a valuable tool in benchmarking equipment/configurations and understanding non-trivial interactions. Encryption restricts the ability to explore interactions between functions at different protocol layers. This is a side-effect of not allowing a choice of the vantage point from which this information is observed. This can be important (e.g., in examining collateral impact of flows sharing a bottleneck, or where the intention is to understand the interaction between a layer 2 function (e.g., radio resource management policy, a channel impairment, an AQM configuration, a Per Hop Behaviour (PHB) or scheduling method, and a transport protocol).

Common specifications: Since the introduction of congestion control, TCP has continued to be the predominate transport, with a consistent approach to avoiding congestion collapse. There is a risk that the diversity of transport mechanisms could also increase, with incentives to use a wide range of methods, this is not in itself a problem, nor is this a direct result of encryption. Encryption of all headers places the onus on validation in the hands of developers. While there is little to doubt that developers will seek to produce high quality code for their target use, it is not clear whether there is sufficient incentive to ensure good practice that benefits the wide diversity of requirements from the Internet community as a whole. The use of encryption needs to be weighed against the reduced visibility of the interactions between traffic, the network and the mechanisms. Especially, if a development cycle could focus on specific protocols/applications and then offer incentives for optimisations that could prove suboptimal for users or operators that utilise a network segments with different characteristics than targeted by the developer.

Restricting research and development: The use of encryption may impede independent research into new mechanisms, measurement of behaviour, and development initiatives. Experience shows that transport protocols are complicated to design and complex to deploy, and that individual mechanisms need to be evaluated while considering other mechanism, across a broad range of network topologies and with attention to the impact on traffic sharing the capacity. Adopting pervasive encryption of transport information could eliminate the independent self-checks that have previously been in place from research and academic contributors (e.g., the role of the IRTF ICCRG, and research publications in reviewing new

transport mechanisms and assessing the impact of their experimental deployment).

Pervasive use of transport header encryption can impact the ways that future protocols are designed and deployed. The choice of whether candidate transport designs should encrypt their protocol headers therefore needs to be taken based not just on security considerations, but also on the impact on operating networks and the constrictions this may place on evolution of Internet protocols. While encryption of all transport information can help reduce ossification of the transport layer, it could result in ossification of the network service. There can be advantages in providing a level of ossification of the header in terms of providing a set of open specified header fields that are observable from in-network devices.

8. Acknowledgements

The author would like to thank all who have talked to him face-to-face or via email. ...

This work has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 688421. The opinions expressed and arguments employed reflect only the authors' view. The European Commission is not responsible for any use that may be made of that information.

9. IANA Considerations

XX RFC ED - PLEASE REMOVE THIS SECTION XXX

This memo includes no request to IANA.

10. Security Considerations

This document is about design and deployment considerations for transport protocols. Authentication, confidentiality protection, and integrity protection are identified as Transport Features by RFC8095". As currently deployed in the Internet, these features are generally provided by a protocol or layer on top of the transport protocol; no current full-featured standards-track transport protocol provides these features on its own. Therefore, these features are not considered in this document, with the exception of native authentication capabilities of TCP and SCTP for which the security considerations in RFC4895.

Like congestion control mechanisms, security mechanisms are difficult to design and implement correctly. It is hence recommended that

applications employ well-known standard security mechanisms such as DTLS, TLS or IPsec, rather than inventing their own.

11. References

11.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.

11.2. Informative References

- [I-D.dolson-plus-middlebox-benefits]
Dolson, D., Snellman, J., Boucadair, M., and C. Jacquenet, "Beneficial Functions of Middleboxes", draft-dolson-plus-middlebox-benefits-03 (work in progress), March 2017.
- [I-D.ietf-aqm-codel]
Nichols, K., Jacobson, V., McGregor, A., and J. Jana, "Controlled Delay Active Queue Management", draft-ietf-aqm-codel-00 (work in progress), October 2014.
- [I-D.ietf-aqm-fq-codel]
Hoeiland-Joergensen, T., McKenney, P., Taht, D., Gettys, J., and E. Dumazet, "FlowQueue-Codel", draft-ietf-aqm-fq-codel-00 (work in progress), January 2015.
- [I-D.ietf-aqm-pie]
Pan, R., Natarajan, P., Baker, F., and G. White, "PIE: A Lightweight Control Scheme To Address the Bufferbloat Problem", draft-ietf-aqm-pie-00 (work in progress), October 2014.
- [I-D.ietf-ippm-6man-pdm-option]
Elkins, N., Hamilton, R., and m. mackermann@bcbsm.com, "IPv6 Performance and Diagnostic Metrics (PDM) Destination Option", draft-ietf-ippm-6man-pdm-option-10 (work in progress), May 2017.
- [I-D.ietf-quic-transport]
Iyengar, J. and M. Thomson, "QUIC: A UDP-Based Multiplexed and Secure Transport", draft-ietf-quic-transport-03 (work in progress), May 2017.

- [I-D.ietf-tcpm-accurate-ecn]
Briscoe, B., Kuehlewind, M., and R. Scheffenegger, "More Accurate ECN Feedback in TCP", draft-ietf-tcpm-accurate-ecn-00 (work in progress), December 2015.
- [I-D.ietf-tcpm-dctcp]
Bensley, S., Thaler, D., Balasubramanian, P., Eggert, L., and G. Judd, "Datacenter TCP (DCTCP): TCP Congestion Control for Datacenters", draft-ietf-tcpm-dctcp-06 (work in progress), May 2017.
- [I-D.ietf-tsvwg-l4s-arch]
Briscoe, B., Schepper, K., and M. Bagnulo, "Low Latency, Low Loss, Scalable Throughput (L4S) Internet Service: Architecture", draft-ietf-tsvwg-l4s-arch-00 (work in progress), May 2017.
- [I-D.mm-wg-effect-encrypt]
Moriarty, K. and A. Morton, "Effect of Pervasive Encryption on Operators", draft-mm-wg-effect-encrypt-11 (work in progress), April 2017.
- [I-D.trammell-plus-abstract-mech]
Trammell, B., "Abstract Mechanisms for a Cooperative Path Layer under Endpoint Control", draft-trammell-plus-abstract-mech-00 (work in progress), September 2016.
- [I-D.trammell-plus-statefulness]
Kuehlewind, M., Trammell, B., and J. Hildebrand, "Transport-Independent Path Layer State Management", draft-trammell-plus-statefulness-02 (work in progress), December 2016.
- [Latency] Briscoe, B., "Reducing Internet Latency: A Survey of Techniques and Their Merits", November 2014.
- [Measure] Fairhurst, G., Kuehlewind, M., and D. Lopez, "Measurement-based Protocol Design", June 2017.
- [RFC2474] Nichols, K., Blake, S., Baker, F., and D. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers", RFC 2474, DOI 10.17487/RFC2474, December 1998, <<https://www.rfc-editor.org/info/rfc2474>>.

- [RFC3135] Border, J., Kojo, M., Griner, J., Montenegro, G., and Z. Shelby, "Performance Enhancing Proxies Intended to Mitigate Link-Related Degradations", RFC 3135, DOI 10.17487/RFC3135, June 2001, <<https://www.rfc-editor.org/info/rfc3135>>.
- [RFC3168] Ramakrishnan, K., Floyd, S., and D. Black, "The Addition of Explicit Congestion Notification (ECN) to IP", RFC 3168, DOI 10.17487/RFC3168, September 2001, <<https://www.rfc-editor.org/info/rfc3168>>.
- [RFC3234] Carpenter, B. and S. Brim, "Middleboxes: Taxonomy and Issues", RFC 3234, DOI 10.17487/RFC3234, February 2002, <<https://www.rfc-editor.org/info/rfc3234>>.
- [RFC3449] Balakrishnan, H., Padmanabhan, V., Fairhurst, G., and M. Sooriyabandara, "TCP Performance Implications of Network Path Asymmetry", BCP 69, RFC 3449, DOI 10.17487/RFC3449, December 2002, <<https://www.rfc-editor.org/info/rfc3449>>.
- [RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, DOI 10.17487/RFC3550, July 2003, <<https://www.rfc-editor.org/info/rfc3550>>.
- [RFC3819] Karn, P., Ed., Bormann, C., Fairhurst, G., Grossman, D., Ludwig, R., Mahdavi, J., Montenegro, G., Touch, J., and L. Wood, "Advice for Internet Subnetwork Designers", BCP 89, RFC 3819, DOI 10.17487/RFC3819, July 2004, <<https://www.rfc-editor.org/info/rfc3819>>.
- [RFC4301] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, DOI 10.17487/RFC4301, December 2005, <<https://www.rfc-editor.org/info/rfc4301>>.
- [RFC4302] Kent, S., "IP Authentication Header", RFC 4302, DOI 10.17487/RFC4302, December 2005, <<https://www.rfc-editor.org/info/rfc4302>>.
- [RFC4303] Kent, S., "IP Encapsulating Security Payload (ESP)", RFC 4303, DOI 10.17487/RFC4303, December 2005, <<https://www.rfc-editor.org/info/rfc4303>>.
- [RFC4585] Ott, J., Wenger, S., Sato, N., Burmeister, C., and J. Rey, "Extended RTP Profile for Real-time Transport Control Protocol (RTCP)-Based Feedback (RTP/AVPF)", RFC 4585, DOI 10.17487/RFC4585, July 2006, <<https://www.rfc-editor.org/info/rfc4585>>.

- [RFC4737] Morton, A., Ciavattone, L., Ramachandran, G., Shalunov, S., and J. Perser, "Packet Reordering Metrics", RFC 4737, DOI 10.17487/RFC4737, November 2006, <<https://www.rfc-editor.org/info/rfc4737>>.
- [RFC5236] Jayasumana, A., Piratla, N., Banka, T., Bare, A., and R. Whitner, "Improved Packet Reordering Metrics", RFC 5236, DOI 10.17487/RFC5236, June 2008, <<https://www.rfc-editor.org/info/rfc5236>>.
- [RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, DOI 10.17487/RFC5246, August 2008, <<https://www.rfc-editor.org/info/rfc5246>>.
- [RFC5559] Eardley, P., Ed., "Pre-Congestion Notification (PCN) Architecture", RFC 5559, DOI 10.17487/RFC5559, June 2009, <<https://www.rfc-editor.org/info/rfc5559>>.
- [RFC5925] Touch, J., Mankin, A., and R. Bonica, "The TCP Authentication Option", RFC 5925, DOI 10.17487/RFC5925, June 2010, <<https://www.rfc-editor.org/info/rfc5925>>.
- [RFC6347] Rescorla, E. and N. Modadugu, "Datagram Transport Layer Security Version 1.2", RFC 6347, DOI 10.17487/RFC6347, January 2012, <<https://www.rfc-editor.org/info/rfc6347>>.
- [RFC6437] Amante, S., Carpenter, B., Jiang, S., and J. Rajahalme, "IPv6 Flow Label Specification", RFC 6437, DOI 10.17487/RFC6437, November 2011, <<https://www.rfc-editor.org/info/rfc6437>>.
- [RFC6679] Westerlund, M., Johansson, I., Perkins, C., O'Hanlon, P., and K. Carlberg, "Explicit Congestion Notification (ECN) for RTP over UDP", RFC 6679, DOI 10.17487/RFC6679, August 2012, <<https://www.rfc-editor.org/info/rfc6679>>.
- [RFC7258] Farrell, S. and H. Tschofenig, "Pervasive Monitoring Is an Attack", BCP 188, RFC 7258, DOI 10.17487/RFC7258, May 2014, <<https://www.rfc-editor.org/info/rfc7258>>.
- [RFC7525] Sheffer, Y., Holz, R., and P. Saint-Andre, "Recommendations for Secure Use of Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS)", BCP 195, RFC 7525, DOI 10.17487/RFC7525, May 2015, <<https://www.rfc-editor.org/info/rfc7525>>.

- [RFC7567] Baker, F., Ed. and G. Fairhurst, Ed., "IETF Recommendations Regarding Active Queue Management", BCP 197, RFC 7567, DOI 10.17487/RFC7567, July 2015, <<https://www.rfc-editor.org/info/rfc7567>>.
- [RFC7624] Barnes, R., Schneier, B., Jennings, C., Hardie, T., Trammell, B., Huitema, C., and D. Borkmann, "Confidentiality in the Face of Pervasive Surveillance: A Threat Model and Problem Statement", RFC 7624, DOI 10.17487/RFC7624, August 2015, <<https://www.rfc-editor.org/info/rfc7624>>.
- [RFC7713] Mathis, M. and B. Briscoe, "Congestion Exposure (ConEx) Concepts, Abstract Mechanism, and Requirements", RFC 7713, DOI 10.17487/RFC7713, December 2015, <<https://www.rfc-editor.org/info/rfc7713>>.
- [RFC7928] Kuhn, N., Ed., Natarajan, P., Ed., Khademi, N., Ed., and D. Ros, "Characterization Guidelines for Active Queue Management (AQM)", RFC 7928, DOI 10.17487/RFC7928, July 2016, <<https://www.rfc-editor.org/info/rfc7928>>.
- [RFC8084] Fairhurst, G., "Network Transport Circuit Breakers", BCP 208, RFC 8084, DOI 10.17487/RFC8084, March 2017, <<https://www.rfc-editor.org/info/rfc8084>>.
- [RFC8085] Eggert, L., Fairhurst, G., and G. Shepherd, "UDP Usage Guidelines", BCP 145, RFC 8085, DOI 10.17487/RFC8085, March 2017, <<https://www.rfc-editor.org/info/rfc8085>>.
- [RFC8086] Yong, L., Ed., Crabbe, E., Xu, X., and T. Herbert, "GRE-in-UDP Encapsulation", RFC 8086, DOI 10.17487/RFC8086, March 2017, <<https://www.rfc-editor.org/info/rfc8086>>.
- [RFC8087] Fairhurst, G. and M. Welzl, "The Benefits of Using Explicit Congestion Notification (ECN)", RFC 8087, DOI 10.17487/RFC8087, March 2017, <<https://www.rfc-editor.org/info/rfc8087>>.
- [Tor] The Tor Project, ., "<<https://www.torproject.org>>", June 2017.

Appendix A. Revision information

-00 This is an individual draft for the IETF community

-01 This draft was a result of walking away from the text for a few days and then reorganising the content.

-02 This draft fixes textual errors.

-03 This draft follows feedback from people reading this draft.

Comments from the community are welcome on the text and recommendations.

Author's Address

Godred Fairhurst
University of Aberdeen
Department of Engineering
Fraser Noble Building
Aberdeen AB24 3UE
Scotland

Email: gorry@erg.abdn.ac.uk
URI: <http://www.erg.abdn.ac.uk/>

TSVWG
Internet-Draft
Intended status: Informational
Expires: March 29, 2018

G. Fairhurst
University of Aberdeen
C.S. Perkins
University of Glasgow
September 27, 2017

The Impact of Transport Header Encryption on Operation and Evolution of
the Internet
draft-fairhurst-tsvwg-transport-encrypt-04

Abstract

This document describes implications of applying end-to-end encryption at the transport layer. It identifies some in-network uses of transport layer header information that can be used with a transport header integrity check. It reviews the implication of developing encrypted end-to-end transport protocols and examines the implication of developing and deploying encrypted end-to-end transport protocols. Since transport measurement and analysis of the impact of network characteristics have been important to the design of current transport protocols, it also considers some anticipated implications on transport and application evolution.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on March 29, 2018.

Copyright Notice

Copyright (c) 2017 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
1.1. Current uses of Transport Headers within the Network	6
1.1.1. Observing Transport Information in the Network	7
1.1.1.1. Flow Identification	7
1.1.1.2. Metrics derived from Transport Layer Headers	7
1.1.1.3. Metrics derived from Network Layer Headers	10
1.1.2. Transport Measurement	12
1.1.2.1. Point of Measurement	12
1.1.2.2. Use by Operators to Plan and Provision Networks . .	13
1.1.2.3. Service Performance Measurement	13
1.1.2.4. Measuring Transport to Support Network Operations .	13
1.1.3. Use for Network Diagnostics and Troubleshooting	15
1.1.4. Observing Headers to Implement Network Policy	15
2. Encryption and Authentication of Transport Headers	15
2.1. Authenticating the Transport Protocol Header	17
2.2. Encrypting the Transport Payload	17
2.3. Encrypting the Transport Header	18
2.4. Authenticating Transport Information and Selectively Encrypting the Transport Header	18
2.5. Adding Transport Information to Network-Layer Protocol Headers	18
3. Implications of Protecting the Transport Headers	19
3.1. Independent Measurement	19
3.2. Characterising "Unknown" Network Traffic	20
3.3. Accountability and Internet Transport Protocols	20
3.4. Impact on Research, Development and Deployment	21
4. Acknowledgements	21
5. Security Considerations	22
6. IANA Considerations	22
7. References	22
7.1. Normative References	22
7.2. Informative References	22
Appendix A. Revision information	26
Authors' Addresses	27

1. Introduction

This document discusses the implications of end-to-end encryption applied at the transport layer, and examines the impact on transport protocol design, usage, and network operations and management. It also considers anticipated implications on transport and application evolution.

The transport layer provides the first end-to-end interactions across the Internet. Transport protocols layer directly over the network-layer service and are sent in the payload of network-layer packets. They support end-to-end communication between applications, supported by higher-layer protocols, running on the end systems (or transport endpoint). This simple architectural view hides one of the core functions of the transport, however - to discover and adapt to the properties of the Internet path that is currently being used. The design of Internet transport protocols is as much about trying to avoid the unwanted side effects of congestion on a flow and other capacity-sharing flows, avoiding congestion collapse, adapting to changes in the path characteristics, etc., as it is about end-to-end feature negotiation, flow control and optimising for performance of a specific application.

To achieve stable Internet operations the IETF transport community has to date relied heavily on measurement and insights of the network operations community to understand the trade-offs, and to inform selection of select appropriate mechanisms, to ensure a safe, reliable and robust Internet. In turn, the network operations community relies on being able to understand the traffic passing over the Internet, both in aggregate and at the flow level -- inspecting transport layer headers to help understand traffic dynamics.

There are many motivations for deploying encrypted transports, and encryption of transport payloads. The increasing public concerns about the interference with Internet traffic have led to a rapidly expanding deployment of encryption to protect end-user privacy, in protocols like QUIC. At the same time, network operators and access providers, especially in mobile networks, have come to rely on the in-network measurement of transport properties and the functionality provided by middleboxes to both support network operations and enhance performance.

This document considers some implications of working with encrypted transport protocols, and discusses trade-offs around authentication, encryption of transport protocol headers. It describes some of the architectural challenges and considerations in the way transport protocols are designed when using encryption [Measure].

Encryption of the transport layer brings some well-known privacy and security benefits, but also introduces various costs that need to be considered. Specifically, it can impact the following activities that rely on measurement and analysis of traffic flows:

- o Network Operations and Research: Observable transport headers enable operators and the research community to measure and analyse protocol performance, network anomalies, and failure pathologies. This information can help inform capacity planning, and assist in determining the need for equipment and/or configuration changes by network operators. This data also can inform Internet engineering research, and help the develop of new protocols and procedures. Encryption of the entire transport protocol, including header information, will restrict the availability of data, and might lead to the development of alternative, and potentially more intrusive, methods to acquire the needed data. Encrypting the transport payload, but leaving some, or all, of the transport headers unencrypted but authenticated can provide the majority of the privacy and security benefits while allowing some measurement.
- o Network Troubleshooting and diagnostics: Encrypting transport header information eliminates the incentive for operators to troubleshoot what they cannot interpret. A flow experiencing packet loss looks like an unaffected flow when only observing network layer headers (if transport sequence numbers and flow identifiers are obscured). This limits understanding of the impact of packet loss on the flows that share a network segment. Encrypted traffic therefore implies "don't touch", and a likely trouble-shooting response will be "can't help, no trouble found". The additional mechanisms that will need to be introduced to help reconstruct transport-level metrics add complexity and operational costs [I-D.mmm-wg-effect-encrypt].
- o Network Traffic Analysis: The use of encryption can make it harder to determine which transport protocols and features are being used across a network segment. The trends in usage. This could impact the ability for an operator to anticipate the need for network upgrades and roll-out. It can also impact the on-going traffic engineering activities performed by operators. While the impact may, in many cases, be small there are scenarios where operators directly support particular services (e.g., in radio links, or to troubleshoot issues realting to Quality of Service, QoS). The more complex the underlying infrastructure the more important this impact.

- o Open and Verifiable Network Data: The use of transport header encryption reduces the range of actors that can capture useful measurement data. This is, of course, its goal. Doing so, however, limits the information sources available to the Internet community to understand the operation of transport protocols, so preventing access to the information necessary to inform design decisions and standards for new protocols and related operational practices. There are dangers in a model where only endpoints (i.e., at user devices and within service platforms) can observe performance, and this cannot be independently verified. To ensure the health of the standards and research communities, we need independently captured data to develop on the behaviour of the transports. Independently verifiable performance metrics might also be important in order to demonstrate regulatory compliance in some jurisdictions.

The last point leads us to consider the impact of encrypting all the transport headers the specification and development of protocols and standards. It has potential impact on:

- o Understanding Feature Interactions: An appropriate vantage point, coupled with timing information about traffic flows, provides a valuable tool for benchmarking equipment and/or configurations, and to understand complex feature interactions. Transport header encryption limits the ability to diagnose and explore interactions between features at different protocol layers, a side-effect of not allowing a choice of vantage point from which this information is observed.
- o Supporting Common Specifications: The Transmission Control Protocol (TCP) is the predominant transport protocol. Its many variants have broadly consistent approaches to avoiding congestion collapse, and to ensuring the stability of the network. Increased use of transport layer encryption can overcome ossification, allowing deployment of new transports with different types of congestion control. This flexibility can be beneficial, but it comes at the cost of fragmenting the ecosystem. There's little doubt that developers will try to produce high quality transports for their target uses, but it is not clear there are sufficient incentives to ensure good practice that benefits the wide diversity of requirements for the Internet community as a whole. Increased diversity, and the ability to innovate without public scrutiny, risks point solutions that optimise for specific needs, but accidentally disrupt operations of/in different parts of the network. The social compact that maintains the stability of the network relies on accepting common specifications, and on the ability to verify that others also conform.

- o Operational practice: Published transport specifications allow operators to check compliance. This can bring assurance to those operating networks, often avoiding the need to deploy complex techniques that routinely monitor and manage TCP/IP traffic flows (e.g. Avoiding the capital and operational costs of deploying flow rate-limiting and network circuit-breaker methods). This should continue when encrypted transport headers are used, but methods need to confirm that the traffic produced conforms to the expectations of the operator or developer.
- o Restricting research and development: The use of encryption may impede independent research into new mechanisms, measurement of behaviour, and development initiatives. Experience shows that transport protocols are complicated to design and complex to deploy, and that individual mechanisms need to be evaluated while considering other mechanism, across a broad range of network topologies and with attention to the impact on traffic sharing the capacity. Adopting pervasive encryption of transport information could eliminate the independent self-checks that have previously been in place from research and academic contributors (e.g., the role of the IRTF ICCRG, and research publications in reviewing new transport mechanisms and assessing the impact of their experimental deployment).

Pervasive use of transport header encryption can impact the ways that protocols are designed, standardised, deployed, and operated. The choice of whether future transport protocols encrypt their protocol headers therefore needs to be taken based not solely on security and privacy considerations, but also taking into account the impact on operations, standards, and research. A network that is secure but unusable due to persistent congestion collapse is not an improvement, and while that would be an extreme outcome proposals that impose high costs for very limited benefits need to be considered carefully, to ensure the benefits outweigh the costs.

1.1. Current uses of Transport Headers within the Network

The transport layer is the first end-to-end layer in the network stack. Despite headers having end-to-end meaning, some transport headers have come to be used in various ways within the Internet. In response to pervasive monitoring [RFC7624] revelations and the IETF consensus that "Pervasive Monitoring is an Attack" [RFC7258], efforts are underway to increase encryption of Internet traffic, which would prevent visibility of transport headers. This affects on how network protocols are designed and used [I-D.mm-wg-effect-encrypt]. To understand these implications, it is first necessary to understand how transport layer headers are currently observed and/or modified by middleboxes within the network.

Transport protocols can be designed to encrypt or authenticate transport header fields. Authentication methods at the transport layer can be used to detect any changes to an immutable header field that were made by a network device along a path. The intentional modification of transport headers by middleboxes (such as Network Address Translation with Protocol Translation, NAT-PT, or Firewalls) is not considered.

1.1.1.1. Observing Transport Information in the Network

In-network observation of transport protocol headers requires knowledge of the format of the transport header:

- o Flows need to be identified at the level required for monitoring;
- o The protocol and version of the header need to be observable. As protocols evolve over time and there may be a need to introduce new transport headers. This may require interpretation of protocol version information or connection setup information;
- o Location and syntax of any transport headers to be observed. IETF transport protocols specify this information.

The following subsections describe various ways that observable transport information may be utilised.

1.1.1.1.1. Flow Identification

Transport protocol header information can identify a flow and the connection state of the flow, together with the protocol options being used. In some usages, a low-numbered (well-known) port can identify a protocol (although port information alone is not sufficient to guarantee identification of a protocol). Transport protocols, such as TCP and Stream Control Transport Protocol (SCTP) specify a standard base header that includes sequence number information and other data, with the possibility to negotiate additional headers at connection setup, identified by an option number in the transport header. UDP-based protocols can use, but sometimes do not use, well-known ports. Some can instead be identified by signalling protocols or through the use of magic numbers placed in the first byte(s) of the datagram payload.

1.1.1.1.2. Metrics derived from Transport Layer Headers

Some actors have a need to characterise the performance of link/network segments. Passive monitoring uses observed traffic to make inferences from transport headers to derive these measurements. A variety of open source and commercial tools have been deployed that utilise this information. The following metrics can be derived from transport header information:

Traffic Rate and Volume: Header information may allow derivation of volume measures per-application, to characterise the traffic that uses a network segment or the pattern of network usage. This may be measured per endpoint or aggregate of endpoint (e.g., by an operator to assess subscriber usage). It can also be used to trigger measurement-based traffic shaping and to implement QoS support within the network and lower layers. Volume measures can be valuable for capacity planning (providing detail of trends rather than the volume per subscriber).

Loss Rate and Loss Pattern: Flow loss rate may be derived and is often used as a metric for performance assessment and to characterise transport behaviour. Understanding the root cause of loss can help an operator determine whether this requires corrective action.

There are various cause of loss, including: corruption on a link (e.g., interference on a radio link), buffer overflow (e.g., due to congestion), policing (traffic management), buffer management (e.g., Active Queue Management, AQM). Understanding flow loss rate requires either maintaining per flow packet counters or by observing sequence numbers in transport headers. Loss can be monitored at the interface level by devices in the network. It is often important to understand the conditions under which packet loss occurs. This usually requires relating loss to the traffic flowing on the network segment at the time of loss.

Observation of transport feedback information (observing loss reports, e.g., RTP Control Protocol (RTCP), TCP SACK) can increase understanding of the impact of loss and help identify cases where loss may have been wrongly identified, or the transport did not require the lost packet. It is sometimes more important to understand the pattern of loss, than the loss rate - since losses can often occur as bursts, rather than randomly-timed events.

Throughput and Goodput: The throughput observed by a flow can be determined even when a flow is encrypted, providing the individual flow can be identified. Goodput [RFC7928] is a measure of useful data exchanged (the ratio of useful/total volume of traffic sent by a flow), which requires ability to differentiate loss and retransmission of packets (e.g., by observing packet sequence numbers in the TCP or the Real Time Protocol, RTP, headers

[RFC3550])).

Latency: Latency is a key performance metric that impacts application response time and user-perceived response time. It often indirectly impacts throughput and flow completion time. Latency determines the reaction time of the transport protocol itself, impacting flow setup, congestion control, loss recovery, and other transport mechanisms. The observed latency can have many components [Latency]. Of these, unnecessary/unwanted queuing in network buffers has often been observed as a significant factor. Once the cause of unwanted latency has been identified, this can often be eliminated, and determining latency metrics is a key driver in the deployment of AQM [RFC7567], DiffServ [RFC2474], and Explicit Congestion Notification (ECN) [RFC3168] [RFC8087].

To measure latency across a part of the path, an observation point can measure the experienced round trip time (RTT) using packet sequence numbers, and acknowledgements, or by observing header timestamp information. Such information allows an observation point in the network to determine not only the path RTT, but also to measure the upstream and downstream contribution to the RTT. This may be used to locate a source of latency, e.g., by observing cases where the ratio of median to minimum RTT is large for a part of a path.

An example usage of this method could identify excessive buffers to help deploy or configure AQM [RFC7567] [RFC7928] to effectively eliminate unnecessary queuing in routers and other devices. AQM methods need to be deployed at the capacity bottleneck, but are often deployed in combination with other techniques, such as scheduling [RFC7567] [I-D.ietf-aqm-fq-codel] and although parameter-less methods are desired [RFC7567], current methods [I-D.ietf-aqm-fq-codel] [I-D.ietf-aqm-codel] [I-D.ietf-aqm-pie] often cannot scale across all possible deployment scenarios. The service offered by operators can therefore benefit from latency information to understand the impact of deployment and tune deployed services.

Jitter: Some network applications are sensitive to changes in packet timing. For such applications, it can be necessary to measure the jitter observed along a portion of the path. The requirements to measure jitter resemble those for the measurement of latency.

Flow Reordering: Significant flow reordering can impact time-critical applications and can be interpreted as loss by reliable transports. Many transport protocol techniques are impacted by reordering (e.g., triggering TCP retransmission, or re-buffering

of real-time applications). Packet reordering can occur for many reasons (from equipment design to misconfiguration of forwarding rules).

As in the drive to reduce network latency, there is a need for operational tools to detect mis-ordered packet flows and quantify the degree of reordering. Techniques for measuring reordering typically observe packet sequence numbers. Metrics have been defined that evaluate whether a network has maintained packet order on a packet-by-packet basis [RFC4737] and [RFC5236].

There has been initiatives in the IETF transport area to reduce the impact of reordering within a transport flow, possibly leading to reduced the requirements for ordering. These have promise to simplify network equipment design as well as the potential to improve robustness of the transport service. Measurements of reordering can help understand the level of reordering within deployed infrastructure, and inform decisions about how to progress such mechanisms.

Some protocols provide in-built monitoring and reporting functions. Transport fields in the RTP header [RFC3550] [RFC4585] can be observed to derive traffic volume measurements and provide information on the progress and quality of a session using RTP. Key performance indicators are retransmission rate, packet drop rate, sector utilization level, a measure of reordering, peak rate, the CE-marking rate, etc. Metadata is often important to understand the context under which the data was collected, including the time, observation point, and way in which metrics were accumulated. The RTCP protocol directly reports some of this information in a form that can be directly visible in the network. A user of summary measurement data needs to trust the source of this data and the method used to generate the summary information.

When encryption conceals information in packet headers, measurements need to rely on pattern inferences and other heuristics grows, and accuracy suffers [I-D.mm-wg-effect-encrypt].

1.1.1.3. Metrics derived from Network Layer Headers

Some transport information is made visible in the network-layer protocol header. These header fields are not encrypted and can be used to make flow observations.

Use of IPv6 Network-Layer Flow Label: Endpoints are encouraged expose flow information in the IPv6 Flow Label field of the network-layer header (e.g. [RFC8085]). This can be used to inform network-layer queuing, forwarding (e.g., for equal cost multi-path (ECMP) routing, and Link Aggregation, LAG). This can provide useful information to assign packets to flows in the data collected by measurement campaigns. Although important to characterising a path, it does not directly provide any performance data.

Use Network-Layer Differentiated Services Code Point Point: Applications can expose their delivery expectations to the network by setting the Differentiated Services Code Point (DSCP) field of IPv4 and IPv6 packets. This can be used to inform network-layer queuing and forwarding, and can also provide information on the relative importance of packet information collected by measurement campaigns, but does not directly provide any performance data.

This field provides explicit information that can be used in place of inferring traffic requirements (e.g., by inferring QoS requirements from port information via a multi-field classifier). The DSCP value can therefore impact the quality of experience for a flow. Observations of service performance need to consider this field when a network path has support for differentiated service treatment.

Use of Explicit Congestion Marking: ECN[RFC3168] is an optional transport mechanism that uses a code point in the network-layer header. Use of ECN can offer gains in terms of increased throughput, reduced delay, and other benefits when used over a path that includes equipment that supports an AQM method that performs Congestion Experienced (CE) marking of IP packets [RFC8087].

ECN exposes the presence of congestion on a network path to the transport and network layer. The reception of CE-marked packets can therefore be used to monitor the presence and estimate the level of incipient congestion on the upstream portion of the path from the point of observation (Section 2.5 of [RFC8087]). Because ECN marks carried in the IP protocol header, it is much easier to measure ECN than metering packet loss. However, interpreting the marking behaviour (i.e., assessing congestion and diagnosing faults) requires context from the transport layer (path RTT, visibility of loss - that could be due to queue overflow, congestion response, etc) [RFC7567].

Some ECN-capable network devices can provide richer (more frequent and fine-grained) indication of their congestion state. Setting congestion marks proportional to the level of congestion (e.g., Data Center TCP, DCTP [I-D.ietf-tcpm-dctcp], and Low Latency Low Loss Scalable throughput, L4S, [I-D.ietf-tsvwg-l4s-arch]).

Use of ECN requires feedback a transport to feed back reception information on the path towards the data sender. Exposure of this Transport ECN feedback provides an additional powerful tool to understand ECN-enabled AQM-based networks [RFC8087].

AQM and ECN offer a range of algorithms and configuration options, it is therefore important for tools to be available to network operators and researchers to understand the implication of configuration choices and transport behaviour as use of ECN increases and new methods emerge [RFC7567] [RFC8087]. ECN-monitoring is expected to become important as AQM is deployed that supports ECN [RFC8087].

1.1.2. Transport Measurement

The common language between network operators and application/content providers/users is packet transfer performance at a layer that all can view and analyse. For most packets, this has been transport layer, until the emergence of QUIC, with the obvious exception of VPNs and IPsec. When encryption conceals more layers in a packet, people seeking understanding of the network operation need to rely more on pattern inferences and other heuristics. The accuracy of measurements therefore suffers, as does the ability to investigate and troubleshoot interactions between different anomalies. For example, the traffic patterns between a web server and a browser are dependent on browser supplier and version, even use of the application (e.g., web e-mail access). Even when measurement datasets are made available (e.g., from endpoints) additional metadata, such as the state of the network, is often required to interpret the data. Collecting and coordinating such metadata is more difficult when the observation point is at a different location to the bottleneck/device under evaluation.

Packet sampling techniques can be used to scale the processing involved in observing packets on high rate links. This exports only the packet header information of (randomly) selected packets. The utility of these measurements depends on the type of bearer and number of mechanisms used by network devices. Simple routers are relatively easy to manage, a device with more complexity demands understanding of the choice of many system parameters. This level of complexity exists when several network methods are combined.

This section discusses topics concerning observation of transport flows, with a focus on transport measurement.

1.1.2.1. Point of Measurement

Often measurements can only be understood in the context of the other flows that share a bottleneck. A simple example is monitoring of AQM. For example, FQ-CODEL [I-D.ietf-aqm-fq-codel], combines sub queues (statistically assigned per flow), management of the queue length (CODEL), flow-scheduling, and a starvation prevention mechanism. Usually such algorithms are designed to be self-tuning, but current methods typically employ heuristics that can result in more loss under certain path conditions (e.g., large RTT, effects of multiple bottlenecks [RFC7567]).

In-network measurements can distinguish between upstream and downstream metrics with respect to the measurement point. These are particularly useful for locating the source of problems or to assess the performance of a network segment or a particular device configuration.

By correlating observations at multiple points along the path (e.g., at the ingress and egress of a network segment), an observer can determine the contribution of a portion of the path to an observed metric (to locate a source of delay, jitter, loss, reordering, congestion marking, etc.).

1.1.2.2. Use by Operators to Plan and Provision Networks

Traffic measurements (e.g., traffic volume, loss, latency) is used by operators to help plan deployment of new equipment and configurations in their networks. Data is also important to equipment vendors who need to understand traffic trends and patterns of usage as inputs to decisions about planning products and provisioning for new deployments. This measurement information can also be correlated with billing information when this is also collected by an operator.

A network operator supporting traffic that uses transport header encryption may not have access to per-flow measurement data. Trends in aggregate traffic can be observed and can be related this to the endpoint addresses being used, but it may not be possible to correlate patterns in measurements with changes in transport protocols (e.g., the impact of changes in introducing a new transport protocol mechanism). This increases the dependency on other indirect sources of information to inform planning and provisioning.

1.1.2.3. Service Performance Measurement

Traffic measurements (e.g., traffic volume, loss, latency) can be used by various actors to help analyse the performance available to users of a network segment, and inform operational practice. While active measurements may be used in-network passive measurements can have advantages in terms of eliminating unproductive traffic, reducing the influence of test traffic on the overall traffic mix, and the ability to choose the point of measurement Section 1.1.2.1.

1.1.2.4. Measuring Transport to Support Network Operations

Information provided by tools observing transport headers can help determine whether mechanisms are needed in the network to prevent flows from acquiring excessive network capacity. Operators can implement operational practices to manage traffic flows (e.g., to prevent flows from acquiring excessive network capacity under severe congestion) by deploying rate-limiters, traffic shaping or network transport circuit breakers [RFC8084].

Congestion Control Compliance of Traffic: Congestion control is a key transport function. Many network operators implicitly accept that TCP traffic to comply with a behaviour that is acceptable for use in the shared Internet. TCP algorithms have been continuously improved over decades, and they have reached a level of efficiency and correctness that custom application-layer mechanisms will struggle to easily duplicate [RFC8085].

A standards-compliant TCP stack provides congestion control may therefore be judged safe for use across the Internet. Applications developed on top of well-designed transports can be expected to appropriately control their network usage, reacting when the network experiences congestion, by back-off and reduce the load placed on the network. This is the normal expected behaviour for TCP and SCTP.

However when anomalies are detected, tools can interpret the transport protocol header information to help understand the impact of specific transport protocols (or protocol mechanisms) on the other traffic that shares a network. An observation in the network can gain understanding of the dynamics of a flow and its congestion control behaviour. Analysing observed packet sequence numbers can be used to help build confidence that an application flow backs-off its share of the network load in the face of persistent congestion, and hence to understand whether the behaviour is appropriate for sharing limited network capacity. For example, it is common to visualise plots of TCP sequence numbers versus time for a flow to understand how a flow shares available capacity, deduce its dynamics in response to congestion, etc.

Congestion Control Compliance for UDP Traffic UDP provides a minimal message-passing transport that has no inherent congestion control mechanisms. Because congestion control is critical to the stable operation of the Internet, applications and other protocols that choose to use UDP as an Internet transport are required to employ mechanisms to prevent congestion collapse, avoid unacceptable contributions to jitter/latency, and to establish an acceptable share of capacity with concurrent traffic [RFC8085].

A network operator needs tools to understand if UDP flows comply with congestion control expectations and therefore whether there

is a need to deploy methods such as rate-limiters, transport circuit breakers or other methods to enforce acceptable usage for the offered service.

UDP flows that expose a well-known header by specifying the format of header fields can allow information to be observed to gain understanding of the dynamics of a flow and its congestion control behaviour. For example, tools exist to monitor various aspects of the RTP and RTCP header information of real-time flows (see Section 1.1.1.2).

1.1.1.3. Use for Network Diagnostics and Troubleshooting

Transport header information is useful for a variety of operational tasks [I-D.mm-wg-effect-encrypt]: to diagnose network problems, assess performance, capacity planning, management of denial of service threats, and responding to user performance questions. These tasks seldom involve the need to determine the contents of the transport payload, or other application details.

A network operator supporting traffic that uses transport header encryption can see only encrypted transport headers. This prevents deployment of performance measurement tools that rely on transport protocol information. Choosing to encrypt all information may be expected to reduce the ability for networks to "help" (e.g., in response to tracing issues, making appropriate Quality of Service, QoS, decisions). For some this will be blessing, for others it may be a curse. For example, operational performance data about encrypted flows needs to be determined by traffic pattern analysis, rather than relying on traditional tools. This can impact the ability of the operator to respond to faults, it could require reliance on endpoint diagnostic tools or user involvement in diagnosing and troubleshooting unusual use cases or non-trivial problems. A key need here is that tools need to provide useful information during network anomalies (e.g., significant reordering, high or intermittent loss). Although many network operators utilise transport information as a part of their operational practice, the network will not break because transport headers are encrypted.

1.1.1.4. Observing Headers to Implement Network Policy

Information from the transport protocol can be used by a multi-field classifier as a part of policy framework. Policies are commonly used for QoS management for resource-constrained networks and by firewalls that use the information to implement access rules. Traffic that cannot be classified, will typically receive a default treatment.

2. Encryption and Authentication of Transport Headers

End-to-end encryption can be applied at various protocol layers. It can be applied above the transport to encrypt the transport payload. Encryption methods can hide information from an eavesdropper in the network. Encryption can also help protect the privacy of a user, by hiding data relating to user/device identity or location. Neither an integrity check nor encryption methods prevent traffic analysis, and usage needs to reflect that profiling of users, identification of location and fingerprinting of behaviour can take place even on encrypted traffic flows.

One motive to use encryption is a response to perceptions that the network has become ossified by over-reliance on middleboxes that prevent new protocols and mechanisms from being deployed. This has lead to a common perception that there is too much "manipulation" of protocol headers within the network, and that designing to deploy in such networks is preventing transport evolution. In the light of this, a method that authenticates transport headers may help improve the pace of transport development, by eliminating the need to always consider deployed middleboxes [I-D.trammell-plus-abstract-mech], or potentially to only explicitly enable middlebox use for particular paths with particular middleboxes that are deliberately deployed to realise a useful function for the network and/or users[RFC3135].

Another motivation stems from increased concerns about privacy and surveillance. Some Internet users have valued the ability to protect identity, user location, and defend against traffic analysis, and have used methods such as IPsec ESP and Tor [Tor]. Revelations about the use of pervasive surveillance [RFC7624] have, to some extent, eroded trust in the service offered by network operators, and following the Snowden revelation in the USA in 2013 has led to an increased desire for people to employ encryption to avoid unwanted "eavesdropping" on their communications. Whatever the reasons, there are now activities in the IETF to design new protocols that may include some form of transport header encryption (e.g., QUIC [I-D.ietf-quic-transport]).

Authentication methods (that provide integrity checks of protocols fields) have also been specified at the network layer, and this also protects transport header fields. The network layer itself carries protocol header fields that are increasingly used to help forwarding decisions reflect the need of transport protocols, such the IPv6 Flow Label [RFC6437], the Differentiated Services Code Point (DSCP) [RFC2474] and Explicit Congestion Notification (ECN) [RFC3168].

The use of transport layer authentication and encryption exposes a tussle between middlebox vendors, operators, applications developers and users.

- o On the one hand, future Internet protocols that enable large-scale encryption assist in the restoration of the end-to-end nature of the Internet by returning complex processing to the endpoints, since middleboxes cannot modify what they cannot see.

- o On the other hand, encryption of transport layer header information has implications for people who are responsible for operating networks and researchers and analysts seeking to understand the dynamics of protocols and traffic patterns.

Whatever the motives, a decision to use pervasive of transport header encryption will have implications on the way in which design and evaluation is performed, and which can in turn impact the direction of evolution of the TCP/IP stack.

The next subsections briefly review some security design options for transport protocols.

2.1. Authenticating the Transport Protocol Header

Transport layer header information can be authenticated. An integrity check that protects the immutable transport header fields, but can still expose the transport protocol header information in the clear, allowing in-network devices to observe these fields. An integrity check can not prevent in-network modification, but can avoid a receiving accepting changes and avoid impact on the transport protocol operation.

An example transport authentication mechanism is TCP-Authentication (TCP-AO) [RFC5925]. This TCP option authenticates TCP segments, including the IP pseudo header, TCP header, and TCP data. TCP-AO protects the transport layer, preventing attacks from disabling the TCP connection itself. TCP-AO may interact with middleboxes, depending on their behaviour [RFC3234].

The IPsec Authentication Header (AH) [RFC4302] works at the network layer and authenticates the IP payload. This therefore also authenticates all transport headers, and verifies their integrity at the receiver, preventing in-network modification.

2.2. Encrypting the Transport Payload

The transport layer payload can be encrypted to protect the content of transport segments. This leaves transport protocol header information in the clear. The integrity of immutable transport

header fields could be protected by combining this with an integrity check (Section 2.1).

Examples of encrypting the payload include Transport Layer Security (TLS) over TCP [RFC5246] [RFC7525] or Datagram TLS (DTLS) over UDP [RFC6347] [RFC7525].

2.3. Encrypting the Transport Header

The network layer payload could be encrypted (including the entire transport header and payload). This method does not expose any transport information to devices in the network, which also prevents modification along the network path.

The IPsec Encapsulating Security Payload (ESP) [RFC4303] is an example of encryption at the network layer, it encrypts and authenticates all transport headers, preventing visibility of the headers by in-network devices. Some Virtual Private Network (VPN) methods also encrypt these headers.

2.4. Authenticating Transport Information and Selectively Encrypting the Transport Header

A transport protocol design can encrypt selected header fields, while also choosing to authenticate fields in the transport header. This allows specific transport header fields to be made observable by network devices. End-to end integrity checks can prevent an endpoint from undetected modification of the immutable transport headers.

The choice of which fields to expose and which to encrypt is a design choice for the transport protocol. Any selective encryption method requires trading two conflicting goals for a transport protocol designer to decide which header fields to encrypt. On the one hand, security work typically employs a design technique that seeks to expose only what is needed. On the other hand, there may be performance and operational benefits in exposing selected information to network tools.

Mutable fields in the transport header provide opportunities for middleboxes to modify the transport behaviour (e.g., the extended headers described in [I-D.trammell-plus-abstract-mech]). This considers only immutable fields in the transport headers, that is, fields that may be authenticated end-to-end across a path.

An example of a method that encrypts some, but not all, transport information is GRE-in-UDP [RFC8086] when used with GRE encryption.

2.5. Adding Transport Information to Network-Layer Protocol Headers

The transport information can be made visible in a network-layer header. This has the advantage that this information can then be observed by in-network devices. This has the advantage that a single header can support all transport protocols, but there may also be less desirable implications of separating the operation of the transport protocol from the measurement framework.

Some measurements may be made by adding additional protocol headers carrying operations, administration and management (OAM) information to packets at the ingress to a maintenance domain (e.g., an Ethernet protocol header with timestamps and sequence number information using a method such as 802.1lag) and removing the additional header at the egress of the maintenance domain. This approach enables some types of measurements, but does not cover the entire range of measurements described in this document.

Another example of a network-layer approach is the IPv6 Performance and Diagnostic Metrics (PDM) Destination Option [I-D.ietf-ippm-6man-pdm-option]. This allows a sender to optionally include a destination option that carries header fields that can be used to observe timestamps and packet sequence numbers. This information could be authenticated by receiving transport endpoints when the information is added at the sender and visible at the receiving endpoint, although methods to do this have not currently been proposed. This method needs to be explicitly enabled at the sender.

A drawback of using extension headers is that IPv4 network options are often not supported (or are carried on a slower processing path) and some IPv6 networks are also known to drop packets that set an IPv6 header extension. Another disadvantage is that protocols that separately expose header information do not necessarily have an advantage to expose the information that is utilised by the protocol itself, and could manipulate this header information to gain an advantage from the network.

3. Implications of Protecting the Transport Headers

This section explores key implications of working with encrypted transport protocols.

3.1. Independent Measurement

Independent observation by multiple actors is important for scientific analysis. Encrypting transport header encryption changes the ability for other actors to collect and independently analyse data. Internet transport protocols employ a set of mechanisms. Some

of these need to work in cooperation with the network layer - loss detection and recovery, congestion detection and congestion control, some of these need to work only end-to-end (e.g., parameter negotiation, flow-control).

When encryption conceals information in the transport header, it could be possible for an applications to provide summary data on performance and usage of the network. This data could be made available to other actors. However, this data needs to contain sufficient detail to understand (and possibly reconstruct the network traffic pattern for further testing) and to be correlated with the configuration of the network paths being measured. Sharing information between actors needs also to consider the privacy of the user and the incentives for providing accurate and detailed information. Protocols that expose the state information used by the transport protocol in their header information (e.g., timestamps used to calculate the RTT, packet numbers used to asses congestion and requests for retransmission) provide an incentive for the sending endpoint to provide correct information, increasing confidence that the observer understands the transport interaction with the network. This becomes important when considering changes to transport protocols, changes in network infrastructure, or the emergence of new traffic patterns.

3.2. Characterising "Unknown" Network Traffic

The patterns and types of traffic that share Internet capacity changes with time as networked applications, usage patterns and protocols continue to evolve.

If "unknown" or "uncharacterised" traffic patterns form a small part of the traffic aggregate passing through a network device or segment of the network the path, the dynamics of the uncharacterised traffic may not have a significant collateral impact on the performance of other traffic that shares this network segment. Once the proportion of this traffic increases, the need to monitor the traffic and determine if appropriate safety measures need to be put in place.

Tracking the impact of new mechanisms and protocols requires traffic volume to be measured and new transport behaviours to be identified. This is especially true of protocols operating over a UDP substrate. The level and style of encryption needs to be considered in determining how this activity is performed. On a shorter timescale, information may also need to be collected to manage denial of service attacks against the infrastructure.

3.3. Accountability and Internet Transport Protocols

Information provided by tools observing transport headers can help determine whether mechanisms are needed in the network to prevent flows from acquiring excessive network capacity, and where needed to deploy appropriate tools Section 1.1.2.4. Obfuscating or hiding this information using encryption is expected to lead operators and maintainers of middleboxes (firewalls, etc.) to seek other methods to classify and mechanisms to condition network traffic. A lack of data seems likely to reduce the level of precision with which these mechanisms are applied, and this needs to be considered when evaluating the impact of designs for transport encryption.

3.4. Impact on Research, Development and Deployment

Measurement data is increasingly being used to inform design decisions in networking research, during development of new mechanisms and protocols and in standardisation. Measurement has a critical role in the design of transport protocol mechanisms and their acceptance by the wider community (e.g., as a method to judge the safety for Internet deployment). Observation of pathologies are also important in understanding the interactions between cooperating protocols and network mechanism, the implications of sharing capacity with other traffic and the impact of different patterns of usage.

Attention needs to be paid to the expected scale of deployment of new protocols and protocol mechanisms. Whatever the mechanism, experience has shown that it is often difficult to correctly implement combination of mechanisms [RFC8085]. These mechanisms therefore typically evolve as a protocol matures, or in response to changes in network conditions, changes in network traffic or changes to application usage.

The growth and diversity of applications and protocols using the Internet continues to expand - and there has been recent interest in a wide range of new transport methods, e.g., Larger Initial Window, Proportional Rate Reduction (PRR), congestion control methods based on measuring bottleneck bandwidth and round-trip propagation time, the introduction of AQM techniques and new forms of ECN response (e.g., Data Centre TCP, DCTP [I-D.ietf-tcpm-dctcp], and methods proposed for Low Latency Low Loss Scalable throughput, L4S). For each new method it is desirable to build a body of data reflecting its behaviour under a wide range of deployment scenarios, traffic load, and interactions with other deployed/candidate methods.

Open standards motivate a desire for this evaluation to include independent observation and evaluation of performance data, which in turn suggests control over where and when measurement samples are collected. This requires consideration of the appropriate balance between encrypting all and no transport information.

4. Acknowledgements

The author would like to thank all who have talked to him face-to-face or via email. ...

This work has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 688421. The opinions expressed and arguments employed reflect only the authors' view. The European Commission is not responsible for any use that may be made of that information.

5. Security Considerations

This document is about design and deployment considerations for transport protocols. Authentication, confidentiality protection, and integrity protection are identified as Transport Features by RFC8095". As currently deployed in the Internet, these features are generally provided by a protocol or layer on top of the transport protocol; no current full-featured standards-track transport protocol provides these features on its own. Therefore, these features are not considered in this document, with the exception of native authentication capabilities of TCP and SCTP for which the security considerations in RFC4895.

Open data, and accessibility to tools that can help understand trends in application deployment, network traffic and usage patterns can all contribute to understanding security challenges. Standard protocols and understanding of the interactions between mechanisms and traffic patterns can also provide valuable insight into appropriate security design. Like congestion control mechanisms, security mechanisms are difficult to design and implement correctly. It is hence recommended that applications employ well-known standard security mechanisms such as DTLS, TLS or IPsec, rather than inventing their own.

6. IANA Considerations

XX RFC ED - PLEASE REMOVE THIS SECTION XXX

This memo includes no request to IANA.

7. References

7.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<http://www.rfc-editor.org/info/rfc2119>>.

7.2. Informative References

[I-D.dolson-plus-middlebox-benefits]
Dolson, D., Snellman, J., Boucadair, M. and C. Jacquenet,
"Beneficial Functions of Middleboxes", Internet-Draft
draft-dolson-plus-middlebox-benefits-03, March 2017.

[I-D.ietf-aqm-codel]

Nichols, K., Jacobson, V., McGregor, A. and J. Jana,
"Controlled Delay Active Queue Management", Internet-Draft
draft-ietf-aqm-codel-00, October 2014.

[I-D.ietf-aqm-fq-codel]
Hoeiland-Joergensen, T., McKenney, P., Taht, D., Gettys,
J. and E. Dumazet, "FlowQueue-Codel", Internet-Draft
draft-ietf-aqm-fq-codel-00, January 2015.

[I-D.ietf-aqm-pie]
Pan, R., Natarajan, P., Baker, F. and G. White, "PIE: A
Lightweight Control Scheme To Address the Bufferbloat
Problem", Internet-Draft draft-ietf-aqm-pie-00, October
2014.

[I-D.ietf-ippm-6man-pdm-option]
Elkins, N., Hamilton, R. and m. mackermann@bcbsm.com,
"IPv6 Performance and Diagnostic Metrics (PDM) Destination
Option", Internet-Draft draft-ietf-ippm-6man-pdm-
option-10, May 2017.

[I-D.ietf-quic-transport]
Iyengar, J. and M. Thomson, "QUIC: A UDP-Based Multiplexed
and Secure Transport", Internet-Draft draft-ietf-quic-
transport-03, May 2017.

[I-D.ietf-tcpm-accurate-ecn]
Briscoe, B., Kuehlewind, M. and R. Scheffenegger, "More
Accurate ECN Feedback in TCP", Internet-Draft draft-ietf-
tcpm-accurate-ecn-00, December 2015.

[I-D.ietf-tcpm-dctcp]
Bensley, S., Thaler, D., Balasubramanian, P., Eggert, L.
and G. Judd, "Datacenter TCP (DCTCP): TCP Congestion
Control for Datacenters", Internet-Draft draft-ietf-tcpm-
dctcp-06, May 2017.

[I-D.ietf-tsvwg-l4s-arch]
Briscoe, B., Schepper, K. and M. Bagnulo, "Low Latency,
Low Loss, Scalable Throughput (L4S) Internet Service:
Architecture", Internet-Draft draft-ietf-tsvwg-l4s-
arch-00, May 2017.

[I-D.mm-wg-effect-encrypt]
Moriarty, K. and A. Morton, "Effect of Pervasive
Encryption on Operators", Internet-Draft draft-mm-wg-
effect-encrypt-11, April 2017.

[I-D.trammell-plus-abstract-mech]
Trammell, B., "Abstract Mechanisms for a Cooperative Path
Layer under Endpoint Control", Internet-Draft draft-
trammell-plus-abstract-mech-00, September 2016.

[I-D.trammell-plus-statefulness]

Kuehlewind, M., Trammell, B. and J. Hildebrand,
"Transport-Independent Path Layer State Management",
Internet-Draft draft-trammell-plus-statefulness-02,
December 2016.

- [Latency] Briscoe, B., "Reducing Internet Latency: A Survey of
Techniques and Their Merits", November 2014.
- [Measure] Fairhurst, G., Kuehlewind, M. and D. Lopez, "Measurement-
based Protocol Design", June 2017.
- [RFC2474] Nichols, K., Blake, S., Baker, F. and D. Black,
"Definition of the Differentiated Services Field (DS
Field) in the IPv4 and IPv6 Headers", RFC 2474, DOI
10.17487/RFC2474, December 1998, <[http://www.rfc-
editor.org/info/rfc2474](http://www.rfc-editor.org/info/rfc2474)>.
- [RFC3135] Border, J., Kojo, M., Griner, J., Montenegro, G. and Z.
Shelby, "Performance Enhancing Proxies Intended to
Mitigate Link-Related Degradations", RFC 3135, DOI
10.17487/RFC3135, June 2001, <[http://www.rfc-editor.org/
info/rfc3135](http://www.rfc-editor.org/info/rfc3135)>.
- [RFC3168] Ramakrishnan, K., Floyd, S. and D. Black, "The Addition of
Explicit Congestion Notification (ECN) to IP", RFC 3168,
DOI 10.17487/RFC3168, September 2001, <[http://www.rfc-
editor.org/info/rfc3168](http://www.rfc-editor.org/info/rfc3168)>.
- [RFC3234] Carpenter, B. and S. Brim, "Middleboxes: Taxonomy and
Issues", RFC 3234, DOI 10.17487/RFC3234, February 2002,
<<http://www.rfc-editor.org/info/rfc3234>>.
- [RFC3449] Balakrishnan, H., Padmanabhan, V., Fairhurst, G. and M.
Sooriyabandara, "TCP Performance Implications of Network
Path Asymmetry", BCP 69, RFC 3449, DOI 10.17487/RFC3449,
December 2002, <<http://www.rfc-editor.org/info/rfc3449>>.
- [RFC3550] Schulzrinne, H., Casner, S., Frederick, R. and V.
Jacobson, "RTP: A Transport Protocol for Real-Time
Applications", STD 64, RFC 3550, DOI 10.17487/RFC3550,
July 2003, <<http://www.rfc-editor.org/info/rfc3550>>.
- [RFC3819] Karn, P., Ed., Bormann, C., Fairhurst, G., Grossman, D.,
Ludwig, R., Mahdavi, J., Montenegro, G., Touch, J. and L.
Wood, "Advice for Internet Subnetwork Designers", BCP 89,
RFC 3819, DOI 10.17487/RFC3819, July 2004, <[http://www
.rfc-editor.org/info/rfc3819](http://www.rfc-editor.org/info/rfc3819)>.
- [RFC4301] Kent, S. and K. Seo, "Security Architecture for the
Internet Protocol", RFC 4301, DOI 10.17487/RFC4301,
December 2005, <<http://www.rfc-editor.org/info/rfc4301>>.

- [RFC4302] Kent, S., "IP Authentication Header", RFC 4302, DOI 10.17487/RFC4302, December 2005, <<http://www.rfc-editor.org/info/rfc4302>>.
- [RFC4303] Kent, S., "IP Encapsulating Security Payload (ESP)", RFC 4303, DOI 10.17487/RFC4303, December 2005, <<http://www.rfc-editor.org/info/rfc4303>>.
- [RFC4585] Ott, J., Wenger, S., Sato, N., Burmeister, C. and J. Rey, "Extended RTP Profile for Real-time Transport Control Protocol (RTCP)-Based Feedback (RTP/AVPF)", RFC 4585, DOI 10.17487/RFC4585, July 2006, <<http://www.rfc-editor.org/info/rfc4585>>.
- [RFC4737] Morton, A., Ciavattone, L., Ramachandran, G., Shalunov, S. and J. Perser, "Packet Reordering Metrics", RFC 4737, DOI 10.17487/RFC4737, November 2006, <<http://www.rfc-editor.org/info/rfc4737>>.
- [RFC5236] Jayasumana, A., Piratla, N., Banka, T., Bare, A. and R. Whitner, "Improved Packet Reordering Metrics", RFC 5236, DOI 10.17487/RFC5236, June 2008, <<http://www.rfc-editor.org/info/rfc5236>>.
- [RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, DOI 10.17487/RFC5246, August 2008, <<http://www.rfc-editor.org/info/rfc5246>>.
- [RFC5559] Eardley, P., Ed., "Pre-Congestion Notification (PCN) Architecture", RFC 5559, DOI 10.17487/RFC5559, June 2009, <<http://www.rfc-editor.org/info/rfc5559>>.
- [RFC5925] Touch, J., Mankin, A. and R. Bonica, "The TCP Authentication Option", RFC 5925, DOI 10.17487/RFC5925, June 2010, <<http://www.rfc-editor.org/info/rfc5925>>.
- [RFC6347] Rescorla, E. and N. Modadugu, "Datagram Transport Layer Security Version 1.2", RFC 6347, DOI 10.17487/RFC6347, January 2012, <<http://www.rfc-editor.org/info/rfc6347>>.
- [RFC6437] Amante, S., Carpenter, B., Jiang, S. and J. Rajahalme, "IPv6 Flow Label Specification", RFC 6437, DOI 10.17487/RFC6437, November 2011, <<http://www.rfc-editor.org/info/rfc6437>>.
- [RFC6679] Westerlund, M., Johansson, I., Perkins, C., O'Hanlon, P. and K. Carlberg, "Explicit Congestion Notification (ECN) for RTP over UDP", RFC 6679, DOI 10.17487/RFC6679, August 2012, <<http://www.rfc-editor.org/info/rfc6679>>.

- [RFC7258] Farrell, S. and H. Tschofenig, "Pervasive Monitoring Is an Attack", BCP 188, RFC 7258, DOI 10.17487/RFC7258, May 2014, <<http://www.rfc-editor.org/info/rfc7258>>.
- [RFC7525] Sheffer, Y., Holz, R. and P. Saint-Andre, "Recommendations for Secure Use of Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS)", BCP 195, RFC 7525, DOI 10.17487/RFC7525, May 2015, <<http://www.rfc-editor.org/info/rfc7525>>.
- [RFC7567] Baker, F. Ed., and G. Fairhurst, Ed., "IETF Recommendations Regarding Active Queue Management", BCP 197, RFC 7567, DOI 10.17487/RFC7567, July 2015, <<http://www.rfc-editor.org/info/rfc7567>>.
- [RFC7624] Barnes, R., Schneier, B., Jennings, C., Hardie, T., Trammell, B., Huitema, C. and D. Borkmann, "Confidentiality in the Face of Pervasive Surveillance: A Threat Model and Problem Statement", RFC 7624, DOI 10.17487/RFC7624, August 2015, <<http://www.rfc-editor.org/info/rfc7624>>.
- [RFC7713] Mathis, M. and B. Briscoe, "Congestion Exposure (ConEx) Concepts, Abstract Mechanism, and Requirements", RFC 7713, DOI 10.17487/RFC7713, December 2015, <<http://www.rfc-editor.org/info/rfc7713>>.
- [RFC7928] Kuhn, N., Ed., Natarajan, P., Ed., Khademi, N. Ed., and D. Ros, "Characterization Guidelines for Active Queue Management (AQM)", RFC 7928, DOI 10.17487/RFC7928, July 2016, <<http://www.rfc-editor.org/info/rfc7928>>.
- [RFC8084] Fairhurst, G., "Network Transport Circuit Breakers", BCP 208, RFC 8084, DOI 10.17487/RFC8084, March 2017, <<http://www.rfc-editor.org/info/rfc8084>>.
- [RFC8085] Eggert, L., Fairhurst, G. and G. Shepherd, "UDP Usage Guidelines", BCP 145, RFC 8085, DOI 10.17487/RFC8085, March 2017, <<http://www.rfc-editor.org/info/rfc8085>>.
- [RFC8086] Yong, L., Ed., Crabbe, E., Xu, X. and T. Herbert, "GRE-in-UDP Encapsulation", RFC 8086, DOI 10.17487/RFC8086, March 2017, <<http://www.rfc-editor.org/info/rfc8086>>.
- [RFC8087] Fairhurst, G. and M. Welzl, "The Benefits of Using Explicit Congestion Notification (ECN)", RFC 8087, DOI 10.17487/RFC8087, March 2017, <<http://www.rfc-editor.org/info/rfc8087>>.
- [Tor] The Tor Project, ., "<<https://www.torproject.org>>", June 2017.

Appendix A. Revision information

- 00 This is an individual draft for the IETF community.
- 01 This draft was a result of walking away from the text for a few days and then reorganising the content.
- 02 This draft fixes textual errors.
- 03 This draft follows feedback from people reading this draft.
- 04 This adds an additional contributor and includes significant reworking to ready this for review by the wider IETF community Colin Perkins joined the author list.

Comments from the community are welcome on the text and recommendations.

Authors' Addresses

Godred Fairhurst
University of Aberdeen
Department of Engineering
Fraser Noble Building
Aberdeen, AB24 3UE
Scotland

Email: gorry@erg.abdn.ac.uk
URI: <http://www.erg.abdn.ac.uk/>

Colin Perkins
University of Glasgow
School of Computing Science
Glasgow, G12 8QQ
Scotland

Email: csp@csp Perkins.org
URI: <https://csp Perkins.org/>

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: April 27, 2018

E. Lear
Cisco Systems
R. Droms

D. Romascanu
October 24, 2017

Manufacturer Usage Description Specification
draft-ietf-opsawg-mud-13

Abstract

This memo specifies a component-based architecture for manufacturer usage descriptions (MUD). The goal of MUD is to provide a means for Things to signal to the network what sort of access and network functionality they require to properly function. The initial focus is on access control. Later work can delve into other aspects.

This memo specifies two YANG modules, IPv4 and IPv6 DHCP options, an LLDP TLV, a URL suffix specification, an X.509 certificate extension and a means to sign and verify the descriptions.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on April 27, 2018.

Copyright Notice

Copyright (c) 2017 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of

publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	3
1.1. What MUD doesn't do	4
1.2. A Simple Example	5
1.3. Determining Intended Use	5
1.4. Finding A Policy: The MUD URL	5
1.5. Types of Policies	6
1.6. Terminology	8
1.7. The Manufacturer Usage Description Architecture	9
1.8. Order of operations	10
2. The MUD Model and Semantic Meaning	11
2.1. The IETF-MUD YANG Module	11
3. Data Node Definitions	13
3.1. to-device-policy and from-device-policy containers	13
3.2. last-update	14
3.3. cache-validity	14
3.4. is-supported	14
3.5. systeminfo	14
3.6. extensions	14
3.7. manufacturer	15
3.8. same-manufacturer	15
3.9. model	15
3.10. local-networks	15
3.11. controller	15
3.12. my-controller	16
3.13. direction-initiated	16
4. Processing of the MUD file	16
5. What does a MUD URL look like?	17
6. The MUD YANG Model	17
7. The Domain Name Extension to the ACL Model	23
7.1. source-dnsname	24
7.2. destination-dnsname	24
7.3. The ietf-acldns Model	24
8. MUD File Example	25
9. The MUD URL DHCP Option	28
9.1. Client Behavior	28
9.2. Server Behavior	29
9.3. Relay Requirements	29
10. The Manufacturer Usage Description (MUD) URL X.509 Extension	29
11. The Manufacturer Usage Description LLDP extension	31

12. Creating and Processing of Signed MUD Files	33
12.1. Creating a MUD file signature	33
12.2. Verifying a MUD file signature	33
13. Extensibility	34
14. Deployment Considerations	34
15. Security Considerations	35
16. IANA Considerations	37
16.1. YANG Module Registrations	37
16.2. DHCPv4 and DHCPv6 Options	38
16.3. PKIX Extensions	38
16.4. Well Known URI Suffix	38
16.5. MIME Media-type Registration for MUD files	38
16.6. LLDP IANA TLV Subtype Registry	39
16.7. The MUD Well Known Universal Resource Name (URNs)	40
16.8. Extensions Registry	40
17. Acknowledgments	40
18. References	41
18.1. Normative References	41
18.2. Informative References	43
Appendix A. Changes from Earlier Versions	44
Appendix B. Default MUD nodes	47
Appendix C. A Sample Extension: DETNET-indicator	51
Authors' Addresses	55

1. Introduction

The Internet has largely been constructed on general purpose computers, those devices that may be used for a purpose that is specified by those who buy the device. [RFC1984] presumed that an end device would be most capable of protecting itself. This made sense when the typical device was a workstation or a mainframe, and it continues to make sense for general purpose computing devices today, including laptops, smart phones, and tablets.

[RFC7452] discusses design patterns for, and poses questions about, smart objects. Let us then posit a group of objects that are specifically NOT general purpose computers. These devices have a specific purpose. By definition, therefore, all other uses are NOT intended. The combination of these two statements can be restated as a manufacturer usage description (MUD) that can be applied at various points within a network. Although this memo may seem to stress access requirements, usage intent also consists of quality of service needs a device may have.

We use the notion of "manufacturer" loosely in this context to refer to the entity or organization that will state how a device is intended to be used. In the context of a lightbulb, this might indeed be the lightbulb manufacturer. In the context of a smarter

device that has a built in Linux stack, it might be an integrator of that device. The key points are that the device itself is expected to serve a limited purpose, and that there may exist an organization in the supply chain of that device that will take responsibility for informing the network about that purpose.

The intent of MUD is to solve for the following problems:

- o Substantially reduce the threat surface on a device entering a network to those communications intended by the manufacturer.
- o Provide for a means to scale network policies to the ever-increasing number types of devices in the network.
- o Provide a means to address at least some vulnerabilities in a way that is faster than it might take to update systems. This will be particularly true for systems that are no longer supported by their manufacturer.
- o Keep the cost of implementation of such a system to the bare minimum.
- o Provide a means of extensibility for manufacturers to express other device capabilities or requirements.

MUD consists of three architectural building blocks:

- o A classifier that a device emits that can be used to locate a description;
- o The description itself, including how it is interpreted, and;
- o A means for local network management systems to retrieve the description.

In this specification we describe each of these building blocks and how they are intended to be used together. However, they may also be used separately, independent of this specification, by local deployments for their own purposes.

1.1. What MUD doesn't do

MUD is not intended to address network authorization of general purpose computers, as their manufacturers cannot envision a specific communication pattern to describe. In addition, even those devices that have a single or small number of uses might have very broad communication patterns. MUD on its own is not for them either.

No matter how good a MUD-enabled network is, it will never replace the need for manufacturers to patch vulnerabilities. It may, however, provide network administrators with some additional protection when those vulnerabilities exist.

Finally, no matter what the manufacturer specifies in a MUD file, these are not directives, but suggestions. How they are instantiated locally will depend on many factors and will be ultimately up to the local network administrator, who must decide what is appropriate in a given circumstances.

1.2. A Simple Example

A light bulb is intended to light a room. It may be remotely controlled through the network, and it may make use of a rendezvous service of some form that an app on smart phone accesses. What we can say about that light bulb, then, is that all other network access is unwanted. It will not contact a news service, nor speak to the refrigerator, and it has no need of a printer or other devices. It has no social networking friends. Therefore, an access list applied to it that states that it will only connect to the single rendezvous service will not impede the light bulb in performing its function, while at the same time allowing the network to provide both it and other devices an additional layer of protection.

1.3. Determining Intended Use

The notion of intended use is in itself not new. Network administrators apply access lists every day to allow for only such use. This notion of white listing was well described by Chapman and Zwicky in [FW95]. Profiling systems that make use of heuristics to identify types of systems have existed for years as well.

A Thing could just as easily tell the network what sort of access it requires without going into what sort of system it is. This would, in effect, be the converse of [RFC7488]. In seeking a general purpose solution, however, we assume that a device has so few capabilities that it will implement the least necessary capabilities to function properly. This is a basic economic constraint. Unless the network would refuse access to such a device, its developers would have no reason to provide the network any information. To date, such an assertion has held true.

1.4. Finding A Policy: The MUD URL

Our work begins with the device emitting a Universal Resource Locator (URL) [RFC3986]. This URL serves both to classify the device type and to provide a means to locate a policy file.

In this memo three means are defined to emit the MUD URL. One is a DHCP option[RFC2131],[RFC3315] that the DHCP client uses to inform the DHCP server. The DHCP server may take further actions, such as retrieve the URL or otherwise pass it along to network management system or controller. The second method defined is an X.509 constraint. The IEEE has developed [IEEE802.1AR] that provides a certificate-based approach to communicate device characteristics, which itself relies on [RFC5280]. The MUD URL extension is non-critical, as required by IEEE 802.1AR. Various means may be used to communicate that certificate, including Tunnel Extensible Authentication Protocol (TEAP) [RFC7170]. Finally, a Link Layer Discovery Protocol (LLDP) frame is defined [IEEE802.1AB].

It is possible that there may be other means for a MUD URL to be learned by a network. For instance, some devices may already be fielded or have very limited ability to communicate a MUD URL, and yet can be identified through some means, such as a serial number or a public key. In these cases, manufacturers may be able to map those identifiers to particular MUD URLs (or even the files themselves). Similarly, there may be alternative resolution mechanisms available for situations where Internet connectivity is limited or does not exist. Such mechanisms are not described in this memo, but are possible. Implementors should allow for this sort of flexibility of how MUD URLs may be learned.

1.5. Types of Policies

When the MUD URL is resolved, the MUD controller retrieves a file that describes what sort of communications a device is designed to have. The manufacturer may specify either specific hosts for cloud based services or certain classes for access within an operational network. An example of a class might be "devices of a specified manufacturer type", where the manufacturer type itself is indicated simply by the authority component (e.g, the domain name) of the MUD URL. Another example might be to allow or disallow local access. Just like other policies, these may be combined. For example:

- o Allow access to devices of the same manufacturer
- o Allow access to and from controllers via Constrained Application Protocol (COAP)[RFC7252]
- o Allow access to local DNS/NTP
- o Deny all other access

A printer might have a description that states:

- o Allow access for port IPP or port LPD
- o Allow local access for port HTTP
- o Deny all other access

In this way anyone can print to the printer, but local access would be required for the management interface.

The files that are retrieved are intended to be closely aligned to existing network architectures so that they are easy to deploy. We make use of YANG [RFC7950] because of the time and effort spent to develop accurate and adequate models for use by network devices. JSON is used as a serialization for compactness and readability, relative to XML. Other formats may be chosen with later versions of MUD.

While the policy examples given here focus on access control, this is not intended to be the sole focus. By structuring the model described in this document with clear extension points, other descriptions could be included. One that often comes to mind is quality of service.

The YANG modules specified here are extensions of [I-D.ietf-netmod-acl-model]. The extensions to this model allow for a manufacturer to express classes of systems that a manufacturer would find necessary for the proper function of the device. Two modules are specified. The first module specifies a means for domain names to be used in ACLs so that devices that have their controllers in the cloud may be appropriately authorized with domain names, where the mapping of those names to addresses may rapidly change.

The other module abstracts away IP addresses into certain classes that are instantiated into actual IP addresses through local processing. Through these classes, manufacturers can specify how the device is designed to communicate, so that network elements can be configured by local systems that have local topological knowledge. That is, the deployment populates the classes that the manufacturer specifies. The abstractions below map to zero or more hosts, as follows:

Manufacturer: A device made by a particular manufacturer, as identified by the authority component of its MUD URL

same-manufacturer: Devices that have the same authority component of their MUD URL.

Controller: Devices that the local network administrator admits to the particular class.

my-controller: Devices associated with the MUD URL of a device that the administrator admits.

local: The class of IP addresses that are scoped within some administrative boundary. By default it is suggested that this be the local subnet.

The "manufacturer" classes can be easily specified by the manufacturer, whereas controller classes are initially envisioned to be specified by the administrator.

Because manufacturers do not know who will be using their devices, it is important for functionality referenced in usage descriptions to be relatively ubiquitous and mature. For these reasons only a limited subset YANG-based configuration of is permitted in a MUD file.

1.6. Terminology

MUD: manufacturer usage description.

MUD file: a file containing YANG-based JSON that describes a Thing and associated suggested specific network behavior.

MUD file server: a web server that hosts a MUD file.

MUD controller: the system that requests and receives the MUD file from the MUD server. After it has processed a MUD file, it may direct changes to relevant network elements.

MUD URL: a URL that can be used by the MUD controller to receive the MUD file.

Thing: the device emitting a MUD URL.

Manufacturer: the entity that configures the Thing to emit the MUD URL and the one who asserts a recommendation in a MUD file. The manufacturer might not always be the entity that constructs a Thing. It could, for instance, be a systems integrator, or even a component provider.

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

1.7. The Manufacturer Usage Description Architecture

With these components laid out we now have the basis for an architecture. This leads us to ASCII art.

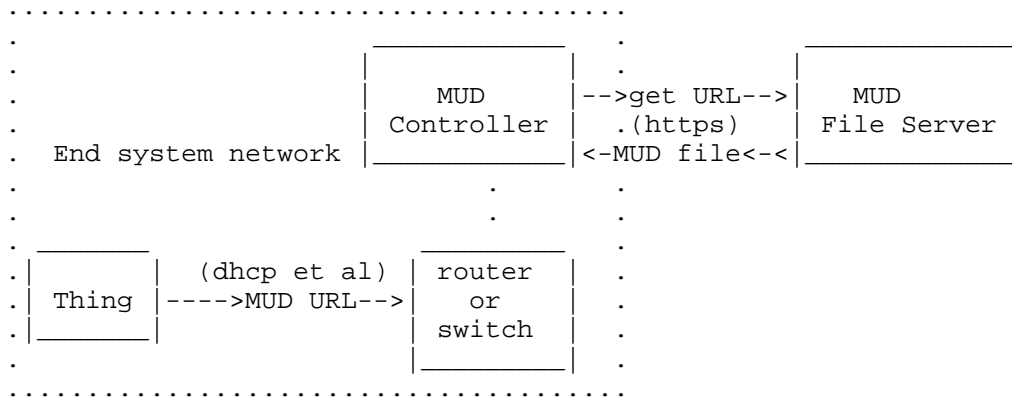


Figure 1: MUD Architecture

In the above diagram, the switch or router collects MUD URLs and forwards them to the network management system for processing. This happens in different ways, depending on how the URL is communicated. For instance, in the case of DHCP, the DHCP server might receive the URL and then process it. In the case of IEEE 802.1X, the switch would carry the URL via a certificate to the authentication server via EAP over Radius[RFC3748], which would then process it. One method to do this is TEAP, described in [RFC7170]. The certificate extension is described below.

The information returned by the web site is valid for the duration of the Thing's connection, or as specified in the description. Thus if the Thing is disconnected, any associated configuration in the switch can be removed. Similarly, from time to time the description may be refreshed, based on new capabilities or communication patterns or vulnerabilities.

The web site is typically run by or on behalf of the manufacturer. Its domain name is that of the authority found in the MUD URL. For legacy cases where Things cannot emit a URL, if the switch is able to determine the appropriate URL, it may proxy it, the trivial cases being a hardcoded MUD-URL on a switch port, or a mapping from some available identifier such as an L2 address or certificate hash to a MUD-URL.

The role of the MUD controller in this environment is to do the following:

- o receive MUD URLs,
- o retrieve MUD files,
- o translate abstractions in the MUD files to specific network element configuration,
- o maintain and update any required mappings of the abstractions, and
- o update network elements with appropriate configuration.

A MUD controller may be a component of a AAA or network management system. Communication within those systems and from those systems to network elements is beyond the scope of this memo.

1.8. Order of operations

As mentioned above, MUD contains architectural building blocks, and so order of operation may vary. However, here is one clear intended example:

1. Thing emits URL.
2. That URL is forwarded to a MUD controller by the nearest switch (how this happens depends on the way in which the MUD URL is emitted).
3. The MUD controller retrieves the MUD file and signature from the MUD file server, assuming it doesn't already have copies. After validating the signature, it may test the URL against a web or domain reputation service, and it may test any hosts within the file against those reputation services, as it deems fit.
4. The MUD controller may query the administrator for permission to add the Thing and associated policy. If the Thing is known or the Thing type is known, it may skip this step.
5. The MUD controller instantiates local configuration based on the abstractions defined in this document.
6. The MUD controller configures the switch nearest the Thing. Other systems may be configured as well.
7. When the Thing disconnects, policy is removed.

2. The MUD Model and Semantic Meaning

A MUD file consists of a YANG model that has been serialized in JSON [RFC7951]. For purposes of MUD, the nodes that can be modified are access lists as augmented by this model. The MUD file is limited to the serialization of only the following YANG schema:

- o ietf-access-control-list [I-D.ietf-netmod-acl-model]
- o ietf-mud (this document)
- o ietf-acldns (this document)

Extensions may be used to add additional schema. This is described further on.

To provide the widest possible deployment, publishers of MUD files SHOULD make use of the abstractions in this memo and avoid the use of IP addresses. A MUD controller SHOULD NOT automatically implement any MUD file that contains IP addresses, especially those that might have local significance. The addressing of one side of an access list is implicit, based on whether it is applied as to-device-policy or from-device-policy.

With the exceptions of "acl-name", "acl-type", "rule-name", and TCP and UDP source and destination port information, publishers of MUD files SHOULD limit the use of ACL model leaf nodes expressed to those found in this specification. Absent any extensions, MUD files are assumed to implement only the following ACL model features:

- o any-acl, mud-acl, icmp-acl, ipv6-acl, tcp-acl, any-acl, udp-acl, ipv4-acl, and ipv6-acl

Furthermore, only "accept" or "drop" actions SHOULD be included. A MUD controller MAY choose to interpret "reject" as "drop". A MUD controller SHOULD ignore all other actions.

In fact, MUD controllers MAY ignore any particular component of a description or MAY ignore the description in its entirety, and SHOULD carefully inspect all MUD descriptions. Publishers of MUD files MUST NOT include other nodes except as described in Section 3.6. See that section for more information.

2.1. The IETF-MUD YANG Module

This module is structured into three parts:

- o The first container "mud" holds information that is relevant to retrieval and validity of the MUD file itself, as well as policy intended to and from the Thing.
- o The second component augments the matching container of the ACL model to add several nodes that are relevant to the MUD URL, or otherwise abstracted for use within a local environment.
- o The third component augments the tcp-acl container of the ACL model to add the ability to match on the direction of initiation of a TCP connection.

A valid MUD file will contain two root objects, a "mud" container and an "access-lists" container. Extensions may add additional root objects as required. As a reminder, when parsing access-lists, elements within a "match" block are logically ANDed. In general, a single abstraction in a match statement should be used. For instance, it makes little sense to match both "my-controller" and "controller" with an argument, since they are highly unlikely to be the same value.

A simplified graphical representation of the data models is used in this document. The meaning of the symbols in these diagrams is explained in [I-D.ietf-netmod-rfc6087bis].

```

module: ietf-mud
  +--rw mud!
    +--rw mud-url          inet:uri
    +--rw last-update      yang:date-and-time
    +--rw cache-validity?  uint8
    +--rw is-supported     boolean
    +--rw systeminfo?     inet:uri
    +--rw extensions*      string
    +--rw from-device-policy
      +--rw access-lists
        +--rw access-list* [acl-name acl-type]
          +--rw acl-name    -> /acl:access-lists/acl/acl-name
          +--rw acl-type    identityref
    +--rw to-device-policy
      +--rw access-lists
        +--rw access-list* [acl-name acl-type]
          +--rw acl-name    -> /acl:access-lists/acl/acl-name
          +--rw acl-type    identityref
  augment /acl:access-lists/acl:acl/acl:aces/
    acl:ace/acl:matches:
      +--rw mud-acl
        +--rw manufacturer?  inet:host
        +--rw same-manufacturer? empty
        +--rw model?         inet:uri
        +--rw local-networks? empty
        +--rw controller?    inet:uri
        +--rw my-controller? empty
      augment /acl:access-lists/acl:acl/acl:aces/
        acl:ace/acl:matches/acl:tcp-acl:
          +--rw direction-initiated? direction

```

3. Data Node Definitions

Note that in this section, when we use the term "match" we are referring to the ACL model "matches" node, and thus returns positive such that an action should be applied.

The following nodes are defined.

3.1. to-device-policy and from-device-policy containers

[I-D.ietf-netmod-acl-model] describes access-lists but does not attempt to indicate where they are applied as that is handled elsewhere in a configuration. However, in this case, a MUD file must be explicit in describing the communication pattern of a Thing, and that includes indicating what is to be permitted or denied in either direction of communication. Hence each of these containers indicate

the appropriate direction of a flow in association with a particular Thing. They contain references to specific access-lists.

3.2. last-update

This is a date-and-time value of when the MUD file was generated. This is akin to a version number. Its form is taken from [RFC6991] which, for those keeping score, in turn was taken from Section 5.6 of [RFC3339], which was taken from [ISO.8601.1988].

3.3. cache-validity

This uint8 is the period of time in hours that a network management station MUST wait since its last retrieval before checking for an update. It is RECOMMENDED that this value be no less than 24 and MUST NOT be more than 168 for any Thing that is supported. This period SHOULD be no shorter than any period determined through HTTP caching directives (e.g., "cache-control" or "Expires"). N.B., expiring of this timer does not require the MUD controller to discard the MUD file, nor terminate access to a Thing. See Section 15 for more information.

3.4. is-supported

This boolean is an indication from the manufacturer to the network administrator as to whether or not the Thing is supported. In this context a Thing is said to NOT be supported if the manufacturer intends never to issue an update to the Thing or never update the MUD file. A MUD controller MAY still periodically check for updates.

3.5. systeminfo

This is a URL that points to a description of the Thing to be connected. The intent is for administrators to be able to see a localized name associated with the Thing. The referenced URL SHOULD be a localized display string, and MAY be in either HTML or a raw UTF-8 text file. It SHOULD NOT exceed 60 characters worth of display space (that is- what the administrator actually sees), but it MAY contain links to other documents (presumably product documentation).

3.6. extensions

This optional leaf-list names MUD extensions that are used in the MUD file. Note that NO MUD extensions may be used in a MUD file prior to the extensions being declared. Implementations MUST ignore any node in this file that they do not understand.

Note that extensions can either extend the MUD file as described in the previous paragraph, or they might reference other work. An extension example can be found in Appendix C.

3.7. manufacturer

This node consists of a hostname that would be matched against the authority component of another Thing's MUD URL. In its simplest form "manufacturer" and "same-manufacturer" may be implemented as access-lists. In more complex forms, additional network capabilities may be used. For example, if one saw the line "manufacturer" : "flobbidy.example.com", then all Things that registered with a MUD URL that contained flobbity.example.com in its authority section would match.

3.8. same-manufacturer

This is an equivalent for when the manufacturer element is used to indicate the authority that is found in another Thing's MUD URL matches that of the authority found in this Thing's MUD URL. For example, if the Thing's MUD URL were https://bl.example.com/.well-known/mud/v1/ThingV1, then all devices that had MUD URL with an authority section of bl.example.com would match.

3.9. model

This string matches the entire MUD URL, thus covering the model that is unique within the context of the authority. It may contain not only model information, but versioning information as well, and any other information that the manufacturer wishes to add. The intended use is for devices of this precise class to match, to permit or deny communication between one another.

3.10. local-networks

This null-valued node expands to include local networks. Its default expansion is that packets must not traverse toward a default route that is received from the router. However, administrators may expand the expression as is appropriate in their deployments.

3.11. controller

This URI specifies a value that a controller will register with the mud controller. The node then is expanded to the set of hosts that are so registered. This node may also be a URN. In this case, the URN describes a well known service, such as DNS or NTP.

Great care should be used when invoking the controller class. For one thing, it requires some understanding by the administrator as to when it is appropriate. Classes that are standardized may make it possible to easily name devices that support standard functions. For instance, the MUD controller could have some knowledge of which DNS servers should be used for any particular group of Things. Non-standard classes will likely require some sort of administrator interaction. Pre-registration in such classes by controllers with the MUD server is encouraged. The mechanism to do that is beyond the scope of this work.

Controller URIs MAY take the form of a URL (e.g. "http[s]://"). However, MUD controllers MUST NOT resolve and retrieve such files, and it is RECOMMENDED that there be no such file at this time, as their form and function may be defined at a point in the future. For now, URLs should serve simply as class names and be populated by the local deployment administrator.

3.12. my-controller

This null-valued node signals to the MUD controller to use whatever mapping it has for this MUD URL to a particular group of hosts. This may require prompting the administrator for class members. Future work should seek to automate membership management.

3.13. direction-initiated

When applied this matches packets when the flow was initiated in the corresponding direction. [RFC6092] specifies IPv6 guidance best practices. While that document is scoped specifically to IPv6, its contents are applicable for IPv4 as well. When this flag is set, and the system has no reason to believe a flow has been initiated it MUST drop the packet. This node may be implemented in its simplest form by looking at naked SYN bits, but may also be implemented through more stateful mechanisms.

4. Processing of the MUD file

To keep things relatively simple in addition to whatever definitions exist, we also apply two additional default behaviors:

- o Anything not explicitly permitted is denied.
- o Local DNS and NTP are, by default, permitted to and from the Thing.

An explicit description of the defaults can be found in Appendix B.

5. What does a MUD URL look like?

To begin with, MUD takes full advantage of both the https: scheme and the use of .well-known. HTTPS is important in this case because a man in the middle attack could otherwise harm the operation of a class of Things. .well-known is used because we wish to add additional structure to the URL, and want to leave open for future versions both the means by which the URL is processed and the format of the MUD file retrieved (there have already been some discussions along these lines). The URL appears as follows:

```
mud-url    = "https://" authority "/" .well-known/mud/" mud-rev
              "/" modelinfo ( "?" extras )
              ; authority is from RFC3986
mud-rev    = "v1"
modelinfo  = segment ; from RFC3986
extras     = query   ; from RFC3986
```

mud-rev signifies the version of the manufacturer usage description file. This memo specifies "v1" of that file. Later versions may permit additional schemas or modify the format. In order to provide for the broadest compatibility for the various transmission mechanisms, the length of the URL for v1 MUST NOT exceed 255 octets.

Taken together with the mud-url, "modelinfo" represents a Thing model as the manufacturer wishes to represent it. It could be a brand name or something more specific. It also may provide a means to indicate what version the product is. Specifically if it has been updated in the field, this is the place where evidence of that update would appear. The field should be changed when the intended communication patterns of a Thing change. While from a controller standpoint, only comparison and matching operations are safe, it is envisioned that updates will require some administrative review. Processing of this URL occurs as specified in [RFC2818] and [RFC3986].

"extras" is intended for use by the MUD controller to provide additional information such as posture about the Thing to the MUD file server. This field MUST NOT be configured on the Thing itself by a manufacturer - that is what "modelinfo" is for. It is left as future work to define the full semantics of this field.

6. The MUD YANG Model

```
<CODE BEGINS>file "ietf-mud@2017-10-07.yang"
module ietf-mud {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-mud";
```



```
prefix ietf-mud;

import ietf-access-control-list {
  prefix acl;
}
import ietf-yang-types {
  prefix yang;
}
import ietf-inet-types {
  prefix inet;
}

organization
  "IETF OPSAWG (Ops Area) Working Group";
contact
  "WG Web: http://tools.ietf.org/wg/opsawg/
  WG List: opsawg@ietf.org
  Author: Eliot Lear
  lear@cisco.com
  Author: Ralph Droms
  rdroms@gmail.com
  Author: Dan Romascanu
  dromasca@gmail.com

  ";
description
  "This YANG module defines a component that augments the
  IETF description of an access list.  This specific module
  focuses on additional filters that include local, model,
  and same-manufacturer.

  This module is intended to be serialized via JSON and stored
  as a file, as described in RFC XXXX [RFC Editor to fill in with
  this document #].

  Copyright (c) 2016,2017 IETF Trust and the persons
  identified as the document authors.  All rights reserved.
  Redistribution and use in source and binary forms, with or
  without modification, is permitted pursuant to, and subject
  to the license terms contained in, the Simplified BSD
  License set forth in Section 4.c of the IETF Trust's Legal
  Provisions Relating to IETF Documents
  (http://trustee.ietf.org/license-info).
  This version of this YANG module is part of RFC XXXX; see
  the RFC itself for full legal notices."

revision 2017-10-07 {
  description
```

```
    "Initial proposed standard.";
  reference
    "RFC XXXX: Manufacturer Usage Description
    Specification";
}

typedef direction {
  type enumeration {
    enum "to-device" {
      description
        "packets or flows destined to the target
        Thing";
    }
    enum "from-device" {
      description
        "packets or flows destined from
        the target Thing";
    }
  }
}
description
  "Which way are we talking about?";
}

container mud {
  presence "Enabled for this particular MUD URL";
  description
    "MUD related information, as specified
    by RFC-XXXX [RFC Editor to fill in].";
  uses mud-grouping;
}

grouping mud-grouping {
  description
    "Information about when support end(ed), and
    when to refresh";
  leaf mud-url {
    type inet:uri;
    mandatory true;
    description
      "This is the MUD URL associated with the entry found
      in a MUD file.";
  }
  leaf last-update {
    type yang:date-and-time;
    mandatory true;
    description
      "This is intended to be when the current MUD file
      was generated.  MUD Controllers SHOULD NOT check
```

```
        for updates between this time plus cache validity";
    }
    leaf cache-validity {
        type uint8 {
            range "1..168";
        }
        units "hours";
        default "48";
        description
            "The information retrieved from the MUD server is
            valid for these many hours, after which it should
            be refreshed.  N.B. MUD controller implementations
            need not discard MUD files beyond this period.";
    }
    leaf is-supported {
        type boolean;
        mandatory true;
        description
            "This boolean indicates whether or not the Thing is
            currently supported by the manufacturer.";
    }
    leaf systeminfo {
        type inet:uri;
        description
            "A URL to a description of this Thing.  This
            should be a brief localized description.  The
            reference text should be no more than octets.
            systeminfo may be displayed to the user to
            determine whether to allow the Thing on the
            network.";
    }
    leaf-list extensions {
        type string {
            length "1..40";
        }
        description
            "A list of extension names that are used in this MUD
            file.  Each name is registered with the IANA and
            described in an RFC.";
    }
    container from-device-policy {
        description
            "The policies that should be enforced on traffic
            coming from the device. These policies are not
            necessarily intended to be enforced at a single
            point, but may be rendered by the controller to any
            relevant enforcement points in the network or
            elsewhere.";
```

```
    uses access-lists;
  }
  container to-device-policy {
    description
      "The policies that should be enforced on traffic
      going to the device. These policies are not
      necessarily intended to be enforced at a single
      point, but may be rendered by the controller to any
      relevant enforcement points in the network or
      elsewhere.";
    uses access-lists;
  }
}

grouping access-lists {
  description
    "A grouping for access lists in the context of device
    policy.";
  container access-lists {
    description
      "The access lists that should be applied to traffic
      to or from the device.";
    list access-list {
      key "acl-name acl-type";
      description
        "Each entry on this list refers to an ACL that
        should be present in the overall access list
        data model. Each ACL is identified by name and
        type.";
      leaf acl-name {
        type leafref {
          path "/acl:access-lists/acl:acl/acl:acl-name";
        }
        description
          "The name of the ACL for this entry.";
      }
      leaf acl-type {
        type identityref {
          base acl:acl-base;
        }
        description
          "The type of the ACL for this entry. The name is
          scoped ONLY to the MUD file, and may not be unique
          in any other circumstance.";
      }
    }
  }
}
```

```
augment "/acl:access-lists/acl:acl/acl:aces/acl:ace/acl:matches" {
  description
    "adding abstractions to avoid need of IP addresses";
  container mud-acl {
    description
      "MUD-specific matches.";
    leaf manufacturer {
      type inet:host;
      description
        "A domain that is intended to match the authority
        section of the MUD URL. This node is used to specify
        one or more manufacturers a device should
        be authorized to access.";
    }
    leaf same-manufacturer {
      type empty;
      description
        "This node matches the authority section of the MUD URL
        of a Thing. It is intended to grant access to all
        devices with the same authority section.";
    }
    leaf model {
      type inet:uri;
      description
        "Devices of the specified model type will match if
        they have an identical MUD URL.";
    }
    leaf local-networks {
      type empty;
      description
        "IP addresses will match this node if they are
        considered local addresses. A local address may be
        a list of locally defined prefixes and masks
        that indicate a particular administrative scope.";
    }
    leaf controller {
      type inet:uri;
      description
        "This node names a class that has associated with it
        zero or more IP addresses to match against. These
        may be scoped to a manufacturer or via a standard
        URN.";
    }
    leaf my-controller {
      type empty;
      description
        "This node matches one or more network elements that
        have been configured to be the controller for this
```

```

        Thing, based on its MUD URL.";
    }
}
}
augment "/acl:access-lists/acl:acl/acl:aces/" +
  "acl:ace/acl:matches/acl:tcp-acl" {
  description
    "Adding domain names to matching";
  leaf direction-initiated {
    type direction;
    description
      "This node matches based on which direction a
       connection was initiated. The means by which that
       is determined is discussed in this document.";
  }
}
}
}
}

<CODE ENDS>

```

7. The Domain Name Extension to the ACL Model

This module specifies an extension to IETF-ACL model such that domain names may be referenced by augmenting the "matches" node. Different implementations may deploy differing methods to maintain the mapping between IP address and domain name, if indeed any are needed. However, the intent is that resources that are referred to using a name should be authorized (or not) within an access list.

The structure of the change is as follows:

```

module: ietf-acldns
  augment /acl:access-lists/acl:acl/acl:aces/acl:ace/
    acl:matches/acl:ipv4-acl:
      +--rw src-dnsname?   inet:host
      +--rw dst-dnsname?   inet:host
  augment /acl:access-lists/acl:acl/acl:aces/acl:ace/
    acl:matches/acl:ipv6-acl:
      +--rw src-dnsname?   inet:host
      +--rw dst-dnsname?   inet:host

```

The choice of these particular points in the access-list model is based on the assumption that we are in some way referring to IP-related resources, as that is what the DNS returns. A domain name in our context is defined in [RFC6991]. The augmentations are

replicated across IPv4 and IPv6 to allow MUD file authors the ability to control the IP version that the Thing may utilize.

The following node are defined.

7.1. source-dnsname

The argument corresponds to a domain name of a source as specified by `inet:host`. A number of means may be used to resolve hosts. What is important is that such resolutions be consistent with ACLs required by Things to properly operate.

7.2. destination-dnsname

The argument corresponds to a domain name of a destination as specified by `inet:host` See the previous section relating to resolution.

7.3. The ietf-acldns Model

```
<CODE BEGINS>file "ietf-acldns@2017-10-07.yang"
module ietf-acldns {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-acldns";
  prefix "ietf-acldns";

  import ietf-access-control-list {
    prefix "acl";
  }

  import ietf-inet-types {
    prefix "inet";
  }

  organization
    "IETF OPSAWG (Ops Area) Working Group";

  contact
    "WG Web: http://tools.ietf.org/wg/opsawg/
    WG List: opsawg@ietf.org
    Author: Eliot Lear
    lear@cisco.com
    Author: Ralph Droms
    rdroms@gmail.com
    Author: Dan Romascanu
    dromasca@gmail.com
    ";
```

```
description
  "This YANG module defines a component that augments the
   IETF description of an access list to allow dns names
   as matching criteria.";

revision "2017-10-07" {
  description "Base version of dnsname extension of ACL model";
  reference "RFC XXXX: Manufacturer Usage Description
            Specification";
}

grouping dns-matches {
  description "Domain names for matching.";

  leaf src-dnsname {
    type inet:host;
    description "domain name to be matched against";
  }
  leaf dst-dnsname {
    type inet:host;
    description "domain name to be matched against";
  }
}

augment "/acl:access-lists/acl:acl/acl:aces/acl:ace/" +
  "acl:matches/acl:ipv4-acl" {
  description "Adding domain names to matching";
  uses dns-matches;
}

augment "/acl:access-lists/acl:acl/" +
  "acl:aces/acl:ace/" +
  "acl:matches/acl:ipv6-acl" {
  description "Adding domain names to matching";
  uses dns-matches;
}
}
<CODE ENDS>
```

8. MUD File Example

This example contains two access lists that are intended to provide outbound access to a cloud service on TCP port 443.

```
{
```



```

"ietf-mud:mud": {
  "mud-url":
"https://bms.example.com/.well-known/mud/v1/lightbulb2000",
  "last-update": "2017-10-07T12:16:24+02:00",
  "cache-validity": 48,
  "is-supported": true,
  "systeminfo":
"https://bms.example.com/descriptions/lightbulb2000",
  "from-device-policy": {
    "access-lists": {
      "access-list": [
        {
          "acl-name": "mud-14377-v6fr",
          "acl-type": "ietf-access-control-list:ipv6-acl"
        }
      ]
    }
  },
  "to-device-policy": {
    "access-lists": {
      "access-list": [
        {
          "acl-name": "mud-14377-v6to",
          "acl-type": "ietf-access-control-list:ipv6-acl"
        }
      ]
    }
  }
},
"ietf-access-control-list:access-lists": {
  "acl": [
    {
      "acl-name": "mud-14377-v6to",
      "acl-type": "ipv6-acl",
      "access-list-entries": {
        "ace": [
          {
            "rule-name": "cl0-todev",
            "matches": {
              "ipv6-acl": {
                "ietf-acldns:src-dnsname":
"service.bms.example.com",
                "protocol": 6,
                "source-port-range": {
                  "lower-port": 443,
                  "upper-port": 443
                }
              }
            }
          }
        ]
      }
    }
  ],

```

```

        "tcp-acl": {
            "ietf-mud:direction-initiated": "from-device"
        },
        "actions": {
            "forwarding": "accept"
        }
    ]
}
},
{
    "acl-name": "mud-14377-v6fr",
    "acl-type": "ipv6-acl",
    "access-list-entries": {
        "ace": [
            {
                "rule-name": "cl0-frdev",
                "matches": {
                    "ipv6-acl": {
                        "ietf-acldns:dst-dnsname":
                            "service.bms.example.com",
                        "protocol": 6,
                        "destination-port-range": {
                            "lower-port": 443,
                            "upper-port": 443
                        }
                    }
                },
                "tcp-acl": {
                    "ietf-mud:direction-initiated": "from-device"
                }
            },
            {
                "actions": {
                    "forwarding": "accept"
                }
            }
        ]
    }
}
]
}
}
}

```

In this example, two policies are declared, one from the Thing and the other to the Thing. Each policy names an access list that applies to the Thing, and one that applies from. Within each access list, access is permitted to packets flowing to or from the Thing

that can be mapped to the domain name of "service.bms.example.com". For each access list, the enforcement point should expect that the Thing initiated the connection.

9. The MUD URL DHCP Option

The IPv4 MUD URL client option has the following format:

```
+-----+-----+-----+
| code | len | MUD URL
+-----+-----+-----+
```

Code `OPTION_MUD_URL_V4` (161) is assigned by IANA. `len` is a single octet that indicates the length of the URL in octets. MUD URL is a URL. MUD URLs MUST NOT exceed 255 octets.

The IPv6 MUD URL client option has the following format:

```

0          1          2          3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+
|          OPTION_MUD_URL_V6          |          option-length          |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     MUD URL                                     |
|                                     ...                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+
```

`OPTION_MUD_URL_V6` (112; assigned by IANA).

`option-length` contains the length of the URL in octets.

The intent of this option is to provide both a new Thing classifier to the network as well as some recommended configuration to the routers that implement policy. However, it is entirely the purview of the network system as managed by the network administrator to decide what to do with this information. The key function of this option is simply to identify the type of Thing to the network in a structured way such that the policy can be easily found with existing toolsets.

9.1. Client Behavior

A DHCPv4 client MAY emit a DHCPv4 option and a DHCPv6 client MAY emit DHCPv6 option. These options are singletons, as specified in [RFC7227]. Because clients are intended to have at most one MUD URL associated with them, they may emit at most one MUD URL option via

DHCPv4 and one MUD URL option via DHCPv6. In the case where both v4 and v6 DHCP options are emitted, the same URL MUST be used.

Clients SHOULD log or otherwise report improper acknowledgments from servers, but they MUST NOT modify their MUD URL configuration based on a server's response. The server's response is only an acknowledgment that the server has processed the option, and promises no specific network behavior to the client. In particular, it may not be possible for the server to retrieve the file associated with the MUD URL, or the local network administration may not wish to use the usage description. Neither of these situations should be considered in any way exceptional.

9.2. Server Behavior

A DHCP server may ignore these options or take action based on receipt of these options. If a server successfully parses the option and the URL, it MUST return the option with length field set to zero and a corresponding null URL field as an acknowledgment. Even in this circumstance, no specific network behavior is guaranteed. When a server consumes this option, it will either forward the URL and relevant client information (such as the gateway address or giaddr) to a network management system, or it will retrieve the usage description itself by resolving the URL.

DHCP servers may implement MUD functionality themselves or they may pass along appropriate information to a network management system or MUD controller. A DHCP server that does process the MUD URL MUST adhere to the process specified in [RFC2818] and [RFC5280] to validate the TLS certificate of the web server hosting the MUD file. Those servers will retrieve the file, process it, create and install the necessary configuration on the relevant network element. Servers SHOULD monitor the gateway for state changes on a given interface. A DHCP server that does not provide MUD functionality and has forwarded a MUD URL to a MUD controller MUST notify the MUD controller of any corresponding change to the DHCP state of the client (such as expiration or explicit release of a network address lease).

9.3. Relay Requirements

There are no additional requirements for relays.

10. The Manufacturer Usage Description (MUD) URL X.509 Extension

This section defines an X.509 non-critical certificate extension that contains a single Uniform Resource Locator (URL) that points to an on-line Manufacturer Usage Description concerning the certificate

subject. URI must be represented as described in Section 7.4 of [RFC5280].

Any Internationalized Resource Identifiers (IRIs) MUST be mapped to URIs as specified in Section 3.1 of [RFC3987] before they are placed in the certificate extension.

The semantics of the URL are defined Section 5 of this document.

The choice of id-pe is based on guidance found in Section 4.2.2 of [RFC5280]:

These extensions may be used to direct applications to on-line information about the issuer or the subject.

The MUD URL is precisely that: online information about the particular subject.

The new extension is identified as follows:

<CODE BEGINS>

```

MUDURLExtnModule-2016 { iso(1) identified-organization(3) dod(6)
                        internet(1) security(5) mechanisms(5) pkix(7)
                        id-mod(0) id-mod-mudURLExtn2016(88) }

DEFINITIONS IMPLICIT TAGS ::= BEGIN

-- EXPORTS ALL --

IMPORTS
    EXTENSION
    FROM PKIX-CommonTypes-2009
        { iso(1) identified-organization(3) dod(6) internet(1)
          security(5) mechanisms(5) pkix(7) id-mod(0)
          id-mod-pkixCommon-02(57) }

    id-pe
    FROM PKIX1Explicit-2009
        { iso(1) identified-organization(3) dod(6) internet(1)
          security(5) mechanisms(5) pkix(7) id-mod(0)
          id-mod-pkix1-explicit-02(51) } ;
    MUDCertExtensions EXTENSION ::= { ext-MUDURL, ... }
    ext-MUDURL EXTENSION ::= { SYNTAX MUDURLSyntax
    IDENTIFIED BY id-pe-mud-url }

    id-pe-mud-url OBJECT IDENTIFIER ::= { id-pe 25 }

    MUDURLSyntax ::= IA5String

END

```

<CODE ENDS>

While this extension can appear in either an 802.AR manufacturer certificate (IDevID) or deployment certificate (LDevID), of course it is not guaranteed in either, nor is it guaranteed to be carried over. It is RECOMMENDED that MUD controller implementations maintain a table that maps a Thing to its MUD URL based on IDevIDs.

11. The Manufacturer Usage Description LLDP extension

The IEEE802.1AB Link Layer Discovery Protocol (LLDP) is a one hop vendor-neutral link layer protocol used by end hosts network Things for advertising their identity, capabilities, and neighbors on an IEEE 802 local area network. Its Type-Length-Value (TLV) design allows for 'vendor-specific' extensions to be defined. IANA has a registered IEEE 802 organizationally unique identifier (OUI) defined

as documented in [RFC7042]. The MUD LLDP extension uses a subtype defined in this document to carry the MUD URL.

The LLDP vendor specific frame has the following format:

TLV Type	len	OUI	subtype	MUD URL
=127		= 00 00 5E	= 1	
(7 bits)	(9 bits)	(3 octets)	(1 octet)	(1-255 octets)

where:

- o TLV Type = 127 indicates a vendor-specific TLV
- o len - indicates the TLV string length
- o OUI = 00 00 5E is the organizationally unique identifier of IANA
- o subtype = 1 (to be assigned by IANA for the MUD URL)
- o MUD URL - the length MUST NOT exceed 255 octets

The intent of this extension is to provide both a new Thing classifier to the network as well as some recommended configuration to the routers that implement policy. However, it is entirely the purview of the network system as managed by the network administrator to decide what to do with this information. The key function of this extension is simply to identify the type of Thing to the network in a structured way such that the policy can be easily found with existing toolsets.

Hosts, routers, or other network elements that implement this option are intended to have at most one MUD URL associated with them, so they may transmit at most one MUD URL value.

Hosts, routers, or other network elements that implement this option may ignore these options or take action based on receipt of these options. For example they may fill in information in the respective extensions of the LLDP Management Information Base (LLDP MIB). LLDP operates in a one-way direction. LLDPDUs are not exchanged as information requests by one Thing and response sent by another Thing. The other Things do not acknowledge LLDP information received from a Thing. No specific network behavior is guaranteed. When a Thing consumes this extension, it may either forward the URL and relevant remote Thing information to a MUD controller, or it will retrieve the

usage description by resolving the URL in accordance with normal HTTP semantics.

12. Creating and Processing of Signed MUD Files

Because MUD files contain information that may be used to configure network access lists, they are sensitive. To insure that they have not been tampered with, it is important that they be signed. We make use of DER-encoded Cryptographic Message Syntax (CMS) [RFC5652] for this purpose.

12.1. Creating a MUD file signature

A MUD file MUST be signed using CMS as an opaque binary object. In order to make successful verification more likely, intermediate certificates SHOULD be included. The signature is stored at the same location as the MUD URL but with the suffix of ".p7s". Signatures are transferred using content-type "application/pkcs7-signature".

For example:

```
% openssl cms -sign -signer mancertfile -inkey mankey \  
-in mudfile -binary -outform DER - \  
-certfile intermediatecert -out mudfile.p7s
```

Note: A MUD file may need to be re-signed if the signature expires.

12.2. Verifying a MUD file signature

Prior to retrieving a MUD file the MUD controller SHOULD retrieve the MUD signature file using the MUD URL with a suffix of ".p7s". For example, if the MUD URL is "https://example.com/.well-known/v1/modela", the MUD signature URL will be "https://example.com/.well-known/v1/modela.p7s".

Upon retrieving a MUD file, a MUD controller MUST validate the signature of the file before continuing with further processing. A MUD controller MUST cease processing of that file if it cannot validate the chain of trust to a known trust anchor until an administrator has given approval.

The purpose of the signature on the file is to assign accountability to an entity, whose reputation can be used to guide administrators on whether or not to accept a given MUD file. It is already common place to check web reputation on the location of a server on which a file resides. While it is likely that the manufacturer will be the signer of the file, this is not strictly necessary, and may not be desirable. For one thing, in some environments, integrators may

install their own certificates. For another, what is more important is the accountability of the recommendation, and not the cryptographic relationship between the device and the file.

An example:

```
% openssl cms -verify -in mudfile.p7s -inform DER -content mudfile
```

Note the additional step of verifying the common trust root.

13. Extensibility

One of our design goals is to see that MUD files are able to be understood by as broad a cross-section of systems as is possible. Coupled with the fact that we have also chosen to leverage existing mechanisms, we are left with no ability to negotiate extensions and a limited desire for those extensions in any event. As such, a two-tier extensibility framework is employed, as follows:

1. At a coarse grain, a protocol version is included in a MUD URL. This memo specifies MUD version 1. Any and all changes are entertained when this version is bumped. Transition approaches between versions would be a matter for discussion in future versions.
2. At a finer grain, only extensions that would not incur additional risk to the Thing are permitted. Specifically, adding nodes to the mud container is permitted with the understanding that such additions will be ignored by unaware implementations. Any such extensions SHALL be standardized through the IETF process, and MUST be named in the "extensions" list. MUD controllers MUST ignore YANG nodes they do not understand and SHOULD create an exception to be resolved by an administrator, so as to avoid any policy inconsistencies.

14. Deployment Considerations

Because MUD consists of a number of architectural building blocks, it is possible to assemble different deployment scenarios. One key aspect is where to place policy enforcement. In order to protect the Thing from other Things within a local deployment, policy can be enforced on the nearest switch or access point. In order to limit unwanted traffic within a network, it may also be advisable to enforce policy as close to the Internet as possible. In some circumstances, policy enforcement may not be available at the closest hop. At that point, the risk of so-called east-west infection is increased to the number of Things that are able to communicate without protection.

A caution about some of the classes: admission of a Thing into the "manufacturer" and "same-manufacturer" class may have impact on access of other Things. Put another way, the admission may grow the access-list on switches connected to other Things, depending on how access is managed. Some care should be given on managing that access-list growth. Alternative methods such as additional network segmentation can be used to keep that growth within reason.

Because as of this writing MUD is a new concept, one can expect a great many devices to not have implemented it. It remains a local deployment decision as to whether a device that is first connected should be allowed broad or limited access. Furthermore, as mentioned in the introduction, a deployment may choose to ignore a MUD policy in its entirety, but simply taken into account the MUD URL as a classifier to be used as part of a local policy decision.

15. Security Considerations

Based on how a MUD URL is emitted, a Thing may be able to lie about what it is, thus gaining additional network access. There are several means to limit risk in this case. The most obvious is to only believe Things that make use of certificate-based authentication such as IEEE 802.1AR certificates. When those certificates are not present, Things claiming to be of a certain manufacturer SHOULD NOT be included in that manufacturer grouping without additional validation of some form. This will occur when it makes use of primitives such as "manufacturer" for the purpose of accessing Things of a particular type. Similarly, network management systems may be able to fingerprint the Thing. In such cases, the MUD URL can act as a classifier that can be proven or disproven. Fingerprinting may have other advantages as well: when 802.1AR certificates are used, because they themselves cannot change, fingerprinting offers the opportunity to add artifacts to the MUD URL. The meaning of such artifacts is left as future work.

Network management systems SHOULD NOT accept a usage description for a Thing with the same MAC address that has indicated a change of authority without some additional validation (such as review by a network administrator). New Things that present some form of unauthenticated MUD URL SHOULD be validated by some external means when they would be otherwise be given increased network access.

It may be possible for a rogue manufacturer to inappropriately exercise the MUD file parser, in order to exploit a vulnerability. There are three recommended approaches to address this threat. The first is to validate the signature of the MUD file. The second is to have a system do a primary scan of the file to ensure that it is both parseable and believable at some level. MUD files will likely be

relatively small, to start with. The number of ACEs used by any given Thing should be relatively small as well. It may also be useful to limit retrieval of MUD URLs to only those sites that are known to have decent web or domain reputations.

Use of a URL necessitates the use of domain names. If a domain name changes ownership, the new owner of that domain may be able to provide MUD files that MUD controllers would consider valid. There are a few approaches that can mitigate this attack. First, MUD controllers SHOULD cache certificates used by the MUD file server. When a new certificate is retrieved for whatever reason, the MUD controller should check to see if ownership of the domain has changed. A fair programmatic approximation of this is when the name servers for the domain have changed. If the actual MUD file has changed, the controller MAY check the WHOIS database to see if registration ownership of a domain has changed. If a change has occurred, or if for some reason it is not possible to determine whether ownership has changed, further review may be warranted. Note, this remediation does not take into account the case of a Thing that was produced long ago and only recently fielded, or the case where a new MUD controller has been installed.

It may not be possible for a MUD controller to retrieve a MUD file at any given time. Should a MUD controller fail to retrieve a MUD file, it SHOULD consider the existing one safe to use, at least for a time. After some period, it SHOULD log that it has been unable to retrieve the file. There may be very good reasons for such failures, including the possibility that the MUD controller is in an off-line environment, the local Internet connection has failed, or the remote Internet connection has failed. It is also possible that an attacker is attempting to prevent onboarding of a device. It is a local deployment decision as to whether or not devices may be onboarded in the face of such failures.

The release of a MUD URL by a Thing reveals what the Thing is, and provides an attacker with guidance on what vulnerabilities may be present.

While the MUD URL itself is not intended to be unique to a specific Thing, the release of the URL may aid an observer in identifying individuals when combined with other information. This is a privacy consideration.

In addressing both of these concerns, implementors should take into account what other information they are advertising through mechanisms such as mDNS[RFC6872], how a Thing might otherwise be identified, perhaps through how it behaves when it is connected to the network, whether a Thing is intended to be used by individuals or

carry personal identifying information, and then apply appropriate data minimization techniques. One approach is to make use of TEAP [RFC7170] as the means to share information with authorized components in the network. Network elements may also assist in limiting access to the MUD URL through the use of mechanisms such as DHCPv6-Shield [RFC7610].

Please note that the security considerations mentioned in Section 4.7 of [I-D.ietf-netmod-rfc6087bis] are not applicable in this case because the YANG serialization is not intended to be accessed via NETCONF. However, for those who try to instantiate this model in a network element via NETCONF, all objects in each model in this draft exhibit similar security characteristics as [I-D.ietf-netmod-acl-model]. The basic purpose of MUD is to configure access, and so by its very nature can be disruptive if used by unauthorized parties.

16. IANA Considerations

16.1. YANG Module Registrations

The following YANG modules are requested to be registered in the "IANA Module Names" registry:

The ietf-mud module:

- o Name: ietf-mud
- o XML Namespace: urn:ietf:params:xml:ns:yang:ietf-mud
- o Prefix: ief-mud
- o Reference: This memo

The ietf-acldns module:

- o Name: ietf-acldns
- o XML Namespace: urn:ietf:params:xml:ns:yang:ietf-acldns
- o Prefix: ietf-acldns
- o Reference: This memo

16.2. DHCPv4 and DHCPv6 Options

The IANA has allocated option 161 in the Dynamic Host Configuration Protocol (DHCP) and Bootstrap Protocol (BOOTP) Parameters registry for the MUD DHCPv4 option.

IANA is requested to allocated the DHCPv4 and v6 options as specified in Section 9.

16.3. PKIX Extensions

IANA is kindly requested to make the following assignments for:

- o The MUDURLExtnModule-2016 ASN.1 module in the "SMI Security for PKIX Module Identifier" registry (1.3.6.1.5.5.7.0).

- o id-pe-mud-url object identifier from the "SMI Security for PKIX Certificate Extension" registry (1.3.6.1.5.5.7.1).

The use of these values is specified in Section 10.

16.4. Well Known URI Suffix

The IANA has allocated the URL suffix of "mud" as follows:

- o URI Suffix: "mud" o Specification documents: this document o Related information: n/a

16.5. MIME Media-type Registration for MUD files

The following media-type is defined for transfer of MUD file:

- o Type name: application
- o Subtype name: mud+json
- o Required parameters: n/a
- o Optional parameters: n/a
- o Encoding considerations: 8bit; application/mud+json values are represented as a JSON object; UTF-8 encoding SHOULD be employed.
- o Security considerations: See Security Considerations of this document.
- o Interoperability considerations: n/a
- o Published specification: this document
- o Applications that use this media type: MUD controllers as specified by this document.
- o Fragment identifier considerations: n/a
- o Additional information:
 - Magic number(s): n/a
 - File extension(s): n/a
 - Macintosh file type code(s): n/a
- o Person & email address to contact for further information: Eliot Lear <lear@cisco.com>, Ralph Droms <rdroms@cisco.com>
- o Intended usage: COMMON
- o Restrictions on usage: none
- o Author:
 - Eliot Lear <lear@cisco.com>
 - Ralph Droms <rdroms@cisco.com>
- o Change controller: IESG
- o Provisional registration? (standards tree only): No.

16.6. LLDP IANA TLV Subtype Registry

IANA is requested to create a new registry for IANA Link Layer Discovery Protocol (LLDP) TLV subtype values. The recommended policy for this registry is Expert Review. The maximum number of entries in the registry is 256.

IANA is required to populate the initial registry with the value:

LLDP subtype value = 1 (All the other 255 values should be initially marked as 'Unassigned'.)

Description = the Manufacturer Usage Description (MUD) Uniform Resource Locator (URL)

Reference = < this document >

16.7. The MUD Well Known Universal Resource Name (URNs)

The following parameter registry is requested to be added in accordance with [RFC3553]

Registry name: "urn:ietf:params:mud" is requested.
Specification: this document
Repository: this document
Index value: Encoded identically to a TCP/UDP port service name, as specified in Section 5.1 of [RFC6335]

The following entries should be added to the "urn:ietf:params:mud" name space:

"urn:ietf:params:mud:dns" refers to the service specified by [RFC1123]. "urn:ietf:params:mud:ntp" refers to the service specified by [RFC5905].

16.8. Extensions Registry

The IANA is requested to establish a registry of extensions as follows:

Registry name: MUD extensions registry
Registry policy: Standards action
Standard reference: document
Extension name: UTF-8 encoded string, not to exceed 40 characters.

Each extension MUST follow the rules specified in this specification. As is usual, the IANA issues early allocations based in accordance with [RFC7120].

17. Acknowledgments

The authors would like to thank Einar Nilsen-Nygaard, who singlehandedly updated the model to match the updated ACL model, Bernie Volz, Tom Gindin, Brian Weis, Sandeep Kumar, Thorsten Dahm, John Bashinski, Steve Rich, Jim Bieda, Dan Wing, Joe Clarke, Henk Birkholz, Adam Montville, and Robert Sparks for their valuable advice and reviews. Russ Housley entirely rewrote Section 10 to be a complete module. Adrian Farrel provided the basis for privacy considerations text. Kent Watsen provided a thorough review of the architecture and the YANG model. The remaining errors in this work are entirely the responsibility of the authors.

18. References

18.1. Normative References

- [I-D.ietf-netmod-acl-model]
Jethanandani, M., Huang, L., Agarwal, S., and D. Blair,
"Network Access Control List (ACL) YANG Data Model",
draft-ietf-netmod-acl-model-14 (work in progress), October
2017.
- [IEEE8021AB]
Institute for Electrical and Electronics Engineers, "IEEE
Standard for Local and Metropolitan Area Networks--
Station and Media Access Control Connectivity Discovery",
n.d..
- [RFC1123] Braden, R., Ed., "Requirements for Internet Hosts -
Application and Support", STD 3, RFC 1123,
DOI 10.17487/RFC1123, October 1989, <<https://www.rfc-editor.org/info/rfc1123>>.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate
Requirement Levels", BCP 14, RFC 2119,
DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC2131] Droms, R., "Dynamic Host Configuration Protocol",
RFC 2131, DOI 10.17487/RFC2131, March 1997,
<<https://www.rfc-editor.org/info/rfc2131>>.
- [RFC2818] Rescorla, E., "HTTP Over TLS", RFC 2818,
DOI 10.17487/RFC2818, May 2000, <<https://www.rfc-editor.org/info/rfc2818>>.
- [RFC3315] Droms, R., Ed., Bound, J., Volz, B., Lemon, T., Perkins,
C., and M. Carney, "Dynamic Host Configuration Protocol
for IPv6 (DHCPv6)", RFC 3315, DOI 10.17487/RFC3315, July
2003, <<https://www.rfc-editor.org/info/rfc3315>>.
- [RFC3748] Aboba, B., Blunk, L., Vollbrecht, J., Carlson, J., and H.
Levkowetz, Ed., "Extensible Authentication Protocol
(EAP)", RFC 3748, DOI 10.17487/RFC3748, June 2004,
<<https://www.rfc-editor.org/info/rfc3748>>.
- [RFC3986] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform
Resource Identifier (URI): Generic Syntax", STD 66,
RFC 3986, DOI 10.17487/RFC3986, January 2005,
<<https://www.rfc-editor.org/info/rfc3986>>.

- [RFC3987] Duerst, M. and M. Suignard, "Internationalized Resource Identifiers (IRIs)", RFC 3987, DOI 10.17487/RFC3987, January 2005, <<https://www.rfc-editor.org/info/rfc3987>>.
- [RFC5280] Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., and W. Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 5280, DOI 10.17487/RFC5280, May 2008, <<https://www.rfc-editor.org/info/rfc5280>>.
- [RFC5652] Housley, R., "Cryptographic Message Syntax (CMS)", STD 70, RFC 5652, DOI 10.17487/RFC5652, September 2009, <<https://www.rfc-editor.org/info/rfc5652>>.
- [RFC5905] Mills, D., Martin, J., Ed., Burbank, J., and W. Kasch, "Network Time Protocol Version 4: Protocol and Algorithms Specification", RFC 5905, DOI 10.17487/RFC5905, June 2010, <<https://www.rfc-editor.org/info/rfc5905>>.
- [RFC6335] Cotton, M., Eggert, L., Touch, J., Westerlund, M., and S. Cheshire, "Internet Assigned Numbers Authority (IANA) Procedures for the Management of the Service Name and Transport Protocol Port Number Registry", BCP 165, RFC 6335, DOI 10.17487/RFC6335, August 2011, <<https://www.rfc-editor.org/info/rfc6335>>.
- [RFC6991] Schoenwaelder, J., Ed., "Common YANG Data Types", RFC 6991, DOI 10.17487/RFC6991, July 2013, <<https://www.rfc-editor.org/info/rfc6991>>.
- [RFC7120] Cotton, M., "Early IANA Allocation of Standards Track Code Points", BCP 100, RFC 7120, DOI 10.17487/RFC7120, January 2014, <<https://www.rfc-editor.org/info/rfc7120>>.
- [RFC7227] Hankins, D., Mrugalski, T., Siodelski, M., Jiang, S., and S. Krishnan, "Guidelines for Creating New DHCPv6 Options", BCP 187, RFC 7227, DOI 10.17487/RFC7227, May 2014, <<https://www.rfc-editor.org/info/rfc7227>>.
- [RFC7610] Gont, F., Liu, W., and G. Van de Velde, "DHCPv6-Shield: Protecting against Rogue DHCPv6 Servers", BCP 199, RFC 7610, DOI 10.17487/RFC7610, August 2015, <<https://www.rfc-editor.org/info/rfc7610>>.
- [RFC7950] Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language", RFC 7950, DOI 10.17487/RFC7950, August 2016, <<https://www.rfc-editor.org/info/rfc7950>>.

- [RFC7951] Lhotka, L., "JSON Encoding of Data Modeled with YANG", RFC 7951, DOI 10.17487/RFC7951, August 2016, <<https://www.rfc-editor.org/info/rfc7951>>.

18.2. Informative References

- [FW95] Chapman, D. and E. Zwicky, "Building Internet Firewalls", January 1995.
- [I-D.ietf-netmod-rfc6087bis] Bierman, A., "Guidelines for Authors and Reviewers of YANG Data Model Documents", draft-ietf-netmod-rfc6087bis-14 (work in progress), September 2017.
- [IEEE8021AR] Institute for Electrical and Electronics Engineers, "Secure Device Identity", 1998.
- [ISO.8601.1988] International Organization for Standardization, "Data elements and interchange formats - Information interchange - Representation of dates and times", ISO Standard 8601, June 1988.
- [RFC1984] IAB and IESG, "IAB and IESG Statement on Cryptographic Technology and the Internet", BCP 200, RFC 1984, DOI 10.17487/RFC1984, August 1996, <<https://www.rfc-editor.org/info/rfc1984>>.
- [RFC3339] Klyne, G. and C. Newman, "Date and Time on the Internet: Timestamps", RFC 3339, DOI 10.17487/RFC3339, July 2002, <<https://www.rfc-editor.org/info/rfc3339>>.
- [RFC3553] Mealling, M., Masinter, L., Hardie, T., and G. Klyne, "An IETF URN Sub-namespace for Registered Protocol Parameters", BCP 73, RFC 3553, DOI 10.17487/RFC3553, June 2003, <<https://www.rfc-editor.org/info/rfc3553>>.
- [RFC6092] Woodyatt, J., Ed., "Recommended Simple Security Capabilities in Customer Premises Equipment (CPE) for Providing Residential IPv6 Internet Service", RFC 6092, DOI 10.17487/RFC6092, January 2011, <<https://www.rfc-editor.org/info/rfc6092>>.

- [RFC6872] Gurbani, V., Ed., Burger, E., Ed., Anjali, T., Abdelnur, H., and O. Festor, "The Common Log Format (CLF) for the Session Initiation Protocol (SIP): Framework and Information Model", RFC 6872, DOI 10.17487/RFC6872, February 2013, <<https://www.rfc-editor.org/info/rfc6872>>.
- [RFC7042] Eastlake 3rd, D. and J. Abley, "IANA Considerations and IETF Protocol and Documentation Usage for IEEE 802 Parameters", BCP 141, RFC 7042, DOI 10.17487/RFC7042, October 2013, <<https://www.rfc-editor.org/info/rfc7042>>.
- [RFC7170] Zhou, H., Cam-Winget, N., Salowey, J., and S. Hanna, "Tunnel Extensible Authentication Protocol (TEAP) Version 1", RFC 7170, DOI 10.17487/RFC7170, May 2014, <<https://www.rfc-editor.org/info/rfc7170>>.
- [RFC7252] Shelby, Z., Hartke, K., and C. Bormann, "The Constrained Application Protocol (CoAP)", RFC 7252, DOI 10.17487/RFC7252, June 2014, <<https://www.rfc-editor.org/info/rfc7252>>.
- [RFC7452] Tschofenig, H., Arkko, J., Thaler, D., and D. McPherson, "Architectural Considerations in Smart Object Networking", RFC 7452, DOI 10.17487/RFC7452, March 2015, <<https://www.rfc-editor.org/info/rfc7452>>.
- [RFC7488] Boucadair, M., Penno, R., Wing, D., Patil, P., and T. Reddy, "Port Control Protocol (PCP) Server Selection", RFC 7488, DOI 10.17487/RFC7488, March 2015, <<https://www.rfc-editor.org/info/rfc7488>>.

Appendix A. Changes from Earlier Versions

RFC Editor to remove this section prior to publication.

Draft -10 to -12:

These are based on WGLC comments:

- o Correct examples based on ACL model changes.
- o Change ordering nodes.
- o Additional explanatory text around systeminfo.
- o Change ordering in examples.

- o Make it VERY VERY VERY VERY clear that these are recommendations, not mandates.
- o DHCP -> NTP in some of the intro text.
- o Remove masa-server
- o "Things" to "network elements" in a few key places.
- o Reference to JSON YANG RFC added.

Draft -10 to -11:

- o Example corrections
- o Typo
- o Fix two lists.
- o Addition of 'any-acl' and 'mud-acl' in the list of allowed features.
- o Clarification of what should be in a MUD file.

Draft -09 to -10:

- o AD input.
- o Correct dates.
- o Add compliance sentence as to which ACL module features are implemented.

Draft -08 to -09:

- o Resolution of Security Area review, IoT directorate review, GenART review, YANG doctors review.
- o change of YANG structure to address mandatory nodes.
- o Terminology cleanup.
- o specify out extra portion of MUD-URL.
- o consistency changes.
- o improved YANG descriptions.

- o Remove extra revisions.
- o Track ACL model changes.
- o Additional cautions on use of ACL model; further clarifications on extensions.

Draft -07 to -08:

- o a number of editorials corrected.
- o definition of MUD file tweaked.

Draft -06 to -07:

- o Examples updated.
- o Additional clarification for direction-initiated.
- o Additional implementation guidance given.

Draft -06 to -07:

- o Update models to match new ACL model
- o extract directionality from the ACL, introducing a new device container.

Draft -05 to -06:

- o Make clear that this is a component architecture (Polk and Watson)
- o Add order of operations (Watson)
- o Add extensions leaf-list (Pritikin)
- o Remove previous-mud-file (Watson)
- o Modify text in last-update (Watson)
- o Clarify local networks (Weis, Watson)
- o Fix contact info (Watson)
- o Terminology clarification (Weis)
- o Advice on how to handle LDevIDs (Watson)

- o Add deployment considerations (Watson)
- o Add some additional text about fingerprinting (Watson)
- o Appropriate references to 6087bis (Watson)
- o Change systeminfo to a URL to be referenced (Lear)

Draft -04 to -05: * syntax error correction

Draft -03 to -04: * Re-add my-controller

Draft -02 to -03: * Additional IANA updates * Format correction in YANG. * Add reference to TEAP.

Draft -01 to -02: * Update IANA considerations * Accept Russ Housley rewrite of X.509 text * Include privacy considerations text * Redo the URL limit. Still 255 bytes, but now stated in the URL definition. * Change URI registration to be under urn:ietf:params

Draft -00 to -01: * Fix cert trust text. * change supportInformation to meta-info * Add an informational element in. * add urn registry and create first entry * add default elements

Appendix B. Default MUD nodes

What follows is the portion of a MUD file that permits DNS traffic to a controller that is registered with the URN "urn:ietf:params:mud:dns" and traffic NTP to a controller that is registered "urn:ietf:params:mud:ntp". This is considered the default behavior and the ACEs are in effect appended to whatever other "ace" entries that a MUD file contains. To block DNS or NTP one repeats the matching statement but replaces the "forwarding" action "accept" with "drop". Because ACEs are processed in the order they are received, the defaults would not be reached. A MUD controller might further decide to optimize to simply not include the defaults when they are overridden.

Four of "acl" list entries that implement default MUD nodes is listed below. Two are for IPv4 and two are for IPv6 (one in each direction for both versions of IP).

```
"ietf-access-control-list:access-lists": {  
  "acl": [  
    {  
      "acl-name": "mud-v4-default-to-device",  
      "acl-type": "ipv4-acl",  
      "aces": {
```

```

    "ace": [
      {
        "rule-name": "ent0-todev",
        "matches": {
          "ietf-mud:mud-acl": {
            "controller": "urn:ietf:params:mud:dns"
          },
          "ipv4-acl": {
            "protocol": 17,
            "source-port-range": {
              "lower-port": 53,
              "upper-port": 53
            }
          }
        },
        "actions": {
          "forwarding": "accept"
        }
      },
      {
        "rule-name": "ent1-todev",
        "matches": {
          "ietf-mud:mud-acl": {
            "controller": "urn:ietf:params:mud:ntp"
          },
          "ipv4-acl": {
            "protocol": 17,
            "source-port-range": {
              "lower-port": 123,
              "upper-port": 123
            }
          }
        },
        "actions": {
          "forwarding": "accept"
        }
      }
    ]
  },
  {
    "acl-name": "mud-v4-default-from-device",
    "acl-type": "ipv4-acl",
    "aces": {
      "ace": [
        {
          "rule-name": "ent0-frdev",
          "matches": {

```

```

        "ietf-mud:mud-acl": {
            "controller": "urn:ietf:params:mud:dns"
        },
        "ipv4-acl": {
            "protocol": 17,
            "destination-port-range": {
                "lower-port": 53,
                "upper-port": 53
            }
        }
    },
    "actions": {
        "forwarding": "accept"
    }
},
{
    "rule-name": "ent1-frdev",
    "matches": {
        "ietf-mud:mud-acl": {
            "controller": "urn:ietf:params:mud:ntp"
        },
        "ipv4-acl": {
            "protocol": 17,
            "destination-port-range": {
                "lower-port": 123,
                "upper-port": 123
            }
        }
    },
    "actions": {
        "forwarding": "accept"
    }
}
]
}
},
{
    "acl-name": "mud-v6-default-to-device",
    "acl-type": "ipv6-acl",
    "access-list-entries": {
        "ace": [
            {
                "rule-name": "ent0-todev",
                "matches": {
                    "ietf-mud:mud-acl": {
                        "controller": "urn:ietf:params:mud:dns"
                    },
                    "ipv6-acl": {

```



```

        "protocol": 17,
        "source-port-range": {
            "lower-port": 53,
            "upper-port": 53
        }
    },
    "actions": {
        "forwarding": "accept"
    }
},
{
    "rule-name": "ent1-todev",
    "matches": {
        "ietf-mud:mud-acl": {
            "controller": "urn:ietf:params:mud:ntp"
        },
        "ipv6-acl": {
            "protocol": 17,
            "source-port-range": {
                "lower-port": 123,
                "upper-port": 123
            }
        }
    },
    "actions": {
        "forwarding": "accept"
    }
}
]
}
},
{
    "acl-name": "mud-v6-default-from-device",
    "acl-type": "ipv6-acl",
    "access-list-entries": {
        "ace": [
            {
                "rule-name": "ent0-frdev",
                "matches": {
                    "ietf-mud:mud-acl": {
                        "controller": "urn:ietf:params:mud:dns"
                    },
                    "ipv6-acl": {
                        "protocol": 17,
                        "destination-port-range": {
                            "lower-port": 53,
                            "upper-port": 53
                        }
                    }
                }
            }
        ]
    }
}

```

```

    }
  },
  "actions": {
    "forwarding": "accept"
  }
},
{
  "rule-name": "ent1-frdev",
  "matches": {
    "ietf-mud:mud-acl": {
      "controller": "urn:ietf:params:mud:ntp"
    },
    "ipv6-acl": {
      "protocol": 17,
      "destination-port-range": {
        "lower-port": 123,
        "upper-port": 123
      }
    }
  },
  "actions": {
    "forwarding": "accept"
  }
}
]
}
]
}

```

Appendix C. A Sample Extension: DETNET-indicator

In this sample extension we augment the core MUD model to indicate whether the device implements DETNET. If a device later attempts to make use of DETNET, an notification or exception might be generated. Note that this example is intended only for illustrative purposes.

Extension Name: "Example-Extension" (to be used in the extensions list)
 Standard: this document (but do not register the example)

This extension augments the MUD model to include a single node, using the following sample module that has the following tree structure:

```
module: ietf-mud-detext-example
  augment /ietf-mud:mud:
    +--rw is-detnet-required?    boolean
```

The model is defined as follows:

```
<CODE BEGINS>file "ietf-mud-detext-example@2016-09-07.yang"
module ietf-mud-detext-example {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-mud-detext-example";
  prefix ietf-mud-detext-example;

  import ietf-mud {
    prefix ietf-mud;
  }

  organization
    "IETF OPSAWG (Ops Area) Working Group";
  contact
    "WG Web: http://tools.ietf.org/wg/opsawg/
    WG List: opsawg@ietf.org
    Author: Eliot Lear
    lear@cisco.com
    Author: Ralph Droms
    rdroms@gmail.com
    Author: Dan Romascanu
    dromasca@gmail.com

    ";
  description
    "Sample extension to a MUD module to indicate a need
    for DETNET support.";

  revision 2017-09-05 {
    description
      "Initial revision.";
    reference
      "RFC XXXX: Manufacturer Usage Description
      Specification";
  }

  augment "/ietf-mud:mud" {
    description
      "This adds a simple extension for a manufacturer
      to indicate whether DETNET is required by a
      device.";
    leaf is-detnet-required {
```

```
        type boolean;
        description
            "This value will equal true if a device requires
             detnet to properly function";
    }
}
}
<CODE ENDS>
```

Using the previous example, we now show how the extension would be expressed:

```
{
  "ietf-mud:mud": {
    "mud-url": "https://bms.example.com/.well-known/mud/v1/lightbulb",
    "last-update": "2017-09-20T15:49:18+02:00",
    "cache-validity": 48,
    "is-supported": true,
    "systeminfo": "https://bms.example.com/descriptions/lightbulb",
    "extensions": [
      "ietf-mud-detext-example"
    ],
    "ietf-mud-detext-example:is-detnet-required": "false",
    "from-device-policy": {
      "access-lists": {
        "access-list": [
          {
            "acl-name": "mud-54684-v6fr",
            "acl-type": "ietf-access-control-list:ipv6-acl"
          }
        ]
      }
    },
    "to-device-policy": {
      "access-lists": {
        "access-list": [
          {
            "acl-name": "mud-54684-v6to",
            "acl-type": "ietf-access-control-list:ipv6-acl"
          }
        ]
      }
    }
  },
  "ietf-access-control-list:access-lists": {
    "acl": [
      {
        "acl-name": "mud-54684-v6to",
```

```
"acl-type": "ipv6-acl",
"access-list-entries": {
  "ace": [
    {
      "rule-name": "cl0-todev",
      "matches": {
        "ipv6-acl": {
          "ietf-acldns:src-dnsname": "service.bms.example.com",
          "protocol": 6,
          "source-port-range": {
            "lower-port": 443,
            "upper-port": 443
          }
        },
        "tcp-acl": {
          "ietf-mud:direction-initiated": "from-device"
        }
      },
      "actions": {
        "forwarding": "accept"
      }
    }
  ]
},
{
  "acl-name": "mud-54684-v6fr",
  "acl-type": "ipv6-acl",
  "access-list-entries": {
    "ace": [
      {
        "rule-name": "cl0-frdev",
        "matches": {
          "ipv6-acl": {
            "ietf-acldns:dst-dnsname": "service.bms.example.com",
            "protocol": 6,
            "destination-port-range": {
              "lower-port": 443,
              "upper-port": 443
            }
          },
          "tcp-acl": {
            "ietf-mud:direction-initiated": "from-device"
          }
        },
        "actions": {
          "forwarding": "accept"
        }
      }
    ]
  }
}
```

```
}  
]  
}  
}  
]  
}  
}
```

Authors' Addresses

Eliot Lear
Cisco Systems
Richtistrasse 7
Wallisellen CH-8304
Switzerland

Phone: +41 44 878 9200
Email: lear@cisco.com

Ralph Droms

Phone: +1 978 376 3731
Email: rdroms@gmail.com

Dan Romascanu

Phone: +972 54 5555347
Email: dromasca@gmail.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: December 17, 2018

E. Lear
Cisco Systems
R. Droms
Google
D. Romascanu
June 15, 2018

Manufacturer Usage Description Specification
draft-ietf-opsawg-mud-25

Abstract

This memo specifies a component-based architecture for manufacturer usage descriptions (MUD). The goal of MUD is to provide a means for end devices to signal to the network what sort of access and network functionality they require to properly function. The initial focus is on access control. Later work can delve into other aspects.

This memo specifies two YANG modules, IPv4 and IPv6 DHCP options, an LLDP TLV, a URL, an X.509 certificate extension and a means to sign and verify the descriptions.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on December 17, 2018.

Copyright Notice

Copyright (c) 2018 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of

publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	3
1.1. What MUD Doesn't Do	5
1.2. A Simple Example	5
1.3. Terminology	5
1.4. Determining Intended Use	6
1.5. Finding A Policy: The MUD URL	6
1.6. Processing of the MUD URL	7
1.7. Types of Policies	8
1.8. The Manufacturer Usage Description Architecture	10
1.9. Order of operations	11
2. The MUD Model and Semantic Meaning	12
2.1. The IETF-MUD YANG Module	13
3. MUD model definitions for the root mud container	14
3.1. mud-version	14
3.2. mud-url	15
3.3. to-device-policy and from-device-policy containers	15
3.4. last-update	15
3.5. cache-validity	15
3.6. is-supported	15
3.7. systeminfo	15
3.8. mfg-name, software-rev, model-name firmware-rev	16
3.9. extensions	16
4. Augmentation to the ACL Model	16
4.1. manufacturer	16
4.2. same-manufacturer	16
4.3. documentation	17
4.4. model	17
4.5. local-networks	17
4.6. controller	17
4.7. my-controller	18
4.8. direction-initiated	18
5. Processing of the MUD file	18
6. What does a MUD URL look like?	18
7. The MUD YANG Model	19
8. The Domain Name Extension to the ACL Model	25
8.1. src-dnsname	26
8.2. dst-dnsname	26
8.3. The ietf-acldns Model	26
9. MUD File Example	28

10. The MUD URL DHCP Option	30
10.1. Client Behavior	31
10.2. Server Behavior	31
10.3. Relay Requirements	32
11. The Manufacturer Usage Description (MUD) URL X.509 Extension	32
12. The Manufacturer Usage Description LLDP extension	34
13. Creating and Processing of Signed MUD Files	35
13.1. Creating a MUD file signature	36
13.2. Verifying a MUD file signature	36
14. Extensibility	37
15. Deployment Considerations	37
16. Security Considerations	38
17. IANA Considerations	40
17.1. YANG Module Registrations	41
17.2. DHCPv4 and DHCPv6 Options	41
17.3. PKIX Extensions	41
17.4. MIME Media-type Registration for MUD files	42
17.5. LLDP IANA TLV Subtype Registry	42
17.6. The MUD Well Known Universal Resource Name (URNs)	43
17.7. Extensions Registry	43
18. Acknowledgments	43
19. References	44
19.1. Normative References	44
19.2. Informative References	47
Appendix A. Changes from Earlier Versions	49
Appendix B. Default MUD nodes	52
Appendix C. A Sample Extension: DETNET-indicator	57
Authors' Addresses	60

1. Introduction

The Internet has largely been constructed for general purpose computers, those devices that may be used for a purpose that is specified by those who own the device. [RFC1984] presumed that an end device would be most capable of protecting itself. This made sense when the typical device was a workstation or a mainframe, and it continues to make sense for general purpose computing devices today, including laptops, smart phones, and tablets.

[RFC7452] discusses design patterns for, and poses questions about, smart objects. Let us then posit a group of objects that are specifically not intended to be used for general purpose computing tasks. These devices, which this memo refers to as Things, have a specific purpose. By definition, therefore, all other uses are not intended. If a small number of communication patterns follows from those small number of uses, the combination of these two statements can be restated as a manufacturer usage description (MUD) that can be applied at various points within a network. MUD primarily addresses

threats to the device rather than the device as a threat. In some circumstances, however, MUD may offer some protection in the latter case, depending on the MUD-URL is communicated, and how devices and their communications are authenticated.

We use the notion of "manufacturer" loosely in this context to refer to the entity or organization that will state how a device is intended to be used. For example, in the context of a lightbulb, this might indeed be the lightbulb manufacturer. In the context of a smarter device that has a built in Linux stack, it might be an integrator of that device. The key points are that the device itself is assumed to serve a limited purpose, and that there exists an organization in the supply chain of that device that will take responsibility for informing the network about that purpose.

The intent of MUD is to provide the following:

- o Substantially reduce the threat surface on a device to those communications intended by the manufacturer.
- o Provide a means to scale network policies to the ever-increasing number of types of devices in the network.
- o Provide a means to address at least some vulnerabilities in a way that is faster than the time it might take to update systems. This will be particularly true for systems that are no longer supported.
- o Keep the cost of implementation of such a system to the bare minimum.
- o Provide a means of extensibility for manufacturers to express other device capabilities or requirements.

MUD consists of three architectural building blocks:

- o A URL that can be used to locate a description;
- o The description itself, including how it is interpreted, and;
- o A means for local network management systems to retrieve the description.

MUD is most effective when the network is able to identify in some way the remote endpoints that Things will talk to.

In this specification we describe each of these building blocks and how they are intended to be used together. However, they may also be

used separately, independent of this specification, by local deployments for their own purposes.

1.1. What MUD Doesn't Do

MUD is not intended to address network authorization of general purpose computers, as their manufacturers cannot envision a specific communication pattern to describe. In addition, even those devices that have a single or small number of uses might have very broad communication patterns. MUD on its own is not for them either.

Although MUD can provide network administrators with some additional protection when device vulnerabilities exist, it will never replace the need for manufacturers to patch vulnerabilities.

Finally, no matter what the manufacturer specifies in a MUD file, these are not directives, but suggestions. How they are instantiated locally will depend on many factors and will be ultimately up to the local network administrator, who must decide what is appropriate in a given circumstances.

1.2. A Simple Example

A light bulb is intended to light a room. It may be remotely controlled through the network, and it may make use of a rendezvous service of some form that an application on a smart phone. What we can say about that light bulb, then, is that all other network access is unwanted. It will not contact a news service, nor speak to the refrigerator, and it has no need of a printer or other devices. It has no social networking friends. Therefore, an access list applied to it that states that it will only connect to the single rendezvous service will not impede the light bulb in performing its function, while at the same time allowing the network to provide both it and other devices an additional layer of protection.

1.3. Terminology

MUD: manufacturer usage description.

MUD file: a file containing YANG-based JSON that describes a Thing and associated suggested specific network behavior.

MUD file server: a web server that hosts a MUD file.

MUD manager: the system that requests and receives the MUD file from the MUD server. After it has processed a MUD file, it may direct changes to relevant network elements.

MUD controller: a synonym that has been used in the past for MUD manager.

MUD URL: a URL that can be used by the MUD manager to receive the MUD file.

Thing: the device emitting a MUD URL.

Manufacturer: the entity that configures the Thing to emit the MUD URL and the one who asserts a recommendation in a MUD file. The manufacturer might not always be the entity that constructs a Thing. It could, for instance, be a systems integrator, or even a component provider.

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

1.4. Determining Intended Use

The notion of intended use is in itself not new. Network administrators apply access lists every day to allow for only such use. This notion of white listing was well described by Chapman and Zwicky in [FW95]. Profiling systems that make use of heuristics to identify types of systems have existed for years as well.

A Thing could just as easily tell the network what sort of access it requires without going into what sort of system it is. This would, in effect, be the converse of [RFC7488]. In seeking a general solution, however, we assume that a device will implement functionality necessary to fulfill its limited purpose. This is basic economic constraint. Unless the network would refuse access to such a device, its developers would have no reason to provide the network any information. To date, such an assertion has held true.

1.5. Finding A Policy: The MUD URL

Our work begins with the device emitting a Universal Resource Locator (URL) [RFC3986]. This URL serves both to classify the device type and to provide a means to locate a policy file.

MUD URLs MUST use the HTTPS scheme [RFC7230].

In this memo three means are defined to emit the MUD URL, as follows:

- o A DHCP option[RFC2131],[RFC3315] that the DHCP client uses to inform the DHCP server. The DHCP server may take further actions, such as act as the MUD manager or otherwise pass the MUD URL along to the MUD manager.
- o An X.509 constraint. The IEEE has developed [IEEE8021AR] that provides a certificate-based approach to communicate device characteristics, which itself relies on [RFC5280]. The MUD URL extension is non-critical, as required by IEEE 802.1AR. Various means may be used to communicate that certificate, including Tunnel Extensible Authentication Protocol (TEAP) [RFC7170].
- o Finally, a Link Layer Discovery Protocol (LLDP) frame is defined [IEEE8021AB].

It is possible that there may be other means for a MUD URL to be learned by a network. For instance, some devices may already be fielded or have very limited ability to communicate a MUD URL, and yet can be identified through some means, such as a serial number or a public key. In these cases, manufacturers may be able to map those identifiers to particular MUD URLs (or even the files themselves). Similarly, there may be alternative resolution mechanisms available for situations where Internet connectivity is limited or does not exist. Such mechanisms are not described in this memo, but are possible. Implementors are encouraged to allow for this sort of flexibility of how MUD URLs may be learned.

1.6. Processing of the MUD URL

MUD managers that are able to do so SHOULD retrieve MUD URLs and signature files as per [RFC7230], using the GET method [RFC7231]. They MUST validate the certificate using the rules in [RFC2818], Section 3.1.

Requests for MUD URLs SHOULD include an "Accept" header ([RFC7231], Section 5.3.2) containing "application/mud+json", an "Accept-Language" header field ([RFC7231], Section 5.3.5), and a "User-Agent" header ([RFC7231], Section 5.5.3).

MUD managers SHOULD automatically process 3xx response status codes.

If a MUD manager is not able to fetch a MUD URL, other means MAY be used to import MUD files and associated signature files. So long as the signature of the file can be validated, the file can be used. In such environments, controllers SHOULD warn administrators when cache-validity expiry is approaching so that they may check for new files.

It may not be possible for a MUD manager to retrieve a MUD file at any given time. Should a MUD manager fail to retrieve a MUD file, it SHOULD consider the existing one safe to use, at least for a time. After some period, it SHOULD log that it has been unable to retrieve the file. There may be very good reasons for such failures, including the possibility that the MUD manager is in an off-line environment, the local Internet connection has failed, or the remote Internet connection has failed. It is also possible that an attacker is attempting to interfere with the deployment of a device. It is a local decision as to how to handle such circumstances.

1.7. Types of Policies

When the MUD URL is resolved, the MUD manager retrieves a file that describes what sort of communications a device is designed to have. The manufacturer may specify either specific hosts for cloud based services or certain classes for access within an operational network. An example of a class might be "devices of a specified manufacturer type", where the manufacturer type itself is indicated simply by the authority component (e.g, the domain name) of the MUD URL. Another example might be to allow or disallow local access. Just like other policies, these may be combined. For example:

- o Allow access to devices of the same manufacturer
- o Allow access to and from controllers via Constrained Application Protocol (COAP)[RFC7252]
- o Allow access to local DNS/NTP
- o Deny all other access

A printer might have a description that states:

- o Allow access for port IPP or port LPD
- o Allow local access for port HTTP
- o Deny all other access

In this way anyone can print to the printer, but local access would be required for the management interface.

The files that are retrieved are intended to be closely aligned to existing network architectures so that they are easy to deploy. We make use of YANG [RFC7950] because it provides accurate and adequate models for use by network devices. JSON[RFC8259] is used as a

serialization format for compactness and readability, relative to XML. Other formats may be chosen with later versions of MUD.

While the policy examples given here focus on access control, this is not intended to be the sole focus. By structuring the model described in this document with clear extension points, other descriptions could be included. One that often comes to mind is quality of service.

The YANG modules specified here are extensions of [I-D.ietf-netmod-acl-model]. The extensions to this model allow for a manufacturer to express classes of systems that a manufacturer would find necessary for the proper function of the device. Two modules are specified. The first module specifies a means for domain names to be used in ACLs so that devices that have their controllers in the cloud may be appropriately authorized with domain names, where the mapping of those names to addresses may rapidly change.

The other module abstracts away IP addresses into certain classes that are instantiated into actual IP addresses through local processing. Through these classes, manufacturers can specify how the device is designed to communicate, so that network elements can be configured by local systems that have local topological knowledge. That is, the deployment populates the classes that the manufacturer specifies. The abstractions below map to zero or more hosts, as follows:

Manufacturer: A device made by a particular manufacturer, as identified by the authority component of its MUD URL

same-manufacturer: Devices that have the same authority component of their MUD URL.

controller: Devices that the local network administrator admits to the particular class.

my-controller: Devices intended to serve as controllers for the MUD-URL that the Thing emitted.

local: The class of IP addresses that are scoped within some administrative boundary. By default it is suggested that this be the local subnet.

The "manufacturer" classes can be easily specified by the manufacturer, whereas controller classes are initially envisioned to be specified by the administrator.

Because manufacturers do not know who will be using their devices, it is important for functionality referenced in usage descriptions to be relatively ubiquitous and mature. For these reasons the YANG-based configuration in a MUD file is limited to either the modules specified or referenced in this document, or those specified in documented extensions.

1.8. The Manufacturer Usage Description Architecture

With these components laid out we now have the basis for an architecture. This leads us to ASCII art.

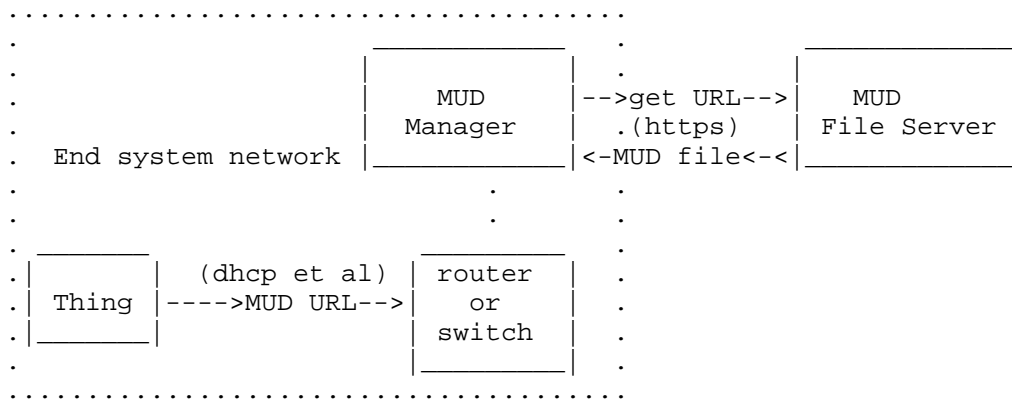


Figure 1: MUD Architecture

In the above diagram, the switch or router collects MUD URLs and forwards them to the MUD manager (a network management system) for processing. This happens in different ways, depending on how the URL is communicated. For instance, in the case of DHCP, the DHCP server might receive the URL and then process it. In the case of IEEE 802.1X [IEEE8021X], the switch would carry the URL via a certificate to the authentication server via EAP over Radius[RFC3748], which would then process it. One method to do this is TEAP, described in [RFC7170]. The certificate extension is described below.

The information returned by the MUD file server is valid for as long as the Thing is connected. There is no expiry. However, if the MUD manager has detected that the MUD file for a Thing has changed, it SHOULD update the policy expeditiously, taking into account whatever approval flow is required in a deployment. In this way, new recommendations from the manufacturer can be processed in a timely fashion.

The information returned by the MUD file server (a web server) is valid for the duration of the Thing's connection, or as specified in the description. Thus if the Thing is disconnected, any associated configuration in the switch can be removed. Similarly, from time to time the description may be refreshed, based on new capabilities or communication patterns or vulnerabilities.

The web server is typically run by or on behalf of the manufacturer. Its domain name is that of the authority found in the MUD URL. For legacy cases where Things cannot emit a URL, if the switch is able to determine the appropriate URL, it may proxy it. In the trivial case it may hardcode MUD-URL on a switch port or a map from some available identifier such as an L2 address or certificate hash to a MUD-URL.

The role of the MUD manager in this environment is to do the following:

- o receive MUD URLs,
- o fetch MUD files,
- o translate abstractions in the MUD files to specific network element configuration,
- o maintain and update any required mappings of the abstractions, and
- o update network elements with appropriate configuration.

A MUD manager may be a component of a AAA or network management system. Communication within those systems and from those systems to network elements is beyond the scope of this memo.

1.9. Order of operations

As mentioned above, MUD contains architectural building blocks, and so order of operation may vary. However, here is one clear intended example:

1. Thing emits URL.
2. That URL is forwarded to a MUD manager by the nearest switch (how this happens depends on the way in which the MUD URL is emitted).
3. The MUD manager retrieves the MUD file and signature from the MUD file server, assuming it doesn't already have copies. After validating the signature, it may test the URL against a web or domain reputation service, and it may test any hosts within the file against those reputation services, as it deems fit.

4. The MUD manager may query the administrator for permission to add the Thing and associated policy. If the Thing is known or the Thing type is known, it may skip this step.
5. The MUD manager instantiates local configuration based on the abstractions defined in this document.
6. The MUD manager configures the switch nearest the Thing. Other systems may be configured as well.
7. When the Thing disconnects, policy is removed.

2. The MUD Model and Semantic Meaning

A MUD file consists of a YANG model instance that has been serialized in JSON [RFC7951]. For purposes of MUD, the nodes that can be modified are access lists as augmented by this model. The MUD file is limited to the serialization of only the following YANG schema:

- o ietf-access-control-list [I-D.ietf-netmod-acl-model]
- o ietf-mud (this document)
- o ietf-acldns (this document)

Extensions may be used to add additional schema. This is described further on.

To provide the widest possible deployment, publishers of MUD files SHOULD make use of the abstractions in this memo and avoid the use of IP addresses. A MUD manager SHOULD NOT automatically implement any MUD file that contains IP addresses, especially those that might have local significance. The addressing of one side of an access list is implicit, based on whether it is applied as to-device-policy or from-device-policy.

With the exceptions of "name" of the ACL, "type", "name" of the ACE, and TCP and UDP source and destination port information, publishers of MUD files SHOULD limit the use of ACL model leaf nodes expressed to those found in this specification. Absent any extensions, MUD files are assumed to implement only the following ACL model features:

- o match-on-ipv4, match-on-ipv6, match-on-tcp, match-on-udp, match-on-icmp

Furthermore, only "accept" or "drop" actions SHOULD be included. A MUD manager MAY choose to interpret "reject" as "drop". A MUD manager SHOULD ignore all other actions. This is because

manufacturers do not have sufficient context within a local deployment to know whether reject is appropriate. That is a decision that should be left to a network administrator.

Given that MUD does not deal with interfaces, the support of the "ietf-interfaces" module [RFC8343] is not required. Specifically, the support of interface-related features and branches (e.g., interface-attachment and interface-stats) of the ACL YANG module is not required.

In fact, MUD managers MAY ignore any particular component of a description or MAY ignore the description in its entirety, and SHOULD carefully inspect all MUD descriptions. Publishers of MUD files MUST NOT include other nodes except as described in Section 3.9. See that section for more information.

2.1. The IETF-MUD YANG Module

This module is structured into three parts:

- o The first component, the "mud" container, holds information that is relevant to retrieval and validity of the MUD file itself, as well as policy intended to and from the Thing.
- o The second component augments the matching container of the ACL model to add several nodes that are relevant to the MUD URL, or otherwise abstracted for use within a local environment.
- o The third component augments the tcp-acl container of the ACL model to add the ability to match on the direction of initiation of a TCP connection.

A valid MUD file will contain two root objects, a "mud" container and an "acls" container. Extensions may add additional root objects as required. As a reminder, when parsing acls, elements within a "match" block are logically ANDed. In general, a single abstraction in a match statement should be used. For instance, it makes little sense to match both "my-controller" and "controller" with an argument, since they are highly unlikely to be the same value.

A simplified graphical representation of the data models is used in this document. The meaning of the symbols in these diagrams is explained in [RFC8340].

```

module: ietf-mud
  +--rw mud!
    +--rw mud-version          uint8
    +--rw mud-url              inet:uri
    +--rw last-update          yang:date-and-time
    +--rw mud-signature?       inet:uri
    +--rw cache-validity?      uint8
    +--rw is-supported         boolean
    +--rw systeminfo?          string
    +--rw mfg-name?            string
    +--rw model-name?          string
    +--rw firmware-rev?        string
    +--rw software-rev?        string
    +--rw documentation?       inet:uri
    +--rw extensions*          string
    +--rw from-device-policy
      | +--rw acls
      | | +--rw access-list* [name]
      | | | +--rw name      -> /acl:acls/acl/name
      | +--rw to-device-policy
      | | +--rw acls
      | | | +--rw access-list* [name]
      | | | | +--rw name      -> /acl:acls/acl/name
    augment /acl:acls/acl:acl/acl:aces/acl:ace/acl:matches:
      +--rw mud
        +--rw manufacturer?    inet:host
        +--rw same-manufacturer? empty
        +--rw model?           inet:uri
        +--rw local-networks?  empty
        +--rw controller?      inet:uri
        +--rw my-controller?    empty
      augment
        /acl:acls/acl:acl/acl:aces/acl:ace/acl:matches
        /acl:l4/acl:tcp/acl:tcp:
          +--rw direction-initiated?  direction

```

3. MUD model definitions for the root mud container

3.1. mud-version

This node specifies the integer version of the MUD specification.
 This memo specifies version 1.

3.2. mud-url

This URL identifies the MUD file. This is useful when the file and associated signature are manually uploaded, say, in an offline mode.

3.3. to-device-policy and from-device-policy containers

[I-D.ietf-netmod-acl-model] describes access-lists. In the case of MUD, a MUD file must be explicit in describing the communication pattern of a Thing, and that includes indicating what is to be permitted or denied in either direction of communication. Hence each of these containers indicates the appropriate direction of a flow in association with a particular Thing. They contain references to specific access-lists.

3.4. last-update

This is a date-and-time value of when the MUD file was generated. This is akin to a version number. Its form is taken from [RFC6991] which, for those keeping score, in turn was taken from Section 5.6 of [RFC3339], which was taken from [ISO.8601.1988].

3.5. cache-validity

This uint8 is the period of time in hours that a network management station MUST wait since its last retrieval before checking for an update. It is RECOMMENDED that this value be no less than 24 and MUST NOT be more than 168 for any Thing that is supported. This period SHOULD be no shorter than any period determined through HTTP caching directives (e.g., "cache-control" or "Expires"). N.B., expiring of this timer does not require the MUD manager to discard the MUD file, nor terminate access to a Thing. See Section 16 for more information.

3.6. is-supported

This boolean is an indication from the manufacturer to the network administrator as to whether or not the Thing is supported. In this context a Thing is said to not be supported if the manufacturer intends never to issue a firmware or software update to the Thing or never update the MUD file. A MUD manager MAY still periodically check for updates.

3.7. systeminfo

This is a textual UTF-8 description of the Thing to be connected. The intent is for administrators to be able to see a brief

displayable description of the Thing. It SHOULD NOT exceed 60 characters worth of display space.

3.8. mfg-name, software-rev, model-name firmware-rev

These optional fields are filled in as specified by [RFC8348]. Note that firmware-rev and software-rev MUST NOT be populated in a MUD file if the device can be upgraded but the MUD-URL cannot be. This would be the case, for instance, with MUD-URLs that are contained in 802.1AR certificates.

3.9. extensions

This optional leaf-list names MUD extensions that are used in the MUD file. Note that MUD extensions MUST NOT be used in a MUD file without the extensions being declared. Implementations MUST ignore any node in this file that they do not understand.

Note that extensions can either extend the MUD file as described in the previous paragraph, or they might reference other work. An extension example can be found in Appendix C.

4. Augmentation to the ACL Model

Note that in this section, when we use the term "match" we are referring to the ACL model "matches" node.

4.1. manufacturer

This node consists of a hostname that would be matched against the authority component of another Thing's MUD URL. In its simplest form "manufacturer" and "same-manufacturer" may be implemented as access-lists. In more complex forms, additional network capabilities may be used. For example, if one saw the line "manufacturer" : "flobbidy.example.com", then all Things that registered with a MUD URL that contained flobbity.example.com in its authority section would match.

4.2. same-manufacturer

This null-valued node is an equivalent for when the manufacturer element is used to indicate the authority that is found in another Thing's MUD URL matches that of the authority found in this Thing's MUD URL. For example, if the Thing's MUD URL were <https://bl.example.com/ThingV1>, then all devices that had MUD URL with an authority section of bl.example.com would match.

4.3. documentation

This URI consists of a URL that points to documentation relating to the device and the MUD file. This can prove particularly useful when the "controller" class is used, so that its use can be explained.

4.4. model

This string matches the entire MUD URL, thus covering the model that is unique within the context of the authority. It may contain not only model information, but versioning information as well, and any other information that the manufacturer wishes to add. The intended use is for devices of this precise class to match, to permit or deny communication between one another.

4.5. local-networks

This null-valued node expands to include local networks. Its default expansion is that packets must not traverse toward a default route that is received from the router. However, administrators may expand the expression as is appropriate in their deployments.

4.6. controller

This URI specifies a value that a controller will register with the MUD manager. The node then is expanded to the set of hosts that are so registered. This node may also be a URN. In this case, the URN describes a well known service, such as DNS or NTP, that has been standardized. Both of those URNs may be found in Section 17.6.

When "my-controller" is used, it is possible that the administrator will be prompted to populate that class for each and every model. Use of "controller" with a named class allows the user to populate that class only once for many different models that a manufacturer may produce.

Controller URIs MAY take the form of a URL (e.g. "http[s]://"). However, MUD managers MUST NOT resolve and retrieve such files, and it is RECOMMENDED that there be no such file at this time, as their form and function may be defined at a point in the future. For now, URLs should serve simply as class names and may be populated by the local deployment administrator.

Great care should be taken by MUD managers when invoking the controller class in the form of URLs. For one thing, it requires some understanding by the administrator as to when it is appropriate. Pre-registration in such classes by controllers with the MUD server

is encouraged. The mechanism to do that is beyond the scope of this work.

4.7. my-controller

This null-valued node signals to the MUD manager to use whatever mapping it has for this MUD URL to a particular group of hosts. This may require prompting the administrator for class members. Future work should seek to automate membership management.

4.8. direction-initiated

This MUST only be applied to TCP. This matches the direction in which a TCP connection is initiated. When direction initiated is "from-device", packets that are transmitted in the direction of a thing MUST be dropped unless the thing has first initiated a TCP connection. By way of example, this node may be implemented in its simplest form by looking at naked SYN bits, but may also be implemented through more stateful mechanisms.

When applied this matches packets when the flow was initiated in the corresponding direction. [RFC6092] specifies IPv6 guidance best practices. While that document is scoped specifically to IPv6, its contents are applicable for IPv4 as well.

5. Processing of the MUD file

To keep things relatively simple in addition to whatever definitions exist, we also apply two additional default behaviors:

- o Anything not explicitly permitted is denied.
- o Local DNS and NTP are, by default, permitted to and from the Thing.

An explicit description of the defaults can be found in Appendix B. These are applied AFTER all other explicit rules. Thus, a default behavior can be changed with a "drop" action.

6. What does a MUD URL look like?

MUD URLs are required to use the HTTPS scheme, in order to establish the MUD file server's identity and assure integrity of the MUD file.

Any "https://" URL can be a MUD URL. For example:


```
https://things.example.org/product_abc123/v5
https://www.example.net/mudfiles/temperature_sensor/
https://example.com/lightbulbs/colour/v1
```

A manufacturer may construct a MUD URL in any way, so long as it makes use of the "https" schema.

7. The MUD YANG Model

```
<CODE BEGINS>file "ietf-mud@2018-06-15.yang"
module ietf-mud {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-mud";
  prefix ietf-mud;

  import ietf-access-control-list {
    prefix acl;
  }
  import ietf-yang-types {
    prefix yang;
  }
  import ietf-inet-types {
    prefix inet;
  }

  organization
    "IETF OPSAWG (Ops Area) Working Group";
  contact
    "WG Web: http://tools.ietf.org/wg/opsawg/
    WG List: opsawg@ietf.org
    Author: Eliot Lear
    lear@cisco.com
    Author: Ralph Droms
    rdroms@gmail.com
    Author: Dan Romascanu
    dromasca@gmail.com

    ";
  description
    "This YANG module defines a component that augments the
    IETF description of an access list. This specific module
    focuses on additional filters that include local, model,
    and same-manufacturer.

    This module is intended to be serialized via JSON and stored
    as a file, as described in RFC XXXX [RFC Editor to fill in with
    this document #]."
```

Copyright (c) 2016,2017 IETF Trust and the persons identified as the document authors. All rights reserved. Redistribution and use in source and binary forms, with or without modification, is permitted pursuant to, and subject to the license terms contained in, the Simplified BSD License set forth in Section 4.c of the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>). This version of this YANG module is part of RFC XXXX; see the RFC itself for full legal notices.";

```
revision 2018-06-15 {
  description
    "Initial proposed standard.";
  reference
    "RFC XXXX: Manufacturer Usage Description
    Specification";
}

typedef direction {
  type enumeration {
    enum to-device {
      description
        "packets or flows destined to the target
        Thing";
    }
    enum from-device {
      description
        "packets or flows destined from
        the target Thing";
    }
  }
  description
    "Which way are we talking about?";
}

container mud {
  presence "Enabled for this particular MUD URL";
  description
    "MUD related information, as specified
    by RFC-XXXX [RFC Editor to fill in].";
  uses mud-grouping;
}

grouping mud-grouping {
  description
    "Information about when support end(ed), and
    when to refresh";
```

```
leaf mud-version {
  type uint8;
  mandatory true;
  description
    "This is the version of the MUD
    specification.  This memo specifies version 1.";
}
leaf mud-url {
  type inet:uri;
  mandatory true;
  description
    "This is the MUD URL associated with the entry found
    in a MUD file.";
}
leaf last-update {
  type yang:date-and-time;
  mandatory true;
  description
    "This is intended to be when the current MUD file
    was generated.  MUD Managers SHOULD NOT check
    for updates between this time plus cache validity";
}
leaf mud-signature {
  type inet:uri;
  description
    "A URI that resolves to a signature as
    described in this specification.";
}
leaf cache-validity {
  type uint8 {
    range "1..168";
  }
  units "hours";
  default "48";
  description
    "The information retrieved from the MUD server is
    valid for these many hours, after which it should
    be refreshed.  N.B. MUD manager implementations
    need not discard MUD files beyond this period.";
}
leaf is-supported {
  type boolean;
  mandatory true;
  description
    "This boolean indicates whether or not the Thing is
    currently supported by the manufacturer.";
}
leaf systeminfo {
```

```
    type string;
    description
      "A UTF-8 description of this Thing.  This
       should be a brief description that may be
       displayed to the user to determine whether
       to allow the Thing on the
       network.";
  }
  leaf mfg-name {
    type string;
    description
      "Manufacturer name, as described in
       the ietf-hardware YANG module.";
  }
  leaf model-name {
    type string;
    description
      "Model name, as described in the
       ietf-hardware YANG module.";
  }
  leaf firmware-rev {
    type string;
    description
      "firmware-rev, as described in the
       ietf-hardware YANG module.  Note this field MUST
       NOT be included when the device can be updated
       but the MUD-URL cannot.";
  }
  leaf software-rev {
    type string;
    description
      "software-rev, as described in the
       ietf-hardware YANG module.  Note this field MUST
       NOT be included when the device can be updated
       but the MUD-URL cannot.";
  }
  leaf documentation {
    type inet:uri;
    description
      "This URL points to documentation that
       relates to this device and any classes that it uses
       in its MUD file.  A caution: MUD managers need
       not resolve this URL on their own, but rather simply
       provide it to the administrator.  Parsing HTML is
       not an intended function of a MUD manager.";
  }
  leaf-list extensions {
    type string {
```

```
        length "1..40";
    }
    description
        "A list of extension names that are used in this MUD
        file. Each name is registered with the IANA and
        described in an RFC.";
    }
    container from-device-policy {
        description
            "The policies that should be enforced on traffic
            coming from the device. These policies are not
            necessarily intended to be enforced at a single
            point, but may be rendered by the controller to any
            relevant enforcement points in the network or
            elsewhere.";
        uses access-lists;
    }
    container to-device-policy {
        description
            "The policies that should be enforced on traffic
            going to the device. These policies are not
            necessarily intended to be enforced at a single
            point, but may be rendered by the controller to any
            relevant enforcement points in the network or
            elsewhere.";
        uses access-lists;
    }
}

grouping access-lists {
    description
        "A grouping for access lists in the context of device
        policy.";
    container access-lists {
        description
            "The access lists that should be applied to traffic
            to or from the device.";
        list access-list {
            key "name";
            description
                "Each entry on this list refers to an ACL that
                should be present in the overall access list
                data model. Each ACL is identified by name and
                type.";
            leaf name {
                type leafref {
                    path "/acl:acls/acl:acl/acl:name";
                }
            }
        }
    }
}
```

```
        description
            "The name of the ACL for this entry.";
    }
}
}

augment "/acl:acls/acl:acl/acl:aces/acl:ace/acl:matches" {
    description
        "adding abstractions to avoid need of IP addresses";
    container mud {
        description
            "MUD-specific matches.";
        leaf manufacturer {
            type inet:host;
            description
                "A domain that is intended to match the authority
                 section of the MUD URL. This node is used to specify
                 one or more manufacturers a device should
                 be authorized to access.";
        }
        leaf same-manufacturer {
            type empty;
            description
                "This node matches the authority section of the MUD URL
                 of a Thing. It is intended to grant access to all
                 devices with the same authority section.";
        }
        leaf model {
            type inet:uri;
            description
                "Devices of the specified model type will match if
                 they have an identical MUD URL.";
        }
        leaf local-networks {
            type empty;
            description
                "IP addresses will match this node if they are
                 considered local addresses. A local address may be
                 a list of locally defined prefixes and masks
                 that indicate a particular administrative scope.";
        }
        leaf controller {
            type inet:uri;
            description
                "This node names a class that has associated with it
                 zero or more IP addresses to match against. These
                 may be scoped to a manufacturer or via a standard
```

```

        URN.";
    }
    leaf my-controller {
        type empty;
        description
            "This node matches one or more network elements that
            have been configured to be the controller for this
            Thing, based on its MUD URL.";
    }
}
}
}
augment "/acl:acls/acl:acl/acl:aces/acl:ace/acl:matches" +
    "/acl:l4/acl:tcp/acl:tcp" {
    description
        "add direction-initiated";
    leaf direction-initiated {
        type direction;
        description
            "This node matches based on which direction a
            connection was initiated. The means by which that
            is determined is discussed in this document.";
    }
}
}
}
}
<CODE ENDS>

```

8. The Domain Name Extension to the ACL Model

This module specifies an extension to IETF-ACL model such that domain names may be referenced by augmenting the "matches" node. Different implementations may deploy differing methods to maintain the mapping between IP address and domain name, if indeed any are needed. However, the intent is that resources that are referred to using a name should be authorized (or not) within an access list.

The structure of the change is as follows:

```

module: ietf-acldns
    augment /acl:acls/acl:acl/acl:aces/acl:ace/
        acl:matches/acl:l3/acl:ipv4/acl:ipv4:
            +--rw src-dnsname?    inet:host
            +--rw dst-dnsname?    inet:host
    augment /acl:acls/acl:acl/acl:aces/acl:ace/
        acl:matches/acl:l3/acl:ipv6/acl:ipv6:
            +--rw src-dnsname?    inet:host
            +--rw dst-dnsname?    inet:host

```

The choice of these particular points in the access-list model is based on the assumption that we are in some way referring to IP-related resources, as that is what the DNS returns. A domain name in our context is defined in [RFC6991]. The augmentations are replicated across IPv4 and IPv6 to allow MUD file authors the ability to control the IP version that the Thing may utilize.

The following node are defined.

8.1. src-dnsname

The argument corresponds to a domain name of a source as specified by inet:host. A number of means may be used to resolve hosts. What is important is that such resolutions be consistent with ACLs required by Things to properly operate.

8.2. dst-dnsname

The argument corresponds to a domain name of a destination as specified by inet:host See the previous section relating to resolution.

Note when using either of these with a MUD file, because access is associated with a particular Thing, MUD files MUST NOT contain either a src-dnsname in an ACL associated with from-device-policy or a dst-dnsname associated with to-device-policy.

8.3. The ietf-acldns Model

```
<CODE BEGINS>file "ietf-acldns@2018-06-15.yang"
module ietf-acldns {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-acldns";
  prefix ietf-acldns;

  import ietf-access-control-list {
    prefix acl;
  }
  import ietf-inet-types {
    prefix inet;
  }

  organization
    "IETF OPSAWG (Ops Area) Working Group";
  contact
    "WG Web: http://tools.ietf.org/wg/opsawg/
    WG List: opsawg@ietf.org
    Author: Eliot Lear
```



```
    lear@cisco.com
    Author: Ralph Droms
    rdroms@gmail.com
    Author: Dan Romascanu
    dromasca@gmail.com
    ";
description
    "This YANG module defines a component that augments the
    IETF description of an access list to allow DNS names
    as matching criteria.";

revision 2018-06-15 {
    description
        "Base version of dnsname extension of ACL model";
    reference
        "RFC XXXX: Manufacturer Usage Description
        Specification";
}

grouping dns-matches {
    description
        "Domain names for matching.";
    leaf src-dnsname {
        type inet:host;
        description
            "domain name to be matched against";
    }
    leaf dst-dnsname {
        type inet:host;
        description
            "domain name to be matched against";
    }
}

augment "/acl:acls/acl:acl/acl:aces/acl:ace/acl:matches" +
"/acl:l3/acl:ipv4/acl:ipv4" {
    description
        "Adding domain names to matching";
    uses dns-matches;
}
augment "/acl:acls/acl:acl/acl:aces/acl:ace/acl:matches" +
"/acl:l3/acl:ipv6/acl:ipv6" {
    description
        "Adding domain names to matching";
    uses dns-matches;
}
}
<CODE ENDS>
```

9. MUD File Example

This example contains two access lists that are intended to provide outbound access to a cloud service on TCP port 443.

```
{
  "ietf-mud:mud": {
    "mud-version": 1,
    "mud-url": "https://lighting.example.com/lightbulb2000",
    "last-update": "2018-03-02T11:20:51+01:00",
    "cache-validity": 48,
    "is-supported": true,
    "systeminfo": "The BMS Example Lightbulb",
    "from-device-policy": {
      "access-lists": {
        "access-list": [
          {
            "name": "mud-76100-v6fr"
          }
        ]
      }
    },
    "to-device-policy": {
      "access-lists": {
        "access-list": [
          {
            "name": "mud-76100-v6to"
          }
        ]
      }
    }
  },
  "ietf-access-control-list:acls": {
    "acl": [
      {
        "name": "mud-76100-v6to",
        "type": "ipv6-acl-type",
        "aces": {
          "ace": [
            {
              "name": "cl0-todev",
              "matches": {
                "ipv6": {
                  "ietf-acldns:src-dnsname": "test.example.com",
                  "protocol": 6
                },
              },
              "tcp": {
                "ietf-mud:direction-initiated": "from-device",

```

```

        "source-port": {
            "operator": "eq",
            "port": 443
        }
    },
    "actions": {
        "forwarding": "accept"
    }
}
]
},
{
    "name": "mud-76100-v6fr",
    "type": "ipv6-acl-type",
    "aces": [
        "ace": [
            {
                "name": "cl0-frdev",
                "matches": {
                    "ipv6": {
                        "ietf-acldns:dst-dnsname": "test.example.com",
                        "protocol": 6
                    },
                    "tcp": {
                        "ietf-mud:direction-initiated": "from-device",
                        "destination-port": {
                            "operator": "eq",
                            "port": 443
                        }
                    }
                },
                "actions": {
                    "forwarding": "accept"
                }
            }
        ]
    }
}
]
}
}

```

In this example, two policies are declared, one from the Thing and the other to the Thing. Each policy names an access list that applies to the Thing, and one that applies from. Within each access

list, access is permitted to packets flowing to or from the Thing that can be mapped to the domain name of "service.bms.example.com". For each access list, the enforcement point should expect that the Thing initiated the connection.

10. The MUD URL DHCP Option

The IPv4 MUD URL client option has the following format:

```
+-----+-----+-----+
| code | len | MUDstring
+-----+-----+-----+
```

Code `OPTION_MUD_URL_V4` (161) is assigned by IANA. `len` is a single octet that indicates the length of MUD string in octets. The MUD string is defined as follows:

```
MUDstring = mudurl [ " " reserved ]
mudurl = URI; a URL [RFC3986] that uses the "https" schema [RFC7230]
reserved = 1*( OCTET ) ; from [RFC5234]
```

The entire option MUST NOT exceed 255 octets. If a space follows the MUD URL, a reserved string that will be defined in future specifications follows. MUD managers that do not understand this field MUST ignore it.

The IPv6 MUD URL client option has the following format:

```

0          1          2          3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+
|          OPTION_MUD_URL_V6          |          option-length          |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     MUDstring                             |
+-----+-----+-----+-----+-----+-----+-----+-----+
```

`OPTION_MUD_URL_V6` (112; assigned by IANA).

`option-length` contains the length of the MUDstring, as defined above, in octets.

The intent of this option is to provide both a new Thing classifier to the network as well as some recommended configuration to the routers that implement policy. However, it is entirely the purview

of the network system as managed by the network administrator to decide what to do with this information. The key function of this option is simply to identify the type of Thing to the network in a structured way such that the policy can be easily found with existing toolsets.

10.1. Client Behavior

A DHCPv4 client MAY emit a DHCPv4 option and a DHCPv6 client MAY emit DHCPv6 option. These options are singletons, as specified in [RFC7227]. Because clients are intended to have at most one MUD URL associated with them, they may emit at most one MUD URL option via DHCPv4 and one MUD URL option via DHCPv6. In the case where both v4 and v6 DHCP options are emitted, the same URL MUST be used.

10.2. Server Behavior

A DHCP server may ignore these options or take action based on receipt of these options. When a server consumes this option, it will either forward the URL and relevant client information (such as the gateway address or giaddr and requested IP address, and lease length) to a network management system, or it will retrieve the usage description itself by resolving the URL.

DHCP servers may implement MUD functionality themselves or they may pass along appropriate information to a network management system or MUD manager. A DHCP server that does process the MUD URL MUST adhere to the process specified in [RFC2818] and [RFC5280] to validate the TLS certificate of the web server hosting the MUD file. Those servers will retrieve the file, process it, create and install the necessary configuration on the relevant network element. Servers SHOULD monitor the gateway for state changes on a given interface. A DHCP server that does not provide MUD functionality and has forwarded a MUD URL to a MUD manager MUST notify the MUD manager of any corresponding change to the DHCP state of the client (such as expiration or explicit release of a network address lease).

Should the DHCP server fail, in the case when it implements the MUD manager functionality, any backup mechanisms SHOULD include the MUD state, and the server SHOULD resolve the status of clients upon its restart, similar to what it would do, absent MUD manager functionality. In the case where the DHCP server forwards information to the MUD manager, the MUD manager will either make use of redundant DHCP servers for information, or otherwise clear state based on other network information, such as monitoring port status on a switch via SNMP, Radius accounting, or similar mechanisms.

10.3. Relay Requirements

There are no additional requirements for relays.

11. The Manufacturer Usage Description (MUD) URL X.509 Extension

This section defines an X.509 non-critical certificate extension that contains a single Uniform Resource Locator (URL) that points to an on-line Manufacturer Usage Description concerning the certificate subject. URI must be represented as described in Section 7.4 of [RFC5280].

Any Internationalized Resource Identifiers (IRIs) MUST be mapped to URIs as specified in Section 3.1 of [RFC3987] before they are placed in the certificate extension.

The semantics of the URL are defined Section 6 of this document.

The choice of id-pe is based on guidance found in Section 4.2.2 of [RFC5280]:

These extensions may be used to direct applications to on-line information about the issuer or the subject.

The MUD URL is precisely that: online information about the particular subject.

In addition, a separate new extension is defined as id-pe-mudsigner. This contains the subject field of the signing certificate of the MUD file. Processing of this field is specified in Section 13.2.

The purpose of this signature is to make a claim that the MUD file found on the server is valid for a given device, independent of any other factors. There are several security considerations below in Section 16.

A new content-type id-ct-mud is also defined. While signatures are detached today, should a MUD file be transmitted as part of a CMS message, this content-type SHOULD be used.

The new extension is identified as follows:

```
<CODE BEGINS>
MUDURLExtnModule-2016 { iso(1) identified-organization(3) dod(6)
    internet(1) security(5) mechanisms(5) pkix(7)
    id-mod(0) id-mod-mudURLExtn2016(88) }
DEFINITIONS IMPLICIT TAGS ::= BEGIN
```

```
-- EXPORTS ALL --

IMPORTS

-- RFC 5912
EXTENSION
FROM PKIX-CommonTypes-2009
    { iso(1) identified-organization(3) dod(6) internet(1)
      security(5) mechanisms(5) pkix(7) id-mod(0)
      id-mod-pkixCommon-02(57) }

-- RFC 5912
id-ct
FROM PKIXCRMF-2009
    { iso(1) identified-organization(3) dod(6) internet(1)
      security(5) mechanisms(5) pkix(7) id-mod(0)
      id-mod-crmf2005-02(55) }

-- RFC 6268
CONTENT-TYPE
FROM CryptographicMessageSyntax-2010
    { iso(1) member-body(2) us(840) rsadsi(113549)
      pkcs(1) pkcs-9(9) smime(16) modules(0) id-mod-cms-2009(58) }

-- RFC 5912
id-pe, Name
FROM PKIX1Explicit-2009
    { iso(1) identified-organization(3) dod(6) internet(1)
      security(5) mechanisms(5) pkix(7) id-mod(0)
      id-mod-pkix1-explicit-02(51) } ;

--
-- Certificate Extensions
--

MUDCertExtensions EXTENSION ::=
    { ext-MUDURL | ext-MUDsigner, ... }

ext-MUDURL EXTENSION ::=
    { SYNTAX MUDURLSyntax IDENTIFIED BY id-pe-mud-url }

id-pe-mud-url OBJECT IDENTIFIER ::= { id-pe 25 }

MUDURLSyntax ::= IA5String

ext-MUDsigner EXTENSION ::=
    { SYNTAX MUDsignerSyntax IDENTIFIED BY id-pe-mudsigner }
```

```

id-pe-mudsigner OBJECT IDENTIFIER ::= { id-pe TBD1 }

MUDsignerSyntax ::= Name

--
-- CMS Content Types
--

MUDContentTypes CONTENT-TYPE ::=
    { ct-mud, ... }

ct-mud CONTENT-TYPE ::=
    { -- directly include the content
      IDENTIFIED BY id-ct-mudtype }
    -- The binary data that is in the form
    -- 'application/mud+json' is directly encoded as the
    -- signed data. No additional ASN.1 encoding is added.

id-ct-mudtype OBJECT IDENTIFIER ::= { id-ct TBD2 }

END
<CODE ENDS>

```

While this extension can appear in either an 802.AR manufacturer certificate (IDevID) or deployment certificate (LDevID), of course it is not guaranteed in either, nor is it guaranteed to be carried over. It is RECOMMENDED that MUD manager implementations maintain a table that maps a Thing to its MUD URL based on IDevIDs.

12. The Manufacturer Usage Description LLDP extension

The IEEE802.1AB Link Layer Discovery Protocol (LLDP) is a one hop vendor-neutral link layer protocol used by end hosts network Things for advertising their identity, capabilities, and neighbors on an IEEE 802 local area network. Its Type-Length-Value (TLV) design allows for 'vendor-specific' extensions to be defined. IANA has a registered IEEE 802 organizationally unique identifier (OUI) defined as documented in [RFC7042]. The MUD LLDP extension uses a subtype defined in this document to carry the MUD URL.

The LLDP vendor specific frame has the following format:

TLV Type	len	OUI	subtype	MUDString
=127		= 00 00 5E	= 1	
(7 bits)	(9 bits)	(3 octets)	(1 octet)	(1-255 octets)

where:

- o TLV Type = 127 indicates a vendor-specific TLV
- o len - indicates the TLV string length
- o OUI = 00 00 5E is the organizationally unique identifier of IANA
- o subtype = 1 (to be assigned by IANA for the MUD URL)
- o MUD URL - the length MUST NOT exceed 255 octets

The intent of this extension is to provide both a new Thing classifier to the network as well as some recommended configuration to the routers that implement policy. However, it is entirely the purview of the network system as managed by the network administrator to decide what to do with this information. The key function of this extension is simply to identify the type of Thing to the network in a structured way such that the policy can be easily found with existing toolsets.

Hosts, routers, or other network elements that implement this option are intended to have at most one MUD URL associated with them, so they may transmit at most one MUD URL value.

Hosts, routers, or other network elements that implement this option may ignore these options or take action based on receipt of these options. For example they may fill in information in the respective extensions of the LLDP Management Information Base (LLDP MIB). LLDP operates in a one-way direction. LLDPDUs are not exchanged as information requests by one Thing and response sent by another Thing. The other Things do not acknowledge LLDP information received from a Thing. No specific network behavior is guaranteed. When a Thing consumes this extension, it may either forward the URL and relevant remote Thing information to a MUD manager, or it will retrieve the usage description by resolving the URL in accordance with normal HTTP semantics.

13. Creating and Processing of Signed MUD Files

Because MUD files contain information that may be used to configure network access lists, they are sensitive. To ensure that they have not been tampered with, it is important that they be signed. We make use of DER-encoded Cryptographic Message Syntax (CMS) [RFC5652] for this purpose.

13.1. Creating a MUD file signature

A MUD file MUST be signed using CMS as an opaque binary object. In order to make successful verification more likely, intermediate certificates SHOULD be included. The signature is stored at the location specified in the MUD file. Signatures are transferred using content-type "application/pkcs7-signature".

For example:

```
% openssl cms -sign -signer mancrtfile -inkey mankey \  
-in mudfile -binary -outform DER -binary \  
-certfile intermediatecert -out mudfile.p7s
```

Note: A MUD file may need to be re-signed if the signature expires.

13.2. Verifying a MUD file signature

Prior to processing the rest of a MUD file, the MUD manager MUST retrieve the MUD signature file by retrieving the value of "mud-signature" and validating the signature across the MUD file. The Key Usage Extension in the signing certificate MUST be present and have the bit digitalSignature(0) set. When the id-pe-mudsigner extension is present in a device's X.509 certificate, the MUD signature file MUST have been generated by a certificate whose subject matches the contents of that id-pe-mudsigner extension. If these conditions are not met, or if it cannot validate the chain of trust to a known trust anchor, the MUD manager MUST cease processing the MUD file until an administrator has given approval.

The purpose of the signature on the file is to assign accountability to an entity, whose reputation can be used to guide administrators on whether or not to accept a given MUD file. It is already common place to check web reputation on the location of a server on which a file resides. While it is likely that the manufacturer will be the signer of the file, this is not strictly necessary, and may not be desirable. For one thing, in some environments, integrators may install their own certificates. For another, what is more important is the accountability of the recommendation, and not just the relationship between the Thing and the file.

An example:

```
% openssl cms -verify -in mudfile.p7s -inform DER -content mudfile
```

Note the additional step of verifying the common trust root.

14. Extensibility

One of our design goals is to see that MUD files are able to be understood by as broad a cross-section of systems as is possible. Coupled with the fact that we have also chosen to leverage existing mechanisms, we are left with no ability to negotiate extensions and a limited desire for those extensions in any event. A such, a two-tier extensibility framework is employed, as follows:

1. At a coarse grain, a protocol version is included in a MUD URL. This memo specifies MUD version 1. Any and all changes are entertained when this version is bumped. Transition approaches between versions would be a matter for discussion in future versions.
2. At a finer grain, only extensions that would not incur additional risk to the Thing are permitted. Specifically, adding nodes to the mud container is permitted with the understanding that such additions will be ignored by unaware implementations. Any such extensions SHALL be standardized through the IETF process, and MUST be named in the "extensions" list. MUD managers MUST ignore YANG nodes they do not understand and SHOULD create an exception to be resolved by an administrator, so as to avoid any policy inconsistencies.

15. Deployment Considerations

Because MUD consists of a number of architectural building blocks, it is possible to assemble different deployment scenarios. One key aspect is where to place policy enforcement. In order to protect the Thing from other Things within a local deployment, policy can be enforced on the nearest switch or access point. In order to limit unwanted traffic within a network, it may also be advisable to enforce policy as close to the Internet as possible. In some circumstances, policy enforcement may not be available at the closest hop. At that point, the risk of lateral infection (infection of devices that reside near one another) is increased to the number of Things that are able to communicate without protection.

A caution about some of the classes: admission of a Thing into the "manufacturer" and "same-manufacturer" class may have impact on access of other Things. Put another way, the admission may grow the access-list on switches connected to other Things, depending on how access is managed. Some care should be given on managing that access-list growth. Alternative methods such as additional network segmentation can be used to keep that growth within reason.

Because as of this writing MUD is a new concept, one can expect a great many devices to not have implemented it. It remains a local deployment decision as to whether a device that is first connected should be allowed broad or limited access. Furthermore, as mentioned in the introduction, a deployment may choose to ignore a MUD policy in its entirety, but simply taken into account the MUD URL as a classifier to be used as part of a local policy decision.

Finally, please see directly below regarding device lifetimes and use of domain names.

16. Security Considerations

Based on how a MUD URL is emitted, a Thing may be able to lie about what it is, thus gaining additional network access. This can happen in a number of ways when a device emits a MUD URL using DHCP or LLDP, such as being inappropriately admitted to a class such as "same-manufacturer", given access to a device such as "my-controller", or being permitted access to an Internet resource, where such access would otherwise be disallowed. Whether that is the case will depend on the deployment. Implementations SHOULD be configurable to disallow additive access for devices using MUD-URLs that are not emitted in a secure fashion such as in a certificate. Similarly, implementations SHOULD NOT grant elevated permissions (beyond those of devices presenting no MUD policy) to devices which do not strongly bind their identity to their L2/L3 transmissions. When insecure methods are used by the MUD Manager, the classes SHOULD NOT contain devices that use both insecure and secure methods, in order to prevent privilege escalation attacks, and MUST NOT contain devices with the same MUD-URL that are derived from both strong and weak authentication methods.

Devices may forge source (L2/L3) information. Deployments should apply appropriate protections to bind communications to the authentication that has taken place. For 802.1X authentication, IEEE 802.1AE (MACsec) [IEEE8021AE] is one means by which this may happen. A similar approach can be used with 802.11i (WPA2) [IEEE80211i]. Other means are available with other lower layer technologies. Implementations using session-oriented access that is not cryptographically bound should take care to remove state when any form of break in the session is detected.

A rogue CA may sign a certificate that contains the same subject name as is listed in the MUDsigner field in the manufacturer certificate, thus seemingly permitting a substitute MUD file for a device. There are two mitigations available: first, if the signer changes, this may be flagged as an exception by the MUD manager. If the MUD file also changes, the MUD manager SHOULD seek administrator approval (it

should do this in any case). In all circumstances, the MUD manager MUST maintain a cache of trusted CAs for this purpose. When such a rogue is discovered, it SHOULD be removed.

Additional mitigations are described below.

When certificates are not present, Things claiming to be of a certain manufacturer SHOULD NOT be included in that manufacturer grouping without additional validation of some form. This will be relevant when the MUD manager makes use of primitives such as "manufacturer" for the purpose of accessing Things of a particular type. Similarly, network management systems may be able to fingerprint the Thing. In such cases, the MUD URL can act as a classifier that can be proven or disproven. Fingerprinting may have other advantages as well: when 802.1AR certificates are used, because they themselves cannot change, fingerprinting offers the opportunity to add artifacts to the MUD string in the form of the reserved field discussed in Section 10. The meaning of such artifacts is left as future work.

MUD managers SHOULD NOT accept a usage description for a Thing with the same MAC address that has indicated a change of the URL authority without some additional validation (such as review by a network administrator). New Things that present some form of unauthenticated MUD URL SHOULD be validated by some external means when they would be given increased network access.

It may be possible for a rogue manufacturer to inappropriately exercise the MUD file parser, in order to exploit a vulnerability. There are three recommended approaches to address this threat. The first is to validate that the signer of the MUD file is known to and trusted by the MUD manager. The second is to have a system do a primary scan of the file to ensure that it is both parseable and believable at some level. MUD files will likely be relatively small, to start with. The number of ACEs used by any given Thing should be relatively small as well. It may also be useful to limit retrieval of MUD URLs to only those sites that are known to have decent web or domain reputations.

Use of a URL necessitates the use of domain names. If a domain name changes ownership, the new owner of that domain may be able to provide MUD files that MUD managers would consider valid. There are a few approaches that can mitigate this attack. First, MUD managers SHOULD cache certificates used by the MUD file server. When a new certificate is retrieved for whatever reason, the MUD manager should check to see if ownership of the domain has changed. A fair programmatic approximation of this is when the name servers for the domain have changed. If the actual MUD file has changed, the MUD manager MAY check the WHOIS database to see if registration ownership

of a domain has changed. If a change has occurred, or if for some reason it is not possible to determine whether ownership has changed, further review may be warranted. Note, this remediation does not take into account the case of a Thing that was produced long ago and only recently fielded, or the case where a new MUD manager has been installed.

The release of a MUD URL by a Thing reveals what the Thing is, and provides an attacker with guidance on what vulnerabilities may be present.

While the MUD URL itself is not intended to be unique to a specific Thing, the release of the URL may aid an observer in identifying individuals when combined with other information. This is a privacy consideration.

In addressing both of these concerns, implementors should take into account what other information they are advertising through mechanisms such as mDNS[RFC6872], how a Thing might otherwise be identified, perhaps through how it behaves when it is connected to the network, whether a Thing is intended to be used by individuals or carry personal identifying information, and then apply appropriate data minimization techniques. One approach is to make use of TEAP [RFC7170] as the means to share information with authorized components in the network. Network elements may also assist in limiting access to the MUD URL through the use of mechanisms such as DHCPv6-Shield [RFC7610].

There is the risk of the MUD manager itself being spied on to determine what things are connected to the network. To address this risk, MUD managers may choose to make use of TLS proxies that they trust that would aggregate other information.

Please note that the security considerations mentioned in Section 4.7 of [I-D.ietf-netmod-rfc6087bis] are not applicable in this case because the YANG serialization is not intended to be accessed via NETCONF. However, for those who try to instantiate this model in a network element via NETCONF, all objects in each model in this draft exhibit similar security characteristics as [I-D.ietf-netmod-acl-model]. The basic purpose of MUD is to configure access, and so by its very nature can be disruptive if used by unauthorized parties.

17. IANA Considerations

[There was originally a registry entry for .well-known suffixes. This has been removed from the draft and may be marked as deprecated in the registry. RFC Editor: please remove this comment.]

17.1. YANG Module Registrations

The following YANG modules are requested to be registered in the "IANA Module Names" registry:

The ietf-mud module:

- o Name: ietf-mud
- o URN: urn:ietf:params:xml:ns:yang:ietf-mud
- o Prefix: ietf-mud
- o Registrant contact: The IESG
- o Reference: [RFCXXXX]

The ietf-acldns module:

- o Name: ietf-acldns
- o URI: urn:ietf:params:xml:ns:yang:ietf-acldns
- o Prefix: ietf-acldns
- o Registrant: the IESG
- o Reference: [RFCXXXX]

17.2. DHCPv4 and DHCPv6 Options

The IANA has allocated option 161 in the Dynamic Host Configuration Protocol (DHCP) and Bootstrap Protocol (BOOTP) Parameters registry for the MUD DHCPv4 option, and option 112 for DHCPv6, as described in Section 10.

17.3. PKIX Extensions

IANA is kindly requested to make the following assignments for:

- o The MUDURLExtnModule-2016 ASN.1 module in the "SMI Security for PKIX Module Identifier" registry (1.3.6.1.5.5.7.0).
- o id-pe-mud-url object identifier from the "SMI Security for PKIX Certificate Extension" registry (1.3.6.1.5.5.7.1).
- o id-pe-mudsigner object identifier from the "SMI Security for PKIX Certificate Extension" registry (TBD1).

- o id-ct-mudtype object identifier from the "SMI Security for S/MIME CMS Content Type" registry (TBD2).

The use of these values is specified in Section 11.

17.4. MIME Media-type Registration for MUD files

The following media-type is defined for transfer of MUD file:

- o Type name: application
- o Subtype name: mud+json
- o Required parameters: n/a
- o Optional parameters: n/a
- o Encoding considerations: 8bit; application/mud+json values are represented as a JSON object; UTF-8 encoding MUST be employed. [RFC3629]
- o Security considerations: See Security Considerations of RFCXXXX and [RFC8259] Section 12.
- o Interoperability considerations: n/a
- o Published specification: [RFCXXXX]
- o Applications that use this media type: MUD managers as specified by [RFCXXXX].
- o Fragment identifier considerations: n/a
- o Additional information:
 - Magic number(s): n/a
 - File extension(s): n/a
 - Macintosh file type code(s): n/a
- o Person & email address to contact for further information: Eliot Lear <lear@cisco.com>, Ralph Droms <rdroms@gmail.com>
- o Intended usage: COMMON
- o Restrictions on usage: none
- o Author:
 - Eliot Lear <lear@cisco.com>
 - Ralph Droms <rdroms@gmail.com>
- o Change controller: IESG
- o Provisional registration? (standards tree only): No.

17.5. LLDP IANA TLV Subtype Registry

IANA is requested to create a new registry for IANA Link Layer Discovery Protocol (LLDP) TLV subtype values. The recommended policy for this registry is Expert Review. The maximum number of entries in the registry is 256.

IANA is required to populate the initial registry with the value:

LLDP subtype value = 1 (All the other 255 values should be initially marked as 'Unassigned'.)

Description = the Manufacturer Usage Description (MUD) Uniform Resource Locator (URL)

Reference = < this document >

17.6. The MUD Well Known Universal Resource Name (URNs)

The following parameter registry is requested to be added in accordance with [RFC3553]

Registry name: "urn:ietf:params:mud" is requested.
Specification: this document
Repository: this document
Index value: Encoded identically to a TCP/UDP port service name, as specified in Section 5.1 of [RFC6335]

The following entries should be added to the "urn:ietf:params:mud" name space:

"urn:ietf:params:mud:dns" refers to the service specified by [RFC1123]. "urn:ietf:params:mud:ntp" refers to the service specified by [RFC5905].

17.7. Extensions Registry

The IANA is requested to establish a registry of extensions as follows:

Registry name: MUD extensions registry
Registry policy: Standards action
Standard reference: document
Extension name: UTF-8 encoded string, not to exceed 40 characters.

Each extension MUST follow the rules specified in this specification. As is usual, the IANA issues early allocations based in accordance with [RFC7120].

18. Acknowledgments

The authors would like to thank Einar Nilsen-Nygaard, who singlehandedly updated the model to match the updated ACL model, Bernie Volz, Tom Gindin, Brian Weis, Sandeep Kumar, Thorsten Dahm, John Bashinski, Steve Rich, Jim Bieda, Dan Wing, Joe Clarke, Henk Birkholz, Adam Montville, Jim Schaad, and Robert Sparks for their valuable advice and reviews. Russ Housley entirely rewrote

Section 11 to be a complete module. Adrian Farrel provided the basis for privacy considerations text. Kent Watsen provided a thorough review of the architecture and the YANG model. The remaining errors in this work are entirely the responsibility of the authors.

19. References

19.1. Normative References

- [I-D.ietf-netmod-acl-model]
Jethanandani, M., Huang, L., Agarwal, S., and D. Blair,
"Network Access Control List (ACL) YANG Data Model",
draft-ietf-netmod-acl-model-19 (work in progress), April
2018.
- [IEEE8021AB]
Institute for Electrical and Electronics Engineers, "IEEE
Standard for Local and Metropolitan Area Networks--
Station and Media Access Control Connectivity Discovery",
n.d..
- [RFC1123] Braden, R., Ed., "Requirements for Internet Hosts -
Application and Support", STD 3, RFC 1123,
DOI 10.17487/RFC1123, October 1989,
<<https://www.rfc-editor.org/info/rfc1123>>.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate
Requirement Levels", BCP 14, RFC 2119,
DOI 10.17487/RFC2119, March 1997,
<<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC2131] Droms, R., "Dynamic Host Configuration Protocol",
RFC 2131, DOI 10.17487/RFC2131, March 1997,
<<https://www.rfc-editor.org/info/rfc2131>>.
- [RFC2818] Rescorla, E., "HTTP Over TLS", RFC 2818,
DOI 10.17487/RFC2818, May 2000,
<<https://www.rfc-editor.org/info/rfc2818>>.
- [RFC3315] Droms, R., Ed., Bound, J., Volz, B., Lemon, T., Perkins,
C., and M. Carney, "Dynamic Host Configuration Protocol
for IPv6 (DHCPv6)", RFC 3315, DOI 10.17487/RFC3315, July
2003, <<https://www.rfc-editor.org/info/rfc3315>>.
- [RFC3629] Yergeau, F., "UTF-8, a transformation format of ISO
10646", STD 63, RFC 3629, DOI 10.17487/RFC3629, November
2003, <<https://www.rfc-editor.org/info/rfc3629>>.

- [RFC3748] Aboba, B., Blunk, L., Vollbrecht, J., Carlson, J., and H. Levkowetz, Ed., "Extensible Authentication Protocol (EAP)", RFC 3748, DOI 10.17487/RFC3748, June 2004, <<https://www.rfc-editor.org/info/rfc3748>>.
- [RFC3986] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, DOI 10.17487/RFC3986, January 2005, <<https://www.rfc-editor.org/info/rfc3986>>.
- [RFC3987] Duerst, M. and M. Suignard, "Internationalized Resource Identifiers (IRIs)", RFC 3987, DOI 10.17487/RFC3987, January 2005, <<https://www.rfc-editor.org/info/rfc3987>>.
- [RFC5234] Crocker, D., Ed. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, DOI 10.17487/RFC5234, January 2008, <<https://www.rfc-editor.org/info/rfc5234>>.
- [RFC5280] Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., and W. Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 5280, DOI 10.17487/RFC5280, May 2008, <<https://www.rfc-editor.org/info/rfc5280>>.
- [RFC5652] Housley, R., "Cryptographic Message Syntax (CMS)", STD 70, RFC 5652, DOI 10.17487/RFC5652, September 2009, <<https://www.rfc-editor.org/info/rfc5652>>.
- [RFC5905] Mills, D., Martin, J., Ed., Burbank, J., and W. Kasch, "Network Time Protocol Version 4: Protocol and Algorithms Specification", RFC 5905, DOI 10.17487/RFC5905, June 2010, <<https://www.rfc-editor.org/info/rfc5905>>.
- [RFC5911] Hoffman, P. and J. Schaad, "New ASN.1 Modules for Cryptographic Message Syntax (CMS) and S/MIME", RFC 5911, DOI 10.17487/RFC5911, June 2010, <<https://www.rfc-editor.org/info/rfc5911>>.
- [RFC5912] Hoffman, P. and J. Schaad, "New ASN.1 Modules for the Public Key Infrastructure Using X.509 (PKIX)", RFC 5912, DOI 10.17487/RFC5912, June 2010, <<https://www.rfc-editor.org/info/rfc5912>>.

- [RFC6335] Cotton, M., Eggert, L., Touch, J., Westerlund, M., and S. Cheshire, "Internet Assigned Numbers Authority (IANA) Procedures for the Management of the Service Name and Transport Protocol Port Number Registry", BCP 165, RFC 6335, DOI 10.17487/RFC6335, August 2011, <<https://www.rfc-editor.org/info/rfc6335>>.
- [RFC6991] Schoenwaelder, J., Ed., "Common YANG Data Types", RFC 6991, DOI 10.17487/RFC6991, July 2013, <<https://www.rfc-editor.org/info/rfc6991>>.
- [RFC7120] Cotton, M., "Early IANA Allocation of Standards Track Code Points", BCP 100, RFC 7120, DOI 10.17487/RFC7120, January 2014, <<https://www.rfc-editor.org/info/rfc7120>>.
- [RFC7227] Hankins, D., Mrugalski, T., Siodelski, M., Jiang, S., and S. Krishnan, "Guidelines for Creating New DHCPv6 Options", BCP 187, RFC 7227, DOI 10.17487/RFC7227, May 2014, <<https://www.rfc-editor.org/info/rfc7227>>.
- [RFC7230] Fielding, R., Ed. and J. Reschke, Ed., "Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing", RFC 7230, DOI 10.17487/RFC7230, June 2014, <<https://www.rfc-editor.org/info/rfc7230>>.
- [RFC7231] Fielding, R., Ed. and J. Reschke, Ed., "Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content", RFC 7231, DOI 10.17487/RFC7231, June 2014, <<https://www.rfc-editor.org/info/rfc7231>>.
- [RFC7610] Gont, F., Liu, W., and G. Van de Velde, "DHCPv6-Shield: Protecting against Rogue DHCPv6 Servers", BCP 199, RFC 7610, DOI 10.17487/RFC7610, August 2015, <<https://www.rfc-editor.org/info/rfc7610>>.
- [RFC7950] Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language", RFC 7950, DOI 10.17487/RFC7950, August 2016, <<https://www.rfc-editor.org/info/rfc7950>>.
- [RFC7951] Lhotka, L., "JSON Encoding of Data Modeled with YANG", RFC 7951, DOI 10.17487/RFC7951, August 2016, <<https://www.rfc-editor.org/info/rfc7951>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.

- [RFC8259] Bray, T., Ed., "The JavaScript Object Notation (JSON) Data Interchange Format", STD 90, RFC 8259, DOI 10.17487/RFC8259, December 2017, <<https://www.rfc-editor.org/info/rfc8259>>.
- [RFC8340] Bjorklund, M. and L. Berger, Ed., "YANG Tree Diagrams", BCP 215, RFC 8340, DOI 10.17487/RFC8340, March 2018, <<https://www.rfc-editor.org/info/rfc8340>>.
- [RFC8348] Bierman, A., Bjorklund, M., Dong, J., and D. Romascanu, "A YANG Data Model for Hardware Management", RFC 8348, DOI 10.17487/RFC8348, March 2018, <<https://www.rfc-editor.org/info/rfc8348>>.

19.2. Informative References

- [FW95] Chapman, D. and E. Zwicky, "Building Internet Firewalls", January 1995.
- [I-D.ietf-netmod-rfc6087bis]
Bierman, A., "Guidelines for Authors and Reviewers of YANG Data Model Documents", draft-ietf-netmod-rfc6087bis-20 (work in progress), March 2018.
- [IEEE80211i]
Institute for Electrical and Electronics Engineers, "IEEE Standard for information technology-Telecommunications and information exchange between systems-Local and metropolitan area networks-Specific requirements-Part 11-Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications- Amendment 6- Medium Access Control (MAC) Security Enhancements", 2004.
- [IEEE8021AE]
Institute for Electrical and Electronics Engineers, "IEEE Standard for Local and Metropolitan Area Networks- Media Access Control (MAC) Security", 2006.
- [IEEE8021AR]
Institute for Electrical and Electronics Engineers, "Secure Device Identity", 1998.
- [IEEE8021X]
Institute for Electrical and Electronics Engineers, "IEEE Standard for Local and metropolitan area networks--Port-Based Network Access Control", 2010.

- [ISO.8601.1988] International Organization for Standardization, "Data elements and interchange formats - Information interchange - Representation of dates and times", ISO Standard 8601, June 1988.
- [RFC1984] IAB and IESG, "IAB and IESG Statement on Cryptographic Technology and the Internet", BCP 200, RFC 1984, DOI 10.17487/RFC1984, August 1996, <<https://www.rfc-editor.org/info/rfc1984>>.
- [RFC3339] Klyne, G. and C. Newman, "Date and Time on the Internet: Timestamps", RFC 3339, DOI 10.17487/RFC3339, July 2002, <<https://www.rfc-editor.org/info/rfc3339>>.
- [RFC3553] Mealling, M., Masinter, L., Hardie, T., and G. Klyne, "An IETF URN Sub-namespace for Registered Protocol Parameters", BCP 73, RFC 3553, DOI 10.17487/RFC3553, June 2003, <<https://www.rfc-editor.org/info/rfc3553>>.
- [RFC6092] Woodyatt, J., Ed., "Recommended Simple Security Capabilities in Customer Premises Equipment (CPE) for Providing Residential IPv6 Internet Service", RFC 6092, DOI 10.17487/RFC6092, January 2011, <<https://www.rfc-editor.org/info/rfc6092>>.
- [RFC6872] Gurbani, V., Ed., Burger, E., Ed., Anjali, T., Abdelnur, H., and O. Festor, "The Common Log Format (CLF) for the Session Initiation Protocol (SIP): Framework and Information Model", RFC 6872, DOI 10.17487/RFC6872, February 2013, <<https://www.rfc-editor.org/info/rfc6872>>.
- [RFC7042] Eastlake 3rd, D. and J. Abley, "IANA Considerations and IETF Protocol and Documentation Usage for IEEE 802 Parameters", BCP 141, RFC 7042, DOI 10.17487/RFC7042, October 2013, <<https://www.rfc-editor.org/info/rfc7042>>.
- [RFC7170] Zhou, H., Cam-Winget, N., Salowey, J., and S. Hanna, "Tunnel Extensible Authentication Protocol (TEAP) Version 1", RFC 7170, DOI 10.17487/RFC7170, May 2014, <<https://www.rfc-editor.org/info/rfc7170>>.
- [RFC7252] Shelby, Z., Hartke, K., and C. Bormann, "The Constrained Application Protocol (CoAP)", RFC 7252, DOI 10.17487/RFC7252, June 2014, <<https://www.rfc-editor.org/info/rfc7252>>.

- [RFC7452] Tschofenig, H., Arkko, J., Thaler, D., and D. McPherson, "Architectural Considerations in Smart Object Networking", RFC 7452, DOI 10.17487/RFC7452, March 2015, <<https://www.rfc-editor.org/info/rfc7452>>.
- [RFC7488] Boucadair, M., Penno, R., Wing, D., Patil, P., and T. Reddy, "Port Control Protocol (PCP) Server Selection", RFC 7488, DOI 10.17487/RFC7488, March 2015, <<https://www.rfc-editor.org/info/rfc7488>>.
- [RFC8343] Bjorklund, M., "A YANG Data Model for Interface Management", RFC 8343, DOI 10.17487/RFC8343, March 2018, <<https://www.rfc-editor.org/info/rfc8343>>.

Appendix A. Changes from Earlier Versions

RFC Editor to remove this section prior to publication.

Draft -19: * Edits after discussion with apps area to address reserved field for the future. * Correct systeminfo to be utf8. * Remove "hardware-rev" from list.

Draft -18: * Correct an error in the augment statement * Changes to the ACL model re ports.

Draft -17:

- o One editorial.

Draft -16

- o add mud-signature element based on review comments
- o redo mud-url
- o make clear that systeminfo uses UTF8

Draft -13 to -14:

- o Final WGLC comments and review comments
- o Move version from MUD-URL to Model
- o Have MUD-URL in model
- o Update based on update to draft-ietf-netmod-acl-model
- o Point to tree diagram draft instead of 6087bis.

Draft -12 to -13:

- o Additional WGLC comments

Draft -10 to -12:

These are based on WGLC comments:

- o Correct examples based on ACL model changes.
- o Change ordering nodes.
- o Additional explanatory text around systeminfo.
- o Change ordering in examples.
- o Make it VERY VERY VERY VERY clear that these are recommendations, not mandates.
- o DHCP -> NTP in some of the intro text.
- o Remove masa-server
- o "Things" to "network elements" in a few key places.
- o Reference to JSON YANG RFC added.

Draft -10 to -11:

- o Example corrections
- o Typo
- o Fix two lists.
- o Addition of 'any-acl' and 'mud-acl' in the list of allowed features.
- o Clarification of what should be in a MUD file.

Draft -09 to -10:

- o AD input.
- o Correct dates.
- o Add compliance sentence as to which ACL module features are implemented.

Draft -08 to -09:

- o Resolution of Security Area review, IoT directorate review, GenART review, YANG doctors review.
- o change of YANG structure to address mandatory nodes.
- o Terminology cleanup.
- o specify out extra portion of MUD-URL.
- o consistency changes.
- o improved YANG descriptions.
- o Remove extra revisions.
- o Track ACL model changes.
- o Additional cautions on use of ACL model; further clarifications on extensions.

Draft -07 to -08:

- o a number of editorials corrected.
- o definition of MUD file tweaked.

Draft -06 to -07:

- o Examples updated.
- o Additional clarification for direction-initiated.
- o Additional implementation guidance given.

Draft -06 to -07:

- o Update models to match new ACL model
- o extract directionality from the ACL, introducing a new device container.

Draft -05 to -06:

- o Make clear that this is a component architecture (Polk and Watson)
- o Add order of operations (Watson)

- o Add extensions leaf-list (Pritikin)
- o Remove previous-mud-file (Watson)
- o Modify text in last-update (Watson)
- o Clarify local networks (Weis, Watson)
- o Fix contact info (Watson)
- o Terminology clarification (Weis)
- o Advice on how to handle LDevIDs (Watson)
- o Add deployment considerations (Watson)
- o Add some additional text about fingerprinting (Watson)
- o Appropriate references to 6087bis (Watson)
- o Change systeminfo to a URL to be referenced (Lear)

Draft -04 to -05: * syntax error correction

Draft -03 to -04: * Re-add my-controller

Draft -02 to -03: * Additional IANA updates * Format correction in YANG. * Add reference to TEAP.

Draft -01 to -02: * Update IANA considerations * Accept Russ Housley rewrite of X.509 text * Include privacy considerations text * Redo the URL limit. Still 255 bytes, but now stated in the URL definition. * Change URI registration to be under urn:ietf:params

Draft -00 to -01: * Fix cert trust text. * change supportInformation to meta-info * Add an informational element in. * add urn registry and create first entry * add default elements

Appendix B. Default MUD nodes

What follows is the portion of a MUD file that permits DNS traffic to a controller that is registered with the URN "urn:ietf:params:mud:dns" and traffic NTP to a controller that is registered "urn:ietf:params:mud:ntp". This is considered the default behavior and the ACEs are in effect appended to whatever other "ace" entries that a MUD file contains. To block DNS or NTP one repeats the matching statement but replaces the "forwarding" action "accept" with "drop". Because ACEs are processed in the order they are

received, the defaults would not be reached. A MUD manager might further decide to optimize to simply not include the defaults when they are overridden.

Four "acl" list entries that implement default MUD nodes are listed below. Two are for IPv4 and two are for IPv6 (one in each direction for both versions of IP). Note that neither access-list name nor ace name need be retained or used in any way by local implementations, but are simply there for completeness' sake.

```
"ietf-access-control-list:acls": {
  "acl": [
    {
      "name": "mud-59776-v4to",
      "type": "ipv4-acl-type",
      "aces": {
        "ace": [
          {
            "name": "ent0-todev",
            "matches": {
              "ietf-mud:mud": {
                "controller": "urn:ietf:params:mud:dns"
              },
              "ipv4": {
                "protocol": 17
              },
              "udp": {
                "source-port": {
                  "operator": "eq",
                  "port": 53
                }
              }
            },
            "actions": {
              "forwarding": "accept"
            }
          },
          {
            "name": "ent1-todev",
            "matches": {
              "ietf-mud:mud": {
                "controller": "urn:ietf:params:mud:ntp"
              },
              "ipv4": {
                "protocol": 17
              },
              "udp": {
                "source-port": {
```

```

        "operator": "eq",
        "port": 123
      }
    },
    "actions": {
      "forwarding": "accept"
    }
  ]
},
{
  "name": "mud-59776-v4fr",
  "type": "ipv4-acl-type",
  "aces": {
    "ace": [
      {
        "name": "ent0-frdev",
        "matches": {
          "ietf-mud:mud": {
            "controller": "urn:ietf:params:mud:dns"
          },
          "ipv4": {
            "protocol": 17
          },
          "udp": {
            "destination-port": {
              "operator": "eq",
              "port": 53
            }
          }
        },
        "actions": {
          "forwarding": "accept"
        }
      }
    ],
    "name": "ent1-frdev",
    "matches": {
      "ietf-mud:mud": {
        "controller": "urn:ietf:params:mud:ntp"
      },
      "ipv4": {
        "protocol": 17
      },
      "udp": {
        "destination-port": {

```

```

        "operator": "eq",
        "port": 123
      }
    },
    "actions": {
      "forwarding": "accept"
    }
  ]
},
{
  "name": "mud-59776-v6to",
  "type": "ipv6-acl-type",
  "aces": {
    "ace": [
      {
        "name": "ent0-todev",
        "matches": {
          "ietf-mud:mud": {
            "controller": "urn:ietf:params:mud:dns"
          },
          "ipv6": {
            "protocol": 17
          },
          "udp": {
            "source-port": {
              "operator": "eq",
              "port": 53
            }
          }
        },
        "actions": {
          "forwarding": "accept"
        }
      },
      {
        "name": "ent1-todev",
        "matches": {
          "ietf-mud:mud": {
            "controller": "urn:ietf:params:mud:ntp"
          },
          "ipv6": {
            "protocol": 17
          },
          "udp": {
            "source-port": {

```

```

        "operator": "eq",
        "port": 123
      }
    },
    "actions": {
      "forwarding": "accept"
    }
  ]
},
{
  "name": "mud-59776-v6fr",
  "type": "ipv6-acl-type",
  "aces": {
    "ace": [
      {
        "name": "ent0-frdev",
        "matches": {
          "ietf-mud:mud": {
            "controller": "urn:ietf:params:mud:dns"
          },
          "ipv6": {
            "protocol": 17
          },
          "udp": {
            "destination-port": {
              "operator": "eq",
              "port": 53
            }
          }
        },
        "actions": {
          "forwarding": "accept"
        }
      }
    ],
    "name": "ent1-frdev",
    "matches": {
      "ietf-mud:mud": {
        "controller": "urn:ietf:params:mud:ntp"
      },
      "ipv6": {
        "protocol": 17
      },
      "udp": {
        "destination-port": {

```

```

        "operator": "eq",
        "port": 123
      }
    },
    "actions": {
      "forwarding": "accept"
    }
  ]
}

```

Appendix C. A Sample Extension: DETNET-indicator

In this sample extension we augment the core MUD model to indicate whether the device implements DETNET. If a device claims not to use DETNET, but then later attempts to do so, a notification or exception might be generated. Note that this example is intended only for illustrative purposes.

Extension Name: "Example-Extension" (to be used in the extensions list)
 Standard: this document (but do not register the example)

This extension augments the MUD model to include a single node, using the following sample module that has the following tree structure:

```

module: ietf-mud-detext-example
  augment /ietf-mud:mud:
    +--rw is-detnet-required?  boolean

```

The model is defined as follows:

```

<CODE BEGINS>file "ietf-mud-detext-example@2018-06-15.yang"
module ietf-mud-detext-example {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-mud-detext-example";
  prefix ietf-mud-detext-example;

  import ietf-mud {
    prefix ietf-mud;
  }
}

```

```
organization
  "IETF OPSAWG (Ops Area) Working Group";
contact
  "WG Web: http://tools.ietf.org/wg/opsawg/
  WG List: opsawg@ietf.org
  Author: Eliot Lear
  lear@cisco.com
  Author: Ralph Droms
  rdroms@gmail.com
  Author: Dan Romascanu
  dromasca@gmail.com

  ";
description
  "Sample extension to a MUD module to indicate a need
  for DETNET support.";

revision 2018-06-15 {
  description
    "Initial revision.";
  reference
    "RFC XXXX: Manufacturer Usage Description
    Specification";
}

augment "/ietf-mud:mud" {
  description
    "This adds a simple extension for a manufacturer
    to indicate whether DETNET is required by a
    device.";
  leaf is-detnet-required {
    type boolean;
    description
      "This value will equal true if a device requires
      detnet to properly function";
  }
}
}
}
<CODE ENDS>
```

Using the previous example, we now show how the extension would be expressed:

```
{
  "ietf-mud:mud": {
    "mud-version": 1,
    "mud-url": "https://lighting.example.com/lightbulb2000",
    "last-update": "2018-03-02T11:20:51+01:00",
```



```
"cache-validity": 48,
"extensions": [
  "ietf-mud-detext-example"
],
"ietf-mud-detext-example:is-detnet-required": "false",
"is-supported": true,
"systeminfo": "The BMS Example Lightbulb",
"from-device-policy": {
  "access-lists": {
    "access-list": [
      {
        "name": "mud-76100-v6fr"
      }
    ]
  }
},
"to-device-policy": {
  "access-lists": {
    "access-list": [
      {
        "name": "mud-76100-v6to"
      }
    ]
  }
},
"ietf-access-control-list:acls": {
  "acl": [
    {
      "name": "mud-76100-v6to",
      "type": "ipv6-acl-type",
      "aces": {
        "ace": [
          {
            "name": "cl0-todev",
            "matches": {
              "ipv6": {
                "ietf-acldns:src-dnsname": "test.example.com",
                "protocol": 6
              },
              "tcp": {
                "ietf-mud:direction-initiated": "from-device",
                "source-port": {
                  "operator": "eq",
                  "port": 443
                }
              }
            }
          }
        ]
      }
    }
  ],
}
```

```

        "actions": {
            "forwarding": "accept"
        }
    ]
}
},
{
    "name": "mud-76100-v6fr",
    "type": "ipv6-acl-type",
    "aces": {
        "ace": [
            {
                "name": "cl0-frdev",
                "matches": {
                    "ipv6": {
                        "ietf-acldns:dst-dnsname": "test.example.com",
                        "protocol": 6
                    },
                    "tcp": {
                        "ietf-mud:direction-initiated": "from-device",
                        "destination-port": {
                            "operator": "eq",
                            "port": 443
                        }
                    }
                },
                "actions": {
                    "forwarding": "accept"
                }
            }
        ]
    }
}
]
}
}

```

Authors' Addresses

Eliot Lear
Cisco Systems
Richtistrasse 7
Wallisellen CH-8304
Switzerland

Phone: +41 44 878 9200
Email: lear@cisco.com

Ralph Droms
Google
355 Main St., 5th Floor
Cambridge

Phone: +1 978 376 3731
Email: rdroms@gmail.com

Dan Romascanu

Phone: +972 54 5555347
Email: dromasca@gmail.com

opsec
Internet-Draft
Intended status: Informational
Expires: January 4, 2018

F. Gont
UTN-FRH / SI6 Networks
W. Liu
Huawei Technologies
R. Bonica
Juniper Networks
July 3, 2017

Recommendations on the Filtering of IPv6 Packets Containing IPv6
Extension Headers
draft-ietf-opsec-ipv6-eh-filtering-03

Abstract

It is common operator practice to mitigate security risks by enforcing appropriate packet filtering. This document analyzes both the general security implications of IPv6 Extension Headers and the specific security implications of each Extension Header and Option type. Additionally, it discusses the operational and interoperability implications of discarding packets based on the IPv6 Extension Headers and IPv6 options they contain. Finally, it provides advice on the filtering of such IPv6 packets at transit routers, for those cases in which such filtering is deemed as necessary.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on January 4, 2018.

Copyright Notice

Copyright (c) 2017 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
2. Terminology and Conventions Used in This Document	4
2.1. Terminology	4
2.2. Conventions	4
3. IPv6 Extension Headers	5
3.1. General Discussion	5
3.2. General Security Implications	6
3.3. Summary of Advice on the Handling of IPv6 Packets with Specific IPv6 Extension Headers	6
3.4. Advice on the Handling of IPv6 Packets with Specific IPv6 Extension Headers	7
3.5. Advice on the Handling of Packets with Unknown IPv6 Extension Headers	16
4. IPv6 Options	17
4.1. General Discussion	17
4.2. General Security Implications of IPv6 Options	17
4.3. Advice on the Handling of Packets with Specific IPv6 Options	17
4.4. Advice on the handling of Packets with Unknown IPv6 Options	28
5. IANA Considerations	29
6. Security Considerations	29
7. Acknowledgements	29
8. References	29
8.1. Normative References	29
8.2. Informative References	33
Authors' Addresses	34

1. Introduction

Recent studies (see e.g. [RFC7872]) suggest that there is widespread dropping of IPv6 packets that contain IPv6 Extension Headers (EHs). In some cases, such packet drops occur at transit routers. While some operators "officially" drop packets that contain IPv6 EHs, it is possible that some of the measured packet drops be the result of

improper configuration defaults, or inappropriate advice in this area.

This document analyzes both the general security implications of IPv6 EHs and the specific security implications of each EH and Option type, and provides advice on the filtering of IPv6 packets based on the IPv6 EHs and the IPv6 options they contain. Since various protocols may use IPv6 EHs (possibly with IPv6 options), discarding packets based on the IPv6 EHs or IPv6 options they contain may have implications on the proper functioning of such protocols. Thus, this document also attempts to discuss the operational and interoperability implications of such filtering policies.

The filtering policy typically depends on where in the network such policy is enforced: when the policy is enforced in a transit network, the policy typically follows a "black-list" approach, where only packets with clear negative implications are dropped. On the other hand, when the policy is enforced closer to the destination systems, the policy typically follows a "white-list" approach, where only traffic that is expected to be received is allowed. The advice in this document is aimed only at transit routers that may need to enforce a filtering policy based on the EHs and IPv6 options a packet may contain, following a "black-list" approach, and hence is likely to be much more permissive than a filtering policy to be employed e.g. at the edge of an enterprise network. The advice in this document is meant to improve the current situation of the dropping of packets with IPv6 EHs in the Internet [RFC7872].

This document is similar in nature to [RFC7126], which addresses the same problem for the IPv4 case. However, in IPv6, the problem space is compounded by the fact that IPv6 specifies a number of IPv6 EHs, and a number of IPv6 options which may be valid only when included in specific EH types.

This document completes and complements the considerations for protecting the control plane from packets containing IP options that can be found in [RFC6192].

Section 2 of this document specifies the terminology and conventions employed throughout this document. Section 3 of this document discusses IPv6 EHs and provides advice in the area of filtering IPv6 packets that contain such IPv6 EHs. Section 4 of this document discusses IPv6 options and provides advice in the area of filtering IPv6 packets that contain such options.

2. Terminology and Conventions Used in This Document

2.1. Terminology

The terms "fast path", "slow path", and associated relative terms ("faster path" and "slower path") are loosely defined as in Section 2 of [RFC6398].

The terms "permit" (allow the traffic), "drop" (drop with no notification to sender), and "reject" (drop with appropriate notification to sender) are employed as defined in [RFC3871]. Throughout this document we also employ the term "discard" as a generic term to indicate the act of discarding a packet, irrespective of whether the sender is notified of such drops, and irrespective of whether the specific filtering action is logged.

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

2.2. Conventions

This document assumes that nodes comply with the requirements in [RFC7045]. Namely (from [RFC7045]),

- o If a forwarding node discards a packet containing a standard IPv6 EH, it MUST be the result of a configurable policy and not just the result of a failure to recognise such a header.
- o The discard policy for each standard type of EH MUST be individually configurable.
- o The default configuration SHOULD allow all standard IPv6 EHs.

The advice provided in this document is only meant to guide an operator in configuring forwarding devices, and is **not** to be interpreted as advice regarding default configuration settings for network devices. That is, this document provides advice with respect to operational configurations, but does not change the implementation defaults required by [RFC7045].

We recommend that configuration options are made available to govern the processing of each IPv6 EH type and each IPv6 option type. Such configuration options may include the following possible settings:

- o Permit this IPv6 EH or IPv6 Option type
- o Discard (and log) packets containing this IPv6 EH or option type

- o Reject (and log) packets containing this IPv6 EH or option type (where the packet drop is signaled with an ICMPv6 error message)
- o Rate-limit traffic containing this IPv6 EH or option type
- o Ignore this IPv6 EH or option type (as if it was not present) and forward the packet. We note that if a packet carries forwarding information (e.g., in an IPv6 Routing Header) this might be an inappropriate or undesirable action.

We note that special care needs to be taken when devices log packet drops/rejects. Devices should count the number of packets dropped/rejected, but the logging of drop/reject events should be limited so as to not overburden device resources.

Finally, we note that when discarding packets, it is generally desirable that the sender be signaled of the packet drop, since this is of use for trouble-shooting purposes. However, throughout this document (when recommending that packets be discarded) we generically refer to the action as "discard" without specifying whether the sender is signaled of the packet drop.

3. IPv6 Extension Headers

3.1. General Discussion

IPv6 [RFC2460] EHs allow for the extension of the IPv6 protocol. Since both IPv6 EHs and upper-layer protocols share the same namespace ("Next Header" registry/namespace), [RFC7045] identifies which of the currently assigned Internet Protocol numbers identify IPv6 EHs vs. upper-layer protocols. This document discusses the filtering of packets based on the IPv6 EHs (as specified by [RFC7045]) they contain.

NOTE: [RFC7112] specifies that non-fragmented IPv6 datagrams and IPv6 First-Fragments MUST contain the entire IPv6 header chain [RFC7112]. Therefore, intermediate systems can enforce the filtering policies discussed in this document, or resort to simply discarding the offending packets when they fail to comply with the requirements in [RFC7112]. We note that, in order to implement filtering rules on the fast path, it may be necessary for the filtering device to limit the depth into the packet that can be inspected before giving up. In circumstances where there is such a limitation, it is recommended that implementations discard packets if, when trying to determine whether to discard or permit a packet, the aforementioned limit is encountered.

3.2. General Security Implications

In some specific device architectures, IPv6 packets that contain IPv6 EHs may cause the corresponding packets to be processed on the slow path, and hence may be leveraged for the purpose of Denial of Service (DoS) attacks [I-D.gont-v6ops-ipv6-ehs-packet-drops] [Cisco-EH] [FW-Benchmark].

Operators are urged to consider IPv6 EH filtering and IPv6 options handling capabilities of different devices as they make deployment decisions in future.

3.3. Summary of Advice on the Handling of IPv6 Packets with Specific IPv6 Extension Headers

This section summarizes the advice provided in Section 3.4, providing references to the specific sections in which a detailed analysis can be found.

EH type	Filtering policy	Reference
IPv6 Hop-by-Hop Options (Proto=0)	Drop or Ignore	Section 3.4.1
Routing Header for IPv6 (Proto=43)	Drop only RTH0, Permit other RH Types	Section 3.4.2
Fragment Header for IPv6 (Proto=44)	Permit	Section 3.4.3
Encapsulating Security Payload (Proto=50)	Permit	Section 3.4.4
Authentication Header (Proto=51)	Permit	Section 3.4.5
Destination Options for IPv6 (Proto=60)	Permit	Section 3.4.6
Mobility Header (Proto=135)	Permit	Section 3.4.7
Host Identity Protocol (Proto=139)	Permit	Section 3.4.8
Shim6 Protocol (Proto=140)	Permit	Section 3.4.9
Use for experimentation and testing (Proto=253 and 254)	Drop	Section 3.4.10

Table 1: Summary of Advice on the Handling of IPv6 Packets with Specific IPv6 Extension Headers

3.4. Advice on the Handling of IPv6 Packets with Specific IPv6 Extension Headers

3.4.1. IPv6 Hop-by-Hop Options (Protocol Number=0)

3.4.1.1. Uses

The Hop-by-Hop Options header is used to carry optional information that should be examined by every node along a packet's delivery path.

3.4.1.2. Specification

This EH is specified in [RFC2460], and its processing rules have been updated by [RFC7045]. At the time of this writing, the following options have been specified for the Hop-by-Hop Options EH:

- o Type 0x00: Pad1 [RFC2460]
- o Type 0x01: PadN [RFC2460]
- o Type 0x05: Router Alert [RFC2711]
- o Type 0x07: CALIPSO [RFC5570]
- o Type 0x08: SMF_DPD [RFC6621]
- o Type 0x26: Quick-Start [RFC4782]
- o Type 0x4D: (Deprecated)
- o Type 0x63: RPL Option [RFC6553]
- o Type 0x6D: MPL Option [RFC7731]
- o Type 0x8A: Endpoint Identification (Deprecated) [draft-ietf-nimrod-eid]
- o Type 0xC2: Jumbo Payload [RFC2675]
- o Type 0xEE: IPv6 DFF Header [RFC6971]
- o Type 0x1E: RFC3692-style Experiment [RFC4727]
- o Type 0x3E: RFC3692-style Experiment [RFC4727]
- o Type 0x5E: RFC3692-style Experiment [RFC4727]
- o Type 0x7E: RFC3692-style Experiment [RFC4727]
- o Type 0x9E: RFC3692-style Experiment [RFC4727]
- o Type 0xBE: RFC3692-style Experiment [RFC4727]
- o Type 0xDE: RFC3692-style Experiment [RFC4727]
- o Type 0xFE: RFC3692-style Experiment [RFC4727]

3.4.1.3. Specific Security Implications

Since this EH is required to be processed by all intermediate-systems en route, it can be leveraged to perform Denial of Service attacks against the network infrastructure.

NOTE: Ongoing work essentially aims at requiring the Hop-by-Hop Option EH to be processed only in cases where the intermediate node is making use of any functionality provided by such header (see [I-D.ietf-6man-hbh-header-handling]). However, the deployed base is likely to reflect the traditional behavior for a while, and hence the potential security problems of this EH are still of concern.

3.4.1.4. Operational and Interoperability Impact if Blocked

Discarding packets containing a Hop-by-Hop Options EH would break any of the protocols that rely on it for proper functioning. For example, it would break RSVP [RFC2205] and multicast deployments, and would cause IPv6 jumbograms to be discarded.

3.4.1.5. Advice

The recommended configuration for the processing of these packets depends on the features and capabilities of the underlying platform. On platforms that allow forwarding of packets with HBH Options on the fast path, we recommend that packets with a HBH Options EH be forwarded as normal (for instance, [RFC7045] allows for implementations to ignore the HBH Options EH when forwarding packets). Otherwise, on platforms in which processing of packets with a IPv6 HBH Options EH is carried out in the slow path, and an option is provided to rate-limit these packets, we recommend that this option be selected. Finally, when packets containing a HBH Options EH are processed in the slow-path, and the underlying platform does not have any mitigation options available for attacks based on these packets, we recommend that such platforms discard packets containing IPv6 HBH Options EHs.

Finally, we note that, for obvious reasons, RPL (Routing Protocol for Low-Power and Lossy Networks) [RFC6550] routers must not discard packets based on the presence of an IPv6 Hop-by-Hop Options EH.

3.4.2. Routing Header for IPv6 (Protocol Number=43)

3.4.2.1. Uses

The Routing header is used by an IPv6 source to list one or more intermediate nodes to be "visited" on the way to a packet's destination.

3.4.2.2. Specification

This EH is specified in [RFC2460]. [RFC2460] originally specified the Routing Header Type 0, which has been later obsoleted by [RFC5095].

At the time of this writing, the following Routing Types have been specified:

- o Type 0: Source Route (DEPRECATED) [RFC2460] [RFC5095]
- o Type 1: Nimrod (DEPRECATED)
- o Type 2: Type 2 Routing Header [RFC6275]
- o Type 3: RPL Source Route Header [RFC6554]
- o Types 4-252: Unassigned
- o Type 253: RFC3692-style Experiment 1 [RFC4727]
- o Type 254: RFC3692-style Experiment 2 [RFC4727]
- o Type 255: Reserved

3.4.2.3. Specific Security Implications

The security implications of RHT0 have been discussed in detail in [Biondi2007] and [RFC5095].

3.4.2.4. Operational and Interoperability Impact if Blocked

Blocking packets containing a RHT0 or RHT1 has no operational implications. However, blocking packets employing other routing header types will break the protocols that rely on them.

3.4.2.5. Advice

Intermediate systems should discard packets containing a RHT0 or RHT1. RHT2 and RHT3 should be permitted, as required by [RFC7045]. Other routing header types should be discarded.

3.4.3. Fragment Header for IPv6 (Protocol Number=44)

3.4.3.1. Uses

This EH provides the fragmentation functionality for IPv6.

3.4.3.2. Specification

This EH is specified in [RFC2460].

3.4.3.3. Specific Security Implications

The security implications of the Fragment Header range from Denial of Service attacks (e.g. based on flooding a target with IPv6 fragments) to information leakage attacks [RFC7739].

3.4.3.4. Operational and Interoperability Impact if Blocked

Blocking packets that contain a Fragment Header will break any protocol that may rely on fragmentation (e.g., the DNS [RFC1034]).

3.4.3.5. Advice

Intermediate systems should permit packets that contain a Fragment Header.

3.4.4. Encapsulating Security Payload (Protocol Number=50)

3.4.4.1. Uses

This EH is employed for the IPsec suite [RFC4303].

3.4.4.2. Specification

This EH is specified in [RFC4303].

3.4.4.3. Specific Security Implications

Besides the general implications of IPv6 EHs, this EH could be employed to potentially perform a DoS attack at the destination system by wasting CPU resources in validating the contents of the packet.

3.4.4.4. Operational and Interoperability Impact if Blocked

Discarding packets that employ this EH would break IPsec deployments.

3.4.4.5. Advice

Intermediate systems should permit packets containing the Encapsulating Security Payload EH.

3.4.5. Authentication Header (Protocol Number=51)

3.4.5.1. Uses

The Authentication Header can be employed for provide authentication services in IPv4 and IPv6.

3.4.5.2. Specification

This EH is specified in [RFC4302].

3.4.5.3. Specific Security Implications

Besides the general implications of IPv6 EHs, this EH could be employed to potentially perform a DoS attack at the destination system by wasting CPU resources in validating the contents of the packet.

3.4.5.4. Operational and Interoperability Impact if Blocked

Discarding packets that employ this EH would break IPsec deployments.

3.4.5.5. Advice

Intermediate systems should permit packets containing an Authentication Header.

3.4.6. Destination Options for IPv6 (Protocol Number=60)

3.4.6.1. Uses

The Destination Options header is used to carry optional information that needs be examined only by a packet's destination node(s).

3.4.6.2. Specification

This EH is specified in [RFC2460]. At the time of this writing, the following options have been specified for this EH:

- o Type 0x00: Pad1 [RFC2460]
- o Type 0x01: PadN [RFC2460]

- o Type 0x04: Tunnel Encapsulation Limit [RFC2473]
- o Type 0x4D: (Deprecated)
- o Type 0xC9: Home Address [RFC6275]
- o Type 0x8A: Endpoint Identification (Deprecated) [draft-ietf-nimrod-eid]
- o Type 0x8B: ILNP Nonce [RFC6744]
- o Type 0x8C: Line-Identification Option [RFC6788]
- o Type 0x1E: RFC3692-style Experiment [RFC4727]
- o Type 0x3E: RFC3692-style Experiment [RFC4727]
- o Type 0x5E: RFC3692-style Experiment [RFC4727]
- o Type 0x7E: RFC3692-style Experiment [RFC4727]
- o Type 0x9E: RFC3692-style Experiment [RFC4727]
- o Type 0xBE: RFC3692-style Experiment [RFC4727]
- o Type 0xDE: RFC3692-style Experiment [RFC4727]
- o Type 0xFE: RFC3692-style Experiment [RFC4727]

3.4.6.3. Specific Security Implications

No security implications are known, other than the general implications of IPv6 EHs. For a discussion of possible security implications of specific options specified for the DO header, please see the Section 4.3.

3.4.6.4. Operational and Interoperability Impact if Blocked

Discarding packets that contain a Destination Options header would break protocols that rely on this EH type for conveying information, including protocols such as ILNP [RFC6740] and Mobile IPv6 [RFC6275], and IPv6 tunnels that employ the Tunnel Encapsulation Limit option.

3.4.6.5. Advice

Intermediate systems should permit packets that contain a Destination Options Header.

3.4.7. Mobility Header (Protocol Number=135)

3.4.7.1. Uses

The Mobility Header is an EH used by mobile nodes, correspondent nodes, and home agents in all messaging related to the creation and management of bindings in Mobile IPv6.

3.4.7.2. Specification

This EH is specified in [RFC6275].

3.4.7.3. Specific Security Implications

A thorough security assessment of the security implications of the Mobility Header and related mechanisms can be found in Section 15 of [RFC6275].

3.4.7.4. Operational and Interoperability Impact if Blocked

Discarding packets containing this EH would break Mobile IPv6.

3.4.7.5. Advice

Intermediate systems should permit packets containing this EH.

3.4.8. Host Identity Protocol (Protocol Number=139)

3.4.8.1. Uses

This EH is employed with the Host Identity Protocol (HIP), an experimental protocol that allows consenting hosts to securely establish and maintain shared IP-layer state, allowing separation of the identifier and locator roles of IP addresses, thereby enabling continuity of communications across IP address changes.

3.4.8.2. Specification

This EH is specified in [RFC5201].

3.4.8.3. Specific Security Implications

The security implications of the HIP header are discussed in detail in Section 8 of [RFC6275].

3.4.8.4. Operational and Interoperability Impact if Blocked

Discarding packets that contain the Host Identity Protocol would break HIP deployments.

3.4.8.5. Advice

Intermediate systems should permit packets that contain a Host Identity Protocol EH.

3.4.9. Shim6 Protocol (Protocol Number=140)

3.4.9.1. Uses

This EH is employed by the Shim6 [RFC5533] Protocol.

3.4.9.2. Specification

This EH is specified in [RFC5533].

3.4.9.3. Specific Security Implications

The specific security implications are discussed in detail in Section 16 of [RFC5533].

3.4.9.4. Operational and Interoperability Impact if Blocked

Discarding packets that contain this EH will break Shim6.

3.4.9.5. Advice

Intermediate systems should permit packets containing this EH.

3.4.10. Use for experimentation and testing (Protocol Numbers=253 and 254)

3.4.10.1. Uses

These IPv6 EHs are employed for performing RFC3692-Style experiments (see [RFC3692] for details).

3.4.10.2. Specification

These EHs are specified in [RFC3692] and [RFC4727].

3.4.10.3. Specific Security Implications

The security implications of these EHs will depend on their specific use.

3.4.10.4. Operational and Interoperability Impact if Blocked

For obvious reasons, discarding packets that contain these EHs limits the ability to perform legitimate experiments across IPv6 routers.

3.4.10.5. Advice

Intermediate systems should discard packets containing these EHs. Only in specific scenarios in which RFC3692-Style experiments are to be performed should these EHs be permitted.

3.5. Advice on the Handling of Packets with Unknown IPv6 Extension Headers

We refer to IPv6 EHs that have not been assigned an Internet Protocol Number by IANA (and marked as such) in [IANA-PROTOCOLS] as "unknown IPv6 extension headers" ("unknown IPv6 EHs").

3.5.1. Uses

New IPv6 EHs may be specified as part of future extensions to the IPv6 protocol.

Since IPv6 EHs and Upper-layer protocols employ the same namespace, it is impossible to tell whether an unknown "Internet Protocol Number" is being employed for an IPv6 EH or an Upper-Layer protocol.

3.5.2. Specification

The processing of unknown IPv6 EHs is specified in [RFC2460] and [RFC7045].

3.5.3. Specific Security Implications

For obvious reasons, it is impossible to determine specific security implications of unknown IPv6 EHs. However, from security standpoint, a device should discard IPv6 extension headers for which the security implications cannot be determined. We note that this policy is allowed by [RFC7045].

3.5.4. Operational and Interoperability Impact if Blocked

As noted in [RFC7045], discarding unknown IPv6 EHs may slow down the deployment of new IPv6 EHs and transport protocols. The corresponding IANA registry ([IANA-PROTOCOLS]) should be monitored such that filtering rules are updated as new IPv6 EHs are standardized.

We note that since IPv6 EHs and upper-layer protocols share the same numbering space, discarding unknown IPv6 EHs may result in packets encapsulating unknown upper-layer protocols being discarded.

3.5.5. Advice

Intermediate systems should discard packets containing unknown IPv6 EHs.

4. IPv6 Options

4.1. General Discussion

The following subsections describe specific security implications of different IPv6 options, and provide advice regarding filtering packets that contain such options.

4.2. General Security Implications of IPv6 Options

The general security implications of IPv6 options are closely related to those discussed in Section 3.2 for IPv6 EHs. Essentially, packets that contain IPv6 options might need to be processed by an IPv6 router's general-purpose CPU, and hence could present a DDoS risk to that router's general-purpose CPU (and thus to the router itself). For some architectures, a possible mitigation would be to rate-limit the packets that are to be processed by the general-purpose CPU (see e.g. [Cisco-EH]).

4.3. Advice on the Handling of Packets with Specific IPv6 Options

The following subsections contain a description of each of the IPv6 options that have so far been specified, a summary of the security implications of each of such options, a discussion of possible interoperability implications if packets containing such options are discarded, and specific advice regarding whether packets containing these options should be permitted.

4.3.1. Pad1 (Type=0x00)

4.3.1.1. Uses

This option is used when necessary to align subsequent options and to pad out the containing header to a multiple of 8 octets in length.

4.3.1.2. Specification

This option is specified in [RFC2460].

4.3.1.3. Specific Security Implications

None.

4.3.1.4. Operational and Interoperability Impact if Blocked

Discarding packets that contain this option would potentially break any protocol that relies on IPv6 EHs.

4.3.1.5. Advice

Intermediate systems should not discard packets based on the presence of this option.

4.3.2. PadN (Type=0x01)

4.3.2.1. Uses

This option is used when necessary to align subsequent options and to pad out the containing header to a multiple of 8 octets in length.

4.3.2.2. Specification

This option is specified in [RFC2460].

4.3.2.3. Specific Security Implications

Because of the possible size of this option, it could be leveraged as a large-bandwidth covert channel.

4.3.2.4. Operational and Interoperability Impact if Blocked

Discarding packets that contain this option would potentially break any protocol that relies on IPv6 EHs.

4.3.2.5. Advice

Intermediate systems should not discard IPv6 packets based on the presence of this option.

4.3.3. Jumbo Payload (Type=0XC2)

4.3.3.1. Uses

The Jumbo payload option provides the means of specifying payloads larger than 65535 bytes.

4.3.3.2. Specification

This option is specified in [RFC2675].

4.3.3.3. Specific Security Implications

There are no specific issues arising from this option, except for improper validity checks of the option and associated packet lengths.

4.3.3.4. Operational and Interoperability Impact if Blocked

Discarding packets based on the presence of this option will cause IPv6 jumbograms to be discarded.

4.3.3.5. Advice

Intermediate systems should discard packets that contain this option. An operator should permit this option only in specific scenarios in which support for IPv6 jumbograms is desired.

4.3.4. RPL Option (Type=0x63)

4.3.4.1. Uses

The RPL Option provides a mechanism to include routing information with each datagram that an RPL router forwards.

4.3.4.2. Specification

This option is specified in [RFC6553].

4.3.4.3. Specific Security Implications

Those described in [RFC6553].

4.3.4.4. Operational and Interoperability Impact if Blocked

This option is meant to be employed within an RPL instance. As a result, discarding packets based on the presence of this option (e.g. at an ISP) will not result in interoperability implications.

4.3.4.5. Advice

Non-RPL routers should discard packets that contain an RPL option.

4.3.5. Tunnel Encapsulation Limit (Type=0x04)

4.3.5.1. Uses

The Tunnel Encapsulation Limit option can be employed to specify how many further levels of nesting the packet is permitted to undergo.

4.3.5.2. Specification

This option is specified in [RFC2473].

4.3.5.3. Specific Security Implications

Those described in [RFC2473].

4.3.5.4. Operational and Interoperability Impact if Blocked

Discarding packets based on the presence of this option could result in tunnel traffic being discarded.

4.3.5.5. Advice

Intermediate systems should not discard packets based on the presence of this option.

4.3.6. Router Alert (Type=0x05)

4.3.6.1. Uses

The Router Alert option [RFC2711] is typically employed for the RSVP protocol [RFC2205] and the MLD protocol [RFC2710].

4.3.6.2. Specification

This option is specified in [RFC2711].

4.3.6.3. Specific Security Implications

Since this option causes the contents of the packet to be inspected by the handling device, this option could be leveraged for performing DoS attacks.

4.3.6.4. Operational and Interoperability Impact if Blocked

Discarding packets that contain this option would break RSVP and multicast deployments.

4.3.6.5. Advice

Intermediate systems should discard packets that contain this option. Only in specific environments where support for RSVP, multicast routing, or similar protocols is desired, should this option be permitted.

4.3.7. Quick-Start (Type=0x26)

4.3.7.1. Uses

This IP Option is used in the specification of Quick-Start for TCP and IP, which is an experimental mechanism that allows transport protocols, in cooperation with routers, to determine an allowed sending rate at the start and, at times, in the middle of a data transfer (e.g., after an idle period) [RFC4782].

4.3.7.2. Specification

This option is specified in [RFC4782], on the "Experimental" track.

4.3.7.3. Specific Security Implications

Section 9.6 of [RFC4782] notes that Quick-Start is vulnerable to two kinds of attacks:

- o attacks to increase the routers' processing and state load, and,
- o attacks with bogus Quick-Start Requests to temporarily tie up available Quick-Start bandwidth, preventing routers from approving Quick-Start Requests from other connections.

We note that if routers in a given environment do not implement and enable the Quick-Start mechanism, only the general security implications of IP options (discussed in Section 4.2) would apply.

4.3.7.4. Operational and Interoperability Impact if Blocked

The Quick-Start functionality would be disabled, and additional delays in TCP's connection establishment (for example) could be introduced. (Please see Section 4.7.2 of [RFC4782].) We note, however, that Quick-Start has been proposed as a mechanism that could be of use in controlled environments, and not as a mechanism that would be intended or appropriate for ubiquitous deployment in the global Internet [RFC4782].

4.3.7.5. Advice

Intermediate systems should not discard IPv6 packets based on the presence of this option.

4.3.8. CALIPSO (Type=0x07)

4.3.8.1. Uses

This option is used for encoding explicit packet Sensitivity Labels on IPv6 packets. It is intended for use only within Multi-Level Secure (MLS) networking environments that are both trusted and trustworthy.

4.3.8.2. Specification

This option is specified in [RFC5570].

4.3.8.3. Specific Security Implications

Presence of this option in a packet does not by itself create any specific new threat. Packets with this option ought not normally be seen on the global public Internet.

4.3.8.4. Operational and Interoperability Impact if Blocked

If packets with this option are discarded or if the option is stripped from the packet during transmission from source to destination, then the packet itself is likely to be discarded by the receiver because it is not properly labeled. In some cases, the receiver might receive the packet but associate an incorrect sensitivity label with the received data from the packet whose CALIPSO was stripped by an intermediate router or firewall. Associating an incorrect sensitivity label can cause the received information either to be handled as more sensitive than it really is ("upgrading") or as less sensitive than it really is ("downgrading"), either of which is problematic.

4.3.8.5. Advice

Intermediate systems that do not operate in Multi-Level Secure (MLS) networking environments should discard packets that contain this option.

4.3.9. SMF_DPD (Type=0x08)

4.3.9.1. Uses

This option is employed in the (experimental) Simplified Multicast Forwarding (SMF) for unique packet identification for IPv6 I-DPD, and as a mechanism to guarantee non-collision of hash values for different packets when H-DPD is used.

4.3.9.2. Specification

This option is specified in [RFC6621].

4.3.9.3. Specific Security Implications

None. The use of identifiers is subject to the security and privacy considerations discussed in [I-D.gont-predictable-numeric-ids].

4.3.9.4. Operational and Interoperability Impact if Blocked

Dropping packets containing this option within a MANET domain would break SMF. However, dropping such packets at the border of such domain would have no negative impact.

4.3.9.5. Advice

Intermediate system should discard packets that contain this option.

4.3.10. Home Address (Type=0xC9)

4.3.10.1. Uses

The Home Address option is used by a Mobile IPv6 node while away from home, to inform the recipient of the mobile node's home address.

4.3.10.2. Specification

This option is specified in [RFC6275].

4.3.10.3. Specific Security Implications

No (known) additional security implications than those described in [RFC6275].

4.3.10.4. Operational and Interoperability Impact if Blocked

Discarding IPv6 packets based on the presence of this option will break Mobile IPv6.

4.3.10.5. Advice

Intermediate systems should not discard IPv6 packets based on the presence of this option.

4.3.11. Endpoint Identification (Type=0x8A)

4.3.11.1. Uses

The Endpoint Identification option was meant to be used with the Nimrod routing architecture [NIMROD-DOC], but has never seen widespread deployment.

4.3.11.2. Specification

This option is specified in [NIMROD-DOC].

4.3.11.3. Specific Security Implications

Undetermined.

4.3.11.4. Operational and Interoperability Impact if Blocked

None.

4.3.11.5. Advice

Intermediate systems should discard packets that contain this option.

4.3.12. ILNP Nonce (Type=0x8B)

4.3.12.1. Uses

This option is employed by Identifier-Locator Network Protocol for IPv6 (ILNPv6) for providing protection against off-path attacks for packets when ILNPv6 is in use, and as a signal during initial network-layer session creation that ILNPv6 is proposed for use with this network-layer session, rather than classic IPv6.

4.3.12.2. Specification

This option is specified in [RFC6744].

4.3.12.3. Specific Security Implications

Those described in [RFC6744].

4.3.12.4. Operational and Interoperability Impact if Blocked

Discarding packets that contain this option will break INLPv6 deployments.

4.3.12.5. Advice

Intermediate systems should not discard packets based on the presence of this option.

4.3.13. Line-Identification Option (Type=0x8C)

4.3.13.1. Uses

This option is used by an Edge Router to identify the subscriber premises in scenarios where several subscriber premises may be logically connected to the same interface of an Edge Router.

4.3.13.2. Specification

This option is specified in [RFC6788].

4.3.13.3. Specific Security Implications

Those described in [RFC6788].

4.3.13.4. Operational and Interoperability Impact if Blocked

Since this option is meant to be employed in Router Solicitation messages, discarding packets based on the presence of this option at intermediate systems will result in no interoperability implications.

4.3.13.5. Advice

Intermediate devices should discard packets that contain this option.

4.3.14. Deprecated (Type=0x4D)

4.3.14.1. Uses

No information has been found about this option type.

4.3.14.2. Specification

No information has been found about this option type.

4.3.14.3. Specific Security Implications

No information has been found about this option type, and hence it has been impossible to perform the corresponding security assessment.

4.3.14.4. Operational and Interoperability Impact if Blocked

Unknown.

4.3.14.5. Advice

Intermediate systems should discard packets that contain this option.

4.3.15. MPL Option (Type=0x6D)

4.3.15.1. Uses

This option is used with the Multicast Protocol for Low power and Lossy Networks (MPL), that provides IPv6 multicast forwarding in constrained networks.

4.3.15.2. Specification

This option is specified in [RFC7731], and is meant to be included only in Hop-by-Hop Option headers.

4.3.15.3. Specific Security Implications

Those described in [RFC7731].

4.3.15.4. Operational and Interoperability Impact if Blocked

Dropping packets that contain an MPL option within an MPL network would break the Multicast Protocol for Low power and Lossy Networks (MPL). However, dropping such packets at the border of such networks will have no negative impact.

4.3.15.5. Advice

Intermediate systems should not discard packets based on the presence of this option. However, since this option has been specified for the Hop-by-Hop Options, such systems should consider the discussion in Section 3.4.1.

4.3.16. IP_DFF (Type=0xEE)

4.3.16.1. Uses

This option is employed with the (Experimental) Depth-First Forwarding (DFF) in Unreliable Networks.

4.3.16.2. Specification

This option is specified in [RFC6971].

4.3.16.3. Specific Security Implications

Those specified in [RFC6971].

4.3.16.4. Operational and Interoperability Impact if Blocked

Dropping packets containing this option within a routing domain that is running DFF would break DFF. However, dropping such packets at the border of such domains will have no security implications.

4.3.16.5. Advice

Intermediate systems that do not operate within a routing domain that is running DFF should discard packets containing this option.

4.3.17. RFC3692-style Experiment (Types = 0x1E, 0x3E, 0x5E, 0x7E, 0x9E, 0xBE, 0xDE, 0xFE)

4.3.17.1. Uses

These options can be employed for performing RFC3692-style experiments. It is only appropriate to use these values in explicitly configured experiments; they must not be shipped as defaults in implementations.

4.3.17.2. Specification

Specified in RFC 4727 [RFC4727] in the context of RFC3692-style experiments.

4.3.17.3. Specific Security Implications

The specific security implications will depend on the specific use of these options.

4.3.17.4. Operational and Interoperability Impact if Blocked

For obvious reasons, discarding packets that contain these options limits the ability to perform legitimate experiments across IPv6 routers.

4.3.17.5. Advice

Intermediate systems should discard packets that contain these options. Only in specific environments where RFC3692-style experiments are meant to be performed should these options be permitted.

4.4. Advice on the handling of Packets with Unknown IPv6 Options

We refer to IPv6 options that have not been assigned an IPv6 option type in the corresponding registry ([IANA-IPV6-PARAM]) as "unknown IPv6 options".

4.4.1. Uses

New IPv6 options may be specified as part of future protocol work.

4.4.2. Specification

The processing of unknown IPv6 options is specified in [RFC2460].

4.4.3. Specific Security Implications

For obvious reasons, it is impossible to determine specific security implications of unknown IPv6 options.

4.4.4. Operational and Interoperability Impact if Blocked

Discarding unknown IPv6 options may slow down the deployment of new IPv6 options. As noted in [draft-gont-6man-ipv6-opt-transmit], the corresponding IANA registry ([IANA-IPV6-PARAM]) should be monitored such that IPv6 option filtering rules are updated as new IPv6 options are standardized.

4.4.5. Advice

Enterprise intermediate systems that process the contents of IPv6 EHs should discard packets that contain unknown options. Other intermediate systems that process the contents of IPv6 EHs should permit packets that contain unknown options.

5. IANA Considerations

This document has no actions for IANA.

6. Security Considerations

This document provides advice on the filtering of IPv6 packets that contain IPv6 EHs (and possibly IPv6 options) at IPv6 transit routers. It is meant to improve the current situation of widespread dropping of such IPv6 packets in those cases where the drops result from improper configuration defaults, or inappropriate advice in this area.

7. Acknowledgements

The authors of this document would like to thank (in alphabetical order) Mikael Abrahamsson, Brian Carpenter, Mike Heard, Jen Linkova, Carlos Pignataro, Donald Smith, Gunter Van De Velde, and Erick Vyncke, for providing valuable comments on earlier versions of this document.

This document borrows some text and analysis from [RFC7126], authored by Fernando Gont, Randall Atkinson, and Carlos Pignataro.

8. References

8.1. Normative References

- [draft-gont-6man-ipv6-opt-transmit]
Gont, F., Liu, W., and R. Bonica, "Transmission and Processing of IPv6 Options", IETF Internet Draft, work in progress, August 2014.
- [RFC1034] Mockapetris, P., "Domain names - concepts and facilities", STD 13, RFC 1034, DOI 10.17487/RFC1034, November 1987, <<http://www.rfc-editor.org/info/rfc1034>>.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<http://www.rfc-editor.org/info/rfc2119>>.

- [RFC2205] Braden, R., Ed., Zhang, L., Berson, S., Herzog, S., and S. Jamin, "Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification", RFC 2205, DOI 10.17487/RFC2205, September 1997, <<http://www.rfc-editor.org/info/rfc2205>>.
- [RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, DOI 10.17487/RFC2460, December 1998, <<http://www.rfc-editor.org/info/rfc2460>>.
- [RFC2473] Conta, A. and S. Deering, "Generic Packet Tunneling in IPv6 Specification", RFC 2473, DOI 10.17487/RFC2473, December 1998, <<http://www.rfc-editor.org/info/rfc2473>>.
- [RFC2675] Borman, D., Deering, S., and R. Hinden, "IPv6 Jumbograms", RFC 2675, DOI 10.17487/RFC2675, August 1999, <<http://www.rfc-editor.org/info/rfc2675>>.
- [RFC2710] Deering, S., Fenner, W., and B. Haberman, "Multicast Listener Discovery (MLD) for IPv6", RFC 2710, DOI 10.17487/RFC2710, October 1999, <<http://www.rfc-editor.org/info/rfc2710>>.
- [RFC2711] Partridge, C. and A. Jackson, "IPv6 Router Alert Option", RFC 2711, DOI 10.17487/RFC2711, October 1999, <<http://www.rfc-editor.org/info/rfc2711>>.
- [RFC3692] Narten, T., "Assigning Experimental and Testing Numbers Considered Useful", BCP 82, RFC 3692, DOI 10.17487/RFC3692, January 2004, <<http://www.rfc-editor.org/info/rfc3692>>.
- [RFC4302] Kent, S., "IP Authentication Header", RFC 4302, DOI 10.17487/RFC4302, December 2005, <<http://www.rfc-editor.org/info/rfc4302>>.
- [RFC4303] Kent, S., "IP Encapsulating Security Payload (ESP)", RFC 4303, DOI 10.17487/RFC4303, December 2005, <<http://www.rfc-editor.org/info/rfc4303>>.
- [RFC4304] Kent, S., "Extended Sequence Number (ESN) Addendum to IPsec Domain of Interpretation (DOI) for Internet Security Association and Key Management Protocol (ISAKMP)", RFC 4304, DOI 10.17487/RFC4304, December 2005, <<http://www.rfc-editor.org/info/rfc4304>>.

- [RFC4727] Fenner, B., "Experimental Values In IPv4, IPv6, ICMPv4, ICMPv6, UDP, and TCP Headers", RFC 4727, DOI 10.17487/RFC4727, November 2006, <<http://www.rfc-editor.org/info/rfc4727>>.
- [RFC4782] Floyd, S., Allman, M., Jain, A., and P. Sarolahti, "Quick-Start for TCP and IP", RFC 4782, DOI 10.17487/RFC4782, January 2007, <<http://www.rfc-editor.org/info/rfc4782>>.
- [RFC5095] Abley, J., Savola, P., and G. Neville-Neil, "Deprecation of Type 0 Routing Headers in IPv6", RFC 5095, DOI 10.17487/RFC5095, December 2007, <<http://www.rfc-editor.org/info/rfc5095>>.
- [RFC5201] Moskowitz, R., Nikander, P., Jokela, P., Ed., and T. Henderson, "Host Identity Protocol", RFC 5201, DOI 10.17487/RFC5201, April 2008, <<http://www.rfc-editor.org/info/rfc5201>>.
- [RFC5533] Nordmark, E. and M. Bagnulo, "Shim6: Level 3 Multihoming Shim Protocol for IPv6", RFC 5533, DOI 10.17487/RFC5533, June 2009, <<http://www.rfc-editor.org/info/rfc5533>>.
- [RFC5570] StJohns, M., Atkinson, R., and G. Thomas, "Common Architecture Label IPv6 Security Option (CALIPSO)", RFC 5570, DOI 10.17487/RFC5570, July 2009, <<http://www.rfc-editor.org/info/rfc5570>>.
- [RFC6275] Perkins, C., Ed., Johnson, D., and J. Arkko, "Mobility Support in IPv6", RFC 6275, DOI 10.17487/RFC6275, July 2011, <<http://www.rfc-editor.org/info/rfc6275>>.
- [RFC6398] Le Faucheur, F., Ed., "IP Router Alert Considerations and Usage", BCP 168, RFC 6398, DOI 10.17487/RFC6398, October 2011, <<http://www.rfc-editor.org/info/rfc6398>>.
- [RFC6550] Winter, T., Ed., Thubert, P., Ed., Brandt, A., Hui, J., Kelsey, R., Levis, P., Pister, K., Struik, R., Vasseur, JP., and R. Alexander, "RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks", RFC 6550, DOI 10.17487/RFC6550, March 2012, <<http://www.rfc-editor.org/info/rfc6550>>.
- [RFC6553] Hui, J. and JP. Vasseur, "The Routing Protocol for Low-Power and Lossy Networks (RPL) Option for Carrying RPL Information in Data-Plane Datagrams", RFC 6553, DOI 10.17487/RFC6553, March 2012, <<http://www.rfc-editor.org/info/rfc6553>>.

- [RFC6554] Hui, J., Vasseur, JP., Culler, D., and V. Manral, "An IPv6 Routing Header for Source Routes with the Routing Protocol for Low-Power and Lossy Networks (RPL)", RFC 6554, DOI 10.17487/RFC6554, March 2012, <<http://www.rfc-editor.org/info/rfc6554>>.
- [RFC6621] Macker, J., Ed., "Simplified Multicast Forwarding", RFC 6621, DOI 10.17487/RFC6621, May 2012, <<http://www.rfc-editor.org/info/rfc6621>>.
- [RFC6740] Atkinson, RJ. and SN. Bhatti, "Identifier-Locator Network Protocol (ILNP) Architectural Description", RFC 6740, DOI 10.17487/RFC6740, November 2012, <<http://www.rfc-editor.org/info/rfc6740>>.
- [RFC6744] Atkinson, RJ. and SN. Bhatti, "IPv6 Nonce Destination Option for the Identifier-Locator Network Protocol for IPv6 (ILNPv6)", RFC 6744, DOI 10.17487/RFC6744, November 2012, <<http://www.rfc-editor.org/info/rfc6744>>.
- [RFC6788] Krishnan, S., Kavanagh, A., Varga, B., Ooghe, S., and E. Nordmark, "The Line-Identification Option", RFC 6788, DOI 10.17487/RFC6788, November 2012, <<http://www.rfc-editor.org/info/rfc6788>>.
- [RFC6971] Herberg, U., Ed., Cardenas, A., Iwao, T., Dow, M., and S. Cespedes, "Depth-First Forwarding (DFF) in Unreliable Networks", RFC 6971, DOI 10.17487/RFC6971, June 2013, <<http://www.rfc-editor.org/info/rfc6971>>.
- [RFC7045] Carpenter, B. and S. Jiang, "Transmission and Processing of IPv6 Extension Headers", RFC 7045, DOI 10.17487/RFC7045, December 2013, <<http://www.rfc-editor.org/info/rfc7045>>.
- [RFC7112] Gont, F., Manral, V., and R. Bonica, "Implications of Oversized IPv6 Header Chains", RFC 7112, DOI 10.17487/RFC7112, January 2014, <<http://www.rfc-editor.org/info/rfc7112>>.
- [RFC7731] Hui, J. and R. Kelsey, "Multicast Protocol for Low-Power and Lossy Networks (MPL)", RFC 7731, DOI 10.17487/RFC7731, February 2016, <<http://www.rfc-editor.org/info/rfc7731>>.

8.2. Informative References

[Biondi2007]

Biondi, P. and A. Ebalard, "IPv6 Routing Header Security", CanSecWest 2007 Security Conference, 2007, <http://www.secdev.org/conf/IPv6_RH_security-csw07.pdf>.

[Cisco-EH]

Cisco Systems, "IPv6 Extension Headers Review and Considerations", Whitepaper. October 2006, <http://www.cisco.com/en/US/technologies/tk648/tk872/technologies_white_paper0900aecd8054d37d.pdf>.

[draft-ietf-nimrod-eid]

Lynn, C., "Endpoint Identifier Destination Option", IETF Internet Draft, draft-ietf-nimrod-eid-00.txt, November 1995.

[FW-Benchmark]

Zack, E., "Firewall Security Assessment and Benchmarking IPv6 Firewall Load Tests", IPv6 Hackers Meeting #1, Berlin, Germany. June 30, 2013, <<http://www.ipv6hackers.org/meetings/ipv6-hackers-1/zack-ipv6hackers1-firewall-security-assessment-and-benchmarking.pdf>>.

[I-D.gont-predictable-numeric-ids]

Gont, F. and I. Arce, "Security and Privacy Implications of Numeric Identifiers Employed in Network Protocols", draft-gont-predictable-numeric-ids-01 (work in progress), July 2017.

[I-D.gont-v6ops-ipv6-ehs-packet-drops]

Gont, F., Hilliard, N., Doering, G., (Will), S., and W. Kumari, "Operational Implications of IPv6 Packets with Extension Headers", draft-gont-v6ops-ipv6-ehs-packet-drops-03 (work in progress), March 2016.

[I-D.ietf-6man-hbh-header-handling]

Baker, F. and R. Bonica, "IPv6 Hop-by-Hop Options Extension Header", draft-ietf-6man-hbh-header-handling-03 (work in progress), March 2016.

[IANA-IPV6-PARAM]

Internet Assigned Numbers Authority, "Internet Protocol Version 6 (IPv6) Parameters", December 2013, <<http://www.iana.org/assignments/ipv6-parameters/ipv6-parameters.xhtml>>.

[IANA-PROTOCOLS]

Internet Assigned Numbers Authority, "Protocol Numbers", 2014, <<http://www.iana.org/assignments/protocol-numbers/protocol-numbers.xhtml>>.

[NIMROD-DOC]

Nimrod Documentation Page,
<<http://ana-3.lcs.mit.edu/~jnc/nimrod/>>.

[RFC3871] Jones, G., Ed., "Operational Security Requirements for Large Internet Service Provider (ISP) IP Network Infrastructure", RFC 3871, DOI 10.17487/RFC3871, September 2004, <<http://www.rfc-editor.org/info/rfc3871>>.

[RFC6192] Dugal, D., Pignataro, C., and R. Dunn, "Protecting the Router Control Plane", RFC 6192, DOI 10.17487/RFC6192, March 2011, <<http://www.rfc-editor.org/info/rfc6192>>.

[RFC7126] Gont, F., Atkinson, R., and C. Pignataro, "Recommendations on Filtering of IPv4 Packets Containing IPv4 Options", BCP 186, RFC 7126, DOI 10.17487/RFC7126, February 2014, <<http://www.rfc-editor.org/info/rfc7126>>.

[RFC7739] Gont, F., "Security Implications of Predictable Fragment Identification Values", RFC 7739, DOI 10.17487/RFC7739, February 2016, <<http://www.rfc-editor.org/info/rfc7739>>.

[RFC7872] Gont, F., Linkova, J., Chown, T., and W. Liu, "Observations on the Dropping of Packets with IPv6 Extension Headers in the Real World", RFC 7872, DOI 10.17487/RFC7872, June 2016, <<http://www.rfc-editor.org/info/rfc7872>>.

Authors' Addresses

Fernando Gont
UTN-FRH / SI6 Networks
Evaristo Carriego 2644
Haedo, Provincia de Buenos Aires 1706
Argentina

Phone: +54 11 4650 8472
Email: fgont@si6networks.com
URI: <http://www.si6networks.com>

Will(Shucheng) Liu
Huawei Technologies
Bantian, Longgang District
Shenzhen 518129
P.R. China

Email: liushucheng@huawei.com

Ronald P. Bonica
Juniper Networks
2251 Corporate Park Drive
Herndon, VA 20171
US

Phone: 571 250 5819
Email: rbonica@juniper.net

opsec
Internet-Draft
Intended status: Informational
Expires: 4 November 2022

F. Gont
EdgeUno
W. Liu
Huawei Technologies
3 May 2022

Recommendations on the Filtering of IPv6 Packets Containing IPv6
Extension Headers at Transit Routers
draft-ietf-opsec-ipv6-eh-filtering-10

Abstract

This document analyzes the security implications of IPv6 Extension Headers and associated IPv6 options. Additionally, it discusses the operational and interoperability implications of discarding packets based on the IPv6 Extension Headers and IPv6 options they contain. Finally, it provides advice on the filtering of such IPv6 packets at transit routers for traffic not directed to them, for those cases where such filtering is deemed as necessary.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 4 November 2022.

Copyright Notice

Copyright (c) 2022 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components

extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1. Introduction	2
2. Terminology and Assumptions Employed in This Document	4
2.1. Terminology	4
2.2. Applicability Statement	4
2.3. Router Default Behavior and Features	4
3. IPv6 Extension Headers	5
3.1. General Discussion	5
3.2. General Security Implications	6
3.3. Rationale for Our Advice on the Handling of IPv6 Packets with Specific IPv6 Extension Headers	6
3.4. Summary of Advice on the Handling of IPv6 Packets with Specific IPv6 Extension Headers	6
3.5. Advice on the Handling of IPv6 Packets with Specific IPv6 Extension Headers	7
3.6. Advice on the Handling of Packets with Unknown IPv6 Extension Headers	16
4. IPv6 Options	17
4.1. General Discussion	17
4.2. General Security Implications of IPv6 Options	17
4.3. Summary of Advice on the Handling of IPv6 Packets with Specific IPv6 Extension Headers	18
4.4. Advice on the Handling of Packets with Specific IPv6 Options	19
4.5. Advice on the handling of Packets with Unknown IPv6 Options	32
5. IANA Considerations	32
6. Privacy Considerations	32
7. Security Considerations	32
8. Acknowledgements	33
9. References	33
9.1. Normative References	33
9.2. Informative References	37
Authors' Addresses	39

1. Introduction

IPv6 Extension Headers (EHs) allow for the extension of the IPv6 protocol, and provide support for core functionality such as IPv6 fragmentation. However, common implementation limitations suggest that EHs present a challenge for IPv6 packet routing equipment, particularly when the IPv6 header chain needs to be processed for e.g. enforcing ACLs or implementing other functions [RFC9098].

Several studies (e.g. [Huston-2022], [I-D.vyncke-v6ops-james], and [RFC7872]) suggest that there is widespread dropping of IPv6 packets that contain IPv6 Extension Headers (EHs). In some cases, such packet drops occur at transit routers. While some operators are known to intentionally drop packets that contain IPv6 EHs, it is possible that some of the measured packet drops are the result of inappropriate advice in this area.

This document analyzes both the general security implications of IPv6 EHs, as well as the security implications of specific EH and Option types. It also provides advice on the filtering of IPv6 packets based on the IPv6 EHs and the IPv6 options they contain. Since various protocols may use IPv6 EHs (possibly with IPv6 options), discarding packets based on the IPv6 EHs or IPv6 options they contain can have implications on the proper functioning of such protocols. Thus, this document also attempts to discuss the operational and interoperability implications of such filtering policies.

The resulting packet filtering policy typically depends on where in the network such policy is enforced: when the policy is enforced in a transit network, the policy typically follows a "deny-list" approach, where only packets with clear negative implications are dropped. On the other hand, when the policy is enforced closer to the destination systems, the policy typically follows an "accept-list" approach, where only traffic that is expected to be received is allowed. The advice in this document is aimed only at transit routers that may need to enforce a filtering policy based on the EHs and IPv6 options a packet may contain, following a "deny-list" approach, and hence is likely to be much more permissive than a filtering policy to be employed at e.g. the edge of an enterprise network. The advice in this document is meant to improve the current situation of the dropping of packets with IPv6 EHs in the Internet [RFC7872] in such cases where packets are being dropped due to inappropriate or missing guidelines.

This document is similar in nature to [RFC7126], which addresses the same problem for the IPv4 case. However, in IPv6, the problem space is compounded by the fact that IPv6 specifies a number of IPv6 EHs, and a number of IPv6 options which may be valid only when included in specific EH types.

This document completes and complements the considerations for protecting the control plane from packets containing IP options that can be found in [RFC6192].

Section 2 specifies the terminology and conventions employed throughout this document. Section 3 discusses IPv6 EHs and provides advice in the area of filtering IPv6 packets that contain such IPv6 EHs. Section 4 discusses IPv6 options and provides advice in the area of filtering IPv6 packets that contain such options.

2. Terminology and Assumptions Employed in This Document

2.1. Terminology

The terms "permit" (allow the traffic), "drop" (drop with no notification to sender), and "reject" (drop with appropriate notification to sender) are employed as defined in [RFC3871]. Throughout this document we also employ the term "discard" as a generic term to indicate the act of discarding a packet, irrespective of whether the sender is notified of such drops, and irrespective of whether the specific filtering action is logged.

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

2.2. Applicability Statement

This document provides advice on the filtering of IPv6 packets with EHs at transit routers for traffic not explicitly destined to them, for cases in which such filtering is deemed as necessary.

2.3. Router Default Behavior and Features

This document assumes that nodes comply with the requirements in [RFC7045]. Namely,

"If a forwarding node discards a packet containing a standard IPv6 extension header, it MUST be the result of a configurable policy and not just the result of a failure to recognise such a header. This means that the discard policy for each standard type of extension header MUST be individually configurable. The default configuration SHOULD allow all standard extension headers."

The advice provided in this document is only meant to guide an operator in configuring forwarding devices, and is not to be interpreted as advice regarding default configuration settings for network devices. That is, this document provides advice with respect to operational policies, but does not change the implementation defaults required by [RFC7045].

We recommend that configuration options are made available to govern the processing of each IPv6 EH type and each IPv6 option type. Such configuration options should include the following possible settings:

- * Permit this IPv6 EH or IPv6 Option type.
- * Drop packets containing this IPv6 EH or option type.
- * Reject packets containing this IPv6 EH or option type (where the packet drop is signaled with an ICMPv6 error message).
- * Rate-limit traffic containing this IPv6 EH or option type.
- * Ignore this IPv6 EH or option type (as if it was not present) and process the packet according the rules for the remaining headers. We note that if a packet carries forwarding information (e.g., in an IPv6 Routing Header) this might be an inappropriate or undesirable action.

We note that special care needs to be taken when devices log packet drops/rejects. Devices should count the number of packets dropped/rejected, but the logging of drop/reject events should be limited so as to not overburden device resources.

Finally, we note that when discarding packets, it is generally desirable that the sender be signaled of the packet drop, since this is of use for trouble-shooting purposes. However, throughout this document (when recommending that packets be discarded) we generically refer to the action as "discard" without specifying whether the sender is signaled of the packet drop.

3. IPv6 Extension Headers

3.1. General Discussion

IPv6 [RFC8200] EHs allow for the extension of the IPv6 protocol. Since both IPv6 EHs and upper-layer protocols share the same namespace ("Next Header" registry/namespace), [RFC7045] identifies which of the currently assigned Internet Protocol numbers identify IPv6 EHs vs. upper-layer protocols. This document discusses the filtering of packets based on the IPv6 EHs (as specified by [RFC7045]) they contain.

NOTE: [RFC8200] specifies that non-fragmented IPv6 datagrams and IPv6 First-Fragments must contain the entire IPv6 header chain [RFC7112]. Therefore, intermediate systems can enforce the filtering policies discussed in this document, or resort to simply discarding the offending packets when they fail to comply with the

requirements in [RFC8200]. We note that, in order to implement filtering rules on the fast path, it may be necessary for the filtering device to limit the depth into the packet that can be inspected before giving up. In circumstances where such a limitation exists, it is recommended that implementations provide a configuration option that specifies whether to discard packets if the aforementioned limit is encountered. Operators may then determine according to their own circumstances how such packets will be handled.

3.2. General Security Implications

In some device architectures, IPv6 packets that contain IPv6 EHs can cause the corresponding packets to be processed on the slow path, and hence may be leveraged for the purpose of Denial of Service (DoS) attacks [RFC9098] [Cisco-EH] [FW-Benchmark].

Operators are urged to consider the IPv6 EH and IPv6 options handling capabilities of their devices as they make deployment decisions in the future.

3.3. Rationale for Our Advice on the Handling of IPv6 Packets with Specific IPv6 Extension Headers

- * IPv6 Packets with IPv6 Extension Headers (or options) that are not expected to traverse transit routers should be dropped.
- * IPv6 Packets with IPv6 Extension Headers (or options) that are only expected to traverse transit routers when a specific technology is employed, should be permitted (or dropped) based on the knowledge regarding the use of such technology in transit provider in question (i.e. permit the packets if the technology is employed, or drop them)
- * IPv6 Packets with IPv6 Extension Headers (or options) that represent a concrete attack vector to network infrastructure devices should be dropped.
- * IPv6 packets with any other IPv6 Extension headers (or options) should be permitted. This is an intentional trade-off made to minimize ossification.

3.4. Summary of Advice on the Handling of IPv6 Packets with Specific IPv6 Extension Headers

This section summarizes the advice provided in Section 3.5, providing references to the specific sections in which a detailed analysis can be found.

EH type	Filtering policy	Reference
IPv6 Hop-by-Hop Options (Proto=0)	Drop or Ignore	Section 3.5.1
Routing Header for IPv6 (Proto=43)	Drop only RHT0 and RHT1. Permit other RH Types	Section 3.5.2
Fragment Header for IPv6 (Proto=44)	Permit	Section 3.5.3
Encapsulating Security Payload (Proto=50)	Permit	Section 3.5.4
Authentication Header (Proto=51)	Permit	Section 3.5.5
Destination Options for IPv6 (Proto=60)	Permit	Section 3.5.6
Mobility Header (Proto=135)	Permit	Section 3.5.7
Host Identity Protocol (Proto=139)	Permit	Section 3.5.8
Shim6 Protocol (Proto=140)	Permit	Section 3.5.9
Use for experimentation and testing (Proto=253 and 254)	Drop	Section 3.5.10

Table 1: Summary of Advice on the Handling of IPv6 Packets with Specific IPv6 Extension Headers

3.5. Advice on the Handling of IPv6 Packets with Specific IPv6 Extension Headers

3.5.1. IPv6 Hop-by-Hop Options (Protocol Number=0)

3.5.1.1. Uses

The Hop-by-Hop Options header is used to carry optional information that may be examined by every node along a packet's delivery path. It is expected that nodes will examine the Hop-by-Hop Options header if explicitly configured to do so.

NOTE: A previous revision of the IPv6 core specification, [RFC2460], originally required that all nodes examined and processed the Hop-by-Hop Options header. However, even before the publication of [RFC8200] a number of implementations already provided the option of ignoring this header unless explicitly configured to examine it.

3.5.1.2. Specification

This EH is specified in [RFC8200]. As of May 2022, the following options have been specified for the Hop-by-Hop Options EH:

- * Type 0x00: Pad1 [RFC8200]
- * Type 0x01: PadN [RFC8200]
- * Type 0x05: Router Alert [RFC2711]
- * Type 0x07: CALIPSO [RFC5570]
- * Type 0x08: SMF_DPD [RFC6621]
- * Type 0x23: RPL Option [RFC9008]
- * Type 0x26: Quick-Start [RFC4782]
- * Type 0x4D: (Deprecated)
- * Type 0x63: RPL Option [RFC6553]
- * Type 0x6D: MPL Option [RFC7731]
- * Type 0x8A: Endpoint Identification (Deprecated)
[draft-ietf-nimrod-eid]
- * Type 0xC2: Jumbo Payload [RFC2675]
- * Type 0xEE: IPv6 DFF Header [RFC6971]
- * Type 0x1E: RFC3692-style Experiment [RFC4727]
- * Type 0x3E: RFC3692-style Experiment [RFC4727]

- * Type 0x5E: RFC3692-style Experiment [RFC4727]
- * Type 0x7E: RFC3692-style Experiment [RFC4727]
- * Type 0x9E: RFC3692-style Experiment [RFC4727]
- * Type 0xBE: RFC3692-style Experiment [RFC4727]
- * Type 0xDE: RFC3692-style Experiment [RFC4727]
- * Type 0xFE: RFC3692-style Experiment [RFC4727]

3.5.1.3. Specific Security Implications

Legacy nodes that process this extension header might be subject to Denial of Service attacks.

NOTE: While [RFC8200] has removed this requirement, the deployed base may still reflect the classical behavior for a while, and hence the potential security problems of this EH are still of concern.

3.5.1.4. Operational and Interoperability Impact if Blocked

Discarding packets containing a Hop-by-Hop Options EH would break any of the protocols that rely on it for proper functioning. For example, it would break RSVP [RFC2205] and multicast deployments, and would cause IPv6 jumbograms to be discarded.

3.5.1.5. Advice

Nodes implementing [RFC8200] would already ignore this extension header unless explicitly required to process it. For legacy ([RFC2460]) nodes, the recommended configuration for the processing of these packets depends on the features and capabilities of the underlying platform, the configuration of the platform, and also the deployment environment of the platform. On platforms that allow forwarding of packets with HBH Options on the fast path, we recommend that packets with a HBH Options EH be forwarded as normal. Otherwise, on platforms in which processing of packets with a IPv6 HBH Options EH is carried out in the slow path, and an option is provided to rate-limit these packets, we recommend that this option be selected. Finally, when packets containing a HBH Options EH are processed in the slow-path, and the underlying platform does not have any mitigation options available for attacks based on these packets, we recommend that such platforms discard packets containing IPv6 HBH Options EHs.

Finally, we note that RPL (Routing Protocol for Low-Power and Lossy Networks) routers [RFC6550] must not discard packets based on the presence of an IPv6 Hop-by-Hop Options EH, as this would break RPL.

3.5.2. Routing Header for IPv6 (Protocol Number=43)

3.5.2.1. Uses

The Routing header is used by an IPv6 source to list one or more intermediate nodes to be "visited" on the way to a packet's destination.

3.5.2.2. Specification

This EH is specified in [RFC8200]. [RFC2460] had originally specified the Routing Header Type 0, which was later obsoleted by [RFC5095], and thus removed from [RFC8200].

At of May 2022, the following Routing Types have been specified:

- * Type 0: Source Route (DEPRECATED) [RFC2460] [RFC5095]
- * Type 1: Nimrod (DEPRECATED)
- * Type 2: Type 2 Routing Header [RFC6275]
- * Type 3: RPL Source Route Header [RFC6554]
- * Type 4: Segment Routing Header (SRH) [RFC8754]
- * Types 5-252: Unassigned
- * Type 253: RFC3692-style Experiment 1 [RFC4727]
- * Type 254: RFC3692-style Experiment 2 [RFC4727]
- * Type 255: Reserved

3.5.2.3. Specific Security Implications

The security implications of RHT0 have been discussed in detail in [Biondi2007] and [RFC5095]. RHT1 was never widely implemented. The security implications of RHT2, RHT3, and RHT4 (SRH) are discussed in [RFC6275], [RFC6554], and [RFC8754], respectively.

3.5.2.4. Operational and Interoperability Impact if Blocked

Blocking packets containing a RHT0 or RHT1 has no operational implications, since both have been deprecated. Blocking packets with a RHT2 would break Mobile IPv6. Packets with a RHT3 may be safely blocked at RPL domain boundaries, since RHT3 headers are employed within a single RPL domain. Blocking packets with a RHT4 (SRH) will break Segment Routing (SR) deployments, if the filtering policy is enforced on packets being forwarded within an SR domain.

3.5.2.5. Advice

Intermediate systems should discard packets containing a RHT0, RHT1, or RHT3. Other routing header types should be permitted, as required by [RFC7045].

3.5.3. Fragment Header for IPv6 (Protocol Number=44)

3.5.3.1. Uses

This EH provides the fragmentation functionality for IPv6.

3.5.3.2. Specification

This EH is specified in [RFC8200].

3.5.3.3. Specific Security Implications

The security implications of the Fragment Header range from Denial of Service attacks (e.g. based on flooding a target with IPv6 fragments) to information leakage attacks [RFC7739].

3.5.3.4. Operational and Interoperability Impact if Blocked

Blocking packets that contain a Fragment Header will break any protocol that may rely on fragmentation (e.g., the DNS [RFC1034]). However, IP fragmentation is known to introduce fragility to Internet communication [RFC8900].

3.5.3.5. Advice

Intermediate systems should permit packets that contain a Fragment Header.

3.5.4. Encapsulating Security Payload (Protocol Number=50)

3.5.4.1. Uses

This EH is employed for the IPsec suite [RFC4303].

3.5.4.2. Specification

This EH is specified in [RFC4303].

3.5.4.3. Specific Security Implications

Besides the general implications of IPv6 EHs, this EH could be employed to potentially perform a DoS attack at the destination system by wasting CPU resources in validating the contents of the packet.

3.5.4.4. Operational and Interoperability Impact if Blocked

Discarding packets that employ this EH would break IPsec deployments.

3.5.4.5. Advice

Intermediate systems should permit packets containing the Encapsulating Security Payload EH.

3.5.5. Authentication Header (Protocol Number=51)

3.5.5.1. Uses

The Authentication Header can be employed to provide authentication services in IPv4 and IPv6.

3.5.5.2. Specification

This EH is specified in [RFC4302].

3.5.5.3. Specific Security Implications

Besides the general implications of IPv6 EHs, this EH could be employed to potentially perform a DoS attack at the destination system by wasting CPU resources in validating the contents of the packet.

3.5.5.4. Operational and Interoperability Impact if Blocked

Discarding packets that employ this EH would break IPsec deployments.

3.5.5.5. Advice

Intermediate systems should permit packets containing an Authentication Header.

3.5.6. Destination Options for IPv6 (Protocol Number=60)

3.5.6.1. Uses

The Destination Options header is used to carry optional information that needs be examined only by a packet's destination node(s).

3.5.6.2. Specification

This EH is specified in [RFC8200]. As of May 2022, the following options have been specified for this EH:

- * Type 0x00: Pad1 [RFC8200]
- * Type 0x01: PadN [RFC8200]
- * Type 0x04: Tunnel Encapsulation Limit [RFC2473]
- * Type 0x0F: IPv6 Performance and Diagnostic Metrics (PDM) [RFC8250]
- * Type 0x4D: (Deprecated)
- * Type 0xC9: Home Address [RFC6275]
- * Type 0x8A: Endpoint Identification (Deprecated)
[draft-ietf-nimrod-eid]
- * Type 0x8B: ILNP Nonce [RFC6744]
- * Type 0x8C: Line-Identification Option [RFC6788]
- * Type 0x1E: RFC3692-style Experiment [RFC4727]
- * Type 0x3E: RFC3692-style Experiment [RFC4727]
- * Type 0x5E: RFC3692-style Experiment [RFC4727]
- * Type 0x7E: RFC3692-style Experiment [RFC4727]
- * Type 0x9E: RFC3692-style Experiment [RFC4727]
- * Type 0xBE: RFC3692-style Experiment [RFC4727]

- * Type 0xDE: RFC3692-style Experiment [RFC4727]

- * Type 0xFE: RFC3692-style Experiment [RFC4727]

3.5.6.3. Specific Security Implications

No security implications are known, other than the general implications of IPv6 EHs. For a discussion of possible security implications of specific options specified for the DO header, please see the Section 4.4.

3.5.6.4. Operational and Interoperability Impact if Blocked

Discarding packets that contain a Destination Options header would break protocols that rely on this EH type for conveying information, including protocols such as ILNP [RFC6740] and Mobile IPv6 [RFC6275], and IPv6 tunnels that employ the Tunnel Encapsulation Limit option.

3.5.6.5. Advice

Intermediate systems should permit packets that contain a Destination Options Header.

3.5.7. Mobility Header (Protocol Number=135)

3.5.7.1. Uses

The Mobility Header is an EH used by mobile nodes, correspondent nodes, and home agents in all messaging related to the creation and management of bindings in Mobile IPv6.

3.5.7.2. Specification

This EH is specified in [RFC6275].

3.5.7.3. Specific Security Implications

A thorough security assessment of the security implications of the Mobility Header and related mechanisms can be found in Section 15 of [RFC6275].

3.5.7.4. Operational and Interoperability Impact if Blocked

Discarding packets containing this EH would break Mobile IPv6.

3.5.7.5. Advice

Intermediate systems should permit packets containing this EH.

3.5.8. Host Identity Protocol (Protocol Number=139)

3.5.8.1. Uses

This EH is employed with the Host Identity Protocol (HIP), a protocol that allows consenting hosts to securely establish and maintain shared IP-layer state, allowing separation of the identifier and locator roles of IP addresses, thereby enabling continuity of communications across IP address changes.

3.5.8.2. Specification

This EH is specified in [RFC7401].

3.5.8.3. Specific Security Implications

The security implications of the HIP header are discussed in detail in Section 8 of [RFC6275].

3.5.8.4. Operational and Interoperability Impact if Blocked

Discarding packets that contain the Host Identity Protocol would break HIP deployments.

3.5.8.5. Advice

Intermediate systems should permit packets that contain a Host Identity Protocol EH.

3.5.9. Shim6 Protocol (Protocol Number=140)

3.5.9.1. Uses

This EH is employed by the Shim6 [RFC5533] Protocol.

3.5.9.2. Specification

This EH is specified in [RFC5533].

3.5.9.3. Specific Security Implications

The specific security implications are discussed in detail in Section 16 of [RFC5533].

3.5.9.4. Operational and Interoperability Impact if Blocked

Discarding packets that contain this EH will break Shim6.

3.5.9.5. Advice

Intermediate systems should permit packets containing this EH.

3.5.10. Use for experimentation and testing (Protocol Numbers=253 and 254)

3.5.10.1. Uses

These IPv6 EHs are employed for performing RFC3692-Style experiments (see [RFC3692] for details).

3.5.10.2. Specification

These EHs are specified in [RFC3692] and [RFC4727].

3.5.10.3. Specific Security Implications

The security implications of these EHs will depend on their specific use.

3.5.10.4. Operational and Interoperability Impact if Blocked

For obvious reasons, discarding packets that contain these EHs limits the ability to perform legitimate experiments across IPv6 routers.

3.5.10.5. Advice

Operators should determine according to their own circumstances whether to discard packets containing these EHs.

3.6. Advice on the Handling of Packets with Unknown IPv6 Extension Headers

We refer to IPv6 EHs that have not been assigned an Internet Protocol Number by IANA (and marked as such) in [IANA-PROTOCOLS] as "unknown IPv6 extension headers" ("unknown IPv6 EHs").

3.6.1. Uses

New IPv6 EHs may be specified as part of future extensions to the IPv6 protocol.

Since IPv6 EHs and Upper-layer protocols employ the same namespace, it is impossible to tell whether an unknown "Internet Protocol Number" is being employed for an IPv6 EH or an Upper-Layer protocol.

3.6.2. Specification

The processing of unknown IPv6 EHs is specified in [RFC7045].

3.6.3. Specific Security Implications

For obvious reasons, it is impossible to determine specific security implications of unknown IPv6 EHs.

3.6.4. Operational and Interoperability Impact if Blocked

As noted in [RFC7045], discarding unknown IPv6 EHs may slow down the deployment of new IPv6 EHs and transport protocols. The corresponding IANA registry ([IANA-PROTOCOLS]) should be monitored such that filtering rules are updated as new IPv6 EHs are standardized.

We note that since IPv6 EHs and upper-layer protocols share the same numbering space, discarding unknown IPv6 EHs may result in packets encapsulating unknown upper-layer protocols being discarded.

3.6.5. Advice

Operators should determine according to their own circumstances whether to discard packets containing unknown IPv6 EHs.

4. IPv6 Options

4.1. General Discussion

The following subsections describe specific security implications of different IPv6 options, and provide advice regarding filtering packets that contain such options.

4.2. General Security Implications of IPv6 Options

The general security implications of IPv6 options are closely related to those discussed in Section 3.2 for IPv6 EHs. Essentially, packets that contain IPv6 options might need to be processed by an IPv6 router's general-purpose CPU, and hence could present a DDoS risk to that router's general-purpose CPU (and thus to the router itself). For some architectures, a possible mitigation would be to rate-limit the packets that are to be processed by the general-purpose CPU (see e.g. [Cisco-EH]).

4.3. Summary of Advice on the Handling of IPv6 Packets with Specific IPv6 Extension Headers

This section summarizes the advice provided in Section 3.5, providing references to the specific sections in which a detailed analysis can be found.

Option	Filtering policy	Reference
Pad1 (Type=0x00)	Permit	Section 4.4.1
PadN (Type=0x01)	Permit	Section 4.4.2
Tunnel Encapsulation Limit (Type=0x04)	Permit	Section 4.4.3
Router Alert (Type=0x05)	Permit based on needed functionality	Section 4.4.4
CALIPSO (Type=0x07)	Permit based on needed functionality	Section 4.4.5
SMF_DPD (Type=0x08)	Permit based on needed functionality	Section 4.4.6
PDM Option (Type=0x0F)	Permit	Section 4.4.7
RPL Option (Type=0x23)	Permit	Section 4.4.8
Quick-Start (Type=0x26)	Permit	Section 4.4.9
Deprecated (Type=0x4D)	Drop	Section 4.4.10
MPL Option (Type=0x6D)	Permit	Section 4.4.12
Jumbo Payload (Type=0xC2)	Permit based on needed functionality	Section 4.4.16
RPL Option (Type=0x63)	Drop in non-RPL routers	Section 4.4.11

Endpoint Identification (Type=0x8A)	Drop	Section 4.4.13
ILNP Nonce (Type=0x8B)	Permit	Section 4.4.14
Line-Identification Option (Type=0x8C)	Drop	Section 4.4.15
Home Address (Type=0xC9)	Permit	Section 4.4.17
IP_DFF (Type=0xEE)	Permit based on needed functionality	Section 4.4.18
RFC3692-style Experiment (Types = 0x1E, 0x3E, 0x5E, 0x7E, 0x9E, 0xBE, 0xDE, 0xFE)	Permit based on needed functionality	Section 4.4.19

Table 2: Summary of Advice on the Handling of IPv6 Packets with Specific IPv6 options

4.4. Advice on the Handling of Packets with Specific IPv6 Options

The following subsections contain a description of each of the IPv6 options that have so far been specified, a summary of the security implications of each of such options, a discussion of possible interoperability implications if packets containing such options are discarded, and specific advice regarding whether packets containing these options should be permitted.

4.4.1. Pad1 (Type=0x00)

4.4.1.1. Uses

This option is used when necessary to align subsequent options and to pad out the containing header to a multiple of 8 octets in length.

4.4.1.2. Specification

This option is specified in [RFC8200].

4.4.1.3. Specific Security Implications

None.

4.4.1.4. Operational and Interoperability Impact if Blocked

Discarding packets that contain this option would potentially break any protocol that relies on IPv6 options.

4.4.1.5. Advice

Intermediate systems should not discard packets based on the presence of this option.

4.4.2. PadN (Type=0x01)

4.4.2.1. Uses

This option is used when necessary to align subsequent options and to pad out the containing header to a multiple of 8 octets in length.

4.4.2.2. Specification

This option is specified in [RFC8200].

4.4.2.3. Specific Security Implications

Because of the possible size of this option, it could be leveraged as a large-bandwidth covert channel.

4.4.2.4. Operational and Interoperability Impact if Blocked

Discarding packets that contain this option would potentially break any protocol that relies on IPv6 options.

4.4.2.5. Advice

Intermediate systems should not discard IPv6 packets based on the presence of this option.

4.4.3. Tunnel Encapsulation Limit (Type=0x04)

4.4.3.1. Uses

The Tunnel Encapsulation Limit option can be employed to specify how many further levels of nesting the packet is permitted to undergo.

4.4.3.2. Specification

This option is specified in [RFC2473].

4.4.3.3. Specific Security Implications

Those described in [RFC2473].

4.4.3.4. Operational and Interoperability Impact if Blocked

Discarding packets based on the presence of this option could result in tunnel traffic being discarded.

4.4.3.5. Advice

Intermediate systems should not discard packets based on the presence of this option.

4.4.4. Router Alert (Type=0x05)

4.4.4.1. Uses

The Router Alert option [RFC2711] is employed by a number of protocols, including the Resource reSerVation Protocol (RSVP) [RFC2205], Multicast Listener Discovery (MLD) [RFC2710] [RFC3810], Multicast Router Discovery (MRD) [RFC4286], and General Internet Signaling Transport (GIST) [RFC5971]. Its usage is discussed in detail in [RFC6398].

4.4.4.2. Specification

This option is specified in [RFC2711].

4.4.4.3. Specific Security Implications

Since this option causes the contents of the packet to be inspected by the handling device, this option could be leveraged for performing DoS attacks. The security implications of the Router Alert option are discussed in detail in [RFC6398].

4.4.4.4. Operational and Interoperability Impact if Blocked

Discarding packets that contain this option would break any protocols that rely on them, such as RSVP and multicast deployments. Please see Section 4.4.4.3 for further details.

4.4.4.5. Advice

Packets containing this option should be permitted in environments where support for RSVP, multicast routing, or similar protocols is desired.

4.4.5. CALIPSO (Type=0x07)

4.4.5.1. Uses

This option is used for encoding explicit packet Sensitivity Labels on IPv6 packets. It is intended for use only within Multi-Level Secure (MLS) networking environments that are both trusted and trustworthy.

4.4.5.2. Specification

This option is specified in [RFC5570].

4.4.5.3. Specific Security Implications

Presence of this option in a packet does not by itself create any specific new threat. Packets with this option ought not normally be seen on the global public Internet.

4.4.5.4. Operational and Interoperability Impact if Blocked

If packets with this option are discarded or if the option is stripped from the packet during transmission from source to destination, then the packet itself is likely to be discarded by the receiver because it is not properly labeled. In some cases, the receiver might receive the packet but associate an incorrect sensitivity label with the received data from the packet whose CALIPSO was stripped by a middle-box (such as a packet-scrubber). Associating an incorrect sensitivity label can cause the received information either to be handled as more sensitive than it really is ("upgrading") or as less sensitive than it really is ("downgrading"), either of which is problematic. As noted in [RFC5570], IPsec [RFC4301] [RFC4302] [RFC4303] can be employed to protect the CALIPSO option.

4.4.5.5. Advice

Recommendations for handling the CALIPSO option depend on the deployment environment, rather than whether an intermediate system happens to be deployed as a transit device (e.g., IPv6 transit router).

Explicit configuration is the only method via which an intermediate system can know whether that particular intermediate system has been deployed within a Multi-Level Secure (MLS) environment. In many cases, ordinary commercial intermediate systems (e.g., IPv6 routers and firewalls) are the majority of the deployed intermediate systems inside an MLS network environment.

For Intermediate systems that DO NOT implement [RFC5570], there should be a configuration option to EITHER (a) drop packets containing the CALIPSO option OR (b) to ignore the presence of the CALIPSO option and forward the packets normally. In non-MLS environments, such intermediate systems should have this configuration option set to (a) above. In MLS environments, such intermediate systems should have this option set to (b) above. The default setting for this configuration option should be set to (a) above, because MLS environments are much less common than non-MLS environments.

For Intermediate systems that DO implement [RFC5570], there should be configuration options (a) and (b) from the preceding paragraph and also a third configuration option (c) to process packets containing a CALIPSO option as per [RFC5570]. When deployed in non-MLS environments, such intermediate systems should have this configuration option set to (a) above. When deployed in MLS environments, such intermediate systems should have this set to (c). The default setting for this configuration option MAY be set to (a) above, because MLS environments are much less common than non-MLS environments.

4.4.6. SMF_DPD (Type=0x08)

4.4.6.1. Uses

This option is employed in the (experimental) Simplified Multicast Forwarding (SMF) for unique packet identification for IPv6 I-DPD, and as a mechanism to guarantee non-collision of hash values for different packets when H-DPD is used.

4.4.6.2. Specification

This option is specified in [RFC6621].

4.4.6.3. Specific Security Implications

None. The use of transient numeric identifiers is subject to the security and privacy considerations discussed in [I-D.irtf-pearg-numeric-ids-generation].

4.4.6.4. Operational and Interoperability Impact if Blocked

Dropping packets containing this option within a MANET domain would break SMF. However, dropping such packets at the border of such domain would have no negative impact.

4.4.6.5. Advice

Intermediate systems that are not within a MANET domain should discard packets that contain this option.

4.4.7. PDM (Type=0x0F)

4.4.7.1. Uses

This option is employed to convey sequence numbers and timing information in IPv6 packets as a basis for measurements.

4.4.7.2. Specification

This option is specified in [RFC8250].

4.4.7.3. Specific Security Implications

Those specified in [RFC8250]. Additionally, since the options employs transient numeric identifiers, implementations may be subject to the issues discussed in [I-D.irtf-pearg-numeric-ids-generation].

4.4.7.4. Operational and Interoperability Impact if Blocked

Dropping packets containing this option will result in negative interoperability implications for traffic employing this option as a basis for measurements.

4.4.7.5. Advice

Intermediate systems should not discard packets based on the presence of this option.

4.4.8. RPL Option (Type=0x23)

4.4.8.1. Uses

The RPL Option provides a mechanism to include routing information with each datagram that an RPL router forwards.

4.4.8.2. Specification

This option is specified in [RFC9008].

4.4.8.3. Specific Security Implications

Those described in [RFC9008].

4.4.8.4. Operational and Interoperability Impact if Blocked

This option can survive outside of an RPL instance. As a result, discarding packets based on the presence of this option would break some use cases for RPL (see [RFC9008]).

4.4.8.5. Advice

Intermediate systems should not discard IPv6 packets based on the presence of this option.

4.4.9. Quick-Start (Type=0x26)

4.4.9.1. Uses

This IP Option is used in the specification of Quick-Start for TCP and IP, which is an experimental mechanism that allows transport protocols, in cooperation with routers, to determine an allowed sending rate at the start and, at times, in the middle of a data transfer (e.g., after an idle period) [RFC4782].

4.4.9.2. Specification

This option is specified in [RFC4782], on the "Experimental" track.

4.4.9.3. Specific Security Implications

Section 9.6 of [RFC4782] notes that Quick-Start is vulnerable to two kinds of attacks:

- * attacks to increase the routers' processing and state load, and,
- * attacks with bogus Quick-Start Requests to temporarily tie up available Quick-Start bandwidth, preventing routers from approving Quick-Start Requests from other connections.

We note that if routers in a given environment do not implement and enable the Quick-Start mechanism, only the general security implications of IP options (discussed in Section 4.2) would apply.

4.4.9.4. Operational and Interoperability Impact if Blocked

The Quick-Start functionality would be disabled, and additional delays in TCP's connection establishment (for example) could be introduced. (Please see Section 4.7.2 of [RFC4782].) We note, however, that Quick-Start has been proposed as a mechanism that could be of use in controlled environments, and not as a mechanism that would be intended or appropriate for ubiquitous deployment in the

global Internet [RFC4782].

4.4.9.5. Advice

Intermediate systems should not discard IPv6 packets based on the presence of this option.

4.4.10. Deprecated (Type=0x4D)

4.4.10.1. Uses

No information has been found about this option type.

4.4.10.2. Specification

No information has been found about this option type.

4.4.10.3. Specific Security Implications

No information has been found about this option type, and hence it has been impossible to perform the corresponding security assessment.

4.4.10.4. Operational and Interoperability Impact if Blocked

Unknown.

4.4.10.5. Advice

Intermediate systems should discard packets that contain this option.

4.4.11. RPL Option (Type=0x63)

4.4.11.1. Uses

The RPL Option provides a mechanism to include routing information with each datagram that an RPL router forwards.

4.4.11.2. Specification

This option was originally specified in [RFC6553]. It has been deprecated by [RFC9008].

4.4.11.3. Specific Security Implications

Those described in [RFC9008].

4.4.11.4. Operational and Interoperability Impact if Blocked

This option is meant to be employed within an RPL instance. As a result, discarding packets based on the presence of this option outside of an RPL instance will not result in interoperability implications.

4.4.11.5. Advice

Non-RPL routers should discard packets that contain an RPL option.

4.4.12. MPL Option (Type=0x6D)

4.4.12.1. Uses

This option is used with the Multicast Protocol for Low power and Lossy Networks (MPL), that provides IPv6 multicast forwarding in constrained networks.

4.4.12.2. Specification

This option is specified in [RFC7731], and is meant to be included only in Hop-by-Hop Option headers.

4.4.12.3. Specific Security Implications

Those described in [RFC7731].

4.4.12.4. Operational and Interoperability Impact if Blocked

Dropping packets that contain an MPL option within an MPL network would break the Multicast Protocol for Low power and Lossy Networks (MPL). However, dropping such packets at the border of such networks will have no negative impact.

4.4.12.5. Advice

Intermediate systems should not discard packets based on the presence of this option. However, since this option has been specified for the Hop-by-Hop Options, such systems should consider the discussion in Section 3.5.1.

4.4.13. Endpoint Identification (Type=0x8A)

4.4.13.1. Uses

The Endpoint Identification option was meant to be used with the Nimrod routing architecture [NIMROD-DOC], but has never seen widespread deployment.

4.4.13.2. Specification

This option is specified in [NIMROD-DOC].

4.4.13.3. Specific Security Implications

Undetermined.

4.4.13.4. Operational and Interoperability Impact if Blocked

None.

4.4.13.5. Advice

Intermediate systems should discard packets that contain this option.

4.4.14. ILNP Nonce (Type=0x8B)

4.4.14.1. Uses

This option is employed by Identifier-Locator Network Protocol for IPv6 (ILNPv6) for providing protection against off-path attacks for packets when ILNPv6 is in use, and as a signal during initial network-layer session creation that ILNPv6 is proposed for use with this network-layer session, rather than classic IPv6.

4.4.14.2. Specification

This option is specified in [RFC6744].

4.4.14.3. Specific Security Implications

Those described in [RFC6744].

4.4.14.4. Operational and Interoperability Impact if Blocked

Discarding packets that contain this option will break ILNPv6 deployments.

4.4.14.5. Advice

Intermediate systems should not discard packets based on the presence of this option.

4.4.15. Line-Identification Option (Type=0x8C)

4.4.15.1. Uses

This option is used by an Edge Router to identify the subscriber premises in scenarios where several subscriber premises may be logically connected to the same interface of an Edge Router.

4.4.15.2. Specification

This option is specified in [RFC6788].

4.4.15.3. Specific Security Implications

Those described in [RFC6788].

4.4.15.4. Operational and Interoperability Impact if Blocked

Since this option is meant to be employed in Router Solicitation messages, discarding packets based on the presence of this option at intermediate systems will result in no interoperability implications.

4.4.15.5. Advice

Intermediate devices should discard packets that contain this option.

4.4.16. Jumbo Payload (Type=0XC2)

4.4.16.1. Uses

The Jumbo payload option provides the means of specifying payloads larger than 65535 bytes.

4.4.16.2. Specification

This option is specified in [RFC2675].

4.4.16.3. Specific Security Implications

There are no specific issues arising from this option, except for improper validity checks of the option and associated packet lengths.

4.4.16.4. Operational and Interoperability Impact if Blocked

Discarding packets based on the presence of this option will cause IPv6 jumbograms to be discarded.

4.4.16.5. Advice

An operator should permit this option only in specific scenarios in which support for IPv6 jumbograms is desired.

4.4.17. Home Address (Type=0xC9)

4.4.17.1. Uses

The Home Address option is used by a Mobile IPv6 node while away from home, to inform the recipient of the mobile node's home address.

4.4.17.2. Specification

This option is specified in [RFC6275].

4.4.17.3. Specific Security Implications

No (known) additional security implications than those described in [RFC6275].

4.4.17.4. Operational and Interoperability Impact if Blocked

Discarding IPv6 packets based on the presence of this option will break Mobile IPv6.

4.4.17.5. Advice

Intermediate systems should not discard IPv6 packets based on the presence of this option.

4.4.18. IP_DFF (Type=0xEE)

4.4.18.1. Uses

This option is employed with the (Experimental) Depth-First Forwarding (DFF) in Unreliable Networks.

4.4.18.2. Specification

This option is specified in [RFC6971].

4.4.18.3. Specific Security Implications

Those specified in [RFC6971].

4.4.18.4. Operational and Interoperability Impact if Blocked

Dropping packets containing this option within a routing domain that is running DFF would break DFF. However, dropping such packets at the border of such domains will have no security implications.

4.4.18.5. Advice

Intermediate systems that do not operate within a routing domain that is running DFF should discard packets containing this option.

4.4.19. RFC3692-style Experiment (Types = 0x1E, 0x3E, 0x5E, 0x7E, 0x9E, 0xBE, 0xDE, 0xFE)

4.4.19.1. Uses

These options can be employed for performing RFC3692-style experiments. It is only appropriate to use these values in explicitly configured experiments; they must not be shipped as defaults in implementations.

4.4.19.2. Specification

Specified in RFC 4727 [RFC4727] in the context of RFC3692-style experiments.

4.4.19.3. Specific Security Implications

The specific security implications will depend on the specific use of these options.

4.4.19.4. Operational and Interoperability Impact if Blocked

For obvious reasons, discarding packets that contain these options limits the ability to perform legitimate experiments across IPv6 routers.

4.4.19.5. Advice

Operators should determine according to their own circumstances whether to discard packets containing these IPv6 options.

4.5. Advice on the handling of Packets with Unknown IPv6 Options

We refer to IPv6 options that have not been assigned an IPv6 option type in the corresponding registry ([IANA-IPV6-PARAM]) as "unknown IPv6 options".

4.5.1. Uses

New IPv6 options may be specified as part of future protocol work.

4.5.2. Specification

The processing of unknown IPv6 options is specified in [RFC8200].

4.5.3. Specific Security Implications

For obvious reasons, it is impossible to determine specific security implications of unknown IPv6 options.

4.5.4. Operational and Interoperability Impact if Blocked

Discarding unknown IPv6 options may slow down the deployment of new IPv6 options. As noted in [draft-gont-6man-ipv6-opt-transmit], the corresponding IANA registry ([IANA-IPV6-PARAM]) should be monitored such that IPv6 option filtering rules are updated as new IPv6 options are standardized.

4.5.5. Advice

Operators should determine according to their own circumstances whether to discard packets containing unknown IPv6 options.

5. IANA Considerations

This document has no actions for IANA.

6. Privacy Considerations

There are no privacy considerations associated with this document.

7. Security Considerations

This document provides advice on the filtering of IPv6 packets that contain IPv6 EHs (and possibly IPv6 options) at IPv6 transit routers. It is meant to improve the current situation of widespread dropping of such IPv6 packets in those cases where the drops result from improper configuration defaults, or inappropriate advice in this area.

As discussed in Section 3.3 of this document, one of the underlying principles for the advice provided in this document is that IPv6 packets with specific EHs or options which may represent an attack vector for infrastructure devices should be dropped. While this policy helps mitigate some specific attack vectors, the recommendations in this document will not help to mitigate vulnerabilities based on implementation errors [RFC9098].

We also note that depending on the router architecture, attempts to filter packets based on the presence of IPv6 EHs or options might itself represent an attack vector to network infrastructure devices [RFC9098].

8. Acknowledgements

The authors would like to thank Ron Bonica for his work on earlier versions of this document.

The authors of this document would like to thank (in alphabetical order) Mikael Abrahamsson, Brian Carpenter, Tim Chown, Roman Danyliw, Darren Dukes, Lars Eggert, David Farmer, Mike Heard, Bob Hinden, Christian Huitema, Benjamin Kaduk, Erik Kline, Murray Kucherawy, Jen Linkova, Carlos Pignataro, Alvaro Retana, Maria Ines Robles, Zaheduzzaman Sarker, Donald Smith, Pascal Thubert, Ole Troan, Gunter Van De Velde, Eric Vyncke, and Robert Wilton, for providing valuable comments on earlier versions of this document.

This document borrows some text and analysis from [RFC7126], authored by Fernando Gont, Randall Atkinson, and Carlos Pignataro.

The authors would like to thank Warren Kumari and Eric Vyncke for their guidance during the publication process of this document.

Fernando would also like to thank Brian Carpenter and Ran Atkinson who, over the years, have answered many questions and provided valuable comments that have benefited his protocol-related work (including the present document).

9. References

9.1. Normative References

- [RFC1034] Mockapetris, P., "Domain names - concepts and facilities", STD 13, RFC 1034, DOI 10.17487/RFC1034, November 1987, <<https://www.rfc-editor.org/info/rfc1034>>.

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC2205] Braden, R., Ed., Zhang, L., Berson, S., Herzog, S., and S. Jamin, "Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification", RFC 2205, DOI 10.17487/RFC2205, September 1997, <<https://www.rfc-editor.org/info/rfc2205>>.
- [RFC2473] Conta, A. and S. Deering, "Generic Packet Tunneling in IPv6 Specification", RFC 2473, DOI 10.17487/RFC2473, December 1998, <<https://www.rfc-editor.org/info/rfc2473>>.
- [RFC2675] Borman, D., Deering, S., and R. Hinden, "IPv6 Jumbograms", RFC 2675, DOI 10.17487/RFC2675, August 1999, <<https://www.rfc-editor.org/info/rfc2675>>.
- [RFC2710] Deering, S., Fenner, W., and B. Haberman, "Multicast Listener Discovery (MLD) for IPv6", RFC 2710, DOI 10.17487/RFC2710, October 1999, <<https://www.rfc-editor.org/info/rfc2710>>.
- [RFC2711] Partridge, C. and A. Jackson, "IPv6 Router Alert Option", RFC 2711, DOI 10.17487/RFC2711, October 1999, <<https://www.rfc-editor.org/info/rfc2711>>.
- [RFC3692] Narten, T., "Assigning Experimental and Testing Numbers Considered Useful", BCP 82, RFC 3692, DOI 10.17487/RFC3692, January 2004, <<https://www.rfc-editor.org/info/rfc3692>>.
- [RFC3810] Vida, R., Ed. and L. Costa, Ed., "Multicast Listener Discovery Version 2 (MLDv2) for IPv6", RFC 3810, DOI 10.17487/RFC3810, June 2004, <<https://www.rfc-editor.org/info/rfc3810>>.
- [RFC4286] Haberman, B. and J. Martin, "Multicast Router Discovery", RFC 4286, DOI 10.17487/RFC4286, December 2005, <<https://www.rfc-editor.org/info/rfc4286>>.
- [RFC4301] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, DOI 10.17487/RFC4301, December 2005, <<https://www.rfc-editor.org/info/rfc4301>>.
- [RFC4302] Kent, S., "IP Authentication Header", RFC 4302, DOI 10.17487/RFC4302, December 2005, <<https://www.rfc-editor.org/info/rfc4302>>.

- [RFC4303] Kent, S., "IP Encapsulating Security Payload (ESP)", RFC 4303, DOI 10.17487/RFC4303, December 2005, <<https://www.rfc-editor.org/info/rfc4303>>.
- [RFC4727] Fenner, B., "Experimental Values In IPv4, IPv6, ICMPv4, ICMPv6, UDP, and TCP Headers", RFC 4727, DOI 10.17487/RFC4727, November 2006, <<https://www.rfc-editor.org/info/rfc4727>>.
- [RFC4782] Floyd, S., Allman, M., Jain, A., and P. Sarolahti, "Quick-Start for TCP and IP", RFC 4782, DOI 10.17487/RFC4782, January 2007, <<https://www.rfc-editor.org/info/rfc4782>>.
- [RFC5095] Abley, J., Savola, P., and G. Neville-Neil, "Deprecation of Type 0 Routing Headers in IPv6", RFC 5095, DOI 10.17487/RFC5095, December 2007, <<https://www.rfc-editor.org/info/rfc5095>>.
- [RFC5533] Nordmark, E. and M. Bagnulo, "Shim6: Level 3 Multihoming Shim Protocol for IPv6", RFC 5533, DOI 10.17487/RFC5533, June 2009, <<https://www.rfc-editor.org/info/rfc5533>>.
- [RFC5570] StJohns, M., Atkinson, R., and G. Thomas, "Common Architecture Label IPv6 Security Option (CALIPSO)", RFC 5570, DOI 10.17487/RFC5570, July 2009, <<https://www.rfc-editor.org/info/rfc5570>>.
- [RFC5971] Schulzrinne, H. and R. Hancock, "GIST: General Internet Signalling Transport", RFC 5971, DOI 10.17487/RFC5971, October 2010, <<https://www.rfc-editor.org/info/rfc5971>>.
- [RFC6275] Perkins, C., Ed., Johnson, D., and J. Arkko, "Mobility Support in IPv6", RFC 6275, DOI 10.17487/RFC6275, July 2011, <<https://www.rfc-editor.org/info/rfc6275>>.
- [RFC6398] Le Faucheur, F., Ed., "IP Router Alert Considerations and Usage", BCP 168, RFC 6398, DOI 10.17487/RFC6398, October 2011, <<https://www.rfc-editor.org/info/rfc6398>>.
- [RFC6550] Winter, T., Ed., Thubert, P., Ed., Brandt, A., Hui, J., Kelsey, R., Levis, P., Pister, K., Struik, R., Vasseur, JP., and R. Alexander, "RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks", RFC 6550, DOI 10.17487/RFC6550, March 2012, <<https://www.rfc-editor.org/info/rfc6550>>.

- [RFC6553] Hui, J. and JP. Vasseur, "The Routing Protocol for Low-Power and Lossy Networks (RPL) Option for Carrying RPL Information in Data-Plane Datagrams", RFC 6553, DOI 10.17487/RFC6553, March 2012, <<https://www.rfc-editor.org/info/rfc6553>>.
- [RFC6554] Hui, J., Vasseur, JP., Culler, D., and V. Manral, "An IPv6 Routing Header for Source Routes with the Routing Protocol for Low-Power and Lossy Networks (RPL)", RFC 6554, DOI 10.17487/RFC6554, March 2012, <<https://www.rfc-editor.org/info/rfc6554>>.
- [RFC6621] Macker, J., Ed., "Simplified Multicast Forwarding", RFC 6621, DOI 10.17487/RFC6621, May 2012, <<https://www.rfc-editor.org/info/rfc6621>>.
- [RFC6740] Atkinson, RJ. and SN. Bhatti, "Identifier-Locator Network Protocol (ILNP) Architectural Description", RFC 6740, DOI 10.17487/RFC6740, November 2012, <<https://www.rfc-editor.org/info/rfc6740>>.
- [RFC6744] Atkinson, RJ. and SN. Bhatti, "IPv6 Nonce Destination Option for the Identifier-Locator Network Protocol for IPv6 (ILNPv6)", RFC 6744, DOI 10.17487/RFC6744, November 2012, <<https://www.rfc-editor.org/info/rfc6744>>.
- [RFC6788] Krishnan, S., Kavanagh, A., Varga, B., Ooghe, S., and E. Nordmark, "The Line-Identification Option", RFC 6788, DOI 10.17487/RFC6788, November 2012, <<https://www.rfc-editor.org/info/rfc6788>>.
- [RFC6971] Herberg, U., Ed., Cardenas, A., Iwao, T., Dow, M., and S. Cespedes, "Depth-First Forwarding (DFF) in Unreliable Networks", RFC 6971, DOI 10.17487/RFC6971, June 2013, <<https://www.rfc-editor.org/info/rfc6971>>.
- [RFC7045] Carpenter, B. and S. Jiang, "Transmission and Processing of IPv6 Extension Headers", RFC 7045, DOI 10.17487/RFC7045, December 2013, <<https://www.rfc-editor.org/info/rfc7045>>.
- [RFC7112] Gont, F., Manral, V., and R. Bonica, "Implications of Oversized IPv6 Header Chains", RFC 7112, DOI 10.17487/RFC7112, January 2014, <<https://www.rfc-editor.org/info/rfc7112>>.

- [RFC7401] Moskowitz, R., Ed., Heer, T., Jokela, P., and T. Henderson, "Host Identity Protocol Version 2 (HIPv2)", RFC 7401, DOI 10.17487/RFC7401, April 2015, <<https://www.rfc-editor.org/info/rfc7401>>.
- [RFC7731] Hui, J. and R. Kelsey, "Multicast Protocol for Low-Power and Lossy Networks (MPL)", RFC 7731, DOI 10.17487/RFC7731, February 2016, <<https://www.rfc-editor.org/info/rfc7731>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [RFC8200] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", STD 86, RFC 8200, DOI 10.17487/RFC8200, July 2017, <<https://www.rfc-editor.org/info/rfc8200>>.
- [RFC8250] Elkins, N., Hamilton, R., and M. Ackermann, "IPv6 Performance and Diagnostic Metrics (PDM) Destination Option", RFC 8250, DOI 10.17487/RFC8250, September 2017, <<https://www.rfc-editor.org/info/rfc8250>>.
- [RFC8754] Filts, C., Ed., Dukes, D., Ed., Previdi, S., Leddy, J., Matsushima, S., and D. Voyer, "IPv6 Segment Routing Header (SRH)", RFC 8754, DOI 10.17487/RFC8754, March 2020, <<https://www.rfc-editor.org/info/rfc8754>>.
- [RFC8900] Bonica, R., Baker, F., Huston, G., Hinden, R., Troan, O., and F. Gont, "IP Fragmentation Considered Fragile", BCP 230, RFC 8900, DOI 10.17487/RFC8900, September 2020, <<https://www.rfc-editor.org/info/rfc8900>>.
- [RFC9008] Robles, M.I., Richardson, M., and P. Thubert, "Using RPI Option Type, Routing Header for Source Routes, and IPv6-in-IPv6 Encapsulation in the RPL Data Plane", RFC 9008, DOI 10.17487/RFC9008, April 2021, <<https://www.rfc-editor.org/info/rfc9008>>.

9.2. Informative References

- [Biondi2007] Biondi, P. and A. Ebalard, "IPv6 Routing Header Security", CanSecWest 2007 Security Conference, 2007, <http://www.secdev.org/conf/IPv6_RH_security-csw07.pdf>.

- [Cisco-EH] Cisco Systems, "IPv6 Extension Headers Review and Considerations", Whitepaper. October 2006,
<https://www.cisco.com/en/US/technologies/tk648/tk872/technologies_white_paper0900aecd8054d37d.pdf>.
- [draft-gont-6man-ipv6-opt-transmit]
Gont, F., Liu, W., and R. Bonica, "Transmission and Processing of IPv6 Options", IETF Internet Draft, work in progress, August 2014.
- [draft-ietf-nimrod-eid]
Lynn, C.L., "Endpoint Identifier Destination Option", IETF Internet Draft, draft-ietf-nimrod-eid-00.txt, November 1995.
- [FW-Benchmark]
Zack, E., "Firewall Security Assessment and Benchmarking IPv6 Firewall Load Tests", IPv6 Hackers Meeting #1, Berlin, Germany. June 30, 2013,
<<https://www.ipv6hackers.org/files/meetings/ipv6-hackers-1/zack-ipv6hackers1-firewall-security-assessment-and-benchmarking.pdf>>.
- [Huston-2022]
Huston, G. and J. Damas, "IPv6 Fragmentation and EH Behaviours", IEPG Meeting - March 2022 @ IETF 113, March 2022,
<<https://iepg.org/2022-03-20-ietf113/huston-v6frag.pdf>>.
- [I-D.irtf-pearg-numeric-ids-generation]
Gont, F. and I. Arce, "On the Generation of Transient Numeric Identifiers", Work in Progress, Internet-Draft, draft-irtf-pearg-numeric-ids-generation-08, 31 January 2022, <<https://www.ietf.org/archive/id/draft-irtf-pearg-numeric-ids-generation-08.txt>>.
- [I-D.vyncke-v6ops-james]
Vyncke, É., Léas, R., and J. Iurman, "Just Another Measurement of Extension header Survivability (JAMES)", Work in Progress, Internet-Draft, draft-vyncke-v6ops-james-01, 19 March 2022, <<https://www.ietf.org/archive/id/draft-vyncke-v6ops-james-01.txt>>.
- [IANA-IPV6-PARAM]
Internet Assigned Numbers Authority, "Internet Protocol Version 6 (IPv6) Parameters", December 2013,
<<https://www.iana.org/assignments/ipv6-parameters/ipv6-parameters.xhtml>>.

[IANA-PROTOCOLS]

Internet Assigned Numbers Authority, "Protocol Numbers", 2014, <<https://www.iana.org/assignments/protocol-numbers/protocol-numbers.xhtml>>.

[NIMROD-DOC]

Nimrod Documentation Page,
"http://ana-3.lcs.mit.edu/~jnc/nimrod/".

[RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, DOI 10.17487/RFC2460, December 1998, <<https://www.rfc-editor.org/info/rfc2460>>.

[RFC3871] Jones, G., Ed., "Operational Security Requirements for Large Internet Service Provider (ISP) IP Network Infrastructure", RFC 3871, DOI 10.17487/RFC3871, September 2004, <<https://www.rfc-editor.org/info/rfc3871>>.

[RFC6192] Dugal, D., Pignataro, C., and R. Dunn, "Protecting the Router Control Plane", RFC 6192, DOI 10.17487/RFC6192, March 2011, <<https://www.rfc-editor.org/info/rfc6192>>.

[RFC7126] Gont, F., Atkinson, R., and C. Pignataro, "Recommendations on Filtering of IPv4 Packets Containing IPv4 Options", BCP 186, RFC 7126, DOI 10.17487/RFC7126, February 2014, <<https://www.rfc-editor.org/info/rfc7126>>.

[RFC7739] Gont, F., "Security Implications of Predictable Fragment Identification Values", RFC 7739, DOI 10.17487/RFC7739, February 2016, <<https://www.rfc-editor.org/info/rfc7739>>.

[RFC7872] Gont, F., Linkova, J., Chown, T., and W. Liu, "Observations on the Dropping of Packets with IPv6 Extension Headers in the Real World", RFC 7872, DOI 10.17487/RFC7872, June 2016, <<https://www.rfc-editor.org/info/rfc7872>>.

[RFC9098] Gont, F., Hilliard, N., Doering, G., Kumari, W., Huston, G., and W. Liu, "Operational Implications of IPv6 Packets with Extension Headers", RFC 9098, DOI 10.17487/RFC9098, September 2021, <<https://www.rfc-editor.org/info/rfc9098>>.

Authors' Addresses

Fernando Gont
EdgeUno
Segurola y Habana 4310, 7mo Piso
Villa Devoto
Ciudad Autonoma de Buenos Aires
Argentina
Email: fernando.gont@edgeuno.com
URI: <https://www.edgeuno.com>

Will (Shucheng) Liu
Huawei Technologies
Bantian, Longgang District
Shenzhen
518129
P.R. China
Email: liushucheng@huawei.com

OPSEC
Internet-Draft
Intended status: Informational
Expires: October 13, 2017

K. Chittimaneni
Dropbox Inc.
M. Kaeo
Double Shot Security
E. Vyncke, Ed.
Cisco
April 11, 2017

Operational Security Considerations for IPv6 Networks
draft-ietf-opsec-v6-11

Abstract

Knowledge and experience on how to operate IPv4 securely is available: whether it is the Internet or an enterprise internal network. However, IPv6 presents some new security challenges. RFC 4942 describes the security issues in the protocol but network managers also need a more practical, operations-minded document to enumerate advantages and/or disadvantages of certain choices.

This document analyzes the operational security issues in all places of a network (enterprises, service providers and residential users) and proposes technical and procedural mitigations techniques.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on October 13, 2017.

Copyright Notice

Copyright (c) 2017 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	3
1.1. Requirements Language	3
2. Generic Security Considerations	4
2.1. Addressing Architecture	4
2.1.1. Statically Configured Addresses	4
2.1.2. Use of ULAs	5
2.1.3. Point-to-Point Links	6
2.1.4. Temporary Addresses - Privacy Extensions for SLAAC	6
2.1.5. Privacy consideration of Addresses	7
2.1.6. DHCP/DNS Considerations	7
2.2. Extension Headers	7
2.2.1. Order and Repetition of Extension Headers	8
2.2.2. Hop-by-Hop Extension Header	8
2.2.3. Fragmentation Extension Header	8
2.2.4. IP Security Extension Header	9
2.3. Link-Layer Security	9
2.3.1. SeND and CGA	9
2.3.2. Securing DHCP	10
2.3.3. ND/RA Rate Limiting	10
2.3.4. ND/RA Filtering	11
2.3.5. 3GPP Link-Layer Security	12
2.4. Control Plane Security	13
2.4.1. Control Protocols	14
2.4.2. Management Protocols	14
2.4.3. Packet Exceptions	15
2.5. Routing Security	16
2.5.1. Authenticating Neighbors/Peers	16
2.5.2. Securing Routing Updates Between Peers	17
2.5.3. Route Filtering	17
2.6. Logging/Monitoring	17
2.6.1. Data Sources	18
2.6.2. Use of Collected Data	22
2.6.3. Summary	24
2.7. Transition/Coexistence Technologies	24
2.7.1. Dual Stack	24
2.7.2. Transition Mechanisms	25

2.7.3. Translation Mechanisms	29
2.8. General Device Hardening	30
3. Enterprises Specific Security Considerations	31
3.1. External Security Considerations:	31
3.2. Internal Security Considerations:	32
4. Service Providers Security Considerations	32
4.1. BGP	32
4.1.1. Remote Triggered Black Hole Filtering	32
4.2. Transition Mechanism	33
4.3. Lawful Intercept	33
5. Residential Users Security Considerations	33
6. Further Reading	34
7. Acknowledgements	34
8. IANA Considerations	35
9. Security Considerations	35
10. References	35
10.1. Normative References	35
10.2. Informative References	35
Authors' Addresses	46

1. Introduction

Running an IPv6 network is new for most operators not only because they are not yet used to large scale IPv6 networks but also because there are subtle differences between IPv4 and IPv6 especially with respect to security. For example, all layer-2 interactions are now done using Neighbor Discovery Protocol [RFC4861] rather than using Address Resolution Protocol [RFC0826]. Also, there are subtle differences between NAT44 [RFC2993] and NPTv6 [RFC6296] which are explicitly pointed out in the latter's security considerations section.

IPv6 networks are deployed using a variety of techniques, each of which have their own specific security concerns.

This document complements [RFC4942] by listing all security issues when operating a network utilizing varying transition technologies and updating with ones that have been standardized since 2007. It also provides more recent operational deployment experiences where warranted.

1.1. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119] when they appear in ALL CAPS. These words may also appear in this document in lower case as plain English words, absent their normative meanings.

2. Generic Security Considerations

2.1. Addressing Architecture

IPv6 address allocations and overall architecture are an important part of securing IPv6. Initial designs, even if intended to be temporary, tend to last much longer than expected. Although initially IPv6 was thought to make renumbering easy, in practice, it may be extremely difficult to renumber without a good IP Addresses Management (IPAM) system.

Once an address allocation has been assigned, there should be some thought given to an overall address allocation plan. With the abundance of address space available, an address allocation may be structured around services along with geographic locations, which then can be a basis for more structured security policies to permit or deny services between geographic regions.

A common question is whether companies should use PI vs PA space [RFC7381], but from a security perspective there is little difference. However, one aspect to keep in mind is who has administrative ownership of the address space and who is technically responsible if/when there is a need to enforce restrictions on routability of the space due to malicious criminal activity.

2.1.1. Statically Configured Addresses

When considering how to assign statically configured addresses it is necessary to take into consideration the effectiveness of perimeter security in a given environment. There is a trade-off between ease of operational deployment where some portions of the IPv6 address could be easily recognizable for operational debugging and troubleshooting versus the risk of scanning; [SCANNING] shows that there are scientifically based mechanisms that make scanning for IPv6 reachable nodes more realizable than expected; see also [RFC7707]. The use of common multicast groups which are defined for important networked devices and the use of commonly repeated addresses could make it easy to figure out which devices are name servers, routers or other critical devices.

While in some environments the security is so poor that obfuscating addresses is considered a benefit; it is a better practice to ensure that perimeter rules are actively checked and enforced and that statically configured addresses follow some logical allocation scheme for ease of operation.

2.1.1.2. Use of ULAs

ULAs are intended for scenarios where IP addresses will not have global scope so they should not appear in the global BGP routing table. The implicit expectation from the RFC is that all ULAs will be randomly created as /48s. Any use of ULAs that are not created as a /48 violates RFC4193 [RFC4193].

ULAs could be useful for infrastructure hiding as described in RFC4864 [RFC4864]. Alternatively Link-Local addresses RFC7404 [RFC7404] could also be used. Although ULAs are supposed to be used in conjunction with global addresses for hosts that desire external connectivity, a few operators chose to use ULAs in conjunction with some sort of address translation at the border in order to maintain a perception of parity between their IPv4 and IPv6 setup. Some operators believe that stateful IPv6 Network Address and Port Translation (NAPT) provides some security not provided by NPTv6 (the authors of this document do not share this point of view). The use of stateful IPv6 NAPT would be problematic in trying to track specific machines that may source malware although this is less of an issue if appropriate logging is done which includes utilizing accurate timestamps and logging a node's source ports RFC6302 [RFC6302]. Another typical argument in favor of ULA is that there are too many mistakes made with ACL filters at the edge and the use of ULAs could make things easier to set filters.

The use of ULA does not isolate 'by magic' the part of the network using ULA from other parts of the network (including the Internet). Although section 4.1 of RFC4193 [RFC4193] explicitly states "If BGP is being used at the site border with an ISP, the default BGP configuration must filter out any Local IPv6 address prefixes, both incoming and outgoing.", the operational reality is that this guideline is not always followed. As written, RFC4193 makes no changes to default routing behavior of exterior protocols. Therefore, routers will happily forward packets whose source or destination address is ULA as long as they have a route to the destination and there is no ACL blocking those packets. This means that using ULA does not prevent route and packet filters having to be implemented and monitored. This also means that all Internet transit networks should consider ULA as source or destination as bogons packets and drop them.

It is important to carefully weigh the benefits of using ULAs versus utilizing a section of the global allocation and creating a more effective filtering strategy. It is also important to note that the IETF does not recommend the use of ULA and NPTv6.

2.1.1.3. Point-to-Point Links

RFC6164 [RFC6164] recommends the use of /127 for inter-router point-to-point links. A /127 prevents the ping-pong attack between routers. However, it should be noted that at the time of this writing, there are still many networks out there that follow the advice provided by RFC3627 [RFC3627] (obsoleted and marked Historic by RFC6547 [RFC6547]) and therefore continue to use /64's and/or /112's. We recommend that the guidance provided by RFC6164 be followed.

Some environments are also using link-local addressing for point-to-point links. While this practice could further reduce the attack surface against infrastructure devices, the operational disadvantages need also to be carefully considered RFC7404 [RFC7404].

2.1.1.4. Temporary Addresses - Privacy Extensions for SLAAC

Normal stateless address autoconfiguration (SLAAC) relies on the automatically generated EUI-64 address, which together with the /64 prefix makes up the global unique IPv6 address. The EUI-64 address is generated from the MAC address. Randomly generating an interface ID, as described in [RFC4941], is part of SLAAC with so-called privacy extension addresses and used to address some privacy concerns. Privacy extension addresses a.k.a. temporary addresses may help to mitigate the correlation of activities of a node within the same network, and may also reduce the attack exposure window.

As privacy extension addresses could also be used to obfuscate some malevolent activities (whether on purpose or not), it is advised in scenarios where user attribution is important to rely on a layer-2 authentication mechanism such as IEEE 802.1X [IEEE-802.1X] with the appropriate RADIUS accounting (Section 2.6.1.6) or to disable SLAAC and rely only on DHCPv6. However, in scenarios where anonymity is a strong desire (protecting user privacy is more important than user attribution), privacy extension addresses should be used.

Using privacy extension addresses prevents the operator from building a priori host specific access control lists (ACLs). It must be noted that recent versions of Windows do not use the MAC address anymore to build the stable address but use a mechanism similar to the one described in [RFC7217], this also means that such an ACL cannot be configured based solely on the MAC address of the nodes, diminishing the value of such ACL. On the other hand, different VLANs are often used to segregate users, in this case ACL can rely on a /64 prefix per VLAN rather than a per host ACL entry.

The decision to utilize privacy extension addresses can come down to whether the network is managed versus unmanaged. In some environments full visibility into the network is required at all times which requires that all traffic be attributable to where it is sourced or where it is destined to within a specific network. This situation is dependent on what level of logging is performed. If logging considerations include utilizing accurate timestamps and logging a node's source ports [RFC6302] then there should always exist appropriate user attribution needed to get to the source of any malware originator or source of criminal activity.

Disabling SLAAC and privacy extensions addresses can be done by sending Router Advertisement with a hint to get addresses via DHCPv6 by setting the M-bit but also disabling SLAAC by resetting all A-bits in all prefix information options sent in the Router Advertisement message.

2.1.5. Privacy consideration of Addresses

However, there are several privacy issues still present with [RFC4941] such as host tracking, and address scanning attacks are still possible. More details are provided in Appendix A. of [RFC7217] and in [RFC7721].

2.1.6. DHCP/DNS Considerations

Many environments use DHCPv6 to allocate addresses to ensure auditability and traceability (but see Section 2.6.1.5). A main security concern is the ability to detect and counteract against rogue DHCP servers (Section 2.3.2).

DNS is often used for malware activities and while there are no fundamental differences with IPv4 and IPv6 security concerns, there are specific consideration in DNS64 RFC6147 [RFC6147] environments that need to be understood. Specifically the interactions and potential to interference with DNSsec implementation need to be understood - these are pointed out in detail in Section 2.7.3.2.

2.2. Extension Headers

The extension headers are one of the most critical differentiator between IPv4 and IPv6. They have also become a very controversial topic since forwarding nodes that discard packets containing extension headers are known to cause connectivity failures and deployment problems. Understanding the role of varying extension headers is important and this section enumerates the ones that need careful consideration. The IANA has closed the the existing empty

"Next Header Types" registry to new entries and is redirecting its users to a new "IPv6 Extension Header Types" registry.

A clarification on how intermediate nodes should handle existing packets with extension headers and any extension headers that are defined in the future is found in RFC7045 [RFC7045]. The uniform TLV format to be used for defining future extension headers is described in RFC6564 [RFC6564]. Some observations listed in RFC7872 [RFC7872] seems to indicate that packets with certain extension headers may not traverse the Internet to its intended destination based on operator policies.

It must also be noted that there is no indication in the packet whether the Next Protocol field points to an extension header or to a transport header. This may confuse some filtering rules.

2.2.1. Order and Repetition of Extension Headers

While RFC2460 [RFC2460] defines the order and the maximum repetition of extension headers, there are still IPv6 implementations at the time of writing this document which support a wrong order of headers (such as ESP before routing) or an illegal repetition of headers (such as multiple routing headers). The same applies for options contained in the extension headers (see [I-D.kampanakis-6man-ipv6-eh-parsing]). In some cases, it has lead to nodes crashing when receiving or forwarding wrongly formatted packets.

2.2.2. Hop-by-Hop Extension Header

The hop-by-hop extension header, when present in an IPv6 packet, forces all nodes in the path to inspect this header. This is of course a large avenue for a denial of service as most if not all routers cannot process this kind of packets in hardware but have to 'punt' this packet for software processing. See also [I-D.ietf-6man-hbh-header-handling].

2.2.3. Fragmentation Extension Header

The fragmentation extension header is used by the source when it has to fragment packets. RFC7112 [RFC7112] explains why it is important to:

firewall and security devices should drop first fragment not containing enough of the layer-4 header;

destination node should ignore first fragment not containing the entire IPv6 header chain.

Else, stateless filtering could be bypassed by an hostile party. RFC6980 [RFC6980] applies the same rule to NDP and the RA-guard function.

2.2.4. IP Security Extension Header

The IPsec [RFC4301] [RFC4301] extension headers (AH [RFC4302] and ESP [RFC4303]) are required if IPsec is to be utilized for network level security functionality.

2.3. Link-Layer Security

IPv6 relies heavily on the Neighbor Discovery protocol (NDP) RFC4861 [RFC4861] to perform a variety of link operations such as discovering other nodes on the link, resolving their link-layer addresses, and finding routers on the link. If not secured, NDP is vulnerable to various attacks such as router/neighbor message spoofing, redirect attacks, Duplicate Address Detection (DAD) DoS attacks, etc. many of these security threats to NDP have been documented in IPv6 ND Trust Models and Threats RFC3756 [RFC3756] and in RFC6583 [RFC6583].

2.3.1. SeND and CGA

Secure Neighbor Discovery (SeND), as described in RFC3971 [RFC3971], is a mechanism that was designed to secure ND messages. This approach involves the use of new NDP options to carry public key based signatures. Cryptographically Generated Addresses (CGA), as described in RFC3972 [RFC3972], are used to ensure that the sender of a Neighbor Discovery message is the actual "owner" of the claimed IPv6 address. A new NDP option, the CGA option, was introduced and is used to carry the public key and associated parameters. Another NDP option, the RSA Signature option, is used to protect all messages relating to neighbor and Router discovery.

SeND protects against:

- o Neighbor Solicitation/Advertisement Spoofing
- o Neighbor Unreachability Detection Failure
- o Duplicate Address Detection DoS Attack
- o Router Solicitation and Advertisement Attacks
- o Replay Attacks
- o Neighbor Discovery DoS Attacks

SeND does NOT:

- o Protect statically configured addresses
- o Protect addresses configured using fixed identifiers (i.e. EUI-64)
- o Provide confidentiality for NDP communications
- o Compensate for an unsecured link - SEND does not require that the addresses on the link and Neighbor Advertisements correspond

However, at this time and after many years after their specifications, CGA and SeND do not have wide support from generic operating systems; hence, their usefulness is limited.

2.3.2. Securing DHCP

Dynamic Host Configuration Protocol for IPv6 (DHCPv6), as detailed in RFC3315 [RFC3315], enables DHCP servers to pass configuration parameters such as IPv6 network addresses and other configuration information to IPv6 nodes. DHCP plays an important role in any large network by providing robust stateful configuration and autoregistration of DNS Host Names.

The two most common threats to DHCP clients come from malicious (a.k.a. rogue) or unintentionally misconfigured DHCP servers. A malicious DHCP server is established with the intent of providing incorrect configuration information to the client to cause a denial of service attack or mount a man in the middle attack. While unintentionally, a misconfigured DHCP server can have the same impact. Additional threats against DHCP are discussed in the security considerations section of RFC3315 [RFC3315]DHCP-shield

RFC7610 [RFC7610] specifies a mechanism for protecting connected DHCPv6 clients against rogue DHCPv6 servers. This mechanism is based on DHCPv6 packet-filtering at the layer-2 device; the administrator specifies the interfaces connected to DHCPv6 servers.

It is recommended to use DHCP-shield.

2.3.3. ND/RA Rate Limiting

Neighbor Discovery (ND) can be vulnerable to denial of service (DoS) attacks in which a router is forced to perform address resolution for a large number of unassigned addresses. Possible side effects of this attack preclude new devices from joining the network or even worse rendering the last hop router ineffective due to high CPU

usage. Easy mitigative steps include rate limiting Neighbor Solicitations, restricting the amount of state reserved for unresolved solicitations, and clever cache/timer management.

RFC6583 [RFC6583] discusses the potential for DoS in detail and suggests implementation improvements and operational mitigation techniques that may be used to mitigate or alleviate the impact of such attacks. Here are some feasible mitigation options that can be employed by network operators today:

- o Ingress filtering of unused addresses by ACL, route filtering, longer than /64 prefix; These require static configuration of the addresses.
- o Tuning of NDP process (where supported).

Additionally, IPv6 ND uses multicast extensively for signaling messages on the local link to avoid broadcast messages for on-the-wire efficiency. However, this has some side effects on wifi networks, especially a negative impact on battery life of smartphones and other battery operated devices that are connected to such networks. The following drafts are actively discussing methods to rate limit RAs and other ND messages on wifi networks in order to address this issue:

- o [I-D.thubert-savi-ra-throttler]
- o [I-D.chakrabarti-nordmark-6man-efficient-nd]

2.3.4. ND/RA Filtering

Router Advertisement spoofing is a well-known attack vector and has been extensively documented. The presence of rogue RAs, either intentional or malicious, can cause partial or complete failure of operation of hosts on an IPv6 link. For example, a host can select an incorrect router address which can be used as a man-in-the-middle (MITM) attack or can assume wrong prefixes to be used for stateless address configuration (SLAAC). RFC6104 [RFC6104] summarizes the scenarios in which rogue RAs may be observed and presents a list of possible solutions to the problem. RFC6105 [RFC6105] (RA-Guard) describes a solution framework for the rogue RA problem where network segments are designed around switching devices that are capable of identifying invalid RAs and blocking them before the attack packets actually reach the target nodes.

However, several evasion techniques that circumvent the protection provided by RA-Guard have surfaced. A key challenge to this mitigation technique is introduced by IPv6 fragmentation. An

attacker can conceal the attack by fragmenting his packets into multiple fragments such that the switching device that is responsible for blocking invalid RAs cannot find all the necessary information to perform packet filtering in the same packet. RFC7113 [RFC7113] describes such evasion techniques, and provides advice to RA-Guard implementers such that the aforementioned evasion vectors can be eliminated.

Given that the IPv6 Fragmentation Header can be leveraged to circumvent current implementations of RA-Guard, RFC6980 [RFC6980] updates RFC4861 [RFC4861] such that use of the IPv6 Fragmentation Header is forbidden in all Neighbor Discovery messages except "Certification Path Advertisement", thus allowing for simple and effective measures to counter Neighbor Discovery attacks.

The Source Address Validation Improvements (SAVI) working group has worked on other ways to mitigate the effects of such attacks. RFC7513 [RFC7513] would help in creating bindings between a DHCPv4 RFC2131 [RFC2131] /DHCPv6 RFC3315 [RFC3315] assigned source IP address and a binding anchor RFC7039 [RFC7039] on a SAVI device. Also, RFC6620 [RFC6620] describes how to glean similar bindings when DHCP is not used. The bindings can be used to filter packets generated on the local link with forged source IP address.

It is still recommended that RA-Guard be employed as a first line of defense against common attack vectors including misconfigured hosts.

2.3.5. 3GPP Link-Layer Security

The 3GPP link is a point-to-point like link that has no link-layer address. This implies there can only be an end host (the mobile hand-set) and the first-hop router (i.e., a GPRS Gateway Support Node (GGSN) or a Packet Gateway (PGW)) on that link. The GGSN/PGW never configures a non link-local address on the link using the advertised /64 prefix on it. The advertised prefix must not be used for on-link determination. There is no need for an address resolution on the 3GPP link, since there are no link-layer addresses. Furthermore, the GGSN/PGW assigns a prefix that is unique within each 3GPP link that uses IPv6 stateless address autoconfiguration. This avoids the necessity to perform DAD at the network level for every address built by the mobile host. The GGSN/PGW always provides an IID to the cellular host for the purpose of configuring the link-local address and ensures the uniqueness of the IID on the link (i.e., no collisions between its own link-local address and the mobile host's one).

The 3GPP link model itself mitigates most of the known NDP-related Denial-of-Service attacks. In practice, the GGSN/PGW only needs to route all traffic to the mobile host that falls under the prefix assigned to it. As there is also a single host on the 3GPP link, there is no need to defend that IPv6 address.

See Section 5 of RFC6459 [RFC6459] for a more detailed discussion on the 3GPP link model, NDP on it and the address configuration detail.

2.4. Control Plane Security

RFC6192 [RFC6192] defines the router control plane. This definition is repeated here for the reader's convenience.

Modern router architecture design maintains a strict separation of forwarding and router control plane hardware and software. The router control plane supports routing and management functions. It is generally described as the router architecture hardware and software components for handling packets destined to the device itself as well as building and sending packets originated locally on the device. The forwarding plane is typically described as the router architecture hardware and software components responsible for receiving a packet on an incoming interface, performing a lookup to identify the packet's IP next hop and determine the best outgoing interface towards the destination, and forwarding the packet out through the appropriate outgoing interface.

While the forwarding plane is usually implemented in high-speed hardware, the control plane is implemented by a generic processor (named router processor RP) and cannot process packets at a high rate. Hence, this processor can be attacked by flooding its input queue with more packets than it can process. The control plane processor is then unable to process valid control packets and the router can lose OSPF or BGP adjacencies which can cause a severe network disruption.

The mitigation technique is:

- o To drop non-legit control packet before they are queued to the RP (this can be done by a forwarding plane ACL) and
- o To rate limit the remaining packets to a rate that the RP can sustain. Protocol specific protection should also be done (for example, a spoofed OSPFv3 packet could trigger the execution of the Dijkstra algorithm, therefore the number of Dijkstra execution should be also rate limited).

This section will consider several classes of control packets:

- o Control protocols: routing protocols: such as OSPFv3, BGP and by extension Neighbor Discovery and ICMP
- o Management protocols: SSH, SNMP, IPfix, etc
- o Packet exceptions: which are normal data packets which requires a specific processing such as generating a packet-too-big ICMP message or having the hop-by-hop extension header.

2.4.1. Control Protocols

This class includes OSPFv3, BGP, NDP, ICMP.

An ingress ACL to be applied on all the router interfaces SHOULD be configured such as:

- o drop OSPFv3 (identified by Next-Header being 89) and RIPng (identified by UDP port 521) packets from a non link-local address
- o allow BGP (identified by TCP port 179) packets from all BGP neighbors and drop the others
- o allow all ICMP packets (transit and to the router interfaces)

Note: dropping OSPFv3 packets which are authenticated by IPsec could be impossible on some routers whose ACL are unable to parse the IPsec ESP or AH extension headers.

Rate limiting of the valid packets SHOULD be done. The exact configuration obviously depends on the power of the Route Processor.

2.4.2. Management Protocols

This class includes: SSH, SNMP, syslog, NTP, etc

An ingress ACL to be applied on all the router interfaces SHOULD be configured such as:

- o Drop packets destined to the routers except those belonging to protocols which are used (for example, permit TCP 22 and drop all when only SSH is used);
- o Drop packets where the source does not match the security policy, for example if SSH connections should only be originated from the NOC, then the ACL should permit TCP port 22 packets only from the NOC prefix.

Rate limiting of the valid packets SHOULD be done. The exact configuration obviously depends on the power of the Route Processor.

2.4.3. Packet Exceptions

This class covers multiple cases where a data plane packet is punted to the route processor because it requires specific processing:

- o generation of an ICMP packet-too-big message when a data plane packet cannot be forwarded because it is too large;
- o generation of an ICMP hop-limit-expired message when a data plane packet cannot be forwarded because its hop-limit field has reached 0;
- o generation of an ICMP destination-unreachable message when a data plane packet cannot be forwarded for any reason;
- o processing of the hop-by-hop extension header (see also [I-D.ietf-6man-hbh-header-handling]);
- o or more specific to some router implementation: an oversized extension header chain which cannot be processed by the hardware and force the packet to be punted to the generic router CPU.

On some routers, not everything can be done by the specialized data plane hardware which requires some packets to be 'punted' to the generic RP. This could include for example the processing of a long extension header chain in order to apply an ACL based on layer 4 information. RFC6980 [RFC6980] and more generally RFC7112 [RFC7112] highlights the security implications of oversized extension header chains on routers and updates RFC2460 [RFC2460] such that the first fragment of a packet is required to contain the entire IPv6 header chain.

An ingress ACL cannot help to mitigate a control plane attack using those packet exceptions. The only protection for the RP is to limit the rate of those packet exceptions forwarded to the RP, this means that some data plane packets will be dropped without any ICMP messages back to the source which will cause Path MTU holes. But, there is no other solution.

In addition to limiting the rate of data plane packets queued to the RP, it is also important to limit the generation rate of ICMP messages both to save the RP but also to prevent an amplification attack using the router as a reflector.

2.5. Routing Security

Routing security in general can be broadly divided into three sections:

1. Authenticating neighbors/peers
2. Securing routing updates between peers
3. Route filtering

[RFC7454] covers these sections specifically for BGP in detail.

2.5.1. Authenticating Neighbors/Peers

A basic element of routing is the process of forming adjacencies, neighbor, or peering relationships with other routers. From a security perspective, it is very important to establish such relationships only with routers and/or administrative domains that one trusts. A traditional approach has been to use MD5 HMAC, which allows routers to authenticate each other prior to establishing a routing relationship.

OSPFv3 can rely on IPsec to fulfill the authentication function. However, it should be noted that IPsec support is not standard on all routing platforms. In some cases, this requires specialized hardware that offloads crypto over to dedicated ASICs or enhanced software images (both of which often come with added financial cost) to provide such functionality. An added detail is to determine whether OSPFv3 IPsec implementations use AH or ESP-Null for integrity protection. In early implementations all OSPFv3 IPsec configurations relied on AH since the details weren't specified in RFC5340 [RFC5340] or RFC2740 [RFC2740] that was obsoleted by the former. However, the document which specifically describes how IPsec should be implemented for OSPFv3 RFC4552 [RFC4552] specifically states that ESP-Null MUST and AH MAY be implemented since it follows the overall IPsec standards wordings. OSPFv3 can also use normal ESP to encrypt the OSPFv3 payload to hide the routing information.

RFC7166 [RFC7166] (which obsoletes RFC6506 [RFC6506] changes OSPFv3's reliance on IPsec by appending an authentication trailer to the end of the OSPFv3 packets. This document does not specifically provide for a mechanism that will authenticate the specific originator of a packet. Rather, it will allow a router to confirm that the packet has indeed been issued by a router that had access to the shared authentication key.

With all authentication mechanisms, operators should confirm that implementations can support re-keying mechanisms that do not cause outages. There have been instances where any re-keying cause outages and therefore the tradeoff between utilizing this functionality needs to be weighed against the protection it provides.

2.5.2. Securing Routing Updates Between Peers

IPv6 initially mandated the provisioning of IPsec capability in all nodes. However, in the updated IPv6 Nodes Requirement standard RFC6434 [RFC6434] is now a SHOULD and not MUST implement. Theoretically it is possible, and recommended, that communication between two IPv6 nodes, including routers exchanging routing information be encrypted using IPsec. In practice however, deploying IPsec is not always feasible given hardware and software limitations of various platforms deployed, as described in the earlier section. Additionally, in a protocol such as OSPFv3 where adjacencies are formed on a one-to-many basis, IPsec key management becomes difficult to maintain and is not often utilized.

2.5.3. Route Filtering

Route filtering policies will be different depending on whether they pertain to edge route filtering vs internal route filtering. At a minimum, IPv6 routing policy as it pertains to routing between different administrative domains should aim to maintain parity with IPv4 from a policy perspective e.g.,

- o Filter internal-use, non-globally routable IPv6 addresses at the perimeter
- o Discard packets from and to bogon and reserved space
- o Configure ingress route filters that validate route origin, prefix ownership, etc. through the use of various routing databases, e.g., RADB. There is additional work being done in this area to formally validate the origin ASs of BGP announcements in RFC6810 [RFC6810]

Some good recommendations for filtering can be found from Team CYMRU at [CYMRU].

2.6. Logging/Monitoring

In order to perform forensic research in case of any security incident or to detect abnormal behaviors, network operator should log multiple pieces of information.

This includes:

- o logs of all applications when available (for example web servers);
- o use of IP Flow Information Export [RFC7011] also known as IPfix;
- o use of SNMP MIB [RFC4293];
- o use of the Neighbor cache;
- o use of stateful DHCPv6 [RFC3315] lease cache, especially when a relay agent [RFC6221] in layer-2 switches is used;
- o use of RADIUS [RFC2866] for accounting records.

Please note that there are privacy issues related to how those logs are collected, kept and safely discarded. Operators are urged to check their country legislation.

All those pieces of information will be used for:

- o forensic (Section 2.6.2.1) research to answer questions such as who did what and when?
- o correlation (Section 2.6.2.3): which IP addresses were used by a specific node (assuming the use of privacy extensions addresses [RFC4941])
- o inventory (Section 2.6.2.2): which IPv6 nodes are on my network?
- o abnormal behavior detection (Section 2.6.2.4): unusual traffic patterns are often the symptoms of a abnormal behavior which is in turn a potential attack (denial of services, network scan, a node being part of a botnet, ...)

2.6.1. Data Sources

This section lists the most important sources of data that are useful for operational security.

2.6.1.1. Logs of Applications

Those logs are usually text files where the remote IPv6 address is stored in all characters (not binary). This can complicate the processing since one IPv6 address, 2001:db8::1 can be written in multiple ways such as:

- o 2001:DB8::1 (in uppercase)

- o 2001:0db8::0001 (with leading 0)
- o and many other ways.

RFC 5952 [RFC5952] explains this problem in detail and recommends the use of a single canonical format (in short use lower case and suppress leading 0). This memo recommends the use of canonical format [RFC5952] for IPv6 addresses in all possible cases. If the existing application cannot log under the canonical format, then this memo recommends the use an external program in order to canonicalize all IPv6 addresses.

For example, this perl script can be used:

```
#!/usr/bin/perl -w
use strict ;
use warnings ;
use Socket ;
use Socket6 ;

my (@words, $word, $binary_address) ;

## go through the file one line at a time
while (my $line = <STDIN>) {
    chomp $line;
    foreach my $word (split /\s+/, $line) {
        $binary_address = inet_pton AF_INET6, $word ;
        if ($binary_address) {
            print inet_ntop AF_INET6, $binary_address ;
        } else {
            print $word ;
        }
        print " " ;
    }
    print "\n" ;
}
```

2.6.1.2. IP Flow Information Export by IPv6 Routers

IPfix [RFC7012] defines some data elements that are useful for security:

- o in section 5.4 (IP Header fields): nextHeaderIPv6 and sourceIPv6Address;
- o in section 5.6 (Sub-IP fields) sourceMacAddress.

Moreover, IPfix is very efficient in terms of data handling and transport. It can also aggregate flows by a key such as sourceMacAddress in order to have aggregated data associated with a specific sourceMacAddress. This memo recommends the use of IPfix and aggregation on nextHeaderIPv6, sourceIPv6Address and sourceMacAddress.

2.6.1.3. SNMP MIB by IPv6 Routers

RFC 4293 [RFC4293] defines a Management Information Base (MIB) for the two address families of IP. This memo recommends the use of:

- o ipIfStatsTable table which collects traffic counters per interface;
- o ipNetToPhysicalTable table which is the content of the Neighbor cache, i.e. the mapping between IPv6 and data-link layer addresses.

2.6.1.4. Neighbor Cache of IPv6 Routers

The neighbor cache of routers contains all mappings between IPv6 addresses and data-link layer addresses. It is usually available by two means:

- o the SNMP MIB (Section 2.6.1.3) as explained above;
- o also by connecting over a secure management channel (such as SSH or HTTPS) and explicitly requesting a neighbor cache dump.

The neighbor cache is highly dynamic as mappings are added when a new IPv6 address appears on the network (could be quite often with privacy extension addresses [RFC4941] or when they are removed when the state goes from UNREACH to removed (the default time for a removal per Neighbor Unreachability Detection [RFC4861] algorithm is 38 seconds for a typical host such as Windows 7). This means that the content of the neighbor cache must periodically be fetched every 30 seconds (to be on the safe side) and stored for later use.

This is an important source of information because it is trivial (on a switch not using the SAVI [RFC7039] algorithm) to defeat the mapping between data-link layer address and IPv6 address. Let us rephrase the previous statement: having access to the current and past content of the neighbor cache has a paramount value for forensic and audit trail.

2.6.1.5. Stateful DHCPv6 Lease

In some networks, IPv6 addresses are managed by stateful DHCPv6 server [RFC3315] that leases IPv6 addresses to clients. It is indeed quite similar to DHCP for IPv4 so it can be tempting to use this DHCP lease file to discover the mapping between IPv6 addresses and data-link layer addresses as it was usually done in the IPv4 era.

It is not so easy in the IPv6 era because not all nodes will use DHCPv6 (there are nodes which can only do stateless autoconfiguration) but also because DHCPv6 clients are identified not by their hardware-client address as in IPv4 but by a DHCP Unique ID (DUID) which can have several formats: some being the data-link layer address, some being data-link layer address prepended with time information or even an opaque number which is useless for operation security. Moreover, when the DUID is based on the data-link address, this address can be of any interface of the client (such as the wireless interface while the client actually uses its wired interface to connect to the network).

If a lightweight DHCP relay agent [RFC6221] is used in the layer-2 switches, then the DHCP server also receives the Interface-ID information which could be save in order to identify the interface of the switches which received a specific leased IPv6 address.

In short, the DHCPv6 lease file is less interesting than in the IPv4 era. DHCPv6 servers that keeps the relayed data-link layer address in addition to the DUID in the lease file do not suffer from this limitation. On a managed network where all hosts support DHCPv6, special care must be taken to prevent stateless autoconfiguration anyway (and if applicable) by sending RA with all announced prefixes without the A-bit set.

The mapping between data-link layer address and the IPv6 address can be secured by using switches implementing the SAVI [RFC7513] algorithms. Of course, this also requires that data-link layer address is protected by using layer-2 mechanism such as [IEEE-802.1X].

2.6.1.6. RADIUS Accounting Log

For interfaces where the user is authenticated via a RADIUS [RFC2866] server, and if RADIUS accounting is enabled, then the RADIUS server receives accounting Acct-Status-Type records at the start and at the end of the connection which include all IPv6 (and IPv4) addresses used by the user. This technique can be used notably for Wi-Fi networks with Wi-Fi Protected Address (WPA) or any other IEEE 802.1X [IEEE-802.1X]wired interface on an Ethernet switch.

2.6.1.7. Other Data Sources

There are other data sources that must be kept exactly as in the IPv4 network:

- o historical mapping of IPv6 addresses to users of remote access VPN;
- o historical mapping of MAC address to switch interface in a wired network.

2.6.2. Use of Collected Data

This section leverages the data collected as described before (Section 2.6.1) in order to achieve several security benefits.

2.6.2.1. Forensic

The forensic use case is when the network operator must locate an IPv6 address that was present in the network at a certain time or is still currently in the network.

The source of information can be, in decreasing order, neighbor cache, DHCP lease file. Then, the procedure is:

1. based on the IPv6 prefix of the IPv6 address find the router(s) which are used to reach this prefix;
2. based on this limited set of routers, on the incident time and on IPv6 address to retrieve the data-link address from live neighbor cache, from the historical data of the neighbor cache, or from the DHCP lease file;
3. based on the data-link layer address, look-up on which switch interface was this data-link layer address. In the case of wireless LAN, the RADIUS log should have the mapping between user identification and the MAC address.

At the end of the process, the interface where the malicious user was connected or the username that was used by the malicious user is found.

2.6.2.2. Inventory

RFC 7707 [RFC7707] (which obsoletes RFC 5157 [RFC5157]) is about the difficulties to scan an IPv6 network due to the vast number of IPv6 addresses per link. This has the side effect of making the inventory task difficult in an IPv6 network while it was trivial to do in an

IPv4 network (a simple enumeration of all IPv4 addresses, followed by a ping and a TCP/UDP port scan). Getting an inventory of all connected devices is of prime importance for a secure operation of a network.

There are many ways to do an inventory of an IPv6 network.

The first technique is to use the IPfix information and extract the list of all IPv6 source addresses to find all IPv6 nodes that sent packets through a router. This is very efficient but alas will not discover silent node that never transmitted such packets... Also, it must be noted that link-local addresses will never be discovered by this means.

The second way is again to use the collected neighbor cache content to find all IPv6 addresses in the cache. This process will also discover all link-local addresses. See Section 2.6.1.4.

Another way works only for local network, it consists in sending a ICMP ECHO_REQUEST to the link-local multicast address ff02::1 which is all IPv6 nodes on the network. All nodes should reply to this ECHO_REQUEST per [RFC4443].

Other techniques involve enumerating the DNS zones, parsing log files, leveraging service discovery such as mDNS RFC6762 [RFC6762] and RFC6763 [RFC6763].

Other techniques involve enumerating the DNS zones, especially looking at reverse DNS records and CNAMEs. Or scanning for DNS misconfigurations to find DNS servers that send NXDOMAIN instead of NOERROR for non-existing nodes with children, which violates RFC8020 [RFC8020]. Parsing log files and leveraging service discovery such as mDNS RFC6762 [RFC6762] and RFC6763 [RFC6763] are also added techniques.

2.6.2.3. Correlation

In an IPv4 network, it is easy to correlate multiple logs, for example to find events related to a specific IPv4 address. A simple Unix grep command was enough to scan through multiple text-based files and extract all lines relevant to a specific IPv4 address.

In an IPv6 network, this is slightly more difficult because different character strings can express the same IPv6 address. Therefore, the simple Unix grep command cannot be used. Moreover, an IPv6 node can have multiple IPv6 addresses...

In order to do correlation in IPv6-related logs, it is advised to have all logs with canonical IPv6 addresses. Then, the neighbor cache current (or historical) data set must be searched to find the data-link layer address of the IPv6 address. Then, the current and historical neighbor cache data sets must be searched for all IPv6 addresses associated to this data-link layer address: this is the search set. The last step is to search in all log files (containing only IPv6 address in canonical format) for any IPv6 addresses in the search set.

2.6.2.4. Abnormal Behavior Detection

Abnormal behaviors (such as network scanning, spamming, denial of service) can be detected in the same way as in an IPv4 network

- o sudden increase of traffic detected by interface counter (SNMP) or by aggregated traffic from IPfix records [RFC7012];
- o change of traffic pattern (number of connection per second, number of connection per host...) with the use of IPfix [RFC7012]

2.6.3. Summary

While some data sources (IPfix, MIB, switch CAM tables, logs, ...) used in IPv4 are also used in the secure operation of an IPv6 network, the DHCPv6 lease file is less reliable and the neighbor cache is of prime importance.

The fact that there are multiple ways to express in a character string the same IPv6 address renders the use of filters mandatory when correlation must be done.

2.7. Transition/Coexistence Technologies

Some text

2.7.1. Dual Stack

Dual stack has established itself as the preferred deployment choice for most network operators without an MPLS core where 6PE RFC4798 [RFC4798] is quite common. Dual stacking the network offers many advantages over other transition mechanisms. Firstly, it is easy to turn on without impacting normal IPv4 operations. Secondly, perhaps more importantly, it is easier to troubleshoot when things break. Dual stack allows you to gradually turn IPv4 operations down when your IPv6 network is ready for prime time.

From an operational security perspective, this now means that you have twice the exposure. One needs to think about protecting both protocols now. At a minimum, the IPv6 portion of a dual stacked network should maintain parity with IPv4 from a security policy point of view. Typically, the following methods are employed to protect IPv4 networks at the edge:

- o ACLs to permit or deny traffic
- o Firewalls with stateful packet inspection

It is recommended that these ACLs and/or firewalls be additionally configured to protect IPv6 communications. Also, given the end-to-end connectivity that IPv6 provides, it is also recommended that hosts be fortified against threats. General device hardening guidelines are provided in Section 2.8

2.7.2. Transition Mechanisms

There are many tunnels used for specific usecases. Except when protected by IPsec [RFC4301], all those tunnels have a couple of security issues (most of them being described in RFC 6169 [RFC6169]);

- o tunnel injection: a malevolent person knowing a few pieces of information (for example the tunnel endpoints and the used protocol) can forge a packet which looks like a legit and valid encapsulated packet that will gladly be accepted by the destination tunnel endpoint, this is a specific case of spoofing;
- o traffic interception: no confidentiality is provided by the tunnel protocols (without the use of IPsec), therefore anybody on the tunnel path can intercept the traffic and have access to the clear-text IPv6 packet;
- o service theft: as there is no authorization, even a non authorized user can use a tunnel relay for free (this is a specific case of tunnel injection);
- o reflection attack: another specific use case of tunnel injection where the attacker injects packets with an IPv4 destination address not matching the IPv6 address causing the first tunnel endpoint to re-encapsulate the packet to the destination... Hence, the final IPv4 destination will not see the original IPv4 address but only one IPv4 address of the relay router.
- o bypassing security policy: if a firewall or an IPS is on the path of the tunnel, then it will probably neither inspect nor detect a malevolent IPv6 traffic contained in the tunnel.

To mitigate the bypassing of security policies, it could be helpful to block all default configuration tunnels by denying all IPv4 traffic matching:

- o IP protocol 41: this will block ISATAP (Section 2.7.2.2), 6to4 (Section 2.7.2.4), 6rd (Section 2.7.2.5) as well as 6in4 (Section 2.7.2.1) tunnels;
- o IP protocol 47: this will block GRE (Section 2.7.2.1) tunnels;
- o UDP protocol 3544: this will block the default encapsulation of Teredo (Section 2.7.2.3) tunnels.

Ingress filtering [RFC2827] should also be applied on all tunnel endpoints if applicable to prevent IPv6 address spoofing.

As several of the tunnel techniques share the same encapsulation (i.e. IPv4 protocol 41) and embed the IPv4 address in the IPv6 address, there are a set of well-known looping attacks described in RFC 6324 [RFC6324], this RFC also proposes mitigation techniques.

2.7.2.1. Site-to-Site Static Tunnels

Site-to-site static tunnels are described in RFC 2529 [RFC2529] and in GRE [RFC2784]. As the IPv4 endpoints are statically configured and are not dynamic they are slightly more secure (bi-directional service theft is mostly impossible) but traffic interception and tunnel injection are still possible. Therefore, the use of IPsec [RFC4301] in transport mode and protecting the encapsulated IPv4 packets is recommended for those tunnels. Alternatively, IPsec in tunnel mode can be used to transport IPv6 traffic over a non-trusted IPv4 network.

2.7.2.2. ISATAP

ISATAP tunnels [RFC5214] are mainly used within a single administrative domain and to connect a single IPv6 host to the IPv6 network. This means that endpoints and the tunnel endpoint are usually managed by a single entity; therefore, audit trail and strict anti-spoofing are usually possible and this raises the overall security.

Special care must be taken to avoid looping attack by implementing the measures of RFC 6324 [RFC6324] and of RFC6964 [RFC6964].

IPsec [RFC4301] in transport or tunnel mode can be used to secure the IPv4 ISATAP traffic to provide IPv6 traffic confidentiality and prevent service theft.

2.7.2.3. Teredo

Teredo tunnels [RFC4380] are mainly used in a residential environment because that can easily traverse an IPv4 NAT-PT device thanks to its UDP encapsulation and they connect a single host to the IPv6 Internet. Teredo shares the same issues as other tunnels: no authentication, no confidentiality, possible spoofing and reflection attacks.

IPsec [RFC4301] for the transported IPv6 traffic is recommended.

The biggest threat to Teredo is probably for IPv4-only network as Teredo has been designed to easily traverse IPV4 NAT-PT devices which are quite often co-located with a stateful firewall. Therefore, if the stateful IPv4 firewall allows unrestricted UDP outbound and accept the return UDP traffic, then Teredo actually punches a hole in this firewall for all IPv6 traffic to the Internet and from the Internet. While host policies can be deployed to block Teredo in an IPv4-only network in order to avoid this firewall bypass, it would be more efficient to block all UDP outbound traffic at the IPv4 firewall if deemed possible (of course, at least port 53 should be left open for DNS traffic).

2.7.2.4. 6to4

6to4 tunnels [RFC3056] require a public routable IPv4 address in order to work correctly. They can be used to provide either one IPv6 host connectivity to the IPv6 Internet or multiple IPv6 networks connectivity to the IPv6 Internet. The 6to4 relay is usually the anycast address defined in RFC3068 [RFC3068] which has been deprecated by RFC7526 [RFC7526], and is no more used by recent Operating Systems. Some security considerations are explained in RFC3694 [RFC3964].

RFC6343 [RFC6343] points out that if an operator provides well-managed servers and relays for 6to4, non-encapsulated IPv6 packets will pass through well-defined points (the native IPv6 interfaces of those servers and relays) at which security mechanisms may be applied. Client usage of 6to4 by default is now discouraged, and significant precautions are needed to avoid operational problems.

2.7.2.5. 6rd

While 6rd tunnels share the same encapsulation as 6to4 tunnels (Section 2.7.2.4), they are designed to be used within a single SP domain, in other words they are deployed in a more constrained environment than 6to4 tunnels and have little security issues except

lack of confidentiality. The security considerations (Section 12) of RFC5969 [RFC5969] describes how to secure the 6rd tunnels.

IPsec [RFC4301] for the transported IPv6 traffic can be used if confidentiality is important.

2.7.2.6. 6PE and 6VPE

Organizations using MPLS in their core can also use 6PE [RFC4798] and 6VPE RFC4659 [RFC4659] to enable IPv6 access over MPLS. As 6PE and 6VPE are really similar to BGP/MPLS IP VPN described in RFC4364 [RFC4364], the security of these networks is also similar to the one described in RFC4381 [RFC4381]. It relies on:

- o Address space, routing and traffic separation with the help of VRF (only applicable to 6VPE);
- o Hiding the IPv4 core, hence removing all attacks against P-routers;
- o Securing the routing protocol between CE and PE, in the case of 6PE and 6VPE, link-local addresses (see [RFC7404]) can be used and as these addresses cannot be reached from outside of the link, the security of 6PE and 6VPE is even higher than the IPv4 BGP/MPLS IP VPN.

2.7.2.7. DS-Lite

DS-lite is more a translation mechanism and is therefore analyzed further (Section 2.7.3.3) in this document.

2.7.2.8. Mapping of Address and Port

With the tunnel and encapsulation versions of mapping of Address and Port (MAP-E [RFC7597] and MAP-T [RFC7599]), the access network is purely an IPv6 network and MAP protocols are used to give IPv4 hosts on the subscriber network, access to IPv4 hosts on the Internet. The subscriber router does stateful operations in order to map all internal IPv4 addresses and layer-4 ports to the IPv4 address and the set of layer-4 ports received through MAP configuration process. The SP equipment always does stateless operations (either decapsulation or stateless translation). Therefore, as opposed to Section 2.7.3.3 there is no state-exhaustion DoS attack against the SP equipment because there is no state and there is no operation caused by a new layer-4 connection (no logging operation).

The SP MAP equipment MUST implement all the security considerations of [RFC7597]; notably, ensuring that the mapping of the IPv4 address

and port are consistent with the configuration. As MAP has a predictable IPv4 address and port mapping, the audit logs are easier to manager.

2.7.3. Translation Mechanisms

Translation mechanisms between IPv4 and IPv6 networks are alternative coexistence strategies while networks transition to IPv6. While a framework is described in [RFC6144] the specific security considerations are documented in each individual mechanism. For the most part they specifically mention interference with IPsec or DNSSEC deployments, how to mitigate spoofed traffic and what some effective filtering strategies may be.

2.7.3.1. Carrier-Grade Nat (CGN)

Carrier-Grade NAT (CGN), also called NAT444 CGN or Large Scale NAT (LSN) or SP NAT is described in [RFC6264] and is utilized as an interim measure to prolong the use of IPv4 in a large service provider network until the provider can deploy an effective IPv6 solution. [RFC6598] requested a specific IANA allocated /10 IPv4 address block to be used as address space shared by all access networks using CGN. This has been allocated as 100.64.0.0/10.

Section 13 of [RFC6269] lists some specific security-related issues caused by large scale address sharing. The Security Considerations section of [RFC6598] also lists some specific mitigation techniques for potential misuse of shared address space.

RFC7422 [RFC7422] suggests the use of deterministic address mapping in order to reduce logging requirements for CGN. The idea is to have an algorithm mapping back and forth the internal subscriber to public ports.

2.7.3.2. NAT64/DNS64

Stateful NAT64 translation [RFC6146] allows IPv6-only clients to contact IPv4 servers using unicast UDP, TCP, or ICMP. It can be used in conjunction with DNS64 [RFC6147], a mechanism which synthesizes AAAA records from existing A records. There is also a stateless NAT64 [RFC6145] which is similar for the security aspects with the added benefit of being stateless, so, less prone to a state exhaustion attack.

The Security Consideration sections of [RFC6146] and [RFC6147] list the comprehensive issues. A specific issue with the use of NAT64 is that it will interfere with most IPsec deployments unless UDP

encapsulation is used. DNS64 has an incidence on DNSSEC see section 3.1 of [RFC7050].

2.7.3.3. DS-Lite

Dual-Stack Lite (DS-Lite) [RFC6333] is a transition technique that enables a service provider to share IPv4 addresses among customers by combining two well-known technologies: IP in IP (IPv4-in-IPv6) and Network Address and Port Translation (NAPT).

Security considerations with respect to DS-Lite mainly revolve around logging data, preventing DoS attacks from rogue devices (as the AFTR function is stateful) and restricting service offered by the AFTR only to registered customers.

Section 11 of [RFC6333] describes important security issues associated with this technology.

2.8. General Device Hardening

There are many environments which rely too much on the network infrastructure to disallow malicious traffic to get access to critical hosts. In new IPv6 deployments it has been common to see IPv6 traffic enabled but none of the typical access control mechanisms enabled for IPv6 device access. With the possibility of network device configuration mistakes and the growth of IPv6 in the overall Internet it is important to ensure that all individual devices are hardened against miscreant behavior.

The following guidelines should be used to ensure appropriate hardening of the host, be it an individual computer or router, firewall, load-balancer, server, etc device.

- o Restrict access to the device to authorized individuals
- o Monitor and audit access to the device
- o Turn off any unused services on the end node
- o Understand which IPv6 addresses are being used to source traffic and change defaults if necessary
- o Use cryptographically protected protocols for device management if possible (SCP, SNMPv3, SSH, TLS, etc)
- o Use host firewall capabilities to control traffic that gets processed by upper layer protocols

- o Use virus scanners to detect malicious programs

3. Enterprises Specific Security Considerations

Enterprises generally have robust network security policies in place to protect existing IPv4 networks. These policies have been distilled from years of experiential knowledge of securing IPv4 networks. At the very least, it is recommended that enterprise networks have parity between their security policies for both protocol versions.

Security considerations in the enterprise can be broadly categorized into two sections - External and Internal.

3.1. External Security Considerations:

The external aspect deals with providing security at the edge or perimeter of the enterprise network where it meets the service providers network. This is commonly achieved by enforcing a security policy either by implementing dedicated firewalls with stateful packet inspection or a router with ACLs. A common default IPv4 policy on firewalls that could easily be ported to IPv6 is to allow all traffic outbound while only allowing specific traffic, such as established sessions, inbound (see also [RFC6092]). Here are a few more things that could enhance the default policy:

- o Filter internal-use IPv6 addresses at the perimeter
- o Discard packets from and to bogon and reserved space, see also [CYMRU]
- o Accept certain ICMPv6 messages to allow proper operation of ND and PMTUD, see also [RFC4890]
- o Filter specific extension headers by accepting only the required ones (white list approach) such as ESP, AH (not forgetting the required transport layers: ICMP, TCP, UDP, ...) , where possible at the edge and possibly inside the perimeter; see also [I-D.gont-opsec-ipv6-eh-filtering]
- o Filter packets having an illegal IPv6 headers chain at the perimeter (and possible inside as well), see Section 2.2
- o Filter unneeded services at the perimeter
- o Implement anti-spoofing
- o Implement appropriate rate-limiters and control-plane policers

3.2. Internal Security Considerations:

The internal aspect deals with providing security inside the perimeter of the network, including the end host. The most significant concerns here are related to Neighbor Discovery. At the network level, it is recommended that all security considerations discussed in Section 2.3 be reviewed carefully and the recommendations be considered in-depth as well.

As mentioned in Section 2.6.2, care must be taken when running automated IPv6-in-IPv4 tunnels.

Hosts need to be hardened directly through security policy to protect against security threats. The host firewall default capabilities have to be clearly understood, especially 3rd party ones which can have different settings for IPv4 or IPv6 default permit/deny behavior. In some cases, 3rd party firewalls have no IPv6 support whereas the native firewall installed by default has it. General device hardening guidelines are provided in Section 2.8

It should also be noted that many hosts still use IPv4 for transport for things like RADIUS, TACACS+, SYSLOG, etc. This will require some extra level of due diligence on the part of the operator.

4. Service Providers Security Considerations

4.1. BGP

The threats and mitigation techniques are identical between IPv4 and IPv6. Broadly speaking they are:

- o Authenticating the TCP session;
- o TTL security (which becomes hop-limit security in IPv6);
- o Prefix Filtering.

These are explained in more detail in section Section 2.5.

4.1.1. Remote Triggered Black Hole Filtering

RTBH [RFC5635] works identically in IPv4 and IPv6. IANA has allocated 100::/64 as discard prefix RFC6666 [RFC6666].

4.2. Transition Mechanism

SP will typically use transition mechanisms such as 6rd, 6PE, MAP, DS-Lite which have been analyzed in the transition Section 2.7.2 section.

4.3. Lawful Intercept

The Lawful Intercept requirements are similar for IPv6 and IPv4 architectures and will be subject to the laws enforced in varying geographic regions. The local issues with each jurisdiction can make this challenging and both corporate legal and privacy personnel should be involved in discussions pertaining to what information gets logged and what the logging retention policies will be.

The target of interception will usually be a residential subscriber (e.g. his/her PPP session or physical line or CPE MAC address). With the absence of NAT on the CPE, IPv6 has the provision to allow for intercepting the traffic from a single host (a /128 target) rather than the whole set of hosts of a subscriber (which could be a /48, a /60 or /64).

In contrast, in mobile environments, since the 3GPP specifications allocate a /64 per device, it may be sufficient to intercept traffic from the /64 rather than specific /128's (since each time the device powers up it gets a new IID).

A sample architecture which was written for informational purposes is found in [RFC3924].

5. Residential Users Security Considerations

The IETF Homenet working group is working on how IPv6 residential network should be done; this obviously includes operational security considerations; but, this is still work in progress.

Residential users have usually less experience and knowledge about security or networking. As most of the recent hosts, smartphones, tablets have all IPv6 enabled by default, IPv6 security is important for those users. Even with an IPv4-only ISP, those users can get IPv6 Internet access with the help of Teredo tunnels. Several peer-to-peer programs (notably Bittorrent) support IPv6 and those programs can initiate a Teredo tunnel through the IPv4 residential gateway, with the consequence of making the internal host reachable from any IPv6 host on the Internet. It is therefore recommended that all host security products (personal firewall, ...) are configured with a dual-stack security policy.

If the Residential Gateway has IPv6 connectivity, [RFC7084] (which obsoletes [RFC6204]) defines the requirements of an IPv6 CPE and does not take position on the debate of default IPv6 security policy as defined in [RFC6092]:

- o outbound only: allowing all internally initiated connections and block all externally initiated ones, which is a common default security policy enforced by IPv4 Residential Gateway doing NAT-PT but it also breaks the end-to-end reachability promise of IPv6. [RFC6092] lists several recommendations to design such a CPE;
- o open/transparent: allowing all internally and externally initiated connections, therefore restoring the end-to-end nature of the Internet for the IPv6 traffic but having a different security policy for IPv6 than for IPv4.

[RFC6092] REC-49 states that a choice must be given to the user to select one of those two policies.

There is also an alternate solution which has been deployed notably by Swisscom ([I-D.ietf-v6ops-balanced-ipv6-security]: open to all outbound and inbound connections at the exception of an handful of TCP and UDP ports known as vulnerable.

6. Further Reading

There are several documents that describe in more details the security of an IPv6 network; these documents are not written by the IETF but are listed here for your convenience:

1. Guidelines for the Secure Deployment of IPv6 [NIST]
2. North American IPv6 Task Force Technology Report - IPv6 Security Technology Paper [NAv6TF_Security]
3. IPv6 Security [IPv6_Security_Book]

7. Acknowledgements

The authors would like to thank the following people for their useful comments: Mikael Abrahamsson, Fred Baker, Brian Carpenter, Tim Chown, Markus deBruen, Fernando Gont, Jeffry Handal, Panos Kampanakis, Erik Kline, Jouni Korhonen, Mark Lentczner, Bob Sleight, Tarko Tikan (by alphabetical order).

8. IANA Considerations

This memo includes no request to IANA.

9. Security Considerations

This memo attempts to give an overview of security considerations of operating an IPv6 network both in an IPv6-only network and in utilizing the most widely deployed IPv4/IPv6 coexistence strategies.

10. References

10.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<http://www.rfc-editor.org/info/rfc2119>>.
- [RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, DOI 10.17487/RFC2460, December 1998, <<http://www.rfc-editor.org/info/rfc2460>>.
- [RFC6104] Chown, T. and S. Venaas, "Rogue IPv6 Router Advertisement Problem Statement", RFC 6104, DOI 10.17487/RFC6104, February 2011, <<http://www.rfc-editor.org/info/rfc6104>>.
- [RFC6105] Levy-Abegnoli, E., Van de Velde, G., Popoviciu, C., and J. Mohacsi, "IPv6 Router Advertisement Guard", RFC 6105, DOI 10.17487/RFC6105, February 2011, <<http://www.rfc-editor.org/info/rfc6105>>.

10.2. Informative References

- [CYMRU] "Packet Filter and Route Filter Recommendation for IPv6 at xSP routers", <<http://www.team-cymru.org/ReadingRoom/Templates/IPv6Routers/xsp-recommendations.html>>.
- [I-D.chakrabarti-nordmark-6man-efficient-nd] Chakrabarti, S., Nordmark, E., Thubert, P., and M. Wasserman, "IPv6 Neighbor Discovery Optimizations for Wired and Wireless Networks", draft-chakrabarti-nordmark-6man-efficient-nd-07 (work in progress), February 2015.

- [I-D.gont-opsec-ipv6-eh-filtering]
Gont, F., Will, W., and R. Bonica, "Recommendations on Filtering of IPv6 Packets Containing IPv6 Extension Headers", draft-gont-opsec-ipv6-eh-filtering-02 (work in progress), August 2014.
- [I-D.ietf-6man-hbh-header-handling]
Baker, F. and R. Bonica, "IPv6 Hop-by-Hop Options Extension Header", draft-ietf-6man-hbh-header-handling-03 (work in progress), March 2016.
- [I-D.ietf-v6ops-balanced-ipv6-security]
Gysi, M., Leclanche, G., Vyncke, E., and R. Anfinson, "Balanced Security for IPv6 Residential CPE", draft-ietf-v6ops-balanced-ipv6-security-01 (work in progress), December 2013.
- [I-D.kampanakis-6man-ipv6-eh-parsing]
Kampanakis, P., "Implementation Guidelines for parsing IPv6 Extension Headers", draft-kampanakis-6man-ipv6-eh-parsing-01 (work in progress), August 2014.
- [I-D.thubert-savi-ra-throttler]
Thubert, P., "Throttling RAs on constrained interfaces", draft-thubert-savi-ra-throttler-01 (work in progress), June 2012.
- [IEEE-802.1X]
IEEE, , "IEEE Standard for Local and metropolitan area networks - Port-Based Network Access Control", IEEE Std 802.1X-2010, February 2010.
- [IPv6_Security_Book]
Hogg, and Vyncke, "IPv6 Security", ISBN 1-58705-594-5, Publisher CiscoPress, December 2008.
- [NAv6TF_Security]
Kao, , Green, , Bound, , and Pouffary, "North American IPv6 Task Force Technology Report - IPv6 Security Technology Paper", 2006,
<http://www.ipv6forum.com/dl/white/NAv6TF_Security_Report.pdf>.
- [NIST]
Frankel, , Graveman, , Pearce, , and Rooks, "Guidelines for the Secure Deployment of IPv6", 2010,
<<http://csrc.nist.gov/publications/nistpubs/800-119/sp800-119.pdf>>.

- [RFC0826] Plummer, D., "Ethernet Address Resolution Protocol: Or Converting Network Protocol Addresses to 48.bit Ethernet Address for Transmission on Ethernet Hardware", STD 37, RFC 826, DOI 10.17487/RFC0826, November 1982, <<http://www.rfc-editor.org/info/rfc826>>.
- [RFC2131] Droms, R., "Dynamic Host Configuration Protocol", RFC 2131, DOI 10.17487/RFC2131, March 1997, <<http://www.rfc-editor.org/info/rfc2131>>.
- [RFC2529] Carpenter, B. and C. Jung, "Transmission of IPv6 over IPv4 Domains without Explicit Tunnels", RFC 2529, DOI 10.17487/RFC2529, March 1999, <<http://www.rfc-editor.org/info/rfc2529>>.
- [RFC2740] Coltun, R., Ferguson, D., and J. Moy, "OSPF for IPv6", RFC 2740, DOI 10.17487/RFC2740, December 1999, <<http://www.rfc-editor.org/info/rfc2740>>.
- [RFC2784] Farinacci, D., Li, T., Hanks, S., Meyer, D., and P. Traina, "Generic Routing Encapsulation (GRE)", RFC 2784, DOI 10.17487/RFC2784, March 2000, <<http://www.rfc-editor.org/info/rfc2784>>.
- [RFC2827] Ferguson, P. and D. Senie, "Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing", BCP 38, RFC 2827, DOI 10.17487/RFC2827, May 2000, <<http://www.rfc-editor.org/info/rfc2827>>.
- [RFC2866] Rigney, C., "RADIUS Accounting", RFC 2866, DOI 10.17487/RFC2866, June 2000, <<http://www.rfc-editor.org/info/rfc2866>>.
- [RFC2993] Hain, T., "Architectural Implications of NAT", RFC 2993, DOI 10.17487/RFC2993, November 2000, <<http://www.rfc-editor.org/info/rfc2993>>.
- [RFC3056] Carpenter, B. and K. Moore, "Connection of IPv6 Domains via IPv4 Clouds", RFC 3056, DOI 10.17487/RFC3056, February 2001, <<http://www.rfc-editor.org/info/rfc3056>>.
- [RFC3068] Huitema, C., "An Anycast Prefix for 6to4 Relay Routers", RFC 3068, DOI 10.17487/RFC3068, June 2001, <<http://www.rfc-editor.org/info/rfc3068>>.

- [RFC3315] Droms, R., Ed., Bound, J., Volz, B., Lemon, T., Perkins, C., and M. Carney, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 3315, DOI 10.17487/RFC3315, July 2003, <<http://www.rfc-editor.org/info/rfc3315>>.
- [RFC3627] Savola, P., "Use of /127 Prefix Length Between Routers Considered Harmful", RFC 3627, DOI 10.17487/RFC3627, September 2003, <<http://www.rfc-editor.org/info/rfc3627>>.
- [RFC3756] Nikander, P., Ed., Kempf, J., and E. Nordmark, "IPv6 Neighbor Discovery (ND) Trust Models and Threats", RFC 3756, DOI 10.17487/RFC3756, May 2004, <<http://www.rfc-editor.org/info/rfc3756>>.
- [RFC3924] Baker, F., Foster, B., and C. Sharp, "Cisco Architecture for Lawful Intercept in IP Networks", RFC 3924, DOI 10.17487/RFC3924, October 2004, <<http://www.rfc-editor.org/info/rfc3924>>.
- [RFC3964] Savola, P. and C. Patel, "Security Considerations for 6to4", RFC 3964, DOI 10.17487/RFC3964, December 2004, <<http://www.rfc-editor.org/info/rfc3964>>.
- [RFC3971] Arkko, J., Ed., Kempf, J., Zill, B., and P. Nikander, "SEcure Neighbor Discovery (SEND)", RFC 3971, DOI 10.17487/RFC3971, March 2005, <<http://www.rfc-editor.org/info/rfc3971>>.
- [RFC3972] Aura, T., "Cryptographically Generated Addresses (CGA)", RFC 3972, DOI 10.17487/RFC3972, March 2005, <<http://www.rfc-editor.org/info/rfc3972>>.
- [RFC4193] Hinden, R. and B. Haberman, "Unique Local IPv6 Unicast Addresses", RFC 4193, DOI 10.17487/RFC4193, October 2005, <<http://www.rfc-editor.org/info/rfc4193>>.
- [RFC4293] Routhier, S., Ed., "Management Information Base for the Internet Protocol (IP)", RFC 4293, DOI 10.17487/RFC4293, April 2006, <<http://www.rfc-editor.org/info/rfc4293>>.
- [RFC4301] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, DOI 10.17487/RFC4301, December 2005, <<http://www.rfc-editor.org/info/rfc4301>>.
- [RFC4302] Kent, S., "IP Authentication Header", RFC 4302, DOI 10.17487/RFC4302, December 2005, <<http://www.rfc-editor.org/info/rfc4302>>.

- [RFC4303] Kent, S., "IP Encapsulating Security Payload (ESP)", RFC 4303, DOI 10.17487/RFC4303, December 2005, <<http://www.rfc-editor.org/info/rfc4303>>.
- [RFC4364] Rosen, E. and Y. Rekhter, "BGP/MPLS IP Virtual Private Networks (VPNs)", RFC 4364, DOI 10.17487/RFC4364, February 2006, <<http://www.rfc-editor.org/info/rfc4364>>.
- [RFC4380] Huitema, C., "Teredo: Tunneling IPv6 over UDP through Network Address Translations (NATs)", RFC 4380, DOI 10.17487/RFC4380, February 2006, <<http://www.rfc-editor.org/info/rfc4380>>.
- [RFC4381] Behringer, M., "Analysis of the Security of BGP/MPLS IP Virtual Private Networks (VPNs)", RFC 4381, DOI 10.17487/RFC4381, February 2006, <<http://www.rfc-editor.org/info/rfc4381>>.
- [RFC4443] Conta, A., Deering, S., and M. Gupta, Ed., "Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification", RFC 4443, DOI 10.17487/RFC4443, March 2006, <<http://www.rfc-editor.org/info/rfc4443>>.
- [RFC4552] Gupta, M. and N. Melam, "Authentication/Confidentiality for OSPFv3", RFC 4552, DOI 10.17487/RFC4552, June 2006, <<http://www.rfc-editor.org/info/rfc4552>>.
- [RFC4659] De Clercq, J., Ooms, D., Carugi, M., and F. Le Faucheur, "BGP-MPLS IP Virtual Private Network (VPN) Extension for IPv6 VPN", RFC 4659, DOI 10.17487/RFC4659, September 2006, <<http://www.rfc-editor.org/info/rfc4659>>.
- [RFC4798] De Clercq, J., Ooms, D., Prevost, S., and F. Le Faucheur, "Connecting IPv6 Islands over IPv4 MPLS Using IPv6 Provider Edge Routers (6PE)", RFC 4798, DOI 10.17487/RFC4798, February 2007, <<http://www.rfc-editor.org/info/rfc4798>>.
- [RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, DOI 10.17487/RFC4861, September 2007, <<http://www.rfc-editor.org/info/rfc4861>>.
- [RFC4864] Van de Velde, G., Hain, T., Droms, R., Carpenter, B., and E. Klein, "Local Network Protection for IPv6", RFC 4864, DOI 10.17487/RFC4864, May 2007, <<http://www.rfc-editor.org/info/rfc4864>>.

- [RFC4890] Davies, E. and J. Mohacsi, "Recommendations for Filtering ICMPv6 Messages in Firewalls", RFC 4890, DOI 10.17487/RFC4890, May 2007, <<http://www.rfc-editor.org/info/rfc4890>>.
- [RFC4941] Narten, T., Draves, R., and S. Krishnan, "Privacy Extensions for Stateless Address Autoconfiguration in IPv6", RFC 4941, DOI 10.17487/RFC4941, September 2007, <<http://www.rfc-editor.org/info/rfc4941>>.
- [RFC4942] Davies, E., Krishnan, S., and P. Savola, "IPv6 Transition/Co-existence Security Considerations", RFC 4942, DOI 10.17487/RFC4942, September 2007, <<http://www.rfc-editor.org/info/rfc4942>>.
- [RFC5157] Chown, T., "IPv6 Implications for Network Scanning", RFC 5157, DOI 10.17487/RFC5157, March 2008, <<http://www.rfc-editor.org/info/rfc5157>>.
- [RFC5214] Templin, F., Gleeson, T., and D. Thaler, "Intra-Site Automatic Tunnel Addressing Protocol (ISATAP)", RFC 5214, DOI 10.17487/RFC5214, March 2008, <<http://www.rfc-editor.org/info/rfc5214>>.
- [RFC5340] Coltun, R., Ferguson, D., Moy, J., and A. Lindem, "OSPF for IPv6", RFC 5340, DOI 10.17487/RFC5340, July 2008, <<http://www.rfc-editor.org/info/rfc5340>>.
- [RFC5635] Kumari, W. and D. McPherson, "Remote Triggered Black Hole Filtering with Unicast Reverse Path Forwarding (uRPF)", RFC 5635, DOI 10.17487/RFC5635, August 2009, <<http://www.rfc-editor.org/info/rfc5635>>.
- [RFC5952] Kawamura, S. and M. Kawashima, "A Recommendation for IPv6 Address Text Representation", RFC 5952, DOI 10.17487/RFC5952, August 2010, <<http://www.rfc-editor.org/info/rfc5952>>.
- [RFC5969] Townsley, W. and O. Troan, "IPv6 Rapid Deployment on IPv4 Infrastructures (6rd) -- Protocol Specification", RFC 5969, DOI 10.17487/RFC5969, August 2010, <<http://www.rfc-editor.org/info/rfc5969>>.
- [RFC6092] Woodyatt, J., Ed., "Recommended Simple Security Capabilities in Customer Premises Equipment (CPE) for Providing Residential IPv6 Internet Service", RFC 6092, DOI 10.17487/RFC6092, January 2011, <<http://www.rfc-editor.org/info/rfc6092>>.

- [RFC6144] Baker, F., Li, X., Bao, C., and K. Yin, "Framework for IPv4/IPv6 Translation", RFC 6144, DOI 10.17487/RFC6144, April 2011, <<http://www.rfc-editor.org/info/rfc6144>>.
- [RFC6145] Li, X., Bao, C., and F. Baker, "IP/ICMP Translation Algorithm", RFC 6145, DOI 10.17487/RFC6145, April 2011, <<http://www.rfc-editor.org/info/rfc6145>>.
- [RFC6146] Bagnulo, M., Matthews, P., and I. van Beijnum, "Stateful NAT64: Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers", RFC 6146, DOI 10.17487/RFC6146, April 2011, <<http://www.rfc-editor.org/info/rfc6146>>.
- [RFC6147] Bagnulo, M., Sullivan, A., Matthews, P., and I. van Beijnum, "DNS64: DNS Extensions for Network Address Translation from IPv6 Clients to IPv4 Servers", RFC 6147, DOI 10.17487/RFC6147, April 2011, <<http://www.rfc-editor.org/info/rfc6147>>.
- [RFC6164] Kohno, M., Nitzan, B., Bush, R., Matsuzaki, Y., Colitti, L., and T. Narten, "Using 127-Bit IPv6 Prefixes on Inter-Router Links", RFC 6164, DOI 10.17487/RFC6164, April 2011, <<http://www.rfc-editor.org/info/rfc6164>>.
- [RFC6169] Krishnan, S., Thaler, D., and J. Hoagland, "Security Concerns with IP Tunneling", RFC 6169, DOI 10.17487/RFC6169, April 2011, <<http://www.rfc-editor.org/info/rfc6169>>.
- [RFC6192] Dugal, D., Pignataro, C., and R. Dunn, "Protecting the Router Control Plane", RFC 6192, DOI 10.17487/RFC6192, March 2011, <<http://www.rfc-editor.org/info/rfc6192>>.
- [RFC6204] Singh, H., Beebee, W., Donley, C., Stark, B., and O. Troan, Ed., "Basic Requirements for IPv6 Customer Edge Routers", RFC 6204, DOI 10.17487/RFC6204, April 2011, <<http://www.rfc-editor.org/info/rfc6204>>.
- [RFC6221] Miles, D., Ed., Ooghe, S., Dec, W., Krishnan, S., and A. Kavanagh, "Lightweight DHCPv6 Relay Agent", RFC 6221, DOI 10.17487/RFC6221, May 2011, <<http://www.rfc-editor.org/info/rfc6221>>.
- [RFC6264] Jiang, S., Guo, D., and B. Carpenter, "An Incremental Carrier-Grade NAT (CGN) for IPv6 Transition", RFC 6264, DOI 10.17487/RFC6264, June 2011, <<http://www.rfc-editor.org/info/rfc6264>>.

- [RFC6269] Ford, M., Ed., Boucadair, M., Durand, A., Levis, P., and P. Roberts, "Issues with IP Address Sharing", RFC 6269, DOI 10.17487/RFC6269, June 2011, <<http://www.rfc-editor.org/info/rfc6269>>.
- [RFC6296] Wasserman, M. and F. Baker, "IPv6-to-IPv6 Network Prefix Translation", RFC 6296, DOI 10.17487/RFC6296, June 2011, <<http://www.rfc-editor.org/info/rfc6296>>.
- [RFC6302] Durand, A., Gashinsky, I., Lee, D., and S. Sheppard, "Logging Recommendations for Internet-Facing Servers", BCP 162, RFC 6302, DOI 10.17487/RFC6302, June 2011, <<http://www.rfc-editor.org/info/rfc6302>>.
- [RFC6324] Nakibly, G. and F. Templin, "Routing Loop Attack Using IPv6 Automatic Tunnels: Problem Statement and Proposed Mitigations", RFC 6324, DOI 10.17487/RFC6324, August 2011, <<http://www.rfc-editor.org/info/rfc6324>>.
- [RFC6333] Durand, A., Droms, R., Woodyatt, J., and Y. Lee, "Dual-Stack Lite Broadband Deployments Following IPv4 Exhaustion", RFC 6333, DOI 10.17487/RFC6333, August 2011, <<http://www.rfc-editor.org/info/rfc6333>>.
- [RFC6343] Carpenter, B., "Advisory Guidelines for 6to4 Deployment", RFC 6343, DOI 10.17487/RFC6343, August 2011, <<http://www.rfc-editor.org/info/rfc6343>>.
- [RFC6434] Jankiewicz, E., Loughney, J., and T. Narten, "IPv6 Node Requirements", RFC 6434, DOI 10.17487/RFC6434, December 2011, <<http://www.rfc-editor.org/info/rfc6434>>.
- [RFC6459] Korhonen, J., Ed., Soininen, J., Patil, B., Savolainen, T., Bajko, G., and K. Iisakkila, "IPv6 in 3rd Generation Partnership Project (3GPP) Evolved Packet System (EPS)", RFC 6459, DOI 10.17487/RFC6459, January 2012, <<http://www.rfc-editor.org/info/rfc6459>>.
- [RFC6506] Bhatia, M., Manral, V., and A. Lindem, "Supporting Authentication Trailer for OSPFv3", RFC 6506, DOI 10.17487/RFC6506, February 2012, <<http://www.rfc-editor.org/info/rfc6506>>.
- [RFC6547] George, W., "RFC 3627 to Historic Status", RFC 6547, DOI 10.17487/RFC6547, February 2012, <<http://www.rfc-editor.org/info/rfc6547>>.

- [RFC6564] Krishnan, S., Woodyatt, J., Kline, E., Hoagland, J., and M. Bhatia, "A Uniform Format for IPv6 Extension Headers", RFC 6564, DOI 10.17487/RFC6564, April 2012, <<http://www.rfc-editor.org/info/rfc6564>>.
- [RFC6583] Gashinsky, I., Jaeggli, J., and W. Kumari, "Operational Neighbor Discovery Problems", RFC 6583, DOI 10.17487/RFC6583, March 2012, <<http://www.rfc-editor.org/info/rfc6583>>.
- [RFC6598] Weil, J., Kuarsingh, V., Donley, C., Liljenstolpe, C., and M. Azinger, "IANA-Reserved IPv4 Prefix for Shared Address Space", BCP 153, RFC 6598, DOI 10.17487/RFC6598, April 2012, <<http://www.rfc-editor.org/info/rfc6598>>.
- [RFC6620] Nordmark, E., Bagnulo, M., and E. Levy-Abegnoli, "FCFS SAVI: First-Come, First-Served Source Address Validation Improvement for Locally Assigned IPv6 Addresses", RFC 6620, DOI 10.17487/RFC6620, May 2012, <<http://www.rfc-editor.org/info/rfc6620>>.
- [RFC6666] Hilliard, N. and D. Freedman, "A Discard Prefix for IPv6", RFC 6666, DOI 10.17487/RFC6666, August 2012, <<http://www.rfc-editor.org/info/rfc6666>>.
- [RFC6762] Cheshire, S. and M. Krochmal, "Multicast DNS", RFC 6762, DOI 10.17487/RFC6762, February 2013, <<http://www.rfc-editor.org/info/rfc6762>>.
- [RFC6763] Cheshire, S. and M. Krochmal, "DNS-Based Service Discovery", RFC 6763, DOI 10.17487/RFC6763, February 2013, <<http://www.rfc-editor.org/info/rfc6763>>.
- [RFC6810] Bush, R. and R. Austein, "The Resource Public Key Infrastructure (RPKI) to Router Protocol", RFC 6810, DOI 10.17487/RFC6810, January 2013, <<http://www.rfc-editor.org/info/rfc6810>>.
- [RFC6964] Templin, F., "Operational Guidance for IPv6 Deployment in IPv4 Sites Using the Intra-Site Automatic Tunnel Addressing Protocol (ISATAP)", RFC 6964, DOI 10.17487/RFC6964, May 2013, <<http://www.rfc-editor.org/info/rfc6964>>.
- [RFC6980] Gont, F., "Security Implications of IPv6 Fragmentation with IPv6 Neighbor Discovery", RFC 6980, DOI 10.17487/RFC6980, August 2013, <<http://www.rfc-editor.org/info/rfc6980>>.

- [RFC7011] Claise, B., Ed., Trammell, B., Ed., and P. Aitken, "Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of Flow Information", STD 77, RFC 7011, DOI 10.17487/RFC7011, September 2013, <<http://www.rfc-editor.org/info/rfc7011>>.
- [RFC7012] Claise, B., Ed. and B. Trammell, Ed., "Information Model for IP Flow Information Export (IPFIX)", RFC 7012, DOI 10.17487/RFC7012, September 2013, <<http://www.rfc-editor.org/info/rfc7012>>.
- [RFC7039] Wu, J., Bi, J., Bagnulo, M., Baker, F., and C. Vogt, Ed., "Source Address Validation Improvement (SAVI) Framework", RFC 7039, DOI 10.17487/RFC7039, October 2013, <<http://www.rfc-editor.org/info/rfc7039>>.
- [RFC7045] Carpenter, B. and S. Jiang, "Transmission and Processing of IPv6 Extension Headers", RFC 7045, DOI 10.17487/RFC7045, December 2013, <<http://www.rfc-editor.org/info/rfc7045>>.
- [RFC7050] Savolainen, T., Korhonen, J., and D. Wing, "Discovery of the IPv6 Prefix Used for IPv6 Address Synthesis", RFC 7050, DOI 10.17487/RFC7050, November 2013, <<http://www.rfc-editor.org/info/rfc7050>>.
- [RFC7084] Singh, H., Beebe, W., Donley, C., and B. Stark, "Basic Requirements for IPv6 Customer Edge Routers", RFC 7084, DOI 10.17487/RFC7084, November 2013, <<http://www.rfc-editor.org/info/rfc7084>>.
- [RFC7112] Gont, F., Manral, V., and R. Bonica, "Implications of Oversized IPv6 Header Chains", RFC 7112, DOI 10.17487/RFC7112, January 2014, <<http://www.rfc-editor.org/info/rfc7112>>.
- [RFC7113] Gont, F., "Implementation Advice for IPv6 Router Advertisement Guard (RA-Guard)", RFC 7113, DOI 10.17487/RFC7113, February 2014, <<http://www.rfc-editor.org/info/rfc7113>>.
- [RFC7166] Bhatia, M., Manral, V., and A. Lindem, "Supporting Authentication Trailer for OSPFv3", RFC 7166, DOI 10.17487/RFC7166, March 2014, <<http://www.rfc-editor.org/info/rfc7166>>.

- [RFC7217] Gont, F., "A Method for Generating Semantically Opaque Interface Identifiers with IPv6 Stateless Address Autoconfiguration (SLAAC)", RFC 7217, DOI 10.17487/RFC7217, April 2014, <<http://www.rfc-editor.org/info/rfc7217>>.
- [RFC7381] Chittimaneni, K., Chown, T., Howard, L., Kuarsingh, V., Pouffary, Y., and E. Vyncke, "Enterprise IPv6 Deployment Guidelines", RFC 7381, DOI 10.17487/RFC7381, October 2014, <<http://www.rfc-editor.org/info/rfc7381>>.
- [RFC7404] Behringer, M. and E. Vyncke, "Using Only Link-Local Addressing inside an IPv6 Network", RFC 7404, DOI 10.17487/RFC7404, November 2014, <<http://www.rfc-editor.org/info/rfc7404>>.
- [RFC7422] Donley, C., Grundemann, C., Sarawat, V., Sundaresan, K., and O. Vautrin, "Deterministic Address Mapping to Reduce Logging in Carrier-Grade NAT Deployments", RFC 7422, DOI 10.17487/RFC7422, December 2014, <<http://www.rfc-editor.org/info/rfc7422>>.
- [RFC7454] Durand, J., Pepelnjak, I., and G. Doering, "BGP Operations and Security", BCP 194, RFC 7454, DOI 10.17487/RFC7454, February 2015, <<http://www.rfc-editor.org/info/rfc7454>>.
- [RFC7513] Bi, J., Wu, J., Yao, G., and F. Baker, "Source Address Validation Improvement (SAVI) Solution for DHCP", RFC 7513, DOI 10.17487/RFC7513, May 2015, <<http://www.rfc-editor.org/info/rfc7513>>.
- [RFC7526] Troan, O. and B. Carpenter, Ed., "Deprecating the Anycast Prefix for 6to4 Relay Routers", BCP 196, RFC 7526, DOI 10.17487/RFC7526, May 2015, <<http://www.rfc-editor.org/info/rfc7526>>.
- [RFC7597] Troan, O., Ed., Dec, W., Li, X., Bao, C., Matsushima, S., Murakami, T., and T. Taylor, Ed., "Mapping of Address and Port with Encapsulation (MAP-E)", RFC 7597, DOI 10.17487/RFC7597, July 2015, <<http://www.rfc-editor.org/info/rfc7597>>.
- [RFC7599] Li, X., Bao, C., Dec, W., Ed., Troan, O., Matsushima, S., and T. Murakami, "Mapping of Address and Port using Translation (MAP-T)", RFC 7599, DOI 10.17487/RFC7599, July 2015, <<http://www.rfc-editor.org/info/rfc7599>>.

- [RFC7610] Gont, F., Liu, W., and G. Van de Velde, "DHCPv6-Shield: Protecting against Rogue DHCPv6 Servers", BCP 199, RFC 7610, DOI 10.17487/RFC7610, August 2015, <<http://www.rfc-editor.org/info/rfc7610>>.
- [RFC7707] Gont, F. and T. Chown, "Network Reconnaissance in IPv6 Networks", RFC 7707, DOI 10.17487/RFC7707, March 2016, <<http://www.rfc-editor.org/info/rfc7707>>.
- [RFC7721] Cooper, A., Gont, F., and D. Thaler, "Security and Privacy Considerations for IPv6 Address Generation Mechanisms", RFC 7721, DOI 10.17487/RFC7721, March 2016, <<http://www.rfc-editor.org/info/rfc7721>>.
- [RFC7872] Gont, F., Linkova, J., Chown, T., and W. Liu, "Observations on the Dropping of Packets with IPv6 Extension Headers in the Real World", RFC 7872, DOI 10.17487/RFC7872, June 2016, <<http://www.rfc-editor.org/info/rfc7872>>.
- [RFC8020] Bortzmeyer, S. and S. Huque, "NXDOMAIN: There Really Is Nothing Underneath", RFC 8020, DOI 10.17487/RFC8020, November 2016, <<http://www.rfc-editor.org/info/rfc8020>>.
- [SCANNING] "Mapping the Great Void - Smarter scanning for IPv6", <http://www.caida.org/workshops/isma/1202/slides/aims1202_rbarnes.pdf>.

Authors' Addresses

Kiran K. Chittimaneni
Dropbox Inc.
185 Berry Street, Suite 400
San Francisco, CA 94107
USA

Email: kk@dropbox.com

Merike Kaeo
Double Shot Security
3518 Fremont Ave N 363
Seattle 98103
USA

Phone: +12066696394
Email: merike@doubleshotsecurity.com

Eric Vyncke (editor)
Cisco
De Kleetlaan 6a
Diegem 1831
Belgium

Phone: +32 2 778 4677
Email: evyncke@cisco.com

OPSEC
Internet-Draft
Intended status: Informational
Expires: November 7, 2021

E. Vyncke
Cisco
K. Chittimaneni
Square
M. Kaeo
Double Shot Security
E. Rey
ERNW
May 6, 2021

Operational Security Considerations for IPv6 Networks
draft-ietf-opsec-v6-27

Abstract

Knowledge and experience on how to operate IPv4 networks securely is available: whether it is an Internet Service Provider or an enterprise internal network. However, IPv6 presents some new security challenges. RFC 4942 describes security issues in the protocol, but network managers also need a more practical, operations-minded document to enumerate advantages and/or disadvantages of certain choices.

This document analyzes the operational security issues associated with several types of network and proposes technical and procedural mitigation techniques. This document is only applicable to managed networks, such as enterprise networks, service provider networks, or managed residential networks.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on November 7, 2021.

Copyright Notice

Copyright (c) 2021 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	3
1.1. Applicability Statement	4
2. Generic Security Considerations	4
2.1. Addressing	4
2.1.1. Use of ULAs	5
2.1.2. Point-to-Point Links	5
2.1.3. Loopback Addresses	5
2.1.4. Stable Addresses	6
2.1.5. Temporary Addresses for SLAAC	6
2.1.6. DHCP Considerations	8
2.1.7. DNS Considerations	8
2.1.8. Using a /64 per host	8
2.1.9. Privacy consideration of Addresses	8
2.2. Extension Headers	9
2.2.1. Order and Repetition of Extension Headers	9
2.2.2. Hop-by-Hop Options Header	10
2.2.3. Fragment Header	10
2.2.4. IP Security Extension Header	10
2.3. Link-Layer Security	11
2.3.1. Neighbor Solicitation Rate-Limiting	11
2.3.2. Router and Neighbor Advertisements Filtering	12
2.3.3. Securing DHCP	13
2.3.4. 3GPP Link-Layer Security	14
2.3.5. Impact of Multicast Traffic	15
2.3.6. SeND and CGA	15
2.4. Control Plane Security	16
2.4.1. Control Protocols	17
2.4.2. Management Protocols	18
2.4.3. Packet Exceptions	18
2.5. Routing Security	19
2.5.1. BGP Security	20

2.5.2.	Authenticating OSPFv3 Neighbors	20
2.5.3.	Securing Routing Updates	21
2.5.4.	Route Filtering	21
2.6.	Logging/Monitoring	21
2.6.1.	Data Sources	23
2.6.2.	Use of Collected Data	26
2.6.3.	Summary	29
2.7.	Transition/Coexistence Technologies	29
2.7.1.	Dual Stack	30
2.7.2.	Encapsulation Mechanisms	31
2.7.3.	Translation Mechanisms	35
2.8.	General Device Hardening	37
3.	Enterprises Specific Security Considerations	37
3.1.	External Security Considerations	38
3.2.	Internal Security Considerations	39
4.	Service Providers Security Considerations	40
4.1.	BGP	40
4.1.1.	Remote Triggered Black Hole Filtering (RTBH)	40
4.2.	Transition/Coexistence Mechanism	40
4.3.	Lawful Intercept	40
5.	Residential Users Security Considerations	41
6.	Further Reading	41
7.	Acknowledgements	42
8.	Security Considerations	42
9.	References	42
9.1.	Normative References	42
9.2.	Informative References	42
	Authors' Addresses	57

1. Introduction

Running an IPv6 network is new for most operators not only because they are not yet used to large-scale IPv6 networks but also because there are subtle but critical and important differences between IPv4 and IPv6, especially with respect to security. For example, all layer-2 interactions are now done using Neighbor Discovery Protocol [RFC4861] rather than using Address Resolution Protocol [RFC0826]. Also, there is no Network Address Port Translation (NAPT) defined in [RFC2663] for IPv6 even if [RFC6296] specifies a Network Prefix Translation for IPv6 (NPTv6) which is a 1-to-1 mapping of IPv6 addresses. Another important difference is that IPv6 is extensible with the use of extension headers.

IPv6 networks are deployed using a variety of techniques, each of which have their own specific security concerns.

This document complements [RFC4942] by listing security issues when operating a network (including various transition technologies). It

also provides more recent operational deployment experiences where warranted.

1.1. Applicability Statement

This document is applicable to managed networks, i.e., when the network is operated by the user organization itself. Indeed, many of the recommended mitigation techniques must be configured with detailed knowledge of the network (which are the default routers, the switch trunk ports, etc.). This covers Service Provider (SP), enterprise networks and some knowledgeable-home-user-managed residential networks. This applicability statement especially applies to Section 2.3 and Section 2.5.4.

2. Generic Security Considerations

2.1. Addressing

IPv6 address allocations and overall architecture are an important part of securing IPv6. Initial designs, even if intended to be temporary, tend to last much longer than expected. Although initially IPv6 was thought to make renumbering easy, in practice it may be extremely difficult to renumber without a proper IP Address Management (IPAM) system. [RFC7010] introduces the mechanisms that could be utilized for IPv6 site renumbering and tries to cover most of the explicit issues and requirements associated with IPv6 renumbering.

A key task for a successful IPv6 deployment is to prepare an addressing plan. Because an abundance of address space is available, structuring an address plan around both services and geographic locations allows address space to become a basis for more structured security policies to permit or deny services between geographic regions. [RFC6177] documents some operational considerations of using different prefix sizes for address assignments at end sites.

A common question is whether companies should use Provider Independent (PI) vs. Provider Allocated (PA) space [RFC7381], but from a security perspective there is little difference. However, one aspect to keep in mind is who has administrative ownership of the address space and who is technically responsible if/when there is a need to enforce restrictions on routability of the space, e.g., due to malicious criminal activity originating from it. Relying on PA address space may also increase the perceived need for address translation techniques such as NPTv6 and thereby augmenting the complexity of the operations including the security operations.

In [RFC7934], it is recommended that IPv6 network deployments provide multiple IPv6 addresses from each prefix to general-purpose hosts and it specifically does not recommend limiting a host to only one IPv6 address per prefix. It also recommends that the network give the host the ability to use new addresses without requiring explicit requests (for example by using SLAAC). Privacy Extensions as of [RFC8981] constitute one of the main scenarios where hosts are expected to generate multiple addresses from the same prefix and having multiple IPv6 addresses per interface is a major change compared to the unique IPv4 address per interface for hosts (secondary IPv4 addresses are not common); especially for audits (see section Section 2.6.2.3).

2.1.1. Use of ULAs

Unique Local Addresses (ULAs) [RFC4193] are intended for scenarios where interfaces are not globally reachable, despite being routed within a domain. They formally have global scope, but [RFC4193] specifies that they must be filtered at domain boundaries. ULAs are different from [RFC1918] addresses and have different use cases. One use of ULA is described in [RFC4864], another one is for internal communication stability in networks where external connectivity may come and go (e.g., some ISPs provide ULAs in home networks connected via a cable modem). It should further be kept in mind that ULA /48s from the fd00::/8 space (L=1) MUST be generated with a pseudo-random algorithm, per [RFC4193] section 3.2.1.

2.1.2. Point-to-Point Links

[RFC6164] in section 5.1 specifies the rationale of using /127 for inter-router point-to-point links to prevent the ping-pong issue between routers not correctly implementing [RFC4443] and also prevents a DoS attack on the neighbor cache. The previous recommendation of [RFC3627] has been obsoleted and marked Historic by [RFC6547]).

Some environments are also using link-local addressing for point-to-point links. While this practice could further reduce the attack surface of infrastructure devices, the operational disadvantages also need to be carefully considered; see also [RFC7404].

2.1.3. Loopback Addresses

Many operators reserve a /64 block for all loopback addresses in their infrastructure and allocate a /128 out of this reserved /64 prefix for each loopback interface. This practice facilitates configuration of Access Control List (ACL) rules to enforce a security policy for those loopback addresses.

2.1.4. Stable Addresses

When considering how to assign stable addresses for nodes (either by static configuration or by pre-provisioned DHCPv6 lease Section 2.1.6), it is necessary to take into consideration the effectiveness of perimeter security in a given environment.

There is a trade-off between ease of operation (where some portions of the IPv6 address could be easily recognizable for operational debugging and troubleshooting) versus the risk of trivial scanning used for reconnaissance. [SCANNING] shows that there are scientifically based mechanisms that make scanning for IPv6 reachable nodes more feasible than expected; see also [RFC7707].

Stable addresses also allow easy enforcement of a security policy at the perimeter based on IPv6 addresses. E.g., Manufacturer Usage Description (MUD) [RFC8520] is a mechanism where the perimeter defense can retrieve security policy template based on the type of internal device and apply the right security policy based on the device IPv6 address.

The use of well-known IPv6 addresses (such as ff02::1 for all link-local nodes) or the use of commonly repeated addresses could make it easy to figure out which devices are name servers, routers, or other critical devices; even a simple traceroute will expose most of the routers on a path. There are many scanning techniques possible and operators should not rely on the 'impossible to find because my address is random' paradigm (a.k.a. "security by obscurity"), even if it is common practice to have the stable addresses randomly distributed across /64 subnets and to always use DNS (as IPv6 addresses are hard for human brains to remember).

While in some environments obfuscating addresses could be considered an added benefit, it should not preclude enforcement of perimeter rules. Stable addresses following some logical allocation scheme may ease the operation (as simplicity always helps security).

Typical deployments will have a mix of stable and non-stable addresses; the stable addresses being either predictable (e.g., ::25 for a mail server) or obfuscated (i.e., appearing as a random 64-bit number).

2.1.5. Temporary Addresses for SLAAC

Historically, stateless address autoconfiguration (SLAAC) makes up the globally unique IPv6 address based on an automatically generated 64-bit interface identifier (IID) based on the EUI-64 MAC address combined with the /64 prefix (received in the Prefix Information

Option (PIO) of the Router Advertisement (RA)). The EUI-64 address is generated from the stable 48-bit MAC address and does not change even if the host moves to another network; this is of course bad for privacy as a host can be traced from network (home) to network (office or Wi-Fi in hotels). [RFC8064] recommends against the use of EUI-64 addresses; and it must be noted that most host operating systems do not use EUI-64 addresses anymore and rely on either [RFC8981] or [RFC8064].

Randomly generating an interface ID, as described in [RFC8981], is part of SLAAC with so-called privacy extension addresses and is used to address some privacy concerns. Privacy extension addresses, a.k.a., temporary addresses may help to mitigate the correlation of activities of a node within the same network and may also reduce the attack exposure window. But using [RFC8981] privacy extension addresses might prevent the operator from building host specific access control lists (ACLs). The [RFC8981] privacy extension addresses could also be used to obfuscate some malevolent activities and specific user attribution/accountability procedures should be put in place as described in Section 2.6.

[RFC8064] combined with the address generation mechanism of [RFC7217] specifies another way to generate an address while still keeping the same IID for each network prefix; this allows SLAAC nodes to always have the same stable IPv6 address on a specific network while having different IPv6 addresses on different networks.

In some specific use cases where user accountability is more important than user privacy, network operators may consider disabling SLAAC and relying only on DHCPv6; but not all operating systems support DHCPv6 so some hosts will not get any IPv6 connectivity. Disabling SLAAC and privacy extension addresses can be done for most operating systems by sending RA messages with a hint to get addresses via DHCPv6 by setting the M-bit and disabling SLAAC by resetting all A-bits in all prefix information options. However, attackers could still find ways to bypass this mechanism if not enforced at the switch/router level.

However, in scenarios where anonymity is a strong desire (protecting user privacy is more important than user attribution), privacy extension addresses should be used. When mechanisms recommended by [RFC8064] are available, the stable privacy address is probably a good balance between privacy (among different networks) and security/user attribution (within a network).

2.1.6. DHCP Considerations

Some environments use DHCPv6 to provision addresses and other parameters in order to ensure auditability and traceability (see Section 2.6.1.5 for the limitations of DHCPv6 for auditability).

A main security concern is the ability to detect and counteract rogue DHCP servers (Section 2.3.3). It must be noted that as opposed to DHCPv4, DHCPv6 can lease several IPv6 addresses per client. For DHCPv4, the lease is bound to the 'client identifier', which may contain a hardware address, or it may contain another type of identifier, such as a DNS name. For DHCPv6, the lease is bound to the client DHCP Unique ID (DUID), which may, or may not, be bound to the client link-layer address. [RFC7824] describes the privacy issues associated with the use of DHCPv6 by Internet users. The anonymity profiles [RFC7844] are designed for clients that wish to remain anonymous to the visited network. [RFC7707] recommends that DHCPv6 servers issue addresses randomly from a large pool.

2.1.7. DNS Considerations

While the security concerns of DNS are not fundamentally different between IPv4 and IPv6, there are specific considerations in DNS64 [RFC6147] environments that need to be understood. Specifically, the interactions and the potential of interference with DNSSEC ([RFC4033]) implementation need to be understood - these are pointed out in more detail in Section 2.7.3.2.

2.1.8. Using a /64 per host

An interesting approach is using a /64 per host as proposed in [RFC8273] especially in a shared environment. This allows for easier user attribution (typically based on the host MAC address) as its /64 prefix is stable even if applications within the host can change their IPv6 address within this /64 prefix.

This can also be useful for the generation of ACLs once individual systems (e.g. admin workstations) have their own prefixes.

2.1.9. Privacy consideration of Addresses

Beside the security aspects of IPv6 addresses, there are also privacy considerations: mainly because they are of global scope and visible globally. [RFC7721] goes into more detail on the privacy considerations for IPv6 addresses by comparing the manually configured IPv6 address, DHCPv6, and SLAAC.

2.2. Extension Headers

Extension headers are an important difference between IPv4 and IPv6. In IPv4-based packets, it's trivial to find the upper-layer protocol type and protocol header, while in IPv6 it is more complex since the extension header chain must be parsed completely (even if not processed) in order to find the upper-layer protocol header. IANA has closed the existing empty "Next Header Types" registry to new entries and is redirecting its users to a new "IPv6 Extension Header Types" registry per [RFC7045].

Extension headers have also become a very controversial topic since forwarding nodes that discard packets containing extension headers are known to cause connectivity failures and deployment problems [RFC7872]. Understanding the role of various extension headers is important and this section enumerates the ones that need careful consideration.

A clarification on how intermediate nodes should handle packets with existing or future extension headers is found in [RFC7045]. The uniform TLV format to be used for defining future extension headers is described in [RFC6564]. Sections 5.2 and 5.3 of [RFC8504] provide more information on the processing of extension headers by IPv6 nodes.

Vendors of filtering solutions and operations personnel responsible for implementing packet filtering rules should be aware that the 'Next Header' field in an IPv6 header can both point to an IPv6 extension header or to an upper layer protocol header. This has to be considered when designing the user interface of filtering solutions or during the creation of filtering rule sets.

There is IETF work in progress regarding filtering rules for those extension headers: [I-D.ietf-opsec-ipv6-eh-filtering] for transit routers.

2.2.1. Order and Repetition of Extension Headers

While [RFC8200] recommends the order and the maximum repetition of extension headers, there are still IPv6 implementations, at the time of writing, which support a non-recommended order of headers (such as ESP before routing) or an illegal repetition of headers (such as multiple routing headers). The same applies for options contained in the extension headers (see [I-D.kampanakis-6man-ipv6-eh-parsing]). In some cases, it has led to nodes crashing when receiving or forwarding wrongly formatted packets.

A firewall or edge device should be used to enforce the recommended order and the maximum occurrences of extension headers by dropping non-conforming packets.

2.2.2. Hop-by-Hop Options Header

In the previous IPv6 specification [RFC2460], the hop-by-hop options header, when present in an IPv6 packet, forced all nodes to inspect and possibly process this header. This enabled denial-of-service attacks as most, if not all, routers cannot process this type of packet in hardware but have to process these packets in software and hence compete with other software tasks, such as handling the control and management plane processing.

Section 4.3 of the current Internet Standard for IPv6, [RFC8200], has taken this attack vector into account and made the processing of hop-by-hop options headers by intermediate routers explicitly configurable.

2.2.3. Fragment Header

The fragment header is used by the source (and only the source) when it has to fragment packets. [RFC7112] and section 4.5 of [RFC8200] explain why it is important that:

Firewall and security devices should drop first fragments that do not contain the entire IPv6 header chain (including the transport-layer header).

Destination nodes should discard first fragments that do not contain the entire IPv6 header chain (including the transport-layer header).

If those requirements are not met, stateless filtering could be bypassed by a hostile party. [RFC6980] applies a stricter rule to Neighbor Discovery Protocol (NDP) by enforcing the drop of fragmented NDP packets (except for "Certification Path Advertisement" messages as noted in section Section 2.3.2.1). [RFC7113] describes how the RA-guard function described in [RFC6105] should behave in the presence of fragmented RA packets.

2.2.4. IP Security Extension Header

The IPsec [RFC4301] extension headers (AH [RFC4302] and ESP [RFC4303]) are required if IPsec is to be utilized for network level security. Previously, IPv6 mandated implementation of IPsec but [RFC6434] updated that recommendation by making support of the IPsec

architecture [RFC4301] a SHOULD for all IPv6 nodes which is also retained in the latest IPv6 Nodes Requirement standard [RFC8504].

2.3. Link-Layer Security

IPv6 relies heavily on NDP [RFC4861] to perform a variety of link operations such as discovering other nodes on the link, resolving their link-layer addresses, and finding routers on the link. If not secured, NDP is vulnerable to various attacks, such as router/neighbor message spoofing, redirect attacks, Duplicate Address Detection (DAD) DoS attacks, etc. Many of these security threats to NDP have been documented in IPv6 ND Trust Models and Threats [RFC3756] and in [RFC6583].

Most of the issues are only applicable when the attacker is on the same link but NDP also has security issues when the attacker is off-link, see the section below Section 2.3.1.

2.3.1. Neighbor Solicitation Rate-Limiting

NDP can be vulnerable to remote denial of service (DoS) attacks; for example, when a router is forced to perform address resolution for a large number of unassigned addresses, i.e., when a prefix is scanned by an attacker in a fast manner. This can keep new devices from joining the network or render the last-hop router ineffective due to high CPU usage. Easy mitigative steps include rate-limiting Neighbor Solicitations, restricting the amount of state reserved for unresolved solicitations, and clever cache/timer management.

[RFC6583] discusses the potential for off-link DoS in detail and suggests implementation improvements and operational mitigation techniques that may be used to mitigate or alleviate the impact of such attacks. Here are some feasible mitigation options that can be employed by network operators today:

- o Ingress filtering of unused addresses by ACL. These require stable configuration of the addresses; for example, allocating the addresses out of a /120 and using a specific ACL to only allow traffic to this /120 (of course, the actual hosts are configured with a /64 prefix for the link).
- o Tuning of NDP process (where supported), e.g., enforcing limits on data structures such as the number of neighbor cache entries in 'incomplete' state (e.g., 256 incomplete entries per interface) or the rate of NA per interface (e.g., 100 NA per second).
- o Using a /127 on a point-to-point link, per [RFC6164].

- o Using only link-local addresses on links where there are only routers, see [RFC7404]

2.3.2. Router and Neighbor Advertisements Filtering

2.3.2.1. Router Advertisement Filtering

Router Advertisement spoofing is a well-known on-link attack vector and has been extensively documented. The presence of rogue RAs, either unintentional or malicious, can cause partial or complete failure of operation of hosts on an IPv6 link. For example, a node can select an incorrect router address which can then be used for an on-path attack or the node can assume wrong prefixes to be used for SLAAC. [RFC6104] summarizes the scenarios in which rogue RAs may be observed and presents a list of possible solutions to the problem. [RFC6105] (RA-Guard) describes a solution framework for the rogue RA problem where network segments are designed around switching devices that are capable of identifying invalid RAs and blocking them before the attack packets actually reach the target nodes.

However, several evasion techniques that circumvent the protection provided by RA-Guard have surfaced. A key challenge to this mitigation technique is introduced by IPv6 fragmentation. Attackers can conceal their attack by fragmenting their packets into multiple fragments such that the switching device that is responsible for blocking invalid RAs cannot find all the necessary information to perform packet filtering of the same packet. [RFC7113] describes such evasion techniques and provides advice to RA-Guard implementers such that the aforementioned evasion vectors can be eliminated.

Given that the IPv6 Fragmentation Header can be leveraged to circumvent some implementations of RA-Guard, [RFC6980] updates [RFC4861] such that use of the IPv6 Fragmentation Header is forbidden in all Neighbor Discovery messages except "Certification Path Advertisement", thus allowing for simple and effective measures to counter fragmented NDP attacks.

2.3.2.2. Neighbor Advertisement Filtering

The Source Address Validation Improvements (SAVI) working group has worked on other ways to mitigate the effects of such attacks. [RFC7513] helps in creating bindings between a DHCPv4 [RFC2131] /DHCPv6 [RFC8415] assigned source IP address and a binding anchor [RFC7039] on a SAVI device. Also, [RFC6620] describes how to glean similar bindings when DHCP is not used. The bindings can be used to filter packets generated on the local link with forged source IP addresses.

2.3.2.3. Host Isolation

Isolating hosts for the NDP traffic can be done by using a /64 per host, refer to Section 2.1.8, as NDP is only relevant within a /64 on-link prefix; 3GPP Section 2.3.4 uses a similar mechanism.

A more drastic technique to prevent all NDP attacks is based on isolation of all hosts with specific configurations. In such a scenario, hosts (i.e., all nodes that are not routers) are unable to send data-link layer frames to other hosts, therefore, no host-to-host attacks can happen. This specific setup can be established on some switches or Wi-Fi access points. This is not always feasible when hosts need to communicate with other hosts in the same subnet, e.g., for access to file shares.

2.3.2.4. NDP Recommendations

It is still recommended that RA-Guard and SAVI be employed as a first line of defense against common attack vectors including misconfigured hosts. This recommendation also applies when DHCPv6 is used, as RA messages are used to discover the default router(s) and for on-link prefix determination. This line of defense is most effective when incomplete fragments are dropped by routers and switches as described in Section 2.2.3. The generated log should also be analyzed to identify and act on violations.

Network operators should be aware that RA-Guard and SAVI do not work as expected or could even be harmful in specific network configurations (notably when there could be multiple routers).

Enabling RA-Guard by default in managed networks (e.g., Wi-Fi networks, enterprise campus networks, etc.) should be strongly considered except for specific use cases such as the presence of homenet devices emitting router advertisements.

2.3.3. Securing DHCP

The Dynamic Host Configuration Protocol for IPv6 (DHCPv6), as described in [RFC8415], enables DHCP servers to pass configuration parameters, such as IPv6 network addresses and other configuration information, to IPv6 nodes. DHCP plays an important role in most large networks by providing robust stateful configuration in the context of automated system provisioning.

The two most common threats to DHCP clients come from malicious (a.k.a., rogue) or unintentionally misconfigured DHCP servers. In these scenarios, a malicious DHCP server is established with the intent of providing incorrect configuration information to the

clients to cause a denial-of-service attack or to mount on-path attack. While unintentional, a misconfigured DHCP server can have the same impact. Additional threats against DHCP are discussed in the security considerations section of [RFC8415].

DHCPv6-Shield, [RFC7610], specifies a mechanism for protecting connected DHCPv6 clients against rogue DHCPv6 servers. This mechanism is based on DHCPv6 packet-filtering at the layer-2 device, i.e., the administrator specifies the interfaces connected to DHCPv6 servers. However, extension headers could be leveraged to bypass DHCPv6-Shield unless [RFC7112] is enforced.

It is recommended to use DHCPv6-Shield and to analyze the corresponding log messages.

2.3.4. 3GPP Link-Layer Security

The 3GPP link is a point-to-point like link that has no link-layer address. This implies there can only be one end host (the mobile hand-set) and the first-hop router (i.e., a GPRS Gateway Support Node (GGSN) or a Packet Gateway (PGW)) on that link. The GGSN/PGW never configures a non link-local address on the link using the advertised /64 prefix on it; see Section 2.1.8. The advertised prefix must not be used for on-link determination. There is no need for address resolution on the 3GPP link, since there are no link-layer addresses. Furthermore, the GGSN/PGW assigns a prefix that is unique within each 3GPP link that uses IPv6 stateless address autoconfiguration. This avoids the necessity to perform DAD at the network level for every address generated by the mobile host. The GGSN/PGW always provides an IID to the cellular host for the purpose of configuring the link-local address and ensures the uniqueness of the IID on the link (i.e., no collisions between its own link-local address and the mobile host's address).

The 3GPP link model itself mitigates most of the known NDP-related Denial-of-Service attacks. In practice, the GGSN/PGW only needs to route all traffic to the mobile host that falls under the prefix assigned to it. As there is also a single host on the 3GPP link, there is no need to defend that IPv6 address.

See Section 5 of [RFC6459] for a more detailed discussion on the 3GPP link model, NDP, and the address configuration details. In some mobile networks, DHCPv6 and DHCP-PD are also used.

2.3.5. Impact of Multicast Traffic

IPv6 uses multicast extensively for signaling messages on the local link to avoid broadcast messages for on-the-wire efficiency.

The use of multicast has some side effects on wireless networks, such as a negative impact on battery life of smartphones and other battery-operated devices that are connected to such networks. [RFC7772] and [RFC6775] (for specific wireless networks) discuss methods to rate-limit RAs and other ND messages on wireless networks in order to address this issue.

The use of link-layer multicast addresses (e.g., ff02::1 for the all nodes link-local multicast address) could also be misused for an amplification attack. Imagine, a hostile node sending an ICMPv6 ECHO_REQUEST to ff02::1 with a spoofed source address, then, all link-local nodes will reply with ICMPv6 ECHO_REPLY packets to the source address. This could be a DoS attack for the address owner. This attack is purely local to the layer-2 network as packets with a link-local destination are never forwarded by an IPv6 router.

This is the reason why large Wi-Fi network deployments often limit the use of link-layer multicast either from or to the uplink of the Wi-Fi access point, i.e., Wi-Fi stations are prevented to send link-local multicast to their direct neighboring Wi-Fi stations; this policy also blocks service discovery via mDNS ([RFC6762]) and LLmNR ([RFC4795]).

2.3.6. SeND and CGA

SEcure Neighbor Discovery (SeND), as described in [RFC3971], is a mechanism that was designed to secure ND messages. This approach involves the use of new NDP options to carry public key-based signatures. Cryptographically Generated Addresses (CGA), as described in [RFC3972], are used to ensure that the sender of a Neighbor Discovery message is the actual "owner" of the claimed IPv6 address. A new NDP option, the CGA option, was introduced and is used to carry the public key and associated parameters. Another NDP option, the RSA Signature option, is used to protect all messages relating to neighbor and Router discovery.

SeND protects against:

- o Neighbor Solicitation/Advertisement Spoofing
- o Neighbor Unreachability Detection Failure
- o Duplicate Address Detection DoS Attack

- o Router Solicitation and Advertisement Attacks
- o Replay Attacks
- o Neighbor Discovery DoS Attacks

SeND does NOT:

- o Protect statically configured addresses
- o Protect addresses configured using fixed identifiers (i.e., EUI-64)
- o Provide confidentiality for NDP communications
- o Compensate for an unsecured link - SeND does not require that the addresses on the link and Neighbor Advertisements correspond.

However, at this time and over a decade since their original specifications, CGA and SeND do not have support from widely deployed IPv6 devices; hence, their usefulness is limited and should not be relied upon.

2.4. Control Plane Security

[RFC6192] defines the router control plane and provides detailed guidance to secure it for IPv4 and IPv6 networks. This definition is repeated here for the reader's convenience. Please note that the definition is completely protocol-version agnostic (most of this section applies to IPv6 in the same way as to IPv4).

Preamble: IPv6 control plane security is vastly congruent with its IPv4 equivalent with the exception of OSPFv3 authentication (Section 2.4.1) and some packet exceptions (see Section 2.4.3) that are specific to IPv6.

Modern router architecture design maintains a strict separation of forwarding and router control plane hardware and software. The router control plane supports routing and management functions. It is generally described as the router architecture hardware and software components for handling packets destined to the device itself, as well as, building and sending packets originated locally on the device. The forwarding plane is typically described as the router architecture hardware and software components responsible for receiving a packet on an incoming interface, performing a lookup to identify the packet's IP next hop and best outgoing interface towards the destination, and forwarding the packet through the appropriate outgoing interface.

While the forwarding plane is usually implemented in high-speed hardware, the control plane is implemented by a generic processor (referred to as the route processor (RP)) and cannot process packets at a high rate. Hence, this processor can be attacked by flooding its input queue with more packets than it can process. The control plane processor is then unable to process valid control packets and the router can lose IGP or BGP adjacencies which can cause a severe network disruption.

[RFC6192] provides detailed guidance to protect the router control plane in IPv6 networks. The rest of this section contains simplified guidance.

The mitigation techniques are:

- o To drop non-legit or potentially harmful control packets before they are queued to the RP (this can be done by a forwarding plane ACL) and
- o To rate-limit the remaining packets to a rate that the RP can sustain. Protocol-specific protection should also be done (for example, a spoofed OSPFv3 packet could trigger the execution of the Dijkstra algorithm, therefore, the frequency of Dijkstra calculations should be also rate-limited).

This section will consider several classes of control packets:

- o Control protocols: routing protocols: such as OSPFv3, BGP, RIPng, and by extension NDP and ICMP
- o Management protocols: SSH, SNMP, NETCONF, RESTCONF, IPFIX, etc.
- o Packet exceptions: normal data packets that require a specific processing such as generating a packet-too-big ICMP message or processing the hop-by-hop options header.

2.4.1. Control Protocols

This class includes OSPFv3, BGP, NDP, ICMP.

An ingress ACL to be applied on all the router interfaces for packets to be processed by the RP should be configured to:

- o drop OSPFv3 (identified by Next-Header being 89) and RIPng (identified by UDP port 521) packets from a non link-local address (except for OSPFv3 virtual links)

- o allow BGP (identified by TCP port 179) packets from all BGP neighbors and drop the others
- o allow all ICMP packets (transit and to the router interfaces)

Note: dropping OSPFv3 packets which are authenticated by IPsec could be impossible on some routers that are unable to parse the IPsec ESP or AH extension headers during ACL classification.

Rate-limiting of the valid packets should be done, see also [RFC8541] for a side benefit for OSPv3. The exact configuration will depend on the available resources of the router (CPU, TCAM, ...).

2.4.2. Management Protocols

This class includes: SSH, SNMP, RESTCONF, NETCONF, gRPC, syslog, NTP, etc.

An ingress ACL to be applied on all the router interfaces (or at ingress interfaces of the security perimeter or by using specific features of the platform) should be configured for packets destined to the RP such as:

- o Drop packets destined to the routers except those belonging to protocols which are used (for example, permit TCP 22 and drop all others when only SSH is used);
- o Drop packets where the source does not match the security policy, for example, if SSH connections should only be originated from the Network Operation Center (NOC), then the ACL should permit TCP port 22 packets only from the NOC prefix.

Rate-limiting of valid packets should be done. The exact configuration will depend on the available router resources.

2.4.3. Packet Exceptions

This class covers multiple cases where a data plane packet is punted to the route processor because it requires specific processing:

- o generation of an ICMP packet-too-big message when a data plane packet cannot be forwarded because it is too large (required to discover the Path MTU);
- o generation of an ICMP hop-limit-expired message when a data plane packet cannot be forwarded because its hop-limit field has reached 0 (also used by the traceroute utility);

- o generation of an ICMP destination-unreachable message when a data plane packet cannot be forwarded for any reason;
- o processing of the hop-by-hop options header, new implementations follow section 4.3 of [RFC8200] where this processing is optional;
- o or more specific to some router implementation: an oversized extension header chain which cannot be processed by the hardware and force the packet to be punted to the RP.

On some routers, not everything can be done by the specialized data plane hardware which requires some packets to be 'punted' to the generic RP. This could include for example the processing of a long extension header chain in order to apply an ACL based on layer-4 information. [RFC6980] and more generally [RFC7112] highlight the security implications of oversized extension header chains on routers and updates the original IPv6 specifications, [RFC2460], such that the first fragment of a packet is required to contain the entire IPv6 header chain. Those changes are incorporated in the IPv6 standard [RFC8200]

An ingress ACL cannot mitigate a control plane attack using these packet exceptions. The only protection for the RP is to rate-limit those packet exceptions that are forwarded to the RP, this means that some data plane packets will be dropped without an ICMP message sent to the source which may delay Path MTU discovery and cause drops.

In addition to limiting the rate of data plane packets queued to the RP, it is also important to rate-limit the generation of ICMP messages. This is important both to preserve RP resources and also to prevent an amplification attack using the router as a reflector. It is worth noting that some platforms implement this rate-limiting in hardware. Of course, a consequence of not generating an ICMP message will break some IPv6 mechanisms such as Path MTU discovery or a simple traceroute.

2.5. Routing Security

Preamble: IPv6 routing security is congruent with IPv4 routing security with the exception of OSPv3 neighbor authentication (see Section 2.5.2).

Routing security in general can be broadly divided into three sections:

1. Authenticating neighbors/peers
2. Securing routing updates between peers

3. Route filtering

[RFC5082] is also applicable to IPv6 and can ensure that routing protocol packets are coming from the local network; it must also be noted that in IPv6 all interior gateway protocols use link-local addresses.

As for IPv4, it is recommended to enable a routing protocol only on interfaces where it is required.

2.5.1. BGP Security

As BGP is identical for IPv4 and IPv6 and as [RFC7454] covers all the security aspects for BGP in detail, [RFC7454] is also applicable to IPv6.

2.5.2. Authenticating OSPFv3 Neighbors

OSPFv3 can rely on IPsec to fulfill the authentication function. Operators should note that IPsec support is not standard on all routing platforms. In some cases, this requires specialized hardware that offloads crypto over to dedicated ASICs or enhanced software images (both of which often come with added financial cost) to provide such functionality. An added detail is to determine whether OSPFv3 IPsec implementations use AH or ESP-Null for integrity protection. In early implementations, all OSPFv3 IPsec configurations relied on AH since the details weren't specified in [RFC5340]. However, the document which specifically describes how IPsec should be implemented for OSPFv3 [RFC4552] specifically states that "ESP-Null MUST and AH MAY be implemented" since it follows the overall IPsec standards wording. OSPFv3 can also use normal ESP to encrypt the OSPFv3 payload to provide confidentiality for the routing information.

[RFC7166] changes OSPFv3 reliance on IPsec by appending an authentication trailer to the end of the OSPFv3 packets; it does not specifically authenticate the specific originator of an OSPFv3 packet; rather, it allows a router to confirm that the packet has been issued by a router that had access to the shared authentication key.

With all authentication mechanisms, operators should confirm that implementations can support re-keying mechanisms that do not cause outages. There have been instances where any re-keying causes outages and therefore, the tradeoff between utilizing this functionality needs to be weighed against the protection it provides. [RFC4107] documents some guidelines for crypto keys management.

2.5.3. Securing Routing Updates

IPv6 initially mandated the provisioning of IPsec capability in all nodes. However, in the updated IPv6 Nodes Requirement standard [RFC8504], IPsec is a 'SHOULD' and not a 'MUST' implement. Theoretically, it is possible that all communication between two IPv6 nodes, especially routers exchanging routing information, is encrypted using IPsec. In practice however, deploying IPsec is not always feasible given hardware and software limitations of the various platforms deployed.

Many routing protocols support the use of cryptography to protect the routing updates, the use of this protection is recommended; [RFC8177] is a YANG data model for key chains that includes re-keying functionality.

2.5.4. Route Filtering

Route filtering policies will be different depending on whether they pertain to edge route filtering vs. internal route filtering. At a minimum, IPv6 routing policy as it pertains to routing between different administrative domains should aim to maintain parity with IPv4 from a policy perspective, e.g.,

- o Filter internal-use, non-globally routable IPv6 addresses at the perimeter;
- o Discard routes for bogon [CYMRU] and reserved space (see [RFC8190]);
- o Configure ingress route filters that validate route origin, prefix ownership, etc. through the use of various routing databases, e.g., [RADB]. [RFC8210] formally validates the origin ASs of BGP announcements.

Some good guidance can be found at [RFC7454].

A valid routing table can also be used to apply network ingress filtering (see [RFC2827]).

2.6. Logging/Monitoring

In order to perform forensic research in the cases of a security incident or detecting abnormal behavior, network operators should log multiple pieces of information. In some cases, this requires a frequent poll of devices via a Network Management Station.

This logging should include, but not limited to:

- o logs of all applications using the network (including user space and kernel space) when available (for example web servers that the network operator manages);
- o data from IP Flow Information Export [RFC7011] also known as IPFIX;
- o data from various SNMP MIBs [RFC4293] or YANG data via RESTCONF [RFC8040] or NETCONF [RFC6241];
- o historical data of Neighbor Cache entries;
- o stateful DHCPv6 [RFC8415] lease cache, especially when a relay agent [RFC6221] is used;
- o Source Address Validation Improvement (SAVI) [RFC7039] events, especially the binding of an IPv6 address to a MAC address and a specific switch or router interface;
- o firewall ACL log;
- o authentication server log;
- o RADIUS [RFC2866] accounting records.

Please note that there are privacy issues or regulations related to how these logs are collected, stored, used, and safely discarded. Operators are urged to check their country legislation (e.g., General Data Protection Regulation GDPR [GDPR] in the European Union).

All those pieces of information can be used for:

- o forensic (Section 2.6.2.1) investigations such as who did what and when?
- o correlation (Section 2.6.2.3): which IP addresses were used by a specific node (assuming the use of privacy extensions addresses [RFC8981])
- o inventory (Section 2.6.2.2): which IPv6 nodes are on my network?
- o abnormal behavior detection (Section 2.6.2.4): unusual traffic patterns are often the symptoms of an abnormal behavior which is in turn a potential attack (denial-of-service, network scan, a node being part of a botnet, etc.)

2.6.1. Data Sources

This section lists the most important sources of data that are useful for operational security.

2.6.1.1. Application Logs

Those logs are usually text files where the remote IPv6 address is stored in clear text (not binary). This can complicate the processing since one IPv6 address, for example 2001:db8::1 can be written in multiple ways, such as:

- o 2001:DB8::1 (in uppercase)
- o 2001:0db8::0001 (with leading 0)
- o and many other ways including the reverse DNS mapping into a FQDN (which should not be trusted).

[RFC5952] explains this problem in detail and recommends the use of a single canonical format. This document recommends the use of canonical format [RFC5952] for IPv6 addresses in all possible cases. If the existing application cannot log using the canonical format, then it is recommended to use an external post-processing program in order to canonicalize all IPv6 addresses.

2.6.1.2. IP Flow Information Export by IPv6 Routers

IPFIX [RFC7012] defines some data elements that are useful for security:

- o nextHeaderIPv6, sourceIPv6Address, and destinationIPv6Address;
- o sourceMacAddress and destinationMacAddress.

The IP version is the ipVersion element defined in [IANA-IPFIX].

Moreover, IPFIX is very efficient in terms of data handling and transport. It can also aggregate flows by a key such as sourceMacAddress in order to have aggregated data associated with a specific sourceMacAddress. This memo recommends the use of IPFIX and aggregation on nextHeaderIPv6, sourceIPv6Address, and sourceMacAddress.

2.6.1.3. SNMP MIB and NETCONF/RESTCONF YANG Modules data by IPv6 Routers

RFC 4293 [RFC4293] defines a Management Information Base (MIB) for the two address families of IP. This memo recommends the use of:

- o ipIfStatsTable table which collects traffic counters per interface;
- o ipNetToPhysicalTable table which is the content of the Neighbor cache, i.e., the mapping between IPv6 and data-link layer addresses.

There are also YANG modules relating to the two IP addresses families and can be used with [RFC6241] and [RFC8040]. This memo recommends the use of:

- o interfaces-state/interface/statistics from ietf-interfaces@2018-02-20.yang [RFC8343] which contains counters for interfaces.
- o ipv6/neighbor from ietf-ip@2018-02-22.yang [RFC8344] which is the content of the Neighbor cache, i.e., the mapping between IPv6 and data-link layer addresses.

2.6.1.4. Neighbor Cache of IPv6 Routers

The neighbor cache of routers contains all mappings between IPv6 addresses and data-link layer addresses. There are multiple ways to collect the current entries in the Neighbor Cache, notably but not limited to:

- o the SNMP MIB (Section 2.6.1.3) as explained above;
- o using streaming telemetry or NETCONF [RFC6241] and RESTCONF [RFC8040] to collect the operational state of the neighbor cache;
- o also, by connecting over a secure management channel (such as SSH) and explicitly requesting a neighbor cache dump via the Command Line Interface (CLI) or another monitoring mechanism.

The neighbor cache is highly dynamic as mappings are added when a new IPv6 address appears on the network. This could be quite frequently with privacy extension addresses [RFC8981] or when they are removed when the state goes from UNREACH to removed (the default time for a removal per Neighbor Unreachability Detection [RFC4861] algorithm is 38 seconds for a host using Windows 7). This means that the content of the neighbor cache must periodically be fetched at an interval

which does not exhaust the router resources and still provides valuable information (suggested value is 30 seconds but this should be verified in the actual deployment) and stored for later use.

This is an important source of information because it is trivial (on a switch not using the SAVI [RFC7039] algorithm) to defeat the mapping between data-link layer address and IPv6 address. Let us rephrase the previous statement: having access to the current and past content of the neighbor cache has a paramount value for the forensic and audit trail. It should also be noted that in certain threat models this information is also deemed valuable and could itself be a target.

When using one /64 per host (Section 2.1.8) or DHCP-PD, it is sufficient to keep the history of the allocated prefixes when combined with strict source address prefix enforcement on the routers and layer-2 switches to prevent IPv6 spoofing.

2.6.1.5. Stateful DHCPv6 Lease

In some networks, IPv6 addresses/prefixes are managed by a stateful DHCPv6 server [RFC8415] that leases IPv6 addresses/prefixes to clients. It is indeed quite similar to DHCP for IPv4, so it can be tempting to use this DHCP lease file to discover the mapping between IPv6 addresses/prefixes and data-link layer addresses as is commonly used in IPv4 networking.

It is not so easy in the IPv6 networks, because not all nodes will use DHCPv6 (there are nodes which can only do stateless autoconfiguration) but also because DHCPv6 clients are identified not by their hardware-client address as in IPv4 but by a DHCP Unique ID (DUID), which can have several formats: some being the data-link layer address, some being data-link layer address prepended with time information, or even an opaque number that requires correlation with another data source to be usable for operational security. Moreover, when the DUID is based on the data-link address, this address can be of any client interface (such as the wireless interface while the client actually uses its wired interface to connect to the network).

If a lightweight DHCP relay agent [RFC6221] is used in a layer-2 switch, then the DHCP servers also receive the Interface-ID information which could be saved in order to identify the interface on which the switch received a specific leased IPv6 address. Also, if a 'normal' (not lightweight) relay agent adds the data-link layer address in the option for Relay Agent Remote-ID [RFC4649] or [RFC6939], then the DHCPv6 server can keep track of the data-link and leased IPv6 addresses.

In short, the DHCPv6 lease file is less interesting than for IPv4 networks. If possible, it is recommended to use DHCPv6 servers that keep the relayed data-link layer address in addition to the DUID in the lease file as those servers have the equivalent information to IPv4 DHCP servers.

The mapping between data-link layer address and the IPv6 address can be secured by deploying switches implementing the SAVI [RFC7513] mechanisms. Of course, this also requires that the data-link layer address is protected by using a layer-2 mechanism such as [IEEE-802.1X].

2.6.1.6. RADIUS Accounting Log

For interfaces where the user is authenticated via a RADIUS [RFC2866] server, and if RADIUS accounting is enabled, then the RADIUS server receives accounting Acct-Status-Type records at the start and at the end of the connection which include all IPv6 (and IPv4) addresses used by the user. This technique can be used notably for Wi-Fi networks with Wi-Fi Protected Access (WPA) or other IEEE 802.1X [IEEE-802.1X] wired interface on an Ethernet switch.

2.6.1.7. Other Data Sources

There are other data sources for log information that must be collected (as currently collected in IPv4 networks):

- o historical mapping of IPv6 addresses to users of remote access VPN;
- o historical mappings of MAC addresses to switch ports in a wired network.

2.6.2. Use of Collected Data

This section leverages the data collected as described before (Section 2.6.1) in order to achieve several security benefits. Section 9.1 of [RFC7934] contains more details about host tracking.

2.6.2.1. Forensic and User Accountability

The forensic use case is when the network operator must locate an IPv6 address (and the associated port, access point/switch, or VPN tunnel) that was present in the network at a certain time or is currently in the network.

To locate an IPv6 address in an enterprise network where the operator has control over all resources, the source of information can be the

neighbor cache, or, if not found, the DHCP lease file. Then, the procedure is:

1. Based on the IPv6 prefix of the IPv6 address, find the router(s) which is(are) used to reach this prefix (assuming that anti-spoofing mechanisms are used) perhaps based on an IPAM.
2. Based on this limited set of routers, on the incident time and on the IPv6 address, retrieve the data-link address from the live neighbor cache, from the historical neighbor cache data, or from SAVI events, or retrieve the data-link address from the DHCP lease file (Section 2.6.1.5).
3. Based on the data-link layer address, look-up the switch interface associated with the data-link layer address. In the case of wireless LAN with RADIUS accounting (see Section 2.6.1.6), the RADIUS log has the mapping between the user identification and the MAC address. If a Configuration Management Data Base (CMDB) is used, then it can be used to map the data-link layer address to a switch port.

At the end of the process, the interface of the host originating, or the subscriber identity associated with, the activity in question has been determined.

To identify the subscriber of an IPv6 address in a residential Internet Service Provider, the starting point is the DHCP-PD leased prefix covering the IPv6 address; this prefix can often be linked to a subscriber via the RADIUS log. Alternatively, the Forwarding Information Base (FIB) of the Cable Modem Termination System (CMTS) or Broadband Network Gateway (BNG) indicates the CPE of the subscriber and the RADIUS log can be used to retrieve the actual subscriber.

More generally, a mix of the above techniques can be used in most, if not all, networks.

2.6.2.2. Inventory

RFC 7707 [RFC7707] describes the difficulties for an attacker to scan an IPv6 network due to the vast number of IPv6 addresses per link (and why in some cases it can still be done). While the huge addressing space can sometimes be perceived as a 'protection', it also makes the inventory task difficult in an IPv6 network while it was trivial to do in an IPv4 network (a simple enumeration of all IPv4 addresses, followed by a ping and a TCP/UDP port scan). Getting an inventory of all connected devices is of prime importance for a secure network operation.

There are many ways to do an inventory of an IPv6 network.

The first technique is to use passive inspection such as IPFIX. Using exported IPFIX information and extracting the list of all IPv6 source addresses allows finding all IPv6 nodes that sent packets through a router. This is very efficient but, alas, will not discover silent nodes that never transmitted packets traversing the IPFIX target router. Also, it must be noted that link-local addresses will never be discovered by this means.

The second way is again to use the collected neighbor cache content to find all IPv6 addresses in the cache. This process will also discover all link-local addresses. See Section 2.6.1.4.

Another way that works only for a local network, consists of sending a ICMP ECHO_REQUEST to the link-local multicast address ff02::1 which addresses all IPv6 nodes on the network. All nodes should reply to this ECHO_REQUEST per [RFC4443].

Other techniques involve obtaining data from DNS, parsing log files, leveraging service discovery such as mDNS [RFC6762] and [RFC6763].

Enumerating DNS zones, especially looking at reverse DNS records and CNAMEs, is another common method employed by various tools. As already mentioned in [RFC7707], this allows an attacker to prune the IPv6 reverse DNS tree, and hence enumerate it in a feasible time. Furthermore, authoritative servers that allow zone transfers (AXFR) may be a further information source. An interesting research paper has analysed the entropy in various IPv6 addresses: see [ENTROPYIP].

2.6.2.3. Correlation

In an IPv4 network, it is easy to correlate multiple logs, for example to find events related to a specific IPv4 address. A simple Unix grep command is enough to scan through multiple text-based files and extract all lines relevant to a specific IPv4 address.

In an IPv6 network, this is slightly more difficult because different character strings can express the same IPv6 address. Therefore, the simple Unix grep command cannot be used. Moreover, an IPv6 node can have multiple IPv6 addresses.

In order to do correlation in IPv6-related logs, it is advised to have all logs in a format with only canonical IPv6 addresses [RFC5952]. Then, the neighbor cache current (or historical) data set must be searched to find the data-link layer address of the IPv6 address. Then, the current and historical neighbor cache data sets must be searched for all IPv6 addresses associated with this data-

link layer address to derive the search set. The last step is to search in all log files (containing only IPv6 addresses in canonical format) for any IPv6 addresses in the search set.

Moreover, [RFC7934] recommends using multiple IPv6 addresses per prefix, so, the correlation must also be done among those multiple IPv6 addresses, for example by discovering in the NDP cache (Section 2.6.1.4) all IPv6 addresses associated with the same MAC address and interface.

2.6.2.4. Abnormal Behavior Detection

Abnormal behavior (such as network scanning, spamming, denial-of-service) can be detected in the same way as in an IPv4 network.

- o Sudden increase of traffic detected by interface counter (SNMP) or by aggregated traffic from IPFIX records [RFC7012].
- o Rapid growth of ND cache size.
- o Change in traffic pattern (number of connections per second, number of connections per host...) observed with the use of IPFIX [RFC7012].

2.6.3. Summary

While some data sources (IPFIX, MIB, switch CAM tables, logs, ...) used in IPv4 are also used in the secure operation of an IPv6 network, the DHCPv6 lease file is less reliable and the neighbor cache is of prime importance.

The fact that there are multiple ways to express the same IPv6 address in a character string renders the use of filters mandatory when correlation must be done.

2.7. Transition/Coexistence Technologies

As it is expected that some networks will not run in a pure IPv6-only mode, the different transition mechanisms must be deployed and operated in a secure way. This section proposes operational guidelines for the most known and deployed transition techniques. [RFC4942] also contains security considerations for transition or coexistence scenarios.

2.7.1. Dual Stack

Dual stack is often the first deployment choice for network operators. Dual stacking the network offers some advantages over other transition mechanisms. Firstly, the impact on existing IPv4 operations is reduced. Secondly, in the absence of tunnels or address translation, the IPv4 and IPv6 traffic are native (easier to observe and secure) and should have the same network processing (network path, quality of service, ...). Dual stack enables a gradual termination of the IPv4 operations when the IPv6 network is ready for prime time. On the other hand, the operators have to manage two network stacks with the added complexities.

From an operational security perspective, this now means that the network operator has twice the exposure. One needs to think about protecting both protocols now. At a minimum, the IPv6 portion of a dual-stacked network should be consistent with IPv4 from a security policy point of view. Typically, the following methods are employed to protect IPv4 networks at the edge or security perimeter:

- o ACLs to permit or deny traffic;
- o Firewalls with stateful packet inspection;
- o Application firewalls inspecting the application flows.

It is recommended that these ACLs and/or firewalls be additionally configured to protect IPv6 communications. The enforced IPv6 security must be congruent with the IPv4 security policy, otherwise the attacker will use the protocol version having the more relaxed security policy. Maintaining the congruence between security policies can be challenging (especially over time); it is recommended to use a firewall or an ACL manager that is dual-stack, i.e., a system that can apply a single ACL entry to a mixed group of IPv4 and IPv6 addresses.

Application firewalls work at the application layer and are oblivious to the IP version, i.e., they work as well for IPv6 as for IPv4 and the same application security policy will work for both protocol versions.

Also, given the end-to-end connectivity that IPv6 provides, it is recommended that hosts be fortified against threats. General device hardening guidelines are provided in Section 2.8.

For many years, all host operating systems have IPv6 enabled by default, so, it is possible even in an 'IPv4-only' network to attack layer-2 adjacent victims via their IPv6 link-local address or via a

global IPv6 address when the attacker provides rogue RAs or a rogue DHCPv6 service.

[RFC7123] discusses the security implications of native IPv6 support and IPv6 transition/coexistence technologies on "IPv4-only" networks and describes possible mitigations for the aforementioned issues.

2.7.2. Encapsulation Mechanisms

There are many tunnels used for specific use cases. Except when protected by IPsec [RFC4301] or alternative tunnel encryption methods, all those tunnels have a number of security issues as described in RFC 6169 [RFC6169];

- o tunnel injection: a malevolent actor knowing a few pieces of information (for example the tunnel endpoints and the encapsulation protocol) can forge a packet which looks like a legitimate and valid encapsulated packet that will gladly be accepted by the destination tunnel endpoint. This is a specific case of spoofing;
- o traffic interception: no confidentiality is provided by the tunnel protocols (without the use of IPsec or alternative encryption methods), therefore anybody on the tunnel path can intercept the traffic and have access to the clear-text IPv6 packet; combined with the absence of authentication, an on-path attack can also be mounted;
- o service theft: as there is no authorization, even a non-authorized user can use a tunnel relay for free (this is a specific case of tunnel injection);
- o reflection attack: another specific use case of tunnel injection where the attacker injects packets with an IPv4 destination address not matching the IPv6 address causing the first tunnel endpoint to re-encapsulate the packet to the destination... Hence, the final IPv4 destination will not see the original IPv4 address but only the IPv4 address of the relay router.
- o bypassing security policy: if a firewall or an Intrusion Prevention System (IPS) is on the path of the tunnel, then it may neither inspect nor detect malevolent IPv6 traffic transmitted over the tunnel.

To mitigate the bypassing of security policies, it is often recommended to block all automatic tunnels in default OS configuration (if they are not required) by denying IPv4 packets matching:

- o IP protocol 41: this will block ISATAP (Section 2.7.2.2), 6to4 (Section 2.7.2.7), 6rd (Section 2.7.2.3), as well as, 6in4 (Section 2.7.2.1) tunnels;
- o IP protocol 47: this will block GRE (Section 2.7.2.1) tunnels;
- o UDP port 3544: this will block the default encapsulation of Teredo (Section 2.7.2.8) tunnels.

Ingress filtering [RFC2827] should also be applied on all tunnel endpoints if applicable to prevent IPv6 address spoofing.

The reflection attack cited above should also be prevented by using an IPv6 ACL preventing the hair pinning of the traffic.

As several of the tunnel techniques share the same encapsulation (i.e., IPv4 protocol 41) and embed the IPv4 address in the IPv6 address, there are a set of well-known looping attacks described in RFC 6324 [RFC6324]. This RFC also proposes mitigation techniques.

2.7.2.1. Site-to-Site Static Tunnels

Site-to-site static tunnels are described in RFC 2529 [RFC2529] and in GRE [RFC2784]. As the IPv4 endpoints are statically configured and are not dynamic, they are slightly more secure (bi-directional service theft is mostly impossible) but traffic interception and tunnel injection are still possible. Therefore, the use of IPsec [RFC4301] in transport mode to protect the encapsulated IPv4 packets is recommended for those tunnels. Alternatively, IPsec in tunnel mode can be used to transport IPv6 traffic over a non-trusted IPv4 network.

2.7.2.2. ISATAP

ISATAP tunnels [RFC5214] are mainly used within a single administrative domain and to connect a single IPv6 host to the IPv6 network. This often implies that those systems are usually managed by a single entity; therefore, audit trail and strict anti-spoofing are usually possible and this raises the overall security. Even if ISATAP is no more often used, its security issues are relevant per [KRISTOFF].

Special care must be taken to avoid a looping attack by implementing the measures of [RFC6324] and [RFC6964] (especially the section 3.6).

IPsec [RFC4301] in transport or tunnel mode can be used to secure the IPv4 ISATAP traffic to provide IPv6 traffic confidentiality and prevent service theft.

2.7.2.3. 6rd

While 6rd tunnels share the same encapsulation as 6to4 tunnels (Section 2.7.2.7), they are designed to be used within a single SP domain, in other words, they are deployed in a more constrained environment (e.g., anti-spoofing, protocol 41 filtering at the edge) than 6to4 tunnels and have few security issues other than lack of confidentiality. The security considerations (Section 12) of [RFC5969] describes how to secure 6rd tunnels.

IPsec [RFC4301] for the transported IPv6 traffic can be used if confidentiality is important.

2.7.2.4. 6PE, 6VPE, and LDPv6

Organizations using MPLS in their core can also use 6PE [RFC4798] and 6VPE [RFC4659] to enable IPv6 access over MPLS. As 6PE and 6VPE are really similar to BGP/MPLS IP VPNs described in [RFC4364], the security properties of these networks are also similar to those described in [RFC4381] (please note that this RFC may resemble a published IETF work but it is not based on an IETF review and the IETF disclaims any knowledge of the fitness of this RFC for any purpose). They rely on:

- o Address space, routing, and traffic separation with the help of VRFs (only applicable to 6VPE);
- o Hiding the IPv4 core, hence removing all attacks against P-routers;
- o Securing the routing protocol between CE and PE; in the case of 6PE and 6VPE, link-local addresses (see [RFC7404]) can be used and as these addresses cannot be reached from outside of the link, the security of 6PE and 6VPE is even higher than an IPv4 BGP/MPLS IP VPN.

LDPv6 itself does not induce new risks, see also [RFC7552].

2.7.2.5. DS-Lite

DS-lite is also a translation mechanism and is therefore analyzed further (Section 2.7.3.3) in this document as it includes IPv4 NAT.

2.7.2.6. Mapping of Address and Port

With the encapsulation and translation versions of mapping of Address and Port (MAP) (MAP-E [RFC7597] and MAP-T [RFC7599]), the access network is purely an IPv6 network and MAP protocols are used to

provide IPv4 hosts on the subscriber network access to IPv4 hosts on the Internet. The subscriber router does stateful operations in order to map all internal IPv4 addresses and layer-4 ports to the IPv4 address and the set of layer-4 ports received through the MAP configuration process. The SP equipment always does stateless operations (either decapsulation or stateless translation). Therefore, as opposed to Section 2.7.3.3, there is no state-exhaustion DoS attack against the SP equipment because there is no state and there is no operation caused by a new layer-4 connection (no logging operation).

The SP MAP equipment should implement all the security considerations of [RFC7597]; notably, ensuring that the mapping of the IPv4 address and port are consistent with the configuration. As MAP has a predictable IPv4 address and port mapping, the audit logs are easier to use as there is a clear mapping between the IPv6 address and the IPv4 address and ports.

2.7.2.7. 6to4

In [RFC3056]; 6to4 tunnels require a public routable IPv4 address in order to work correctly. They can be used to provide either single IPv6 host connectivity to the IPv6 Internet or multiple IPv6 networks connectivity to the IPv6 Internet. The 6to4 relay was historically the anycast address defined in [RFC3068] which has been deprecated by [RFC7526] and is no longer used by recent Operating Systems. Some security considerations are explained in [RFC3964].

[RFC6343] points out that if an operator provides well-managed servers and relays for 6to4, non-encapsulated IPv6 packets will pass through well-defined points (the native IPv6 interfaces of those servers and relays) at which security mechanisms may be applied. Client usage of 6to4 by default is now discouraged, and significant precautions are needed to avoid operational problems.

2.7.2.8. Teredo

Teredo tunnels [RFC4380] are mainly used in a residential environment because Teredo easily traverses an IPv4 NAT device thanks to its UDP encapsulation. Teredo tunnels connect a single host to the IPv6 Internet. Teredo shares the same issues as other tunnels: no authentication, no confidentiality, possible spoofing and reflection attacks.

IPsec [RFC4301] for the transported IPv6 traffic is recommended.

The biggest threat to Teredo is probably for an IPv4-only network as Teredo has been designed to easily traverse IPv4 NAT-PT devices which

are quite often co-located with a stateful firewall. Therefore, if the stateful IPv4 firewall allows unrestricted UDP outbound and accepts the return UDP traffic, then Teredo actually punches a hole in this firewall for all IPv6 traffic to the Internet and from the Internet. Host policies can be deployed to block Teredo in an IPv4-only network in order to avoid this firewall bypass. On the IPv4 firewall all outbound UDP should be blocked except for the commonly used services (e.g., port 53 for DNS, port 123 for NTP, port 443 for QUIC, port 500 for IKE, port 3478 for STUN, etc.).

Teredo is now hardly ever used and no longer enabled by default in most environments, so it is less of a threat, however, special consideration must be taken in cases when devices with older or non-updated operating systems may be present and by default were running Teredo.

2.7.3. Translation Mechanisms

Translation mechanisms between IPv4 and IPv6 networks are alternate coexistence strategies while networks transition to IPv6. While a framework is described in [RFC6144], the specific security considerations are documented with each individual mechanism. For the most part, they specifically mention interference with IPsec or DNSSEC deployments, how to mitigate spoofed traffic, and what some effective filtering strategies may be.

While not really a transition mechanism to IPv6, this section also includes the discussion about the use of heavy IPv4-to-IPv4 network address and port translation to prolong the life of IPv4-only networks.

2.7.3.1. Carrier-Grade NAT (CGN)

Carrier-Grade NAT (CGN), also called NAT444 CGN or Large Scale NAT (LSN) or SP NAT is described in [RFC6264] and is utilized as an interim measure to extend the use of IPv4 in a large service provider network until the provider can deploy an effective IPv6 solution. [RFC6598] requested a specific IANA allocated /10 IPv4 address block to be used as address space shared by all access networks using CGN. This has been allocated as 100.64.0.0/10.

Section 13 of [RFC6269] lists some specific security-related issues caused by large scale address sharing. The Security Considerations section of [RFC6598] also lists some specific mitigation techniques for potential misuse of shared address space. Some Law Enforcement Agencies have identified CGN as impeding their cyber-crime investigations (for example Europol press release on CGN [europol-cgn]). Many translation techniques (NAT64, DS-lite, ...)

have the same security issues as CGN when one part of the connection is IPv4-only.

[RFC6302] has recommendations for Internet-facing servers to also log the source TCP or UDP ports of incoming connections in an attempt to help identify the users behind such a CGN.

[RFC7422] suggests the use of deterministic address mapping in order to reduce logging requirements for CGN. The idea is to have a known algorithm for mapping the internal subscriber to/from public TCP and UDP ports.

[RFC6888] lists common requirements for CGNs. [RFC6967] analyzes some solutions to enforce policies on misbehaving nodes when address sharing is used. [RFC7857] also updates the NAT behavioral requirements.

2.7.3.2. NAT64/DNS64 and 464XLAT

Stateful NAT64 translation [RFC6146] allows IPv6-only clients to contact IPv4 servers using unicast UDP, TCP, or ICMP. It can be used in conjunction with DNS64 [RFC6147], a mechanism which synthesizes AAAA records from existing A records. There is also a stateless NAT64 [RFC7915], which has similar security aspects but with the added benefit of being stateless, so, less prone to a state exhaustion attack.

The Security Consideration sections of [RFC6146] and [RFC6147] list the comprehensive issues; in section 8 of [RFC6147] there are some considerations on the interaction between NAT64 and DNSSEC. A specific issue with the use of NAT64 is that it will interfere with most IPsec deployments unless UDP encapsulation is used.

Another translation mechanism relying on a combination of stateful and stateless translation, 464XLAT [RFC6877], can be used to do host local translation from IPv4 to IPv6 and a network provider translation from IPv6 to IPv4, i.e., giving IPv4-only application access to an IPv4-only server over an IPv6-only network. 464XLAT shares the same security considerations as NAT64 and DNS64, however it can be used without DNS64, avoiding the DNSSEC implications.

2.7.3.3. DS-Lite

Dual-Stack Lite (DS-Lite) [RFC6333] is a transition technique that enables a service provider to share IPv4 addresses among customers by combining two well-known technologies: IP in IP (IPv4-in-IPv6) and IPv4 NAPT.

Security considerations with respect to DS-Lite mainly revolve around logging data, preventing DoS attacks from rogue devices (as the Address Family Translation Router (AFTR) [RFC6333] function is stateful) and restricting service offered by the AFTR only to registered customers.

Section 11 of [RFC6333] and section 2 of [RFC7785] describe important security issues associated with this technology.

2.8. General Device Hardening

With almost all devices being IPv6 enabled by default and with many end points having IPv6 connectivity to the Internet, it is critical to also harden those devices against attacks over IPv6.

The same techniques used to protect devices against attack over IPv4 should be used for IPv6 and should include, but not limited to:

- o Restrict device access to authorized individuals
- o Monitor and audit access to the device
- o Turn off any unused services on the end node
- o Understand which IPv6 addresses are being used to source traffic and change defaults if necessary
- o Use cryptographically protected protocols for device management (SCP, SNMPv3, SSH, TLS, etc.)
- o Use host firewall capabilities to control traffic that gets processed by upper-layer protocols
- o apply firmware, OS and application patches/upgrades to the devices in a timely manner
- o use multi-factor credentials to authenticate to devices
- o Use virus scanners to detect malicious programs

3. Enterprises Specific Security Considerations

Enterprises [RFC7381] generally have robust network security policies in place to protect existing IPv4 networks. These policies have been distilled from years of experiential knowledge of securing IPv4 networks. At the very least, it is recommended that enterprise networks have parity between their security policies for both protocol versions. This section also applies to the enterprise part

of all SP networks, i.e., the part of the network where the SP employees are connected.

Security considerations in the enterprise can be broadly categorized into two groups: External and Internal.

3.1. External Security Considerations

The external aspect deals with providing security at the edge or perimeter of the enterprise network where it meets the service provider's network. This is commonly achieved by enforcing a security policy either by implementing dedicated firewalls with stateful packet inspection or a router with ACLs. A common default IPv4 policy on firewalls that could easily be ported to IPv6 is to allow all traffic outbound while only allowing specific traffic, such as established sessions, inbound (see also [RFC6092]). Section 3.2 of [RFC7381] also provides similar recommendations.

Here are a few more things that could enhance the default policy:

- o Filter internal-use IPv6 addresses at the perimeter, this will also mitigate the vulnerabilities listed in [RFC7359]
- o Discard packets from and to bogon and reserved space, see also [CYMRU] and [RFC8190]
- o Accept certain ICMPv6 messages to allow proper operation of ND and PMTUD, see also [RFC4890] or [REY_PF] for hosts
- o Based on the use of the network, filter specific extension headers by accepting only the required ones (permit list approach) such as ESP, AH, and not forgetting the required transport layers: ICMP, TCP, UDP, ... This filtering should be done where applicable at the edge and possibly inside the perimeter; see also [I-D.ietf-opsec-ipv6-eh-filtering]
- o Filter packets having an illegal IPv6 headers chain at the perimeter (and if possible, inside the network as well), see Section 2.2
- o Filter unneeded services at the perimeter
- o Implement ingress and egress anti-spoofing in the forwarding and control planes, see [RFC2827] and [RFC3704]
- o Implement appropriate rate-limiters and control-plane policers based on traffic baselines

Having global IPv6 addresses on all the enterprise sites is different than in IPv4 where [RFC1918] addresses are often used internally and not routed over the Internet. [RFC7359] and [WEBER_VPN] explain that without careful design, there could be IPv6 leakages from layer-3 VPNs.

3.2. Internal Security Considerations

The internal aspect deals with providing security inside the perimeter of the network, including end hosts. Internal networks of enterprises are often different: University campus, wireless guest access, ... so there is no "one size fits all" recommendation.

The most significant concerns here are related to Neighbor Discovery. At the network level, it is recommended that all security considerations discussed in Section 2.3 be reviewed carefully and the recommendations be considered in-depth as well. Section 4.1 of [RFC7381] also provides some recommendations.

As mentioned in Section 2.7.2, care must be taken when running automated IPv6-in-IPv4 tunnels.

When site-to-site VPNs are used it should be kept in mind that, given the global scope of IPv6 global addresses as opposed to the common use of IPv4 private address space [RFC1918], sites might be able to communicate with each other over the Internet even when the VPN mechanism is not available and hence no traffic encryption is performed and traffic could be injected from the Internet into the site, see [WEBER_VPN]. It is recommended to filter at Internet connection(s) packets having a source or destination address belonging to the site internal prefix(es); this should be done for ingress and egress traffic.

Hosts need to be hardened directly through security policy to protect against security threats. The host firewall default capabilities have to be clearly understood. In some cases, 3rd party firewalls have no IPv6 support whereas the native firewall installed by default has IPv6 support. General device hardening guidelines are provided in Section 2.8.

It should also be noted that many hosts still use IPv4 for transporting logs for RADIUS, DIAMETER, TACACS+, SYSLOG, etc. Operators cannot rely on an IPv6-only security policy to secure such protocols that are still using IPv4.

4. Service Providers Security Considerations

4.1. BGP

The threats and mitigation techniques are identical between IPv4 and IPv6. Broadly speaking they are:

- o Authenticating the TCP session;
- o TTL security (which becomes hop-limit security in IPv6) as [RFC5082];
- o bogon AS filtering, see [CYMRU];
- o Prefix filtering.

These are explained in more detail in Section 2.5. Also, the recommendations of [RFC7454] should be considered.

4.1.1. Remote Triggered Black Hole Filtering (RTBH)

RTBH [RFC5635] works identically in IPv4 and IPv6. IANA has allocated the 100::/64 prefix to be used as the discard prefix [RFC6666]

4.2. Transition/Coexistence Mechanism

SPs will typically use transition mechanisms such as 6rd, 6PE, MAP, and NAT64 which have been analyzed in the transition and coexistence Section 2.7 section.

4.3. Lawful Intercept

The Lawful Intercept requirements are similar for IPv6 and IPv4 architectures and will be subject to the laws enforced in different geographic regions. The local issues with each jurisdiction can make this challenging and both corporate legal and privacy personnel should be involved in discussions pertaining to what information gets logged and with regard to the respective log retention policies for this information.

The target of interception will usually be a residential subscriber (e.g., his/her PPP session, physical line, or CPE MAC address). In the absence of IPv6 NAT on the CPE, IPv6 has the possibility to allow for intercepting the traffic from a single host (i.e., a /128 target) rather than the whole set of hosts of a subscriber (which could be a /48, /60, or /64).

In contrast, in mobile environments, since the 3GPP specifications allocate a /64 per device, it may be sufficient to intercept traffic from the /64 rather than specific /128's (since each time the device establishes a data connection it gets a new IID).

5. Residential Users Security Considerations

The IETF Homenet working group is working on standards and guidelines for IPv6 residential networks; this obviously includes operational security considerations; but this is still work in progress. [RFC8520] is an interesting approach on how firewalls could retrieve and apply specific security policies to some residential devices.

Some residential users have less experience and knowledge about security or networking than experimented operators. As most of the recent hosts (e.g., smartphones, tablets) have IPv6 enabled by default, IPv6 security is important for those users. Even with an IPv4-only ISP, those users can get IPv6 Internet access with the help of Teredo (Section 2.7.2.8) tunnels. Several peer-to-peer programs support IPv6 and those programs can initiate a Teredo tunnel through an IPv4 residential gateway, with the consequence of making the internal host reachable from any IPv6 host on the Internet. It is therefore recommended that all host security products (including personal firewalls) are configured with a dual-stack security policy.

If the residential CPE has IPv6 connectivity, [RFC7084] defines the requirements of an IPv6 CPE and does not take a position on the debate of default IPv6 security policy as defined in [RFC6092]:

- o outbound only: allowing all internally initiated connections and block all externally initiated ones, which is a common default security policy enforced by IPv4 Residential Gateway doing NAT but it also breaks the end-to-end reachability promise of IPv6. [RFC6092] lists several recommendations to design such a CPE;
- o open/transparent: allowing all internally and externally initiated connections, therefore restoring the end-to-end nature of the Internet for IPv6 traffic but having a different security policy for IPv6 than for IPv4.

[RFC6092] REC-49 states that a choice must be given to the user to select one of those two policies.

6. Further Reading

There are several documents that describe in more detail the security of an IPv6 network; these documents are not written by the IETF and

some of them are dated but are listed here for the reader's convenience:

1. Guidelines for the Secure Deployment of IPv6 [NIST]
2. North American IPv6 Task Force Technology Report - IPv6 Security Technology Paper [NAv6TF_Security]
3. IPv6 Security [IPv6_Security_Book]

7. Acknowledgements

The authors would like to thank the following people for their useful comments: Mikael Abrahamsson, Fred Baker, Mustafa Suha Botsali, Mohamed Boucadair, Brian Carpenter, Tim Chown, Lorenzo Colitti, Roman Danyliw (IESG review), Markus de Bruen, Lars Eggert (IESG review), Tobias Fiebig, Fernando Gont, Jeffry Handal, Lee Howard, Benjamin Kaduk (IESG review), Panos Kampanakis, Erik Kline, Jouni Korhonen, Warren Kumari (IESG review), Ted Lemon, Mark Lentczner, Acee Lindem (and his detailed nits), Jen Linkova (and her detailed review), Gyan S. Mishra (the document shepherd), Jordi Palet, Alvaro Retana (IESG review), Zaheduzzaman Sarker (IESG review), Bob Sleigh, Donald Smith, Tarko Tikan, Ole Troan, Bernie Volz (by alphabetical order).

8. Security Considerations

This memo attempts to give an overview of security considerations of operating an IPv6 network both for an IPv6-only network and for networks utilizing the most widely deployed IPv4/IPv6 coexistence strategies.

9. References

9.1. Normative References

- [RFC8200] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", STD 86, RFC 8200, DOI 10.17487/RFC8200, July 2017, <<https://www.rfc-editor.org/info/rfc8200>>.

9.2. Informative References

- [CYMRU] Team, C., "The Bogon Reference", Existing in 2021, <<https://team-cymru.com/community-services/bogon-reference/>>.

[ENTROPYIP]

Foremski, P., Plonka, D., and A. Berger, "Entropy/IP: Uncovering Structure in IPv6 Addresses",
<<http://www.entropy-ip.com/>>.

[europol-cgn]

Europol, "ARE YOU SHARING THE SAME IP ADDRESS AS A CRIMINAL? LAW ENFORCEMENT CALL FOR THE END OF CARRIER GRADE NAT (CGN) TO INCREASE ACCOUNTABILITY ONLINE", October 2017,
<<https://www.europol.europa.eu/newsroom/news/are-you-sharing-same-ip-address-criminal-law-enforcement-call-for-end-of-carrier-grade-nat-cgn-to-increase-accountability-online>>.

[GDPR]

Union, O. J. O. T. E., "Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation)", April 2016,
<<https://eur-lex.europa.eu/eli/reg/2016/679/oj>>.

[I-D.ietf-opsec-ipv6-eh-filtering]

Gont, F. and W. Liu, "Recommendations on the Filtering of IPv6 Packets Containing IPv6 Extension Headers at Transit Routers", draft-ietf-opsec-ipv6-eh-filtering-07 (work in progress), January 2021.

[I-D.kampanakis-6man-ipv6-eh-parsing]

Kampanakis, P., "Implementation Guidelines for parsing IPv6 Extension Headers", draft-kampanakis-6man-ipv6-eh-parsing-01 (work in progress), August 2014.

[IANA-IPFIX]

IANA, "IP Flow Information Export (IPFIX) Entities",
<<http://www.iana.org/assignments/ipfix>>.

[IEEE-802.1X]

IEEE, "IEEE Standard for Local and metropolitan area networks - Port-Based Network Access Control", IEEE Std 802.1X-2010, February 2010.

[IPv6_Security_Book]

Hogg, S. and E. Vyncke, "IPv6 Security", ISBN 1-58705-594-5, Publisher CiscoPress, December 2008.

[KRISTOFF]

Kristoff, J., Ghasemisharif, M., Kanich, C., and J. Polakis, "Plight at the End of the Tunnel: Legacy IPv6 Transition Mechanisms in the Wild", March 2021, <<https://dataplane.org/jtk/publications/kgkp-pam-21.pdf>>.

[NAv6TF_Security]

Kaeo, M., Green, D., Bound, J., and Y. Pouffary, "North American IPv6 Task Force Technology Report - IPv6 Security Technology Paper", 2006, <http://www.ipv6forum.com/dl/white/NAv6TF_Security_Report.pdf>.

[NIST]

Frankel, S., Graveman, R., Pearce, J., and M. Rocks, "Guidelines for the Secure Deployment of IPv6", 2010, <<http://csrc.nist.gov/publications/nistpubs/800-119/sp800-119.pdf>>.

[RADB]

INC., M. N., "RADb The Internet Routing Registry", Existing in 2021, <<https://www.radb.net/>>.

[REY_PF]

Rey, E., "Local Packet Filtering with IPv6", July 2017, <https://labs.ripe.net/Members/enno_rey/local-packet-filtering-with-ipv6>.

[RFC0826]

Plummer, D., "An Ethernet Address Resolution Protocol: Or Converting Network Protocol Addresses to 48.bit Ethernet Address for Transmission on Ethernet Hardware", STD 37, RFC 826, DOI 10.17487/RFC0826, November 1982, <<https://www.rfc-editor.org/info/rfc826>>.

[RFC1918]

Rekhter, Y., Moskowitz, B., Karrenberg, D., de Groot, G., and E. Lear, "Address Allocation for Private Internets", BCP 5, RFC 1918, DOI 10.17487/RFC1918, February 1996, <<https://www.rfc-editor.org/info/rfc1918>>.

[RFC2131]

Droms, R., "Dynamic Host Configuration Protocol", RFC 2131, DOI 10.17487/RFC2131, March 1997, <<https://www.rfc-editor.org/info/rfc2131>>.

[RFC2460]

Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, DOI 10.17487/RFC2460, December 1998, <<https://www.rfc-editor.org/info/rfc2460>>.

[RFC2529]

Carpenter, B. and C. Jung, "Transmission of IPv6 over IPv4 Domains without Explicit Tunnels", RFC 2529, DOI 10.17487/RFC2529, March 1999, <<https://www.rfc-editor.org/info/rfc2529>>.

- [RFC2663] Srisuresh, P. and M. Holdrege, "IP Network Address Translator (NAT) Terminology and Considerations", RFC 2663, DOI 10.17487/RFC2663, August 1999, <<https://www.rfc-editor.org/info/rfc2663>>.
- [RFC2784] Farinacci, D., Li, T., Hanks, S., Meyer, D., and P. Traina, "Generic Routing Encapsulation (GRE)", RFC 2784, DOI 10.17487/RFC2784, March 2000, <<https://www.rfc-editor.org/info/rfc2784>>.
- [RFC2827] Ferguson, P. and D. Senie, "Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing", BCP 38, RFC 2827, DOI 10.17487/RFC2827, May 2000, <<https://www.rfc-editor.org/info/rfc2827>>.
- [RFC2866] Rigney, C., "RADIUS Accounting", RFC 2866, DOI 10.17487/RFC2866, June 2000, <<https://www.rfc-editor.org/info/rfc2866>>.
- [RFC3056] Carpenter, B. and K. Moore, "Connection of IPv6 Domains via IPv4 Clouds", RFC 3056, DOI 10.17487/RFC3056, February 2001, <<https://www.rfc-editor.org/info/rfc3056>>.
- [RFC3068] Huitema, C., "An Anycast Prefix for 6to4 Relay Routers", RFC 3068, DOI 10.17487/RFC3068, June 2001, <<https://www.rfc-editor.org/info/rfc3068>>.
- [RFC3627] Savola, P., "Use of /127 Prefix Length Between Routers Considered Harmful", RFC 3627, DOI 10.17487/RFC3627, September 2003, <<https://www.rfc-editor.org/info/rfc3627>>.
- [RFC3704] Baker, F. and P. Savola, "Ingress Filtering for Multihomed Networks", BCP 84, RFC 3704, DOI 10.17487/RFC3704, March 2004, <<https://www.rfc-editor.org/info/rfc3704>>.
- [RFC3756] Nikander, P., Ed., Kempf, J., and E. Nordmark, "IPv6 Neighbor Discovery (ND) Trust Models and Threats", RFC 3756, DOI 10.17487/RFC3756, May 2004, <<https://www.rfc-editor.org/info/rfc3756>>.
- [RFC3924] Baker, F., Foster, B., and C. Sharp, "Cisco Architecture for Lawful Intercept in IP Networks", RFC 3924, DOI 10.17487/RFC3924, October 2004, <<https://www.rfc-editor.org/info/rfc3924>>.
- [RFC3964] Savola, P. and C. Patel, "Security Considerations for 6to4", RFC 3964, DOI 10.17487/RFC3964, December 2004, <<https://www.rfc-editor.org/info/rfc3964>>.

- [RFC3971] Arkko, J., Ed., Kempf, J., Zill, B., and P. Nikander, "SEcure Neighbor Discovery (SEND)", RFC 3971, DOI 10.17487/RFC3971, March 2005, <<https://www.rfc-editor.org/info/rfc3971>>.
- [RFC3972] Aura, T., "Cryptographically Generated Addresses (CGA)", RFC 3972, DOI 10.17487/RFC3972, March 2005, <<https://www.rfc-editor.org/info/rfc3972>>.
- [RFC4033] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "DNS Security Introduction and Requirements", RFC 4033, DOI 10.17487/RFC4033, March 2005, <<https://www.rfc-editor.org/info/rfc4033>>.
- [RFC4107] Bellovin, S. and R. Housley, "Guidelines for Cryptographic Key Management", BCP 107, RFC 4107, DOI 10.17487/RFC4107, June 2005, <<https://www.rfc-editor.org/info/rfc4107>>.
- [RFC4193] Hinden, R. and B. Haberman, "Unique Local IPv6 Unicast Addresses", RFC 4193, DOI 10.17487/RFC4193, October 2005, <<https://www.rfc-editor.org/info/rfc4193>>.
- [RFC4293] Routhier, S., Ed., "Management Information Base for the Internet Protocol (IP)", RFC 4293, DOI 10.17487/RFC4293, April 2006, <<https://www.rfc-editor.org/info/rfc4293>>.
- [RFC4301] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, DOI 10.17487/RFC4301, December 2005, <<https://www.rfc-editor.org/info/rfc4301>>.
- [RFC4302] Kent, S., "IP Authentication Header", RFC 4302, DOI 10.17487/RFC4302, December 2005, <<https://www.rfc-editor.org/info/rfc4302>>.
- [RFC4303] Kent, S., "IP Encapsulating Security Payload (ESP)", RFC 4303, DOI 10.17487/RFC4303, December 2005, <<https://www.rfc-editor.org/info/rfc4303>>.
- [RFC4364] Rosen, E. and Y. Rekhter, "BGP/MPLS IP Virtual Private Networks (VPNs)", RFC 4364, DOI 10.17487/RFC4364, February 2006, <<https://www.rfc-editor.org/info/rfc4364>>.
- [RFC4380] Huitema, C., "Teredo: Tunneling IPv6 over UDP through Network Address Translations (NATs)", RFC 4380, DOI 10.17487/RFC4380, February 2006, <<https://www.rfc-editor.org/info/rfc4380>>.

- [RFC4381] Behringer, M., "Analysis of the Security of BGP/MPLS IP Virtual Private Networks (VPNs)", RFC 4381, DOI 10.17487/RFC4381, February 2006, <<https://www.rfc-editor.org/info/rfc4381>>.
- [RFC4443] Conta, A., Deering, S., and M. Gupta, Ed., "Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification", STD 89, RFC 4443, DOI 10.17487/RFC4443, March 2006, <<https://www.rfc-editor.org/info/rfc4443>>.
- [RFC4552] Gupta, M. and N. Melam, "Authentication/Confidentiality for OSPFv3", RFC 4552, DOI 10.17487/RFC4552, June 2006, <<https://www.rfc-editor.org/info/rfc4552>>.
- [RFC4649] Volz, B., "Dynamic Host Configuration Protocol for IPv6 (DHCPv6) Relay Agent Remote-ID Option", RFC 4649, DOI 10.17487/RFC4649, August 2006, <<https://www.rfc-editor.org/info/rfc4649>>.
- [RFC4659] De Clercq, J., Ooms, D., Carugi, M., and F. Le Faucheur, "BGP-MPLS IP Virtual Private Network (VPN) Extension for IPv6 VPN", RFC 4659, DOI 10.17487/RFC4659, September 2006, <<https://www.rfc-editor.org/info/rfc4659>>.
- [RFC4795] Aboba, B., Thaler, D., and L. Esibov, "Link-local Multicast Name Resolution (LLMNR)", RFC 4795, DOI 10.17487/RFC4795, January 2007, <<https://www.rfc-editor.org/info/rfc4795>>.
- [RFC4798] De Clercq, J., Ooms, D., Prevost, S., and F. Le Faucheur, "Connecting IPv6 Islands over IPv4 MPLS Using IPv6 Provider Edge Routers (6PE)", RFC 4798, DOI 10.17487/RFC4798, February 2007, <<https://www.rfc-editor.org/info/rfc4798>>.
- [RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, DOI 10.17487/RFC4861, September 2007, <<https://www.rfc-editor.org/info/rfc4861>>.
- [RFC4864] Van de Velde, G., Hain, T., Droms, R., Carpenter, B., and E. Klein, "Local Network Protection for IPv6", RFC 4864, DOI 10.17487/RFC4864, May 2007, <<https://www.rfc-editor.org/info/rfc4864>>.

- [RFC4890] Davies, E. and J. Mohacsi, "Recommendations for Filtering ICMPv6 Messages in Firewalls", RFC 4890, DOI 10.17487/RFC4890, May 2007, <<https://www.rfc-editor.org/info/rfc4890>>.
- [RFC4942] Davies, E., Krishnan, S., and P. Savola, "IPv6 Transition/Co-existence Security Considerations", RFC 4942, DOI 10.17487/RFC4942, September 2007, <<https://www.rfc-editor.org/info/rfc4942>>.
- [RFC5082] Gill, V., Heasley, J., Meyer, D., Savola, P., Ed., and C. Pignataro, "The Generalized TTL Security Mechanism (GTSM)", RFC 5082, DOI 10.17487/RFC5082, October 2007, <<https://www.rfc-editor.org/info/rfc5082>>.
- [RFC5214] Templin, F., Gleeson, T., and D. Thaler, "Intra-Site Automatic Tunnel Addressing Protocol (ISATAP)", RFC 5214, DOI 10.17487/RFC5214, March 2008, <<https://www.rfc-editor.org/info/rfc5214>>.
- [RFC5340] Coltun, R., Ferguson, D., Moy, J., and A. Lindem, "OSPF for IPv6", RFC 5340, DOI 10.17487/RFC5340, July 2008, <<https://www.rfc-editor.org/info/rfc5340>>.
- [RFC5635] Kumari, W. and D. McPherson, "Remote Triggered Black Hole Filtering with Unicast Reverse Path Forwarding (uRPF)", RFC 5635, DOI 10.17487/RFC5635, August 2009, <<https://www.rfc-editor.org/info/rfc5635>>.
- [RFC5952] Kawamura, S. and M. Kawashima, "A Recommendation for IPv6 Address Text Representation", RFC 5952, DOI 10.17487/RFC5952, August 2010, <<https://www.rfc-editor.org/info/rfc5952>>.
- [RFC5969] Townsley, W. and O. Troan, "IPv6 Rapid Deployment on IPv4 Infrastructures (6rd) -- Protocol Specification", RFC 5969, DOI 10.17487/RFC5969, August 2010, <<https://www.rfc-editor.org/info/rfc5969>>.
- [RFC6092] Woodyatt, J., Ed., "Recommended Simple Security Capabilities in Customer Premises Equipment (CPE) for Providing Residential IPv6 Internet Service", RFC 6092, DOI 10.17487/RFC6092, January 2011, <<https://www.rfc-editor.org/info/rfc6092>>.
- [RFC6104] Chown, T. and S. Venaas, "Rogue IPv6 Router Advertisement Problem Statement", RFC 6104, DOI 10.17487/RFC6104, February 2011, <<https://www.rfc-editor.org/info/rfc6104>>.

- [RFC6105] Levy-Abegnoli, E., Van de Velde, G., Popoviciu, C., and J. Mohacsi, "IPv6 Router Advertisement Guard", RFC 6105, DOI 10.17487/RFC6105, February 2011, <<https://www.rfc-editor.org/info/rfc6105>>.
- [RFC6144] Baker, F., Li, X., Bao, C., and K. Yin, "Framework for IPv4/IPv6 Translation", RFC 6144, DOI 10.17487/RFC6144, April 2011, <<https://www.rfc-editor.org/info/rfc6144>>.
- [RFC6146] Bagnulo, M., Matthews, P., and I. van Beijnum, "Stateful NAT64: Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers", RFC 6146, DOI 10.17487/RFC6146, April 2011, <<https://www.rfc-editor.org/info/rfc6146>>.
- [RFC6147] Bagnulo, M., Sullivan, A., Matthews, P., and I. van Beijnum, "DNS64: DNS Extensions for Network Address Translation from IPv6 Clients to IPv4 Servers", RFC 6147, DOI 10.17487/RFC6147, April 2011, <<https://www.rfc-editor.org/info/rfc6147>>.
- [RFC6164] Kohno, M., Nitzan, B., Bush, R., Matsuzaki, Y., Colitti, L., and T. Narten, "Using 127-Bit IPv6 Prefixes on Inter-Router Links", RFC 6164, DOI 10.17487/RFC6164, April 2011, <<https://www.rfc-editor.org/info/rfc6164>>.
- [RFC6169] Krishnan, S., Thaler, D., and J. Hoagland, "Security Concerns with IP Tunneling", RFC 6169, DOI 10.17487/RFC6169, April 2011, <<https://www.rfc-editor.org/info/rfc6169>>.
- [RFC6177] Narten, T., Huston, G., and L. Roberts, "IPv6 Address Assignment to End Sites", BCP 157, RFC 6177, DOI 10.17487/RFC6177, March 2011, <<https://www.rfc-editor.org/info/rfc6177>>.
- [RFC6192] Dugal, D., Pignataro, C., and R. Dunn, "Protecting the Router Control Plane", RFC 6192, DOI 10.17487/RFC6192, March 2011, <<https://www.rfc-editor.org/info/rfc6192>>.
- [RFC6221] Miles, D., Ed., Ooghe, S., Dec, W., Krishnan, S., and A. Kavanagh, "Lightweight DHCPv6 Relay Agent", RFC 6221, DOI 10.17487/RFC6221, May 2011, <<https://www.rfc-editor.org/info/rfc6221>>.
- [RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <<https://www.rfc-editor.org/info/rfc6241>>.

- [RFC6264] Jiang, S., Guo, D., and B. Carpenter, "An Incremental Carrier-Grade NAT (CGN) for IPv6 Transition", RFC 6264, DOI 10.17487/RFC6264, June 2011, <<https://www.rfc-editor.org/info/rfc6264>>.
- [RFC6269] Ford, M., Ed., Boucadair, M., Durand, A., Levis, P., and P. Roberts, "Issues with IP Address Sharing", RFC 6269, DOI 10.17487/RFC6269, June 2011, <<https://www.rfc-editor.org/info/rfc6269>>.
- [RFC6296] Wasserman, M. and F. Baker, "IPv6-to-IPv6 Network Prefix Translation", RFC 6296, DOI 10.17487/RFC6296, June 2011, <<https://www.rfc-editor.org/info/rfc6296>>.
- [RFC6302] Durand, A., Gashinsky, I., Lee, D., and S. Sheppard, "Logging Recommendations for Internet-Facing Servers", BCP 162, RFC 6302, DOI 10.17487/RFC6302, June 2011, <<https://www.rfc-editor.org/info/rfc6302>>.
- [RFC6324] Nakibly, G. and F. Templin, "Routing Loop Attack Using IPv6 Automatic Tunnels: Problem Statement and Proposed Mitigations", RFC 6324, DOI 10.17487/RFC6324, August 2011, <<https://www.rfc-editor.org/info/rfc6324>>.
- [RFC6333] Durand, A., Droms, R., Woodyatt, J., and Y. Lee, "Dual-Stack Lite Broadband Deployments Following IPv4 Exhaustion", RFC 6333, DOI 10.17487/RFC6333, August 2011, <<https://www.rfc-editor.org/info/rfc6333>>.
- [RFC6343] Carpenter, B., "Advisory Guidelines for 6to4 Deployment", RFC 6343, DOI 10.17487/RFC6343, August 2011, <<https://www.rfc-editor.org/info/rfc6343>>.
- [RFC6434] Jankiewicz, E., Loughney, J., and T. Narten, "IPv6 Node Requirements", RFC 6434, DOI 10.17487/RFC6434, December 2011, <<https://www.rfc-editor.org/info/rfc6434>>.
- [RFC6459] Korhonen, J., Ed., Soininen, J., Patil, B., Savolainen, T., Bajko, G., and K. Iisakkila, "IPv6 in 3rd Generation Partnership Project (3GPP) Evolved Packet System (EPS)", RFC 6459, DOI 10.17487/RFC6459, January 2012, <<https://www.rfc-editor.org/info/rfc6459>>.
- [RFC6547] George, W., "RFC 3627 to Historic Status", RFC 6547, DOI 10.17487/RFC6547, February 2012, <<https://www.rfc-editor.org/info/rfc6547>>.

- [RFC6564] Krishnan, S., Woodyatt, J., Kline, E., Hoagland, J., and M. Bhatia, "A Uniform Format for IPv6 Extension Headers", RFC 6564, DOI 10.17487/RFC6564, April 2012, <<https://www.rfc-editor.org/info/rfc6564>>.
- [RFC6583] Gashinsky, I., Jaeggli, J., and W. Kumari, "Operational Neighbor Discovery Problems", RFC 6583, DOI 10.17487/RFC6583, March 2012, <<https://www.rfc-editor.org/info/rfc6583>>.
- [RFC6598] Weil, J., Kuarsingh, V., Donley, C., Liljenstolpe, C., and M. Azinger, "IANA-Reserved IPv4 Prefix for Shared Address Space", BCP 153, RFC 6598, DOI 10.17487/RFC6598, April 2012, <<https://www.rfc-editor.org/info/rfc6598>>.
- [RFC6620] Nordmark, E., Bagnulo, M., and E. Levy-Abegnoli, "FCFS SAVI: First-Come, First-Served Source Address Validation Improvement for Locally Assigned IPv6 Addresses", RFC 6620, DOI 10.17487/RFC6620, May 2012, <<https://www.rfc-editor.org/info/rfc6620>>.
- [RFC6666] Hilliard, N. and D. Freedman, "A Discard Prefix for IPv6", RFC 6666, DOI 10.17487/RFC6666, August 2012, <<https://www.rfc-editor.org/info/rfc6666>>.
- [RFC6762] Cheshire, S. and M. Krochmal, "Multicast DNS", RFC 6762, DOI 10.17487/RFC6762, February 2013, <<https://www.rfc-editor.org/info/rfc6762>>.
- [RFC6763] Cheshire, S. and M. Krochmal, "DNS-Based Service Discovery", RFC 6763, DOI 10.17487/RFC6763, February 2013, <<https://www.rfc-editor.org/info/rfc6763>>.
- [RFC6775] Shelby, Z., Ed., Chakrabarti, S., Nordmark, E., and C. Bormann, "Neighbor Discovery Optimization for IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs)", RFC 6775, DOI 10.17487/RFC6775, November 2012, <<https://www.rfc-editor.org/info/rfc6775>>.
- [RFC6877] Mawatari, M., Kawashima, M., and C. Byrne, "464XLAT: Combination of Stateful and Stateless Translation", RFC 6877, DOI 10.17487/RFC6877, April 2013, <<https://www.rfc-editor.org/info/rfc6877>>.
- [RFC6888] Perreault, S., Ed., Yamagata, I., Miyakawa, S., Nakagawa, A., and H. Ashida, "Common Requirements for Carrier-Grade NATs (CGNs)", BCP 127, RFC 6888, DOI 10.17487/RFC6888, April 2013, <<https://www.rfc-editor.org/info/rfc6888>>.

- [RFC6939] Halwasia, G., Bhandari, S., and W. Dec, "Client Link-Layer Address Option in DHCPv6", RFC 6939, DOI 10.17487/RFC6939, May 2013, <<https://www.rfc-editor.org/info/rfc6939>>.
- [RFC6964] Templin, F., "Operational Guidance for IPv6 Deployment in IPv4 Sites Using the Intra-Site Automatic Tunnel Addressing Protocol (ISATAP)", RFC 6964, DOI 10.17487/RFC6964, May 2013, <<https://www.rfc-editor.org/info/rfc6964>>.
- [RFC6967] Boucadair, M., Touch, J., Levis, P., and R. Penno, "Analysis of Potential Solutions for Revealing a Host Identifier (HOST_ID) in Shared Address Deployments", RFC 6967, DOI 10.17487/RFC6967, June 2013, <<https://www.rfc-editor.org/info/rfc6967>>.
- [RFC6980] Gont, F., "Security Implications of IPv6 Fragmentation with IPv6 Neighbor Discovery", RFC 6980, DOI 10.17487/RFC6980, August 2013, <<https://www.rfc-editor.org/info/rfc6980>>.
- [RFC7010] Liu, B., Jiang, S., Carpenter, B., Venaas, S., and W. George, "IPv6 Site Renumbering Gap Analysis", RFC 7010, DOI 10.17487/RFC7010, September 2013, <<https://www.rfc-editor.org/info/rfc7010>>.
- [RFC7011] Claise, B., Ed., Trammell, B., Ed., and P. Aitken, "Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of Flow Information", STD 77, RFC 7011, DOI 10.17487/RFC7011, September 2013, <<https://www.rfc-editor.org/info/rfc7011>>.
- [RFC7012] Claise, B., Ed. and B. Trammell, Ed., "Information Model for IP Flow Information Export (IPFIX)", RFC 7012, DOI 10.17487/RFC7012, September 2013, <<https://www.rfc-editor.org/info/rfc7012>>.
- [RFC7039] Wu, J., Bi, J., Bagnulo, M., Baker, F., and C. Vogt, Ed., "Source Address Validation Improvement (SAVI) Framework", RFC 7039, DOI 10.17487/RFC7039, October 2013, <<https://www.rfc-editor.org/info/rfc7039>>.
- [RFC7045] Carpenter, B. and S. Jiang, "Transmission and Processing of IPv6 Extension Headers", RFC 7045, DOI 10.17487/RFC7045, December 2013, <<https://www.rfc-editor.org/info/rfc7045>>.

- [RFC7050] Savolainen, T., Korhonen, J., and D. Wing, "Discovery of the IPv6 Prefix Used for IPv6 Address Synthesis", RFC 7050, DOI 10.17487/RFC7050, November 2013, <<https://www.rfc-editor.org/info/rfc7050>>.
- [RFC7084] Singh, H., Beebee, W., Donley, C., and B. Stark, "Basic Requirements for IPv6 Customer Edge Routers", RFC 7084, DOI 10.17487/RFC7084, November 2013, <<https://www.rfc-editor.org/info/rfc7084>>.
- [RFC7112] Gont, F., Manral, V., and R. Bonica, "Implications of Oversized IPv6 Header Chains", RFC 7112, DOI 10.17487/RFC7112, January 2014, <<https://www.rfc-editor.org/info/rfc7112>>.
- [RFC7113] Gont, F., "Implementation Advice for IPv6 Router Advertisement Guard (RA-Guard)", RFC 7113, DOI 10.17487/RFC7113, February 2014, <<https://www.rfc-editor.org/info/rfc7113>>.
- [RFC7123] Gont, F. and W. Liu, "Security Implications of IPv6 on IPv4 Networks", RFC 7123, DOI 10.17487/RFC7123, February 2014, <<https://www.rfc-editor.org/info/rfc7123>>.
- [RFC7166] Bhatia, M., Manral, V., and A. Lindem, "Supporting Authentication Trailer for OSPFv3", RFC 7166, DOI 10.17487/RFC7166, March 2014, <<https://www.rfc-editor.org/info/rfc7166>>.
- [RFC7217] Gont, F., "A Method for Generating Semantically Opaque Interface Identifiers with IPv6 Stateless Address Autoconfiguration (SLAAC)", RFC 7217, DOI 10.17487/RFC7217, April 2014, <<https://www.rfc-editor.org/info/rfc7217>>.
- [RFC7359] Gont, F., "Layer 3 Virtual Private Network (VPN) Tunnel Traffic Leakages in Dual-Stack Hosts/Networks", RFC 7359, DOI 10.17487/RFC7359, August 2014, <<https://www.rfc-editor.org/info/rfc7359>>.
- [RFC7381] Chittimaneni, K., Chown, T., Howard, L., Kuarsingh, V., Pouffary, Y., and E. Vyncke, "Enterprise IPv6 Deployment Guidelines", RFC 7381, DOI 10.17487/RFC7381, October 2014, <<https://www.rfc-editor.org/info/rfc7381>>.

- [RFC7404] Behringer, M. and E. Vyncke, "Using Only Link-Local Addressing inside an IPv6 Network", RFC 7404, DOI 10.17487/RFC7404, November 2014, <<https://www.rfc-editor.org/info/rfc7404>>.
- [RFC7422] Donley, C., Grundemann, C., Sarawat, V., Sundaresan, K., and O. Vautrin, "Deterministic Address Mapping to Reduce Logging in Carrier-Grade NAT Deployments", RFC 7422, DOI 10.17487/RFC7422, December 2014, <<https://www.rfc-editor.org/info/rfc7422>>.
- [RFC7454] Durand, J., Pepelnjak, I., and G. Doering, "BGP Operations and Security", BCP 194, RFC 7454, DOI 10.17487/RFC7454, February 2015, <<https://www.rfc-editor.org/info/rfc7454>>.
- [RFC7513] Bi, J., Wu, J., Yao, G., and F. Baker, "Source Address Validation Improvement (SAVI) Solution for DHCP", RFC 7513, DOI 10.17487/RFC7513, May 2015, <<https://www.rfc-editor.org/info/rfc7513>>.
- [RFC7526] Troan, O. and B. Carpenter, Ed., "Deprecating the Anycast Prefix for 6to4 Relay Routers", BCP 196, RFC 7526, DOI 10.17487/RFC7526, May 2015, <<https://www.rfc-editor.org/info/rfc7526>>.
- [RFC7552] Asati, R., Pignataro, C., Raza, K., Manral, V., and R. Papneja, "Updates to LDP for IPv6", RFC 7552, DOI 10.17487/RFC7552, June 2015, <<https://www.rfc-editor.org/info/rfc7552>>.
- [RFC7597] Troan, O., Ed., Dec, W., Li, X., Bao, C., Matsushima, S., Murakami, T., and T. Taylor, Ed., "Mapping of Address and Port with Encapsulation (MAP-E)", RFC 7597, DOI 10.17487/RFC7597, July 2015, <<https://www.rfc-editor.org/info/rfc7597>>.
- [RFC7599] Li, X., Bao, C., Dec, W., Ed., Troan, O., Matsushima, S., and T. Murakami, "Mapping of Address and Port using Translation (MAP-T)", RFC 7599, DOI 10.17487/RFC7599, July 2015, <<https://www.rfc-editor.org/info/rfc7599>>.
- [RFC7610] Gont, F., Liu, W., and G. Van de Velde, "DHCPv6-Shield: Protecting against Rogue DHCPv6 Servers", BCP 199, RFC 7610, DOI 10.17487/RFC7610, August 2015, <<https://www.rfc-editor.org/info/rfc7610>>.

- [RFC7707] Gont, F. and T. Chown, "Network Reconnaissance in IPv6 Networks", RFC 7707, DOI 10.17487/RFC7707, March 2016, <<https://www.rfc-editor.org/info/rfc7707>>.
- [RFC7721] Cooper, A., Gont, F., and D. Thaler, "Security and Privacy Considerations for IPv6 Address Generation Mechanisms", RFC 7721, DOI 10.17487/RFC7721, March 2016, <<https://www.rfc-editor.org/info/rfc7721>>.
- [RFC7772] Yourtchenko, A. and L. Colitti, "Reducing Energy Consumption of Router Advertisements", BCP 202, RFC 7772, DOI 10.17487/RFC7772, February 2016, <<https://www.rfc-editor.org/info/rfc7772>>.
- [RFC7785] Vinapamula, S. and M. Boucadair, "Recommendations for Prefix Binding in the Context of Software Dual-Stack Lite", RFC 7785, DOI 10.17487/RFC7785, February 2016, <<https://www.rfc-editor.org/info/rfc7785>>.
- [RFC7824] Krishnan, S., Mrugalski, T., and S. Jiang, "Privacy Considerations for DHCPv6", RFC 7824, DOI 10.17487/RFC7824, May 2016, <<https://www.rfc-editor.org/info/rfc7824>>.
- [RFC7844] Huitema, C., Mrugalski, T., and S. Krishnan, "Anonymity Profiles for DHCP Clients", RFC 7844, DOI 10.17487/RFC7844, May 2016, <<https://www.rfc-editor.org/info/rfc7844>>.
- [RFC7857] Penno, R., Perreault, S., Boucadair, M., Ed., Sivakumar, S., and K. Naito, "Updates to Network Address Translation (NAT) Behavioral Requirements", BCP 127, RFC 7857, DOI 10.17487/RFC7857, April 2016, <<https://www.rfc-editor.org/info/rfc7857>>.
- [RFC7872] Gont, F., Linkova, J., Chown, T., and W. Liu, "Observations on the Dropping of Packets with IPv6 Extension Headers in the Real World", RFC 7872, DOI 10.17487/RFC7872, June 2016, <<https://www.rfc-editor.org/info/rfc7872>>.
- [RFC7915] Bao, C., Li, X., Baker, F., Anderson, T., and F. Gont, "IP/ICMP Translation Algorithm", RFC 7915, DOI 10.17487/RFC7915, June 2016, <<https://www.rfc-editor.org/info/rfc7915>>.

- [RFC7934] Colitti, L., Cerf, V., Cheshire, S., and D. Schinazi, "Host Address Availability Recommendations", BCP 204, RFC 7934, DOI 10.17487/RFC7934, July 2016, <<https://www.rfc-editor.org/info/rfc7934>>.
- [RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", RFC 8040, DOI 10.17487/RFC8040, January 2017, <<https://www.rfc-editor.org/info/rfc8040>>.
- [RFC8064] Gont, F., Cooper, A., Thaler, D., and W. Liu, "Recommendation on Stable IPv6 Interface Identifiers", RFC 8064, DOI 10.17487/RFC8064, February 2017, <<https://www.rfc-editor.org/info/rfc8064>>.
- [RFC8177] Lindem, A., Ed., Qu, Y., Yeung, D., Chen, I., and J. Zhang, "YANG Data Model for Key Chains", RFC 8177, DOI 10.17487/RFC8177, June 2017, <<https://www.rfc-editor.org/info/rfc8177>>.
- [RFC8190] Bonica, R., Cotton, M., Haberman, B., and L. Vegoda, "Updates to the Special-Purpose IP Address Registries", BCP 153, RFC 8190, DOI 10.17487/RFC8190, June 2017, <<https://www.rfc-editor.org/info/rfc8190>>.
- [RFC8210] Bush, R. and R. Austein, "The Resource Public Key Infrastructure (RPKI) to Router Protocol, Version 1", RFC 8210, DOI 10.17487/RFC8210, September 2017, <<https://www.rfc-editor.org/info/rfc8210>>.
- [RFC8273] Brzozowski, J. and G. Van de Velde, "Unique IPv6 Prefix per Host", RFC 8273, DOI 10.17487/RFC8273, December 2017, <<https://www.rfc-editor.org/info/rfc8273>>.
- [RFC8343] Bjorklund, M., "A YANG Data Model for Interface Management", RFC 8343, DOI 10.17487/RFC8343, March 2018, <<https://www.rfc-editor.org/info/rfc8343>>.
- [RFC8344] Bjorklund, M., "A YANG Data Model for IP Management", RFC 8344, DOI 10.17487/RFC8344, March 2018, <<https://www.rfc-editor.org/info/rfc8344>>.
- [RFC8415] Mrugalski, T., Siodelski, M., Volz, B., Yourtchenko, A., Richardson, M., Jiang, S., Lemon, T., and T. Winters, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 8415, DOI 10.17487/RFC8415, November 2018, <<https://www.rfc-editor.org/info/rfc8415>>.

- [RFC8504] Chown, T., Loughney, J., and T. Winters, "IPv6 Node Requirements", BCP 220, RFC 8504, DOI 10.17487/RFC8504, January 2019, <<https://www.rfc-editor.org/info/rfc8504>>.
- [RFC8520] Lear, E., Droms, R., and D. Romascanu, "Manufacturer Usage Description Specification", RFC 8520, DOI 10.17487/RFC8520, March 2019, <<https://www.rfc-editor.org/info/rfc8520>>.
- [RFC8541] Litkowski, S., Decraene, B., and M. Horneffer, "Impact of Shortest Path First (SPF) Trigger and Delay Strategies on IGP Micro-loops", RFC 8541, DOI 10.17487/RFC8541, March 2019, <<https://www.rfc-editor.org/info/rfc8541>>.
- [RFC8981] Gont, F., Krishnan, S., Narten, T., and R. Draves, "Temporary Address Extensions for Stateless Address Autoconfiguration in IPv6", RFC 8981, DOI 10.17487/RFC8981, February 2021, <<https://www.rfc-editor.org/info/rfc8981>>.
- [SCANNING] Barnes, R., Altmann, R., and D. Kerr, "Mapping the Great Void - Smarter scanning for IPv6", February 2012, <http://www.caida.org/workshops/isma/1202/slides/aims1202_rbarnes.pdf>.
- [WEBER_VPN] Weber, J., "Dynamic IPv6 Prefix - Problems and VPNs", March 2018, <<https://blog.webernetz.net/wp-content/uploads/2018/03/TR18-Johannes-Weber-Dynamic-IPv6-Prefix-Problems-and-VPNs.pdf>>.

Authors' Addresses

Eric Vyncke
Cisco
De Kleetlaan 6a
Diegem 1831
Belgium

Phone: +32 2 778 4677
Email: evyncke@cisco.com

Kiran Kumar
Square
1455 Market Street, Suite 600
San Francisco 94103
United States of America

Email: kk.chittimaneni@gmail.com

Merike Kaeo
Double Shot Security
3518 Fremont Ave N 363
Seattle 98103
United States of America

Phone: +12066696394
Email: merike@doubleshotsecurity.com

Enno Rey
ERNW
Carl-Bosch-Str. 4
Heidelberg, Baden-Wuerttemberg 69115
Germany

Phone: +49 6221 480390
Email: erey@ernw.de

Network Working Group
Internet-Draft
Intended status: Informational
Expires: January 4, 2018

M. Kuehlewind
ETH Zurich
T. Pauly
C. Wood
Apple Inc.
July 03, 2017

Separating Crypto Negotiation and Communication
draft-kuehlewind-taps-crypto-sep-00

Abstract

Due to the latency involved in connection setup and security handshakes, there is an increasing deployment of cryptographic session resumption mechanisms. While cryptographic context and endpoint capabilities need to be known before encrypted application data can be sent, there is otherwise no technical constraint that the crypto handshake must be performed on the same transport connection. This document recommends a logical separation between the mechanism(s) used to negotiate capabilities and set up encryption context (handshake protocol), the application of encryption and authentication state to data (record protocol), and the associated transport connection(s).

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on January 4, 2018.

Copyright Notice

Copyright (c) 2017 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
2. Terminology	3
3. Protocol Interfaces	4
3.1. Handshake-Transport Interface	5
3.2. Handshake-Record Interface	6
3.3. Transport-Record Interface	6
4. Existing Mappings	6
5. Benefits of Separation	8
5.1. Reducing Connection Latency	9
5.2. Protocol Flexibility	9
5.3. Protocol Capability Negotiation	10
6. IANA Considerations	10
7. Security Considerations	10
8. Acknowledgments	10
9. Informative References	10
Authors' Addresses	11

1. Introduction

Secure transport protocols are generally composed of three pieces:

1. A transport protocol to control the transfer of data.
2. A record protocol to frame, encrypt and/or authenticate data
3. A handshake protocol to negotiate cryptographic secrets.

For ease of deployment and standardization, among other reasons, these constituents are often tightly coupled. For example, in TLS [RFC5246], the handshake protocol depends on the record protocol, and vice versa. However, more recent transport protocols such as QUIC [I-D.ietf-quic-tls] keep these pieces separate. QUIC uses TLS to negotiate secrets, and _exports_ those secrets to encrypt packets directly.

Separating these pieces is important, as new secure transport protocols increasingly rely on session resumption mechanisms where cryptographic context can be resumed to transmit application data with the first packet without delay for connection setup and negotiation. In the case where there is no cryptographic context available when an application expresses the need to transmit data to a certain endpoint, it must first run the handshake protocol on a transport connection before being able to transmit application data. If the handshake protocol can be separated from the other components, then it can use another transport connection to establish secrets without blocking the application's main transport connection. This also opens up the possibility to run the handshake protocol well in advance of the need to send application data, to avoid unnecessary delays. For example, a client system could maintain a database of endpoints it is likely to communicate with, and establish keying material with a handshake protocol at periodic intervals to ensure fresh keys for new transport connections.

[I-D.moskowitz-sse] proposes a similar approach. However while [I-D.moskowitz-sse] proposes a new protocol to negotiate and maintain long-term cryptographic sessions, this document relies on the use of existing protocols and only discusses requirements for the evolution of these protocols and exchange of information within one endpoint locally.

2. Terminology

- o Transport Protocol: A protocol that can transport messages between two endpoints. This may represent the service offered to applications to allow them to send and receive data before encryption; and also represent the protocol that can transmit handshake data and encrypted records.
- o Handshake Protocol: A protocol that can validate and authenticate endpoints, encrypt and authenticate its negotiation, and ultimately generate keying material.
- o Record Protocol: A protocol that can use keying material to transform messages. A record will generally add a frame around application data, and authenticate and/or encrypt the data.
- o Keying Material: One or more pre-shared keys that can be used to encrypt and authenticate data, generated by a handshake protocol and used by a record protocol.

3. Protocol Interfaces

In traditional models in which the protocols are not separated out into the three elements of handshake, record, and transport protocols, there are two basic approaches to the interactions:

1. The transport protocol provides data to the security protocol and gets back an encrypted version of the data to be sent (handshake and record protocols are combined)
2. The security protocol provides keying material to the transport protocol, and the transport protocol is responsible for encrypting data (transport and record protocols are combined)

By teasing apart all three portions as separate protocols, there end up being six interface points:

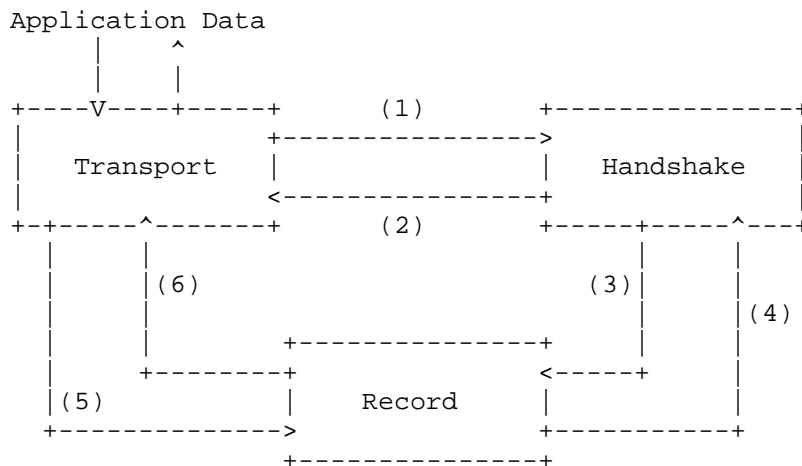


Figure 1: Secure Transport Protocol Components and Interactions

1. A transport protocol depends upon a handshake protocol to establish keying material to protect application data being sent through the transport. The main interface it relies upon is starting the handshake, or ensuring that the material is ready.
2. A handshake protocol depends upon a transport protocol in order to send and receive negotiation messages with the remote peer.
3. A handshake protocol sends its keying material and cryptographic context to the record protocol to use

4. A record protocol may signal state expiration events to a handshake protocol
5. A transport protocol uses a record protocol to send and receive application data
6. A record protocol uses a transport protocol to send and receive encrypted data

3.1. Handshake-Transport Interface

Note that for the purposes of this interface description, it is assumed that the application is primarily interacting with the transport protocol, and thus the handshake protocol interacts with the application primarily through the abstraction of the transport protocol.

- o Start negotiation: The interface MUST provide an indication to start the protocol handshake for key negotiation, and have a way to be notified when the handshake is complete.
- o Identity constraints: The interface MUST allow the application to constrain the identities that it will accept a connection to, such as the hostname it expects to be provided in certificate SAN.
- o Local identities: The interface MUST allow the local identity to be set via a raw private key or interface to one to perform cryptographic operations such as signing and decryption.
- o State changes: The interface SHOULD provide a way for the transport to be notified of important state changes during the protocol execution and session lifetime, e.g., when the handshake begins, ends, or when a key update occurs.
- o Validation: The interface MUST provide a way for the application to participate in the endpoint authentication and validation, which can either be specified as parameters to define how the peer's authentication can be validated, or when the protocol provides the authentication information for the application to inspect directly.
- o Caching domain and lifetime: The application SHOULD be able to specify the instances of the protocol that can share cached keys, as well as the lifetime of cached resources.
- o The protocol SHOULD allow applications to negotiate application protocols and related information.

- o The protocol SHOULD allow applications to specify negotiable cryptographic algorithm suites.
- o The protocol SHOULD expose the peer's identity information.

3.2. Handshake-Record Interface

- o Key export: The interface MUST provide a way to export keying material from a handshake protocol to a record protocol with well-defined cryptographic properties, e.g., "forward-secure" or "perfectly forward secure"
- o Key lifetime and rotation: The interface MUST provide a way for the handshake protocol to define key lifetime bounds in terms of `_time_` or `_bytes encrypted_` and, additionally, provide a way to forcefully update cryptographic session keys at will. The record protocol MUST be able to signal back to the handshake protocol that a lifetime has been reached and that rotation is required. These values SHOULD be configurable by the application.

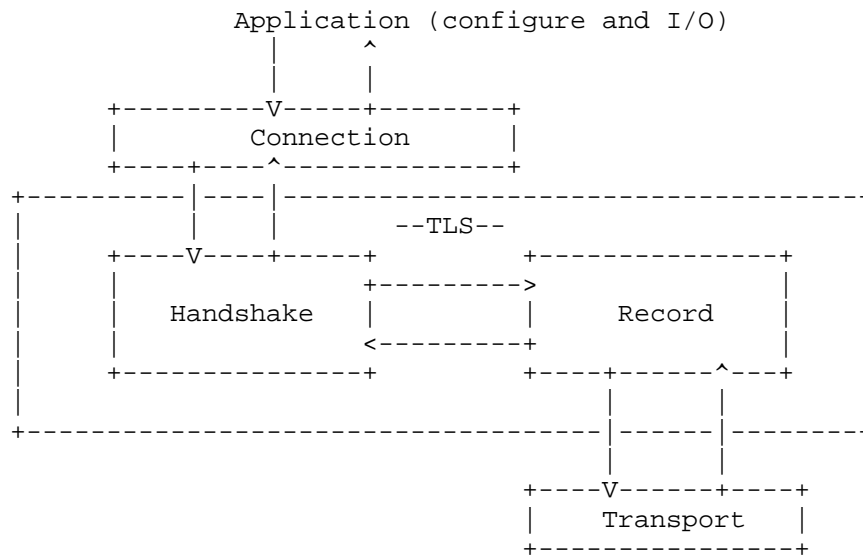
3.3. Transport-Record Interface

- o Transform data: The interface MUST provide a way to send raw application data from the transport protocol to a record protocol to transform it based on the keying material. This data is then sent out by the transport protocol. The same applies for inbound data, in which inbound transport data is transformed by the record protocol into raw application data.
- o Reliability: The transport MUST specify if messages are transmitted reliable and in order.
- o Maximum message size (optional): The transport may specify a maximum message size for the encrypted data if e.g. a datagram transport is used

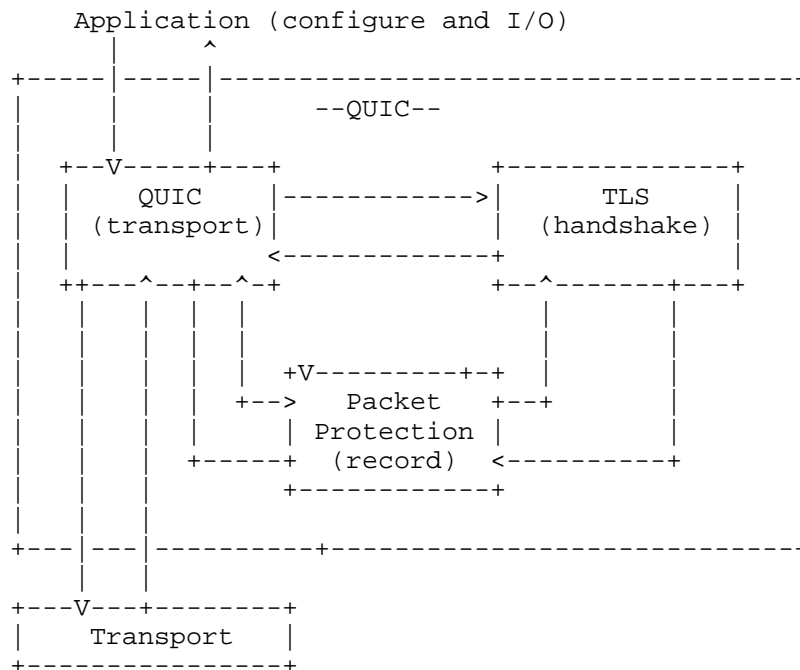
4. Existing Mappings

In this section we document existing mappings between common transport security protocols and the three components described in Section I.

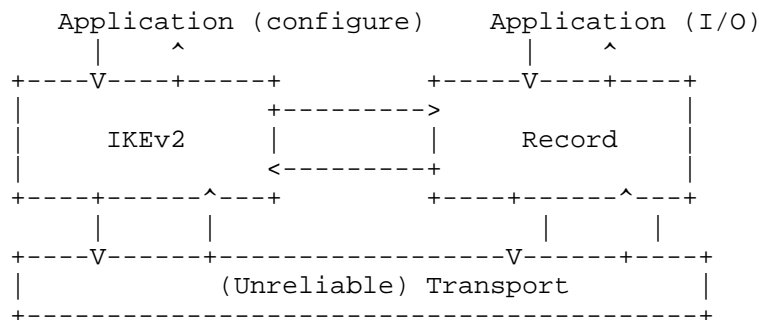
- o TLS/DTLS: TLS [RFC5246] and DTLS [RFC6347] is a combination of a handshake and record protocol, with a dependency on some underlying transport.



- o QUIC + TLS: The emerging QUIC standard is decomposed into the three pieces outlined in Section I [I-D.ietf-quic-tls]. TLS is used as the handshake protocol running on a dedicated QUIC stream, a QUIC-specific record protocol encrypts and encapsulates stream frames, and the main QUIC component handles the transport of these frames.



- o **IKEv2 + ESP:** IKEv2 [RFC7296] is a handshake protocol commonly used to establish keys for use in IPsec (often VPN) deployments. It is already a distinct protocol from its commonly paired record protocol, which is ESP [RFC4303]. ESP encrypts and authenticates IP datagrams, and sends them as datagrams over a transport mechanism such, e.g., IP or UDP.



5. Benefits of Separation

5.1. Reducing Connection Latency

One of the clearest benefits of separating the handshake protocol from the record protocol is that the handshake can be performed out-of-band from the application's data transfer. This should essentially reduce the number of RTTs required before being able to send data by the full length of the handshake (which is commonly 1 or 2 RTTs in the best cases for TLS 1.2 and IKEv2, potentially more if cookie challenges or extended authentication are required).

To avoid long-lived transport connections that wouldn't be actively used, and thus would be vulnerable to timeouts on NATs or firewalls, an obvious approach to separating the handshake and record protocols is to use different transport connections for the early handshake and the data transfer. However, this approach of using separate connections will not always save RTTs if the handshake and data transfer are back-to-back. Each connection may require its own transport protocol handshake, and if the data transfer must wait for two transport protocols to establish and the cryptographic handshake to be finished before sending, then it may experience higher latency. Implementations SHOULD avoid this by either allowing the handshake and record protocols to share a single transport connection or open two connections in parallel when the handshake protocol has not pre-fetched keys. Latency benefits, however, can even be achieved when ensuring that this scenario does not occur by always having the handshake protocol refresh the keys whenever old ones are near expiry.

5.2. Protocol Flexibility

Separation of the handshake, record, and transport protocols also allows for more flexible composition of protocols with one another. If a deployment uses a handshake protocol like TLS, which requires a stream-based transport protocol like TCP, separation of protocols will allow it to use the resulting keys for record protocols that run on datagram transport protocols like UDP.

This flexibility may be useful for implementations that are optimizing for packet size by choosing minimal/lightweight record protocols, while being able to use commonly supported handshake protocols like TLS. One example here is the approach of a VPN tunnel that uses ESP or Diet-ESP [I-D.mglt-ipsecme-diet-esp] to encrypt datagrams, but uses TLS for establishing keys.

5.3. Protocol Capability Negotiation

Enabling the use of a different transport protocol for the actual data transmission than for the cryptographic handshakes opens also the possibility to negotiate protocol capabilities for the data transmission. For TLS, usually TCP is the appropriate transport protocol to use, as it is also widely supported by endpoints. Allowing an endpoint to indicate the support of other, new transport protocols within the TCP connection that is used for the handshake, provides a dynamic transition path to enable easy deployment of new protocols.

6. IANA Considerations

This document has on request to IANA.

7. Security Considerations

(editor's note: this section will be added later. However, this document discusses the use of cryptographic context for transport connections and as such it has security relevant consideration within the whole document.)

8. Acknowledgments

This work is partially supported by the European Commission under Horizon 2020 grant agreement no. 688421 Measurement and Architecture for a Middleboxed Internet (MAMI), and by the Swiss State Secretariat for Education, Research, and Innovation under contract no. 15.0268. This support does not imply endorsement.

9. Informative References

[I-D.ietf-quic-tls]

Thomson, M. and S. Turner, "Using Transport Layer Security (TLS) to Secure QUIC", draft-ietf-quic-tls-04 (work in progress), June 2017.

[I-D.mglt-ipsecme-diet-esp]

Migault, D., Guggemos, T., and C. Bormann, "ESP Header Compression and Diet-ESP", draft-mglt-ipsecme-diet-esp-04 (work in progress), June 2017.

[I-D.moskowitz-sse]

Moskowitz, R., Faynberg, I., Lu, H., Hares, S., and P. Giacomin, "Session Security Envelope", draft-moskowitz-sse-05 (work in progress), June 2017.

- [RFC4303] Kent, S., "IP Encapsulating Security Payload (ESP)", RFC 4303, DOI 10.17487/RFC4303, December 2005, <<http://www.rfc-editor.org/info/rfc4303>>.
- [RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, DOI 10.17487/RFC5246, August 2008, <<http://www.rfc-editor.org/info/rfc5246>>.
- [RFC6347] Rescorla, E. and N. Modadugu, "Datagram Transport Layer Security Version 1.2", RFC 6347, DOI 10.17487/RFC6347, January 2012, <<http://www.rfc-editor.org/info/rfc6347>>.
- [RFC7296] Kaufman, C., Hoffman, P., Nir, Y., Eronen, P., and T. Kivinen, "Internet Key Exchange Protocol Version 2 (IKEv2)", STD 79, RFC 7296, DOI 10.17487/RFC7296, October 2014, <<http://www.rfc-editor.org/info/rfc7296>>.
- [RFC7301] Friedl, S., Popov, A., Langley, A., and E. Stephan, "Transport Layer Security (TLS) Application-Layer Protocol Negotiation Extension", RFC 7301, DOI 10.17487/RFC7301, July 2014, <<http://www.rfc-editor.org/info/rfc7301>>.

Authors' Addresses

Mirja Kuehlewind
ETH Zurich
Gloriastrasse 35
8092 Zurich
Switzerland

Email: mirja.kuehlewind@tik.ee.ethz.ch

Tommy Pauly
Apple Inc.
1 Infinite Loop
Cupertino, California 95014
United States of America

Email: tpauly@apple.com

Christopher A. Wood
Apple Inc.
1 Infinite Loop
Cupertino, California 95014
United States of America

Email: cawood@apple.com

Network Working Group
Internet-Draft
Intended status: Informational
Expires: January 1, 2019

M. Kuehlewind
ETH Zurich
T. Pauly
C. Wood
Apple Inc.
June 30, 2018

Separating Crypto Negotiation and Communication
draft-kuehlewind-taps-crypto-sep-03

Abstract

Secure transport protocols often consist of three logically distinct components: transport, control (handshake), and record protection. Typically, such a protocol contains a single module that is responsible for all three functions. However, in many cases, this coupling is unnecessary. For example, while cryptographic context and endpoint capabilities need to be known before encrypted application data can be sent on a specific transport connection, there is otherwise no technical constraint that a cryptographic handshake must be performed on said connection. This document recommends a logical separation between transport, control, and record components of secure transport protocols. We compare existing protocols such as Transport Layer Security, QUIC, and IKEv2+ESP in the context of this logical separation.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on January 1, 2019.

Copyright Notice

Copyright (c) 2018 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
2. Terminology	3
3. Protocol Interfaces	4
3.1. Control-Transport Interface	5
3.1.1. Passive Configuration Interface	5
3.1.2. Active Control and Introspection Interface	6
3.2. Control-Record Interface	6
3.3. Transport-Record Interface	6
4. Existing Mappings	7
5. Benefits of Separation	10
5.1. Reducing Connection Latency	10
5.2. Protocol Flexibility	11
5.3. Protocol Capability and Upgrade Negotiation	11
6. Transport Service Architecture Integration	11
7. IANA Considerations	12
8. Security Considerations	12
9. Acknowledgments	12
10. Informative References	12
Authors' Addresses	13

1. Introduction

Secure transport protocols are generally composed of three pieces:

1. A transport protocol to handle the transfer of data.
2. A record protocol to frame, encrypt and/or authenticate data
3. A control protocol to perform cryptographic handshakes, negotiate shared secrets, and maintain state during the lifetime of cryptographic session including session resumption and key

refreshment. (In the context of TLS, the control protocol is called the handshake protocol.)

For ease of deployment and standardization, among other reasons, these constituents are often tightly coupled. For example, in TLS [RFC5246], the control protocol depends on the record protocol, and vice versa. However, more recent transport protocols such as QUIC [I-D.ietf-quic-tls] keep these pieces separate. For example, QUIC uses TLS to negotiate secrets, and exports those secrets to encrypt packets independent of TLS.

Separating these pieces is important, as new secure transport protocols increasingly rely on session resumption mechanisms where cryptographic context can be resumed to transmit application data with the first packet without delay for connection setup and negotiation. In the case where there is no cryptographic context available when an application expresses the need to transmit data to a certain endpoint, it must first run the control protocol on a transport connection before being able to transmit application data. If the control protocol can be separated from the other components, then it can use another transport connection to establish secrets without blocking the application's main transport connection. This also opens up the possibility to run the control protocol well in advance of the need to send application data, to avoid unnecessary delays. For example, a client system could maintain a database of endpoints it is likely to communicate with, and establish keying material with a control protocol at periodic intervals to ensure fresh keys for new transport connections.

[I-D.moskowitz-sse] proposes a similar approach. However while [I-D.moskowitz-sse] proposes a new protocol to negotiate and maintain long-term cryptographic sessions, this document relies on the use of existing protocols and only discusses requirements for the evolution of these protocols and exchange of information within one endpoint locally.

2. Terminology

- o Transport Protocol: A protocol that can transport messages between two endpoints. This may represent the service offered to applications to allow them to send and receive data before encryption; and also represent the protocol that can transmit control data and encrypted records.
- o Control Protocol: A protocol that performs a cryptographic handshake and, in addition, can validate and authenticate endpoints, encrypt and authenticate its negotiation, and ultimately generate keying material.

- o Record Protocol: A protocol that can use keying material to transform messages. A record will generally add a frame around application data, and authenticate and/or encrypt the data.
- o Keying Material: A shared secret from which pre-shared keys can be derived and subsequently used to encrypt and authenticate data, generated by a control protocol and used by a record protocol.

3. Protocol Interfaces

In traditional models in which the protocols are not separated out into the three elements of control, record, and transport protocols, there are two basic approaches to the interactions:

1. The transport protocol provides data to the security protocol and gets back an encrypted version of the data to be sent (control and record protocols are combined).
2. The security protocol provides keying material to the transport protocol, and the transport protocol is responsible for encrypting data (transport and record protocols are combined).

By teasing apart all three portions as separate protocols, there end up being six interface points:

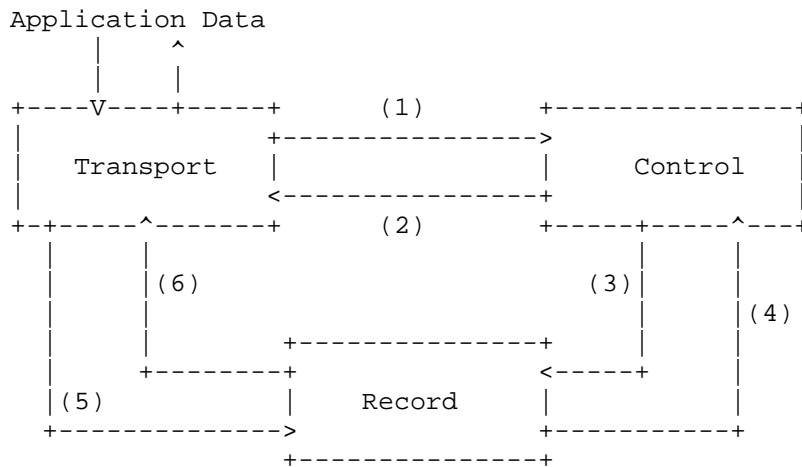


Figure 1: Secure Transport Protocol Components and Interactions

1. A transport protocol depends upon a control protocol to establish keying material to protect application data being sent through the transport. The main interface it relies upon is starting the

control channel, or handshake, or ensuring that the material is ready.

2. A control protocol depends upon a transport protocol in order to send and receive negotiation messages with the remote peer.
3. A control protocol sends its keying material and cryptographic context to the record protocol to use.
4. A record protocol may signal state expiration events to a control protocol.
5. A transport protocol uses a record protocol to send and receive application data.
6. A record protocol uses a transport protocol to send and receive encrypted data.

3.1. Control-Transport Interface

Note that for the purposes of this interface description, it is assumed that the application is primarily interacting with the transport protocol, and thus the control protocol interacts with the application primarily through the abstraction of the transport protocol. Since security protocol interfaces often require pre-connection and active behavior on behalf of clients, we further categorize the following interfaces based on whether they are meant for passive configuration or active control.

3.1.1. Passive Configuration Interface

- o Start negotiation: The interface **MUST** provide an indication to start the protocol handshake for key negotiation, and have a way to be notified when the handshake is complete.
- o Identity constraints: The interface **MUST** allow the application to constrain the identities that it will accept a connection to, such as the hostname it expects to be provided in certificate SAN.
- o Local identities: The interface **MUST** allow the local identity to be set via a raw private key or interface to one to perform cryptographic operations such as signing and decryption.
- o Caching domain and lifetime: The application **SHOULD** be able to specify the instances of the protocol that can share cached keys, as well as the lifetime of cached resources.

- o Pre-shared keying material: The application SHOULD be able to specify pre-share keying material to use to bootstrap connections. The control protocol can pass this directly to the record protocol for use.
- o The protocol SHOULD allow applications to negotiate application protocols and related information.
- o The protocol SHOULD allow applications to specify negotiable cryptographic algorithm suites.

3.1.2. Active Control and Introspection Interface

- o State changes: The interface SHOULD provide a way for the transport to be notified of important state changes during the protocol execution and session lifetime, e.g., when the handshake begins, ends, or when a key update occurs.
- o Validation: The interface MUST provide a way for the application to participate in the endpoint authentication and validation, which can either be specified as parameters to define how the peer's authentication can be validated, or when the protocol provides the authentication information for the application to inspect directly.
- o The protocol SHOULD expose the peer's identity information during and after connection establishment.

3.2. Control-Record Interface

- o Key export: The interface MUST provide a way to export keying material from a control protocol to a record protocol with well-defined cryptographic properties, e.g., "forward-secure."
- o Key lifetime and rotation: The interface MUST provide a way for the control protocol to define key lifetime bounds in terms of `_time_` or `_bytes encrypted_` and, additionally, provide a way to forcefully update cryptographic session keys at will. The record protocol MUST be able to signal back to the control protocol that a lifetime has been reached and that rotation is required. These values SHOULD be configurable by the application.

3.3. Transport-Record Interface

- o Transform data: The interface MUST provide a way to send raw application data from the transport protocol to a record protocol to transform it based on the keying material. This data is then sent out by the transport protocol. The same applies for inbound

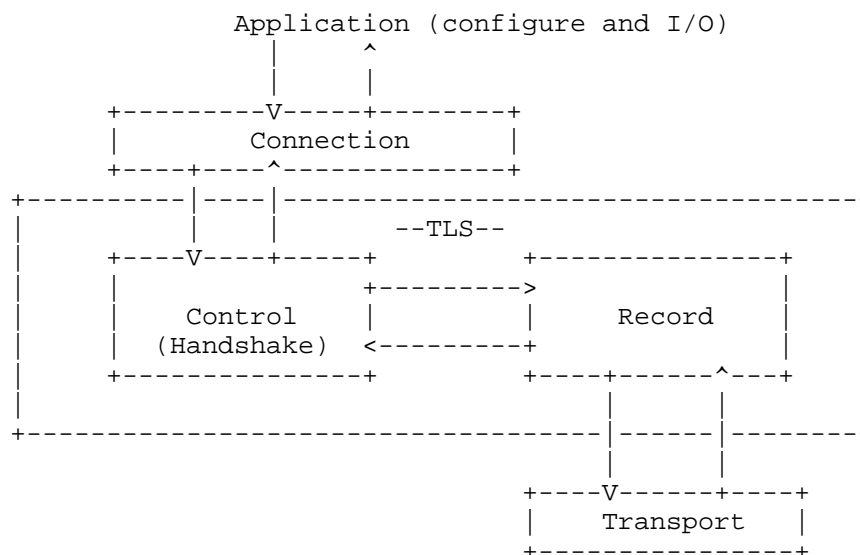
data, in which inbound transport data is transformed by the record protocol into raw application data.

- o Reliability: The transport MUST specify if messages are transmitted reliable and in order.
- o Maximum message size (optional): The transport may specify a maximum message size for the encrypted data if e.g. a datagram transport is used

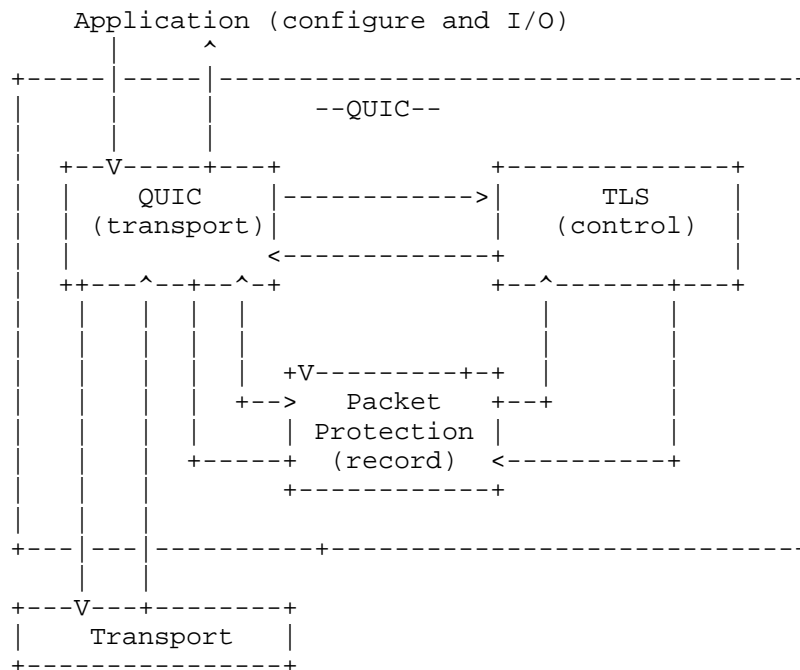
4. Existing Mappings

In this section we document existing mappings between common transport security protocols and the three components described in Section I.

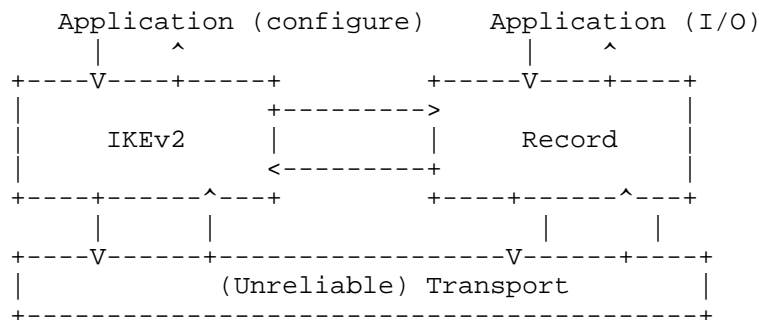
- o TLS/DTLS: TLS [RFC5246] and DTLS [RFC6347] is a combination of a control (handshake) and record protocol, with a dependency on some underlying transport.



- o QUIC + TLS: The emerging QUIC standard is decomposed into the three pieces outlined in Section I [I-D.ietf-quic-tls]. TLS is used as the control protocol running on a dedicated QUIC stream, a QUIC-specific record protocol encrypts and encapsulates stream frames, and the main QUIC component handles the transport of these frames.

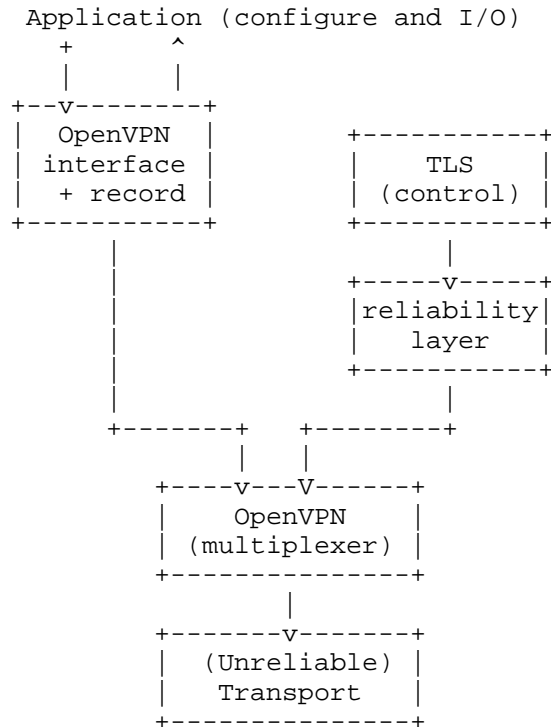


- o IKEv2 + ESP: IKEv2 [RFC7296] is a control protocol commonly used to establish keys for use in IPsec (often VPN) deployments. It is already a distinct protocol from its commonly paired record protocol, which is ESP [RFC4303]. ESP encrypts and authenticates IP datagrams, and sends them as datagrams over a transport mechanism such, e.g., IP or UDP.

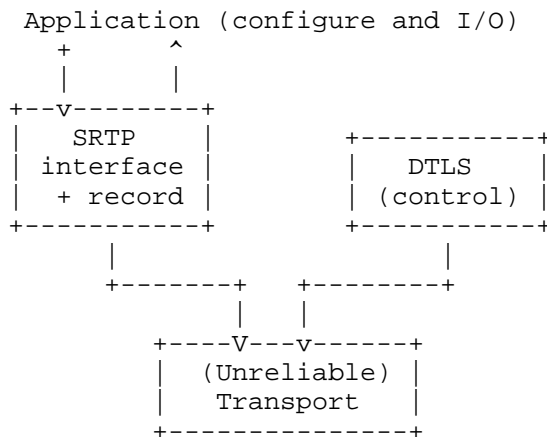


- o OpenVPN [OpenVPN]: OpenVPN consists of two separate stacks - one for TLS, which is used for key exchange and derivation, and the other as an interface to tunnel IP packets over UDP. A common multiplexing layer is used to send TLS and OpenVPN framed packets over an unreliable transport layer. OpenVPN adds a reliability

layer to TLS to ensure packets are sent and processed in order. Running over TCP naturally provides this reliability. After the TLS connection finishes, OpenVPN extracts encryption and authentication keys from TLS, via the PRF, and uses them to encrypt and authenticate IP packets. Packets are framed using a simple length-type-value envelope, wherein the type specifies the contents of the packet, e.g., channel control (TLS ciphertext) bytes.



- o DTLS-SRTP: DTLS [RFC5764] is commonly used as a way to perform mutual authentication and key agreement for SRTP [RFC5763]. (Here, certificates marshal public keys between endpoints. Thus, self-signed certificates may be used if peers do not mutually trust one another, as is common on the Internet.) When DTLS is used, certificate fingerprints are transmitted out-of-band using SIP. Peers typically verify that DTLS-offered certificates match that which are offered over SIP. This prevents active attacks on RTP, but not on the signaling (SIP or WebRTC) channel.



5. Benefits of Separation

5.1. Reducing Connection Latency

One of the clearest benefits of separating the control protocol from the record protocol is that the cryptographic handshake can be performed out-of-band from the application's data transfer. This should essentially reduce the number of RTTs required before being able to send data by the full length of the handshake (which is commonly 1 or 2 RTTs in the best cases for TLS 1.2 and IKEv2, potentially more if cookie challenges or extended authentication are required).

To avoid long-lived transport connections that wouldn't be actively used, and thus would be vulnerable to timeouts on NATs or firewalls, an obvious approach to separating the control and record protocols is to use different transport connections for the early handshake and the data transfer. However, this approach of using separate connections will not always save RTTs if the cryptographic handshake and data transfer are back-to-back. Each connection may require its own transport protocol handshake, and if the data transfer must wait for two transport protocols to establish and the cryptographic handshake to be finished before sending, then it may experience higher latency. Implementations SHOULD avoid this by either allowing the control and record protocols to share a single transport connection or open two connections in parallel when the control protocol has not pre-fetched keys. Latency benefits, however, can even be achieved when ensuring that this scenario does not occur by always having the control protocol refresh the keys whenever old ones are near expiry.

5.2. Protocol Flexibility

Separation of the control, record, and transport protocols also allows for more flexible composition of protocols with one another. If a deployment uses a control protocol like TLS, which requires a stream-based transport protocol like TCP, separation of protocols will allow it to use the resulting keys for record protocols that run on datagram transport protocols like UDP.

This flexibility may be useful for implementations that are optimizing for packet size by choosing minimal/lightweight record protocols, while being able to use commonly supported control protocols like TLS. One example here is the approach of a VPN tunnel that uses ESP or Diet-ESP [I-D.mglt-ipsecme-diet-esp] to encrypt datagrams, but uses TLS for establishing keys. This design is similar to that used by OpenVPN [OpenVPN], as described above.

5.3. Protocol Capability and Upgrade Negotiation

Enabling the use of a different transport protocol for the actual data transmission than for the cryptographic handshakes opens also the possibility to negotiate protocol capabilities for the data transmission. For TLS, usually TCP is the appropriate transport protocol to use, as it is also widely supported by endpoints. Allowing an endpoint to indicate the support of other, new transport protocols within the TCP connection that is used for the cryptographic handshake, provides a dynamic transition path to enable easy deployment of new protocols. Another example is providing an upgrade path from TCP+TLS to QUIC. If TLS could negotiate the use of other transport layers, such as QUIC, applications could perform an abbreviated upgrade from TCP+TLS connections to QUIC, i.e., without doing a full QUIC handshake.

6. Transport Service Architecture Integration

The Transport Services Architecture ([I-D.ietf-taps-arch]) describes a system that can provide transport security functionality behind a common interface. Such systems and their APIs provide applications with the ability to establish connections for sending and receiving data. The lifetime of a connection is comprised of a pre-establishment configuration stage, established (connected) stage, and terminated stage. Pre-establishment properties configured include: Local and Remote Endpoint, protocol selection properties, and specific protocol options. Applications configure security protocols during pre-establishment using the passive interfaces described in Section 3.1. Active control interfaces are exercised during connection establishment, i.e., from pre-establishment to established states. Applications can query connection metadata or state

information, e.g., peer identity information, during and after connection establishment.

7. IANA Considerations

This document has on request to IANA.

8. Security Considerations

(editor's note: this section will be added later. However, this document discusses the use of cryptographic context for transport connections and as such it has security relevant consideration within the whole document.)

9. Acknowledgments

This work is partially supported by the European Commission under Horizon 2020 grant agreement no. 688421 Measurement and Architecture for a Middleboxed Internet (MAMI), and by the Swiss State Secretariat for Education, Research, and Innovation under contract no. 15.0268. This support does not imply endorsement. Thanks to Brian Trammell for reviewing this draft.

10. Informative References

[I-D.ietf-quic-tls]

Thomson, M. and S. Turner, "Using Transport Layer Security (TLS) to Secure QUIC", draft-ietf-quic-tls-13 (work in progress), June 2018.

[I-D.ietf-taps-arch]

Pauly, T., Trammell, B., Brunstrom, A., Fairhurst, G., Perkins, C., Tiesel, P., and C. Wood, "An Architecture for Transport Services", draft-ietf-taps-arch-00 (work in progress), April 2018.

[I-D.mglt-ipsecme-diet-esp]

Migault, D., Guggemos, T., Bormann, C., and D. Schinazi, "ESP Header Compression and Diet-ESP", draft-mglt-ipsecme-diet-esp-06 (work in progress), May 2018.

[I-D.moskowitz-sse]

Moskowitz, R., Faynberg, I., Lu, H., Hares, S., and P. Giacomin, "Session Security Envelope", draft-moskowitz-sse-05 (work in progress), June 2017.

- [OpenVPN] "OpenVPN Security Overview", n.d.,
<<https://openvpn.net/index.php/open-source/documentation/security-overview.html>>.
- [RFC4303] Kent, S., "IP Encapsulating Security Payload (ESP)",
RFC 4303, DOI 10.17487/RFC4303, December 2005,
<<https://www.rfc-editor.org/info/rfc4303>>.
- [RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security
(TLS) Protocol Version 1.2", RFC 5246,
DOI 10.17487/RFC5246, August 2008,
<<https://www.rfc-editor.org/info/rfc5246>>.
- [RFC5763] Fischl, J., Tschofenig, H., and E. Rescorla, "Framework
for Establishing a Secure Real-time Transport Protocol
(SRTP) Security Context Using Datagram Transport Layer
Security (DTLS)", RFC 5763, DOI 10.17487/RFC5763, May
2010, <<https://www.rfc-editor.org/info/rfc5763>>.
- [RFC5764] McGrew, D. and E. Rescorla, "Datagram Transport Layer
Security (DTLS) Extension to Establish Keys for the Secure
Real-time Transport Protocol (SRTP)", RFC 5764,
DOI 10.17487/RFC5764, May 2010,
<<https://www.rfc-editor.org/info/rfc5764>>.
- [RFC6347] Rescorla, E. and N. Modadugu, "Datagram Transport Layer
Security Version 1.2", RFC 6347, DOI 10.17487/RFC6347,
January 2012, <<https://www.rfc-editor.org/info/rfc6347>>.
- [RFC7296] Kaufman, C., Hoffman, P., Nir, Y., Eronen, P., and T.
Kivinen, "Internet Key Exchange Protocol Version 2
(IKEv2)", STD 79, RFC 7296, DOI 10.17487/RFC7296, October
2014, <<https://www.rfc-editor.org/info/rfc7296>>.
- [RFC7301] Friedl, S., Popov, A., Langley, A., and E. Stephan,
"Transport Layer Security (TLS) Application-Layer Protocol
Negotiation Extension", RFC 7301, DOI 10.17487/RFC7301,
July 2014, <<https://www.rfc-editor.org/info/rfc7301>>.

Authors' Addresses

Mirja Kuehlewind
ETH Zurich
Gloriastrasse 35
8092 Zurich
Switzerland

Email: mirja.kuehlewind@tik.ee.ethz.ch

Tommy Pauly
Apple Inc.
One Apple Park Way
Cupertino, California 95014
United States of America

Email: tpauly@apple.com

Christopher A. Wood
Apple Inc.
One Apple Park Way
Cupertino, California 95014
United States of America

Email: cawood@apple.com

Opsec Working Group
Internet-Draft
Intended status: Best Current Practice
Expires: May 3, 2018

K. Sriram
D. Montgomery
US NIST
J. Haas
Juniper Networks, Inc.
October 30, 2017

Enhanced Feasible-Path Unicast Reverse Path Filtering
draft-sriram-opsec-urpf-improvements-02

Abstract

This document identifies a need for improvement of the unicast Reverse Path Filtering techniques (uRPF) [BCP84] for source address validation (SAV) [BCP38]. The strict uRPF is inflexible about directionality, the loose uRPF is oblivious to directionality, and the current feasible-path uRPF attempts to strike a balance between the two [BCP84]. However, as shown in this draft, the existing feasible-path uRPF still has short comings. This document proposes an enhanced feasible-path uRPF technique, which aims to be more flexible (in a meaningful way) about directionality than the feasible-path uRPF. It can potentially alleviate ISPs' concerns about the possibility of disrupting service for their customers, and encourage greater deployment of uRPF techniques.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on May 3, 2018.

Copyright Notice

Copyright (c) 2017 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
1.1. Requirements Language	3
2. Review of Existing Source Address Validation Techniques . . .	3
2.1. SAV using Access Control List	4
2.2. SAV using Strict Unicast Reverse Path Filtering	4
2.3. SAV using Feasible-Path Unicast Reverse Path Filtering .	5
2.4. SAV using Loose Unicast Reverse Path Filtering	6
3. Proposed New Technique: SAV using Enhanced Feasible-Path uRPF	7
3.1. Description of the Method	7
3.2. Operational Recommendations	8
3.3. A Challenging Scenario	9
3.4. Overcoming the Above Challenge: Algorithm with Full Flexibility Across Customer Cone	10
3.5. Implementation Considerations	11
3.5.1. Impact on FIB Memory Size Requirement	11
4. Security Considerations	12
5. IANA Considerations	12
6. Acknowledgements	13
7. Informative References	13
Authors' Addresses	14

1. Introduction

This internet draft identifies a need for improvement of the unicast Reverse Path Filtering (uRPF) techniques [RFC2827] for source address validation (SAV) [RFC3704]. The strict uRPF is inflexible about directionality, the loose uRPF is oblivious to directionality, and the current feasible-path uRPF attempts to strike a balance between the two [RFC3704]. However, as shown in this draft, the existing feasible-path uRPF still has short comings. Even with the feasible-path uRPF, ISPs are often apprehensive that they may be dropping customers' data packets with legitimate source addresses.

This document proposes an enhanced feasible-path uRPF technique, which aims to be more flexible (in a meaningful way) about directionality than the feasible-path uRPF. It is based on the

principle that if BGP updates for multiple prefixes with the same origin AS were received on different interfaces (at border routers), then incoming data packets with source addresses in any of those prefixes should be accepted on any of those interfaces (described in Section 3.1). For some challenging ISP-customer scenarios (see Section 3.3), we further propose (a) Forming a list of all unique prefixes in the collection of routes received on all customer interfaces; and (b) Including that list in the RPF list of each customer interface (described in Section 3.4). Implementation considerations are discussed in Section 3.5.

Note: Definition of Reverse Path Filtering (RPF) list: The list of permissible source address prefixes for incoming data packets on a given interface.

The proposed techniques are expected to add greater operational robustness and efficacy to uRPF, while minimizing ISPs' concerns about accidental service disruption for their customers. It is expected that this will encourage more deployment of uRPF so as to realize its DDoS prevention benefits networkwide.

1.1. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

2. Review of Existing Source Address Validation Techniques

There are various existing techniques for mitigation against DDoS attacks with spoofed addresses [RFC2827] [RFC3704]. There are also some techniques used for mitigating reflection attacks [RRL] [TA14-017A], which are used to amplify the impact in DDoS attacks. Employing a combination of these preventive techniques in enterprise and ISP border routers, DNS servers, broadband and wireless access networks, and data centers provides reasonably effective protection against DDoS attacks.

Source address validation (SAV) is performed in network edge devices such as border routers, Cable Modem Termination Systems (CMTS), Digital Subscriber Line Access Multiplexers (DSLAM), and Packet Data Network (PDN) gateways in mobile networks. Ingress Access Control List (ACL) and unicast Reverse Path Filtering (uRPF) are techniques employed for implementing SAV [RFC2827] [RFC3704] [ISOC].

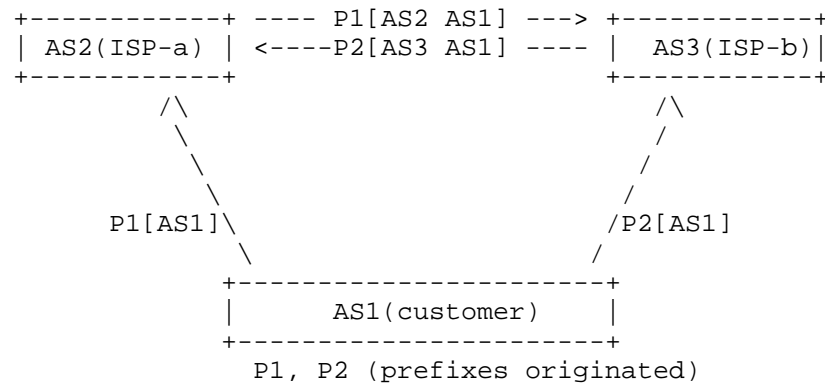
2.1. SAV using Access Control List

Ingress/egress Access Control Lists (ACLs) are maintained which list acceptable (or alternatively, unacceptable) prefixes for the source addresses in the incoming Internet Protocol (IP) packets. Any packet with a source address that does not match the filter is dropped. The ACLs for the ingress/egress filters need to be maintained to keep them up to date. Updating the ACLs is an operator driven manual process, and hence operationally difficult or infeasible.

Typically, the egress ACLs in access aggregation devices (e.g. CMTS, DSLAM) permit source addresses only from the address spaces (prefixes) that are associated with the interface on which the customer network is connected. Ingress ACLs are typically deployed on border routers, and drop ingress packets when the source address is spoofed (i.e. belongs to obviously disallowed prefix blocks, RFC 1918 prefixes, or provider's own prefixes).

2.2. SAV using Strict Unicast Reverse Path Filtering

In the strict unicast Reverse Path Filtering (uRPF) method, an ingress packet at border router is accepted only if the Forwarding Information Base (FIB) contains a prefix that encompasses the source address and forwarding information for that destination prefix points back to the interface over which the packet was received. In other words, the reverse path for routing to that source address (if it were used as a destination address) should use the same interface over which the packet was received. It is well known that this method has limitations when networks are multi-homed and there is asymmetric routing of packets. Asymmetric routing occurs (see Figure 1) when a customer AS announces one prefix (P1) to one transit provider (ISP-a) and a different prefix (P2) to another transit provider (ISP-b), but routes data packets with source addresses in the second prefix (P2) to the first transit provider (ISP-a) or vice versa.



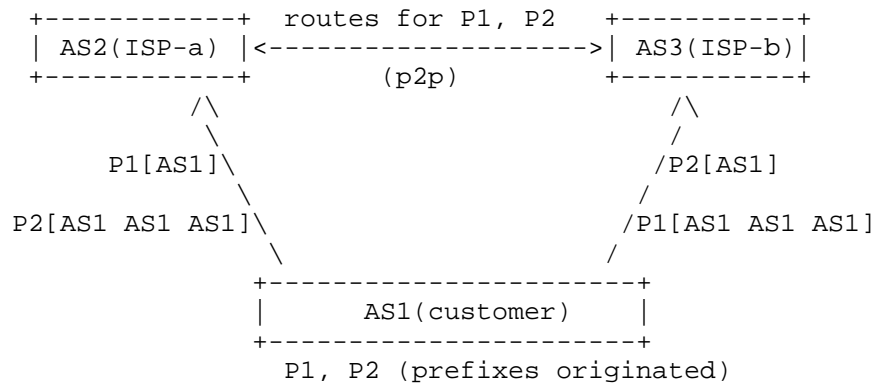
Consider data packets received at AS2
 (1) from AS1 with source address in P2, or
 (2) from AS3 that originated from AS1
 with source address in P1:

- * Strict uRPF fails
- * Feasible-path uRPF fails
- * Loose uRPF works (but ineffective in IPv4)
- * Enhanced Feasible-path uRPF works best

Figure 1: Scenario 1 for illustration of efficacy of uRPF schemes.

2.3. SAV using Feasible-Path Unicast Reverse Path Filtering

The feasible-path uRPF helps partially overcome the problem identified with the strict uRPF in the multi-homing case. The feasible-path uRPF is similar to the strict uRPF, but in addition to inserting the best-path prefix, additional prefixes from alternative announced routes are also included in the RPF table. This method relies on announcements for the same prefixes (albeit some may be prepended to effect lower preference) propagating to all routers performing feasible-path uRPF checks. Therefore, in the multi-homing scenario, if the customer AS announces routes for both prefixes (P1, P2) to both transit providers (with suitable prepends if needed for traffic engineering), then the feasible-path uRPF method works (see Figure 2). It should be mentioned that the feasible-path uRPF works in this scenario only if customer routes are preferred at AS2 and AS3 over a shorter non-customer route.



Consider data packets received at AS2 via AS3 that originated from AS1 and have source address in P1:

- * Feasible-path uRPF works (if customer route to P1 is preferred at AS3 over shorter path)
- * Feasible-path uRPF fails (if shorter path to P1 is preferred at AS3 over customer route)
- * Loose uRPF works (but ineffective in IPv4)
- * Enhanced Feasible-path uRPF works best

Figure 2: Scenario 2 for illustration of efficacy of uRPF schemes.

However, the feasible-path uRPF method has limitations as well. One form of limitation naturally occurs when the recommendation of propagating the same prefixes to all routers is not followed. Another form of limitation can be described as follows. In Scenario 2 (described above, illustrated in Figure 2), it is possible that the second transit provider (ISP-b or AS3) does not propagate the prepended route for prefix P1 to the first transit provider (ISP-a or AS2). This is because AS3's decision policy permits giving priority to a shorter route to prefix P1 via a peer (AS2) over a longer route learned directly from the customer (AS1). In such a scenario, AS3 would not send any route announcement for prefix P1 to AS2. Then a data packet with source address in prefix P1 that originates from AS1 and traverses via AS3 to AS2 will get dropped at AS2.

2.4. SAV using Loose Unicast Reverse Path Filtering

In the loose unicast Reverse Path Filtering (uRPF) method, an ingress packet at the border router is accepted only if the FIB has one or more prefixes that encompass the source address. That is, a packet is dropped if no route exists in the FIB for the source address. Loose uRPF sacrifices directionality. This method is not effective for prevention of address spoofing since there is little unrouted address space in IPv4. It only drops packets if the spoofed address

is unreachable in the current FIB (e.g. RFC 1918, unallocated, allocated but currently not routed).

3. Proposed New Technique: SAV using Enhanced Feasible-Path uRPF

3.1. Description of the Method

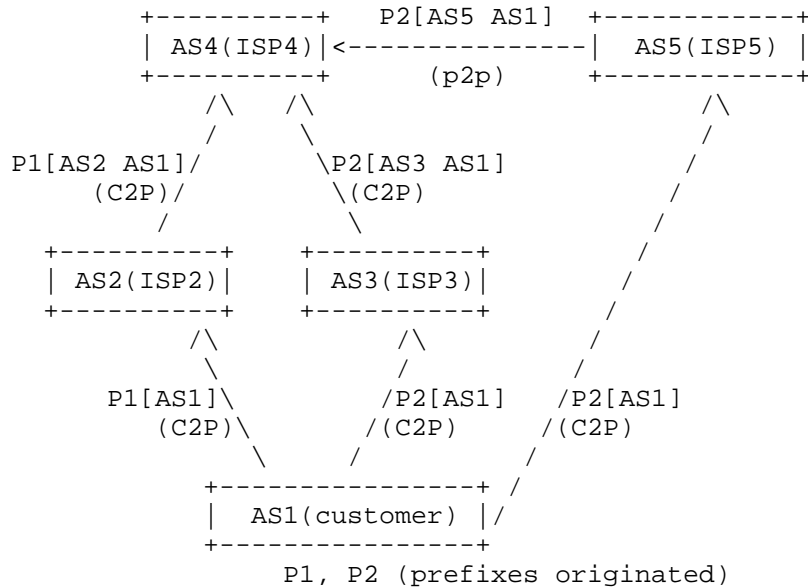
Enhanced feasible-path uRPF adds greater operational robustness and efficacy to existing uRPF methods discussed in Section 2. The proposed technique is based on the principle that if BGP updates for multiple prefixes with the same origin AS were received on different interfaces (at border routers), then incoming data packets with source addresses in any of those prefixes should be accepted on any of those interfaces. It can be best explained with an example as follows:

Let us say, a border router of ISP-A has in its Adj-RIB-in the set of prefixes {Q1, Q2, Q3} each of which has AS-x as its origin and AS-x is in ISP-A's customer cone. Further, the border router received a route for prefix Q1 over a customer facing interface, while it learned routes for prefixes Q2 and Q3 from a lateral peer and an upstream transit provider, respectively. All these routes passed route filtering and/or origin validation (i.e. the origin AS-x is deemed legitimate). That is, the route announcements are considered legitimate. In this example scenario, the enhanced feasible-path uRPF method allows source addresses to belong in {Q1, Q2, Q3} on any of the three specific interfaces in question (customer, peer, provider) on which the three routes were learned.

Thus, enhanced feasible-path uRPF defines feasible paths in a more generalized but precise way (as compared to feasible-path uRPF). In the above example, routes for prefixes Q2 and Q3 were not received on a customer facing interface at the border router, yet data packets with source addresses in Q2 or Q3 are accepted by the router if they come in on the same customer interface on which the route for prefix Q1 was received (based on these prefix routes having the same origin AS).

Looking back at Scenarios 1 and 2 (Figure 1 and Figure 2), the enhanced feasible-path uRPF provides comparable or better performance than the other uRPF methods. Scenario 3 (Figure 3) further illustrates the enhanced feasible-path uRPF method with a more concrete example. In this scenario, the focus is on operation of the feasible-path uRPF at ISP4 (AS4). ISP4 learns a route for prefix P1 via a customer-to-provider (C2P) interface from customer ISP2 (AS2). This route for P1 has origin AS1. ISP4 also learns a route for P2 via another C2P interface from customer ISP3 (AS3). Additionally, AS4 learns an alternate route for P2 via a peer-to-peer (p2p)

interface from ISP5 (AS5). Both routes for P2 have the same origin AS (i.e. AS1) as does the route for P1. Using the proposed enhanced feasible-path uRPF scheme, given the commonality of the origin AS across the above-mentioned routes for P1 and P2, AS4 would permit source addresses belonging to either P1 or P2 in data packets received on any of the three interfaces (from AS2, AS3, and AS5).



Consider that data packets (sourced from AS1) may be received at AS4 with source address in P1 or P2 via any of the neighbors (AS2, AS3, AS5):

- * Feasible-path uRPF fails
- * Loose uRPF works (but not desirable)
- * Enhanced Feasible-path uRPF works best

Figure 3: Scenario 3 for illustration of efficacy of uRPF schemes.

Based on the above, the proposed enhanced feasible-path uRPF method would reduce ISP concerns about possible service disruption affecting their customers and encourage greater adoption of uRPF.

3.2. Operational Recommendations

The following operational recommendations will make the operation of the proposed enhanced feasible-path uRPF robust:

For multi-homed stub AS:

- o A multi-homed stub AS SHOULD announce at least one of the prefixes it originates to each of its transit provider ASes.

For non-stub AS:

- o A non-stub AS SHOULD also announce at least one of the prefixes it originates to each of its transit provider ASes.
- o Additionally, from the routes it has learned from customers, a non-stub AS SHOULD announce at least one route per origin AS to each of its transit provider ASes.

(Note: It is worth noting that in the above recommendations if "at least one" is replaced with "all", then even traditional feasible-path uRPF will work as desired.)

3.3. A Challenging Scenario

It should be observed that in the absence of ASes adhering the above recommendations, the following example scenarios may be constructed which pose a challenge for the enhanced feasible-path uRPF (as well as for traditional feasible-path uRPF). In the scenario illustrated in Figure 4, since routes for neither P1 nor P2 are propagated on the AS2-AS4 interface, the enhanced feasible-path uRPF at AS4 will reject data packets received on that interface with source addresses in P1 or P2.

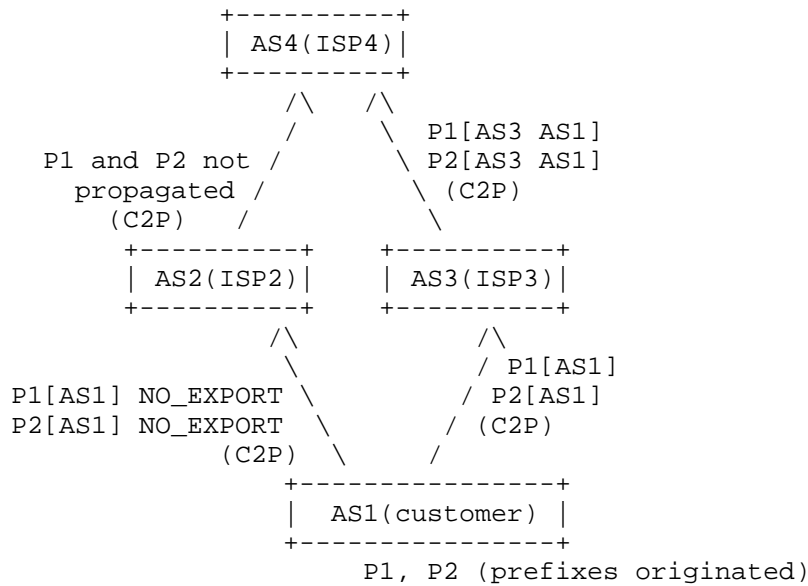


Figure 4: Illustration of a challenging scenario.

3.4. Overcoming the Above Challenge: Algorithm with Full Flexibility Across Customer Cone

Adding further flexibility to the enhanced feasible-path uRPF method can help address the potential limitation identified above using the scenario in Figure 4 (Section 3.3). In the following, "route" refers to a route currently existing in the Adj-RIB-in. Including the additional degree of flexibility, the modified algorithm can be described as follows:

- o Let $I = \{I_1, I_2, \dots, I_n\}$ represent the set of all directly-connected customer interfaces at customer-facing edge routers in a transit provider's AS.
- o Let $P = \{P_1, P_2, \dots, P_m\}$ represent the set of all unique prefixes for which routes were received over the interfaces in Set I .
- o Let $A = \{AS_1, AS_2, \dots, AS_k\}$ represent the set of all unique origin ASes seen in the routes that were received over the interfaces in Set I .
- o Let $Q = \{Q_1, Q_2, \dots, Q_j\}$ represent the set of all unique prefixes for which routes were received over peer or provider interfaces such that each of the routes has its origin AS belonging in Set A .

- o Then, Set Z = Union(P,Q) represents the RPF list for each customer-facing edge router in the AS in question. That is, over each interface in Set I, the edge router SHOULD permit only those ingress data packets that have SA in any of the prefixes in Set Z.

When this algorithmic flexibility is incorporated, then the type of limitation identified in Figure 4 (Section 3.3) goes away. This should significantly reduce the possibility of blocking legitimate customer-data packets in uRPF implementations.

3.5. Implementation Considerations

The existing RPF checks in edge routers take advantage of existing line card implementations to perform the RPF functions. For implementation of the proposed technique, the general necessary feature would be to extend the line cards to take arbitrary RPF lists that are not necessarily the same as the existing FIB contents. For example, in the proposed method, the RPF lists are constructed by applying a set of rules to all received BGP routes (not just those selected as best path and installed in FIB).

3.5.1. Impact on FIB Memory Size Requirement

The proposed technique requires that there should be FIB memory (i.e., TCAM) available to store the RPF lists in line cards. For an ISP's AS, the RPF list size for each line card will roughly and conservatively equal the total number of prefixes in its customer cone (assuming the algorithm in Section 3.4 is used). The following table shows the measured customer cone sizes for various types of ISPs [sriram-ripe63]:

Type of ISP	Measured Customer Cone Size in # Prefixes (in turn this is an estimate for RPF list size on line card)
Very Large Global ISP	32392
Very Large Global ISP	29528
Large Global ISP	20038
Mid-size Global ISP	8661
Regional ISP (in Asia)	1101

Table 1: Customer cone sizes (# prefixes) for various types of ISPs.

For some super large global ISPs that are at the core of the Internet, the customer cone size (# prefixes) can be as high as a few hundred thousand [caida]. But uRPF is most effective when deployed at ASes at the edges of the Internet where the customer cone sizes are smaller as shown in Table 1.

A very large global ISP's router line card is likely to have a FIB size large enough to accommodate 2 to 6 million routes [cisco1]. Similarly, the line cards in routers corresponding to a large global ISP, a mid-size global ISP, and a regional ISP are likely to have FIB sizes large enough to accommodate about 1 million, 0.5 million, and 100K routes, respectively [cisco2]. Comparing these FIB size numbers with the corresponding RPF list size numbers in Table 1, it can be surmised that the conservatively estimated RPF list size is only a small fraction of the anticipated FIB memory size under various ISP scenarios.

4. Security Considerations

This document offers a technique to improve the robustness features of uRPF and thus improve the security of the Internet as a whole. The proposed technique does not warrant any additional security considerations.

5. IANA Considerations

This document does not request new capabilities or attributes. It does not create any new IANA registries.

6. Acknowledgements

The authors would like to thank Job Snijders, Marco Marzetti, Marco d'Itri, Nick Hilliard, Gert Doering, Igor Gashinsky, Barry Greene, and Joel Jaeggli for comments and suggestions.

7. Informative References

- [caida] "Information for AS 174 (COGENT-174)", CAIDA Spoofer Project , <<https://spoofer.caida.org/as.php?asn=174>>.
- [cisco1] "Internet Routing Table Growth Causes ROUTING-FIB-4-RSRC_LOW Message on Trident-Based Line Cards", Cisco Trouble-shooting Tech-notes , January 2014, <<https://www.cisco.com/c/en/us/support/docs/routers/asr-9000-series-aggregation-services-routers/116999-problem-line-card-00.html>>.
- [cisco2] "Cisco Nexus 7000 Series NX-OS Unicast Routing Configuration Guide, Release 5.x (Chapter: Managing the Unicast RIB and FIB)", Cisco Configuration Guides , June 2017, <https://www.cisco.com/c/en/us/td/docs/switches/data-center/sw/5_x/nx-os/unicast/configuration/guide/13_cli_nxos/13_manage-routes.html#22859>.
- [ISOC] Vixie (Ed.), P., "Addressing the challenge of IP spoofing", ISOC report , September 2015, <<https://www.us-cert.gov/ncas/alerts/TA14-017A>>.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC2827] Ferguson, P. and D. Senie, "Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing", BCP 38, RFC 2827, DOI 10.17487/RFC2827, May 2000, <<https://www.rfc-editor.org/info/rfc2827>>.
- [RFC3704] Baker, F. and P. Savola, "Ingress Filtering for Multihomed Networks", BCP 84, RFC 3704, DOI 10.17487/RFC3704, March 2004, <<https://www.rfc-editor.org/info/rfc3704>>.
- [RFC6811] Mohapatra, P., Scudder, J., Ward, D., Bush, R., and R. Austein, "BGP Prefix Origin Validation", RFC 6811, DOI 10.17487/RFC6811, January 2013, <<https://www.rfc-editor.org/info/rfc6811>>.

[RRL] "Response Rate Limiting in the Domain Name System",
Redbarn blog , <<http://www.redbarn.org/dns/ratelimits>>.

[sriram-ripe63]
Sriram, K. and R. Bush, "Estimating CPU Cost of BGPSEC on
a Router", Presented at RIPE-63; also at IETF-83 SIDR WG
Meeting, March 2012,
<[http://www.ietf.org/proceedings/83/slides/
slides-83-sidr-7.pdf](http://www.ietf.org/proceedings/83/slides/slides-83-sidr-7.pdf)>.

[TA14-017A]
"UDP-Based Amplification Attacks", US-CERT alert
TA14-017A , January 2014,
<<https://www.us-cert.gov/ncas/alerts/TA14-017A>>.

Authors' Addresses

Kotikalapudi Sriram
US NIST
100 Bureau Drive
Gaithersburg MD 20899
USA

Email: ksriram@nist.gov

Doug Montgomery
US NIST
100 Bureau Drive
Gaithersburg MD 20899
USA

Email: dougmon@nist.gov

Jeffrey Haas
Juniper Networks, Inc.
1133 Innovation Way
Sunnyvale CA 94089
USA

Email: jhaas@juniper.net

OPSEC Working Group
Internet-Draft
Intended status: Best Current Practice
Expires: September 6, 2018

K. Sriram
D. Montgomery
USA NIST
J. Haas
Juniper Networks, Inc.
March 5, 2018

Enhanced Feasible-Path Unicast Reverse Path Filtering
draft-sriram-opsec-urpf-improvements-03

Abstract

This document identifies a need for improvement of the unicast Reverse Path Filtering techniques (uRPF) [BCP84] for source address validation (SAV) [BCP38]. The strict uRPF is inflexible about directionality, the loose uRPF is oblivious to directionality, and the current feasible-path uRPF attempts to strike a balance between the two [BCP84]. However, as shown in this draft, the existing feasible-path uRPF still has short comings. This document describes an enhanced feasible-path uRPF technique, which aims to be more flexible (in a meaningful way) about directionality than the feasible-path uRPF. It can potentially alleviate ISPs' concerns about the possibility of disrupting service for their customers, and encourage greater deployment of uRPF techniques.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on September 6, 2018.

Copyright Notice

Copyright (c) 2018 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
1.1. Requirements Language	3
2. Review of Existing Source Address Validation Techniques . . .	3
2.1. SAV using Access Control List	4
2.2. SAV using Strict Unicast Reverse Path Filtering	4
2.3. SAV using Feasible-Path Unicast Reverse Path Filtering .	5
2.4. SAV using Loose Unicast Reverse Path Filtering	6
3. SAV using Enhanced Feasible-Path uRPF	7
3.1. Description of the Method	7
3.1.1. Algorithm A: Enhanced Feasible-Path uRPF	8
3.2. Operational Recommendations	9
3.3. A Challenging Scenario	9
3.4. Algorithm B: Enhanced Feasible-Path uRPF with Additional Flexibility Across Customer Cone	10
3.5. Implementation Considerations	11
3.5.1. Impact on FIB Memory Size Requirement	11
3.6. Summary of Recommendations	12
4. Security Considerations	13
5. IANA Considerations	13
6. Acknowledgements	13
7. Informative References	13
Authors' Addresses	15

1. Introduction

This internet draft identifies a need for improvement of the unicast Reverse Path Filtering (uRPF) techniques [RFC2827] for source address validation (SAV) [RFC3704]. The strict uRPF is inflexible about directionality, the loose uRPF is oblivious to directionality, and the current feasible-path uRPF attempts to strike a balance between the two [RFC3704]. However, as shown in this draft, the existing feasible-path uRPF still has short comings. Even with the feasible-path uRPF, ISPs are often apprehensive that they may be dropping customers' data packets with legitimate source addresses.

This document describes an enhanced feasible-path uRPF technique, which aims to be more flexible (in a meaningful way) about directionality than the feasible-path uRPF. It is based on the principle that if BGP updates for multiple prefixes with the same origin AS were received on different interfaces (at border routers), then incoming data packets with source addresses in any of those prefixes should be accepted on any of those interfaces (presented in Section 3). For some challenging ISP-customer scenarios (see Section 3.3), this document also describes a more relaxed version of the enhanced feasible-path uRPF technique (presented in Section 3.4). Implementation considerations are discussed in Section 3.5.

Note: Definition of Reverse Path Filtering (RPF) list: The list of permissible source address prefixes for incoming data packets on a given interface.

Note: Throughout this document, the routes in consideration are assumed to have been vetted based on prefix filtering [RFC7454] and possibly (in the future) origin validation [RFC6811].

The enhanced feasible-path uRPF methods described here are expected to add greater operational robustness and efficacy to uRPF, while minimizing ISPs' concerns about accidental service disruption for their customers. It is expected that this will encourage more deployment of uRPF to help realize its DDoS prevention benefits network wide.

1.1. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

2. Review of Existing Source Address Validation Techniques

There are various existing techniques for mitigation against DDoS attacks with spoofed addresses [RFC2827] [RFC3704]. There are also some techniques used for mitigating reflection attacks [RRL] [TA14-017A], which are used to amplify the impact in DDoS attacks. Employing a combination of these preventive techniques (as applicable) in enterprise and ISP border routers, broadband and wireless access network, data centers, and DNS servers provides reasonably effective protection against DDoS attacks.

Source address validation (SAV) is performed in network edge devices such as border routers, Cable Modem Termination Systems (CMTS), Digital Subscriber Line Access Multiplexers (DSLAM), and Packet Data Network (PDN) gateways in mobile networks. Ingress Access Control

List (ACL) and unicast Reverse Path Filtering (uRPF) are techniques employed for implementing SAV [RFC2827] [RFC3704] [ISOC].

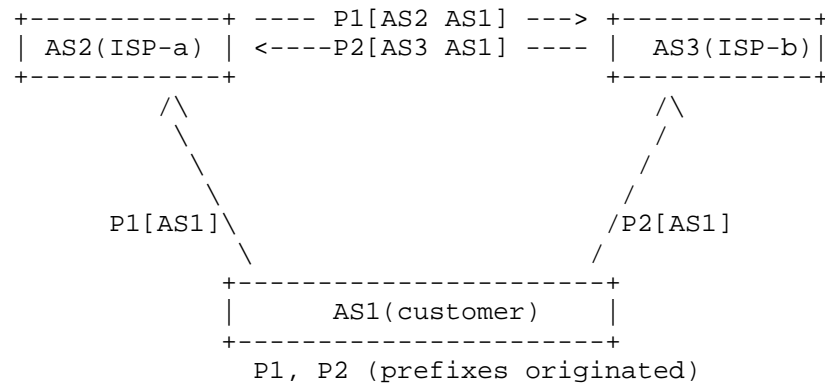
2.1. SAV using Access Control List

Ingress/egress Access Control Lists (ACLs) are maintained which list acceptable (or alternatively, unacceptable) prefixes for the source addresses in the incoming Internet Protocol (IP) packets. Any packet with a source address that does not match the filter is dropped. The ACLs for the ingress/egress filters need to be maintained to keep them up to date. Updating the ACLs is an operator driven manual process, and hence operationally difficult or infeasible.

Typically, the egress ACLs in access aggregation devices (e.g. CMTS, DSLAM) permit source addresses only from the address spaces (prefixes) that are associated with the interface on which the customer network is connected. Ingress ACLs are typically deployed on border routers, and drop ingress packets when the source address is spoofed (i.e. belongs to obviously disallowed prefix blocks, RFC 1918 prefixes, or provider's own prefixes).

2.2. SAV using Strict Unicast Reverse Path Filtering

In the strict unicast Reverse Path Filtering (uRPF) method, an ingress packet at border router is accepted only if the Forwarding Information Base (FIB) contains a prefix that encompasses the source address, and forwarding information for that prefix points back to the interface over which the packet was received. In other words, the reverse path for routing to the source address (if it were used as a destination address) should use the same interface over which the packet was received. It is well known that this method has limitations when networks are multi-homed and there is asymmetric routing of packets. Asymmetric routing occurs (see Figure 1) when a customer AS announces one prefix (P1) to one transit provider (ISP-a) and a different prefix (P2) to another transit provider (ISP-b), but routes data packets with source addresses in the second prefix (P2) to the first transit provider (ISP-a) or vice versa.



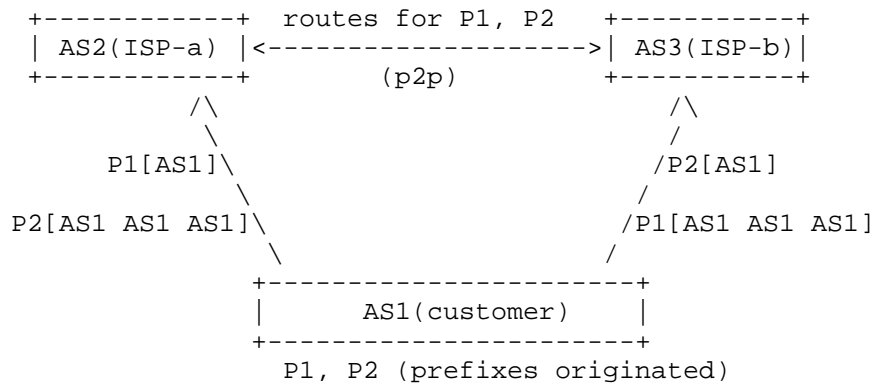
Consider data packets received at AS2
 (1) from AS1 with source address in P2, or
 (2) from AS3 that originated from AS1
 with source address in P1:

- * Strict uRPF fails
- * Feasible-path uRPF fails
- * Loose uRPF works (but ineffective in IPv4)
- * Enhanced Feasible-path uRPF works best

Figure 1: Scenario 1 for illustration of efficacy of uRPF schemes.

2.3. SAV using Feasible-Path Unicast Reverse Path Filtering

The feasible-path uRPF helps partially overcome the problem identified with the strict uRPF in the multi-homing case. The feasible-path uRPF is similar to the strict uRPF, but in addition to inserting the best-path prefix, additional prefixes from alternative announced routes are also included in the RPF table. This method relies on announcements for the same prefixes (albeit some may be prepended to effect lower preference) propagating to all routers performing feasible-path uRPF checks. Therefore, in the multi-homing scenario, if the customer AS announces routes for both prefixes (P1, P2) to both transit providers (with suitable prepends if needed for traffic engineering), then the feasible-path uRPF method works (see Figure 2). It should be mentioned that the feasible-path uRPF works in this scenario only if customer routes are preferred at AS2 and AS3 over a shorter non-customer route.



Consider data packets received at AS2 via AS3 that originated from AS1 and have source address in P1:

- * Feasible-path uRPF works (if customer route to P1 is preferred at AS3 over shorter path)
- * Feasible-path uRPF fails (if shorter path to P1 is preferred at AS3 over customer route)
- * Loose uRPF works (but ineffective in IPv4)
- * Enhanced Feasible-path uRPF works best

Figure 2: Scenario 2 for illustration of efficacy of uRPF schemes.

However, the feasible-path uRPF method has limitations as well. One form of limitation naturally occurs when the recommendation of propagating the same prefixes to all routers is not followed. Another form of limitation can be described as follows. In Scenario 2 (described above, illustrated in Figure 2), it is possible that the second transit provider (ISP-b or AS3) does not propagate the prepended route for prefix P1 to the first transit provider (ISP-a or AS2). This is because AS3's decision policy permits giving priority to a shorter route to prefix P1 via a peer (AS2) over a longer route learned directly from the customer (AS1). In such a scenario, AS3 would not send any route announcement for prefix P1 to AS2. Then a data packet with source address in prefix P1 that originates from AS1 and traverses via AS3 to AS2 will get dropped at AS2.

2.4. SAV using Loose Unicast Reverse Path Filtering

In the loose unicast Reverse Path Filtering (uRPF) method, an ingress packet at the border router is accepted only if the FIB has one or more prefixes that encompass the source address. That is, a packet is dropped if no route exists in the FIB for the source address. Loose uRPF sacrifices directionality. This method is not effective for prevention of address spoofing since there is little unrouted address space in IPv4. It only drops packets if the spoofed address

is unreachable in the current FIB (e.g. RFC 1918, unallocated, allocated but currently not routed).

3. SAV using Enhanced Feasible-Path uRPF

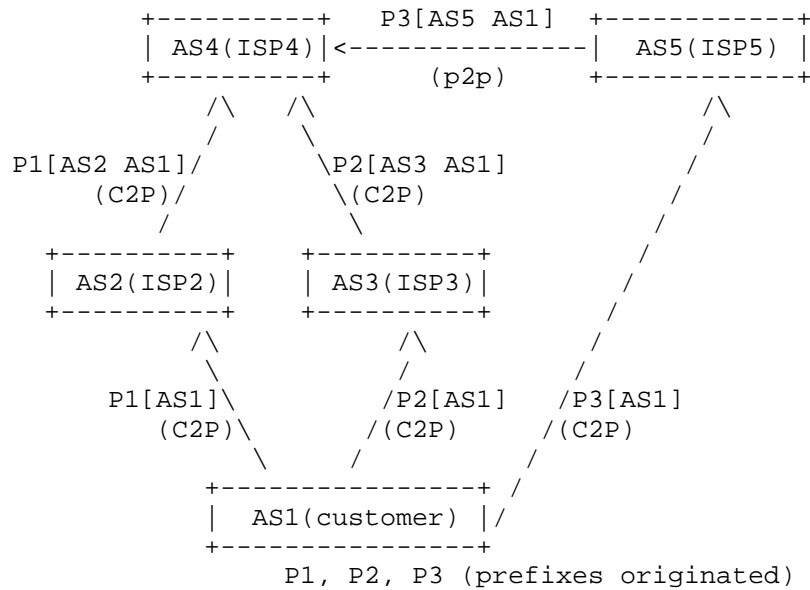
3.1. Description of the Method

Enhanced feasible-path uRPF adds greater operational robustness and efficacy to existing uRPF methods discussed in Section 2. The technique is based on the principle that if BGP updates for multiple prefixes with the same origin AS were received on different interfaces (at border routers), then incoming data packets with source addresses in any of those prefixes should be accepted on any of those interfaces. It can be best explained with an example as follows:

Let us say, a border router of ISP-A has in its Adj-RIB-Ins [RFC4271], the set of prefixes {Q1, Q2, Q3} each of which has AS-x as its origin and AS-x is in ISP-A's customer cone. Further, the border router received a route for prefix Q1 over a customer facing interface, while it learned routes for prefixes Q2 and Q3 from a lateral peer and an upstream transit provider, respectively. In this example scenario, the enhanced feasible-path uRPF method requires Q1, Q2, and Q3 be included in the RPF list for the customer interface in consideration. Loose uRPF (see Section 2.4) is recommended to be applied to the peer and provider interfaces in consideration.

Thus, enhanced feasible-path uRPF defines feasible paths for customer interfaces in a more generalized but precise way (as compared to feasible-path uRPF).

Looking back at Scenarios 1 and 2 (Figure 1 and Figure 2), the enhanced feasible-path uRPF provides comparable or better performance than the other uRPF methods. Scenario 3 (Figure 3) further illustrates the enhanced feasible-path uRPF method with a more concrete example. In this scenario, the focus is on operation of the feasible-path uRPF at ISP4 (AS4). ISP4 learns a route for prefix P1 via a customer-to-provider (C2P) interface from customer ISP2 (AS2). This route for P1 has origin AS1. ISP4 also learns a route for P2 via another C2P interface from customer ISP3 (AS3). Additionally, AS4 learns a route for P3 via a peer-to-peer (p2p) interface from ISP5 (AS5). Routes for all three prefixes have the same origin AS (i.e. AS1). Using the enhanced feasible-path uRPF scheme, given the commonality of the origin AS across the routes for P1, P2 and P3, AS4 includes all of these prefixes to the RPF list for the customer interfaces (from AS2 and AS3).



Consider that data packets (sourced from AS1)
may be received at AS4 with source address
in P1, P2 or P3 via any of the neighbors (AS2, AS3, AS5):

- * Feasible-path uRPF fails
- * Loose uRPF works (but not desirable)
- * Enhanced Feasible-path uRPF works best

Figure 3: Scenario 3 for illustration of efficacy of uRPF schemes.

3.1.1. Algorithm A: Enhanced Feasible-Path uRPF

The underlying algorithm in the solution method described above can be specified as follows (to be implemented in a transit AS):

1. Create the list of unique origin ASes considering only the routes in the Adj-RIB-Ins of customer interfaces. Call it Set A = {AS1, AS2, ..., ASn}.
2. Considering all routes in Adj-RIB-Ins for all interfaces (customer, lateral peer, and provider), form the set of unique prefixes that have a common origin AS1. Call it Set X1.
3. Include set X1 in Reverse Path Filter (RPF) list on all customer interfaces on which one or more of the prefixes in set X1 were received.

4. Repeat Steps 2 and 3 for each of the remaining ASes in Set A (i.e., for AS_i, where $i = 2, \dots, n$).

3.2. Operational Recommendations

The following operational recommendations will make the operation of the enhanced feasible-path uRPF robust:

For multi-homed stub AS:

- o A multi-homed stub AS SHOULD announce at least one of the prefixes it originates to each of its transit provider ASes.

For non-stub AS:

- o A non-stub AS SHOULD also announce at least one of the prefixes it originates to each of its transit provider ASes.
- o Additionally, from the routes it has learned from customers, a non-stub AS SHOULD announce at least one route per origin AS to each of its transit provider ASes.

(Note: It is worth noting that in the above recommendations if "at least one" is replaced with "all", then even traditional feasible-path uRPF will work as effectively.)

3.3. A Challenging Scenario

It should be observed that in the absence of ASes adhering the above recommendations, the following example scenario may be constructed which poses a challenge for the enhanced feasible-path uRPF (as well as for traditional feasible-path uRPF). In the scenario illustrated in Figure 4, since routes for neither P1 nor P2 are propagated on the AS2-AS4 interface, the enhanced feasible-path uRPF at AS4 will reject data packets received on that interface with source addresses in P1 or P2.

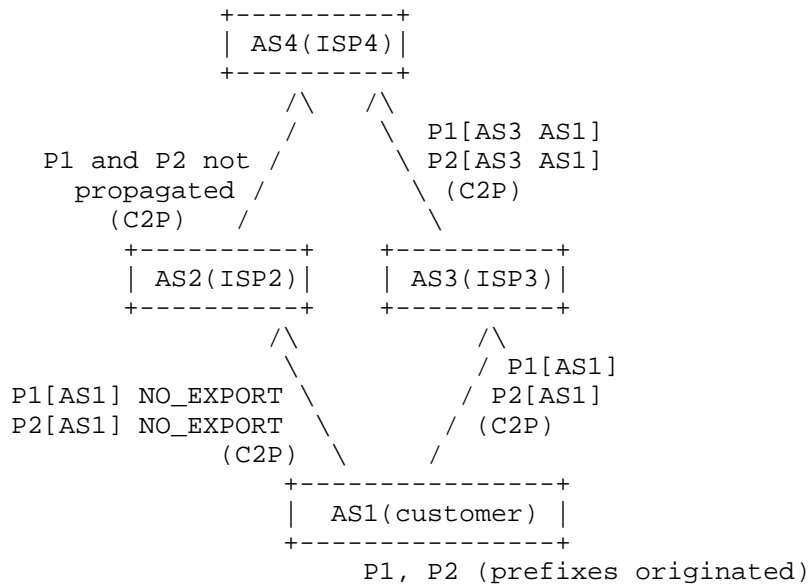


Figure 4: Illustration of a challenging scenario.

3.4. Algorithm B: Enhanced Feasible-Path uRPF with Additional Flexibility Across Customer Cone

Adding further flexibility to the enhanced feasible-path uRPF method can help address the potential limitation identified above using the scenario in Figure 4 (Section 3.3). In the following, "route" refers to a route currently existing in the Adj-RIB-in. Including the additional degree of flexibility, the modified algorithm (implemented in a transit AS) can be described as follows (we call this Algorithm B):

1. Create the set of all directly-connected customer interfaces. Call it Set I = {I1, I2, ..., Ik}.
2. Create the set of all unique prefixes for which routes exist in Adj-RIB-Ins for the interfaces in Set I. Call it Set P = {P1, P2, ..., Pm}.
3. Create the set of all unique origin ASes seen in the routes that exist in Adj-RIB-Ins for the interfaces in Set I. Call it Set A = {AS1, AS2, ..., ASn}.
4. Create the set of all unique prefixes for which routes exist in Adj-RIB-Ins of all lateral peer and provider interfaces such that

each of the routes has its origin AS belonging in Set A. Call it Set $Q = \{Q_1, Q_2, \dots, Q_j\}$.

5. Then, Set $Z = \text{Union}(P, Q)$ represents the RPF list for every customer interface in Set I.
6. Apply loose uRPF method for SAV on all peer and provider interfaces.

When Algorithm B (which is more flexible than Algorithm A) is employed, the type of limitation identified in Figure 4 (Section 3.3) goes away.

3.5. Implementation Considerations

The existing RPF checks in edge routers take advantage of existing line card implementations to perform the RPF functions. For implementation of the enhanced feasible-path uRPF, the general necessary feature would be to extend the line cards to take arbitrary RPF lists that are not necessarily the same as the existing FIB contents. In the algorithms (Section 3.1.1 and Section 3.4) described here, the RPF lists are constructed by applying a set of rules to all received BGP routes (not just those selected as best path and installed in FIB).

3.5.1. Impact on FIB Memory Size Requirement

The techniques described here require that there should be FIB memory (i.e., TCAM) available to store the RPF lists in line cards. For an ISP's AS, the RPF list size for each line card will roughly and conservatively equal the total number of prefixes in its customer cone (assuming the algorithm in Section 3.4 is used). (Note: Most ISP customer cone scenarios would not require the algorithm in Section 3.4, but instead be served best by the algorithm in Section 3.1.1, which requires much less FIB memory.) The following table shows the measured customer cone sizes for various types of ISPs [sriram-ripe63]:

Type of ISP	Measured Customer Cone Size in # Prefixes (in turn this is an estimate for RPF list size on line card)
Very Large Global ISP	32392
Very Large Global ISP	29528
Large Global ISP	20038
Mid-size Global ISP	8661
Regional ISP (in Asia)	1101

Table 1: Customer cone sizes (# prefixes) for various types of ISPs.

For some super large global ISPs that are at the core of the Internet, the customer cone size (# prefixes) can be as high as a few hundred thousand [CAIDA]. But uRPF is most effective when deployed at ASes at the edges of the Internet where the customer cone sizes are smaller as shown in Table 1.

A very large global ISP's router line card is likely to have a FIB size large enough to accommodate 2 to 6 million routes [cisco1]. Similarly, the line cards in routers corresponding to a large global ISP, a mid-size global ISP, and a regional ISP are likely to have FIB sizes large enough to accommodate about 1 million, 0.5 million, and 100K routes, respectively [cisco2]. Comparing these FIB size numbers with the corresponding RPF list size numbers in Table 1, it can be surmised that the conservatively estimated RPF list size is only a small fraction of the anticipated FIB memory size under relevant ISP scenarios.

3.6. Summary of Recommendations

Depending on the scenario, an ISP or enterprise AS operator should follow one of the following recommendations concerning uRPF/SAV:

1. For directly connected networks, i.e., subnets directly connected to the AS and not multi-homed, the AS in consideration SHOULD perform ACL-based SAV.
2. For a directly connected single-homed stub AS (customer), the AS in consideration SHOULD perform SAV based on the strict uRPF method.

3. For all other scenarios:

- * If the scenario does not involve complexity such as NO_EXPORT of routes (see Section 3.3, Figure 4), then the enhanced feasible-path uRPF method in Algorithm A (see Section 3.1.1) SHOULD be applied.
- * Else, if the scenario involves the aforementioned complexity, then the enhanced feasible-path uRPF method in Algorithm B (see Section 3.4) SHOULD be applied.

4. Security Considerations

The security considerations in BCP 38 [RFC2827] and BCP 84 [RFC3704] apply for this document as well. In addition, AS operator should apply the uRPF method that performs best (i.e., with zero or insignificant possibility of dropping legitimate data packets) for the type of peer (customer, provider, etc.) and the nature of customer cone scenario that apply (see Section 3.1.1 and Section 3.4).

5. IANA Considerations

This document does not request new capabilities or attributes. It does not create any new IANA registries.

6. Acknowledgements

The authors would like to thank Job Snijders, Marco Marzetti, Marco d'Itri, Nick Hilliard, Gert Doering, Igor Gashinsky, Barry Greene, and Joel Jaeggli for comments and suggestions.

7. Informative References

- [CAIDA] "Information for AS 174 (COGENT-174)", CAIDA Spoofer Project , <<https://spoofer.caida.org/as.php?asn=174>>.
- [cisco1] "Internet Routing Table Growth Causes ROUTING-FIB-4-RSRC_LOW Message on Trident-Based Line Cards", Cisco Trouble-shooting Tech-notes , January 2014, <<https://www.cisco.com/c/en/us/support/docs/routers/asr-9000-series-aggregation-services-routers/116999-problem-line-card-00.html>>.

- [cisco2] "Cisco Nexus 7000 Series NX-OS Unicast Routing Configuration Guide, Release 5.x (Chapter: Managing the Unicast RIB and FIB)", Cisco Configuration Guides , June 2018, <https://www.cisco.com/c/en/us/td/docs/switches/data_center/sw/5_x/nx-os/unicast/configuration/guide/l3_cli_nxos/l3_manage-routes.html#22859>.
- [ISOC] Vixie (Ed.), P., "Addressing the challenge of IP spoofing", ISOC report , September 2015, <<https://www.us-cert.gov/ncas/alerts/TA14-017A>>.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC2827] Ferguson, P. and D. Senie, "Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing", BCP 38, RFC 2827, DOI 10.17487/RFC2827, May 2000, <<https://www.rfc-editor.org/info/rfc2827>>.
- [RFC3704] Baker, F. and P. Savola, "Ingress Filtering for Multihomed Networks", BCP 84, RFC 3704, DOI 10.17487/RFC3704, March 2004, <<https://www.rfc-editor.org/info/rfc3704>>.
- [RFC4271] Rekhter, Y., Ed., Li, T., Ed., and S. Hares, Ed., "A Border Gateway Protocol 4 (BGP-4)", RFC 4271, DOI 10.17487/RFC4271, January 2006, <<https://www.rfc-editor.org/info/rfc4271>>.
- [RFC6811] Mohapatra, P., Scudder, J., Ward, D., Bush, R., and R. Austein, "BGP Prefix Origin Validation", RFC 6811, DOI 10.17487/RFC6811, January 2013, <<https://www.rfc-editor.org/info/rfc6811>>.
- [RFC7454] Durand, J., Pepelnjak, I., and G. Doering, "BGP Operations and Security", BCP 194, RFC 7454, DOI 10.17487/RFC7454, February 2015, <<https://www.rfc-editor.org/info/rfc7454>>.
- [RRL] "Response Rate Limiting in the Domain Name System", Redbarn blog , <<http://www.redbarn.org/dns/ratelimits>>.

[sriram-ripe63]

Sriram, K. and R. Bush, "Estimating CPU Cost of BGPSEC on a Router", Presented at RIPE-63; also at IETF-83 SIDR WG Meeting, March 2012,
<<http://www.ietf.org/proceedings/83/slides/slides-83-sidr-7.pdf>>.

[TA14-017A]

"UDP-Based Amplification Attacks", US-CERT alert TA14-017A , January 2014,
<<https://www.us-cert.gov/ncas/alerts/TA14-017A>>.

Authors' Addresses

Kotikalapudi Sriram
USA National Institute of Standards and Technology
100 Bureau Drive
Gaithersburg MD 20899
USA

Email: ksriram@nist.gov

Doug Montgomery
USA National Institute of Standards and Technology
100 Bureau Drive
Gaithersburg MD 20899
USA

Email: doug@nist.gov

Jeffrey Haas
Juniper Networks, Inc.
1133 Innovation Way
Sunnyvale CA 94089
USA

Email: jhaas@juniper.net