

SACM
Internet-Draft
Intended status: Informational
Expires: July 25, 2016

C. Coffin
B. Cheikes
C. Schmidt
D. Haynes
The MITRE Corporation
J. Fitzgerald-McKay
Department of Defense
D. Waltermire
National Institute of Standards and Technology
January 22, 2016

SACM Vulnerability Assessment Scenario
draft-coffin-sacm-vuln-scenario-01

Abstract

This document provides a core narrative that walks through an automated enterprise vulnerability assessment scenario. It is aligned with the SACM use cases and begins with an enterprise ingesting vulnerability description data, followed by identifying endpoints on the network and collecting and storing information about them to enable posture assessment, and finally ends with assessing these endpoints against the vulnerability description data to determine which ones are affected. Processes that specifically overlap between this scenario and SACM use cases will be noted where applicable. Specifically, the relationship between this document and the SACM use case building block capabilities and the usage scenarios will be covered.

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 July 25, 2016.

Copyright Notice

Copyright (c) 2016 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. Scope 3
- 2. Assumptions 4
- 3. Endpoint Identification and Initial (Pre-Assessment) Data Collection 5
 - 3.1. Identification 6
 - 3.1.1. SACM Use Case Alignment 6
 - 3.2. Processing Artifacts 6
 - 3.3. Endpoint Data Collection 7
 - 3.3.1. SACM Use Case Alignment 8
 - 3.4. Implementation Examples 9
- 4. Vulnerability Description Data 9
 - 4.1. SACM Use Case Alignment 10
 - 4.2. Implementation Examples 10
- 5. Endpoint Applicability and Assessment 10
 - 5.1. Applicability 11
 - 5.1.1. SACM Use Case Alignment 11
 - 5.2. Secondary Assessment 11
 - 5.2.1. SACM Use Case Alignment 12
 - 5.3. Implementation Examples 13
- 6. Assessment Results 13
 - 6.1. SACM Use Case Alignment 14
 - 6.2. Implementation Examples 15
- 7. IANA Considerations 15
- 8. Security Considerations 15
- 9. Informative References 15
- Appendix A. Change Log 16
 - A.1. Changes in Revision 01 16
- Appendix B. Continuous Vulnerability Assessment 17
- Appendix C. Priority 18
- Appendix D. Data Attribute Table and Definitions 19
 - D.1. Table 19

D.2. Definitions 22

Appendix E. Alignment with Other Existing Works 24

 E.1. Critical Security Controls 24

 E.1.1. Continuous Vulnerability Assessment 24

 E.1.2. Hardware and Software Inventories 26

Appendix F. SACM Usage Scenarios 26

Appendix G. SACM Requirements and Charter - Future Work 28

Authors' Addresses 28

1. Scope

The purpose of this document is to describe a detailed scenario for vulnerability assessment, and identify aspects of this scenario that could be used in the development of an information model. This includes classes of data, major roles, and a high-level description of role interactions. Additionally, this scenario intends to inform engineering work on protocol and data model development. The focus of the document is entirely intra-organizational and covers enterprise handling of vulnerability description data. The document does not attempt to cover the security disclosure itself and any prior activities of the security researcher or discloser, nor does it attempt to cover the specific activities of the vendor whose software is the focus of the vulnerability description data (i.e., the vulnerable software).

For the purposes of this document, the term "vulnerability description data" is intended to mean: "Data intended to alert enterprise IT resources to the existence of a flaw or flaws in software, hardware, and/or firmware, which could potentially have an impact on enterprise functionality and/or security." For the purpose of this scenario, such data also includes information that can be used to determine (to some level of accuracy, although possibly not conclusively) whether or not the flaw is present within an enterprise, when compared to information about the state of the enterprise's endpoints. For those who are familiar with current security practices and terminology, the use of vulnerability description data is also synonymous with security bulletin or advisory.

This document makes no attempt to provide a definition of a normalized data format (e.g. industry standard) for vulnerability description data although there is nothing precluding the development of such a normalized data format. Also, it does not attempt to define procedures by which a vulnerability discoverer coordinates the release of vulnerability description data to other parties.

2. Assumptions

A number of assumptions must be stated in order to further clarify the position and scope of this document.

- o The document begins with the assumption that the enterprise has received vulnerability description data, and that the data has already been processed into a format that the enterprise's security software tools can understand and use. In particular, this document:
 - * Does not discuss how the enterprise identifies potentially relevant vulnerability description data.
 - * Does not discuss how the enterprise collects the vulnerability description data.
 - * Does not discuss how the enterprise assesses the authenticity of the vulnerability description data.
 - * Does not discuss parsing of the vulnerability description data into a usable format.
- o The document assumes that the enterprise has a means of identifying enterprise endpoints. This could mean identifying endpoints as they join the network, actively scanning for connected endpoints, passive scanning of network traffic to identify connected endpoints, or some other method of accounting for the presence of all endpoints in the enterprise. The document also does not distinguish between physical endpoints and virtualized endpoints.
- o The document assumes that the enterprise has a means of extracting relevant information about enterprise endpoints. Moreover, this extracted information is expressed in a format that is compatible with the information extracted from the vulnerability description data. The document:
 - * Does not specify how relevant information is identified.
 - * Does not specify the mechanics of how relevant information is extracted from the data sources (such as the endpoint itself).
 - * Does not specify how extracted endpoint information and vulnerability description data is normalized to be compatible.

Note that having a means of extracting relevant information about enterprise endpoints is within the scope of the SACM Endpoint

Security Posture Assessment process. In the case of this document, this sub-process is assumed to be existent.

- o The document assumes that all information described in the steps below is available in the vulnerability description data and serves as the basis of this assessment. Likewise, the document assumes that the enterprise can provide all relevant information about any endpoint needed to perform the described analysis. The authors recognize that this will not always be the case, but these assumptions are taken in order to show the breadth of data utilization in this scenario. Less complete information may require variations to the described steps.
- o The document assumes that the enterprise has a policy by which assessment of endpoints based on vulnerability description data is prioritized. The document:
 - * Does not specify how prioritization occurs.
 - * Does not specify how prioritization impacts assessment behaviors.
- o The document assumes that the enterprise has a mechanism for long-term storage of vulnerability description data and endpoint assessment results (e.g., a data repository).
- o This document assumes that the enterprise has a procedure for reassessment of endpoints at some point after initial assessment. The document:
 - * Does not specify how a reassessment would impact individual assessment behaviors. (i.e., it is agnostic as to whether the assessment procedure is the same regardless of whether this is the first or a subsequent assessment for some set of vulnerability description data.)
 - * Does not provide recommendations or specifics on reassessment intervals.

3. Endpoint Identification and Initial (Pre-Assessment) Data Collection

The first step in this scenario involves identifying endpoints and collecting the basic or minimum set of system information attributes from them such as operating system type and version. Further examples of system information and attributes can be found below in the section titled Endpoint Data Collection. This identification occurs prior to the receipt of any specific vulnerability description data and is part of the regular, ongoing monitoring of endpoints

within an enterprise. This process is not meant to report on, or gather data for any specific vulnerabilities. The information gathered during this step could be applied in many enterprise automation efforts. Specifically, in addition to vulnerability management, it could be used by configuration and license management tasks. All of the information collected during this step is stored in a central location such as a Repository.

This activity involves the following sub-steps:

3.1. Identification

Prior to any other steps, the identification of endpoints must occur. This involves locating (at least virtually) and distinguishing between endpoints on the network in a way that allows each endpoint to be recognized in future interactions and selected for specific treatment. This not only allows later steps to determine the scope of what endpoints need to be assessed, but also allows for the unique identification of each endpoint. Unique and persistent endpoint IDs are used to allow for endpoints to be tracked over time and between sensors as well as allow for proper counts of assets during inventories and other similar collections. Endpoint identity can be established by collecting certain attributes that allow for unique and persistent tracking of endpoints on the enterprise network. Examples include, but are not limited to, IP address, MAC address, FQDNs, pre-provisioned identifiers such as GUIDs or copies of serial numbers, certificates, hardware identity values, or similar attributes. It is important to note that the persistency of these attributes will likely vary depending on the enterprise. For example, a statically assigned IP address is much more persistent than an IP address assigned via DHCP.

3.1.1. SACM Use Case Alignment

This sub-step aligns with the Endpoint Discovery, Endpoint Characterization, and Endpoint Target Identification building block capabilities. The alignment is due to the fact that the purpose of this sub-step is to discover, identify, and characterize all endpoints on an enterprise network.

3.2. Processing Artifacts

Processing artifacts, such as the date and time the collection was performed, should be collected and stored. This timestamp is extremely important when performing later assessments, as it is needed for data freshness computations. The organization may develop rules for stale data and when a new data collection is required. This metadata is also helpful in correlating information across

multiple data collections. This includes correlating both pre-assessment data and secondary assessment data (sections 4.3 Endpoint Data Collection and 6.2 Secondary Assessment).

3.3. Endpoint Data Collection

The enterprise should perform ongoing collection of basic endpoint information such as operating system and version information, and an installed software inventory. This information is collected for general system monitoring as well as its potential use in activities such as vulnerability assessment.

Some examples of basic information to collect about endpoints in this pre-assessment process could include:

- o Endpoint type - traditional (e.g., workstation, server, etc.) network infrastructure (e.g., switches, routers, etc.), mobile (e.g., cell phones, tablets, laptops, etc.), and constrained (e.g., industrial control systems, Internet of Things, etc.)
- o Hardware version/firmware - e.g., BIOS version, firmware revision, etc.
- o Operating system - e.g., Windows, Linux, Mac OS, Android
- o Operating system attributes - e.g., version, patch level, service pack level, internationalized or localized version, etc.
- o Installed software inventory - Would include the software names and versions and possibly other high-level attributes. Could be used to quickly determine endpoint applicability when new vulnerability description data arrives.

Some additional and more advanced information to collect from endpoints in this pre-assessment process could include:

- o Open ports and enabled services - This would include applications listening for incoming connections on open ports as well as services that are starting, running, suspended, or enabled to run pending some event.
- o Operating system optional component inventory - some OS' have optional components that can be installed which may not show up as separate pieces of software (e.g., web and ftp servers, demo web pages, shared libraries, etc.). Note that this could also occur within third-party applications as well.

- o Endpoint location - physical location (e.g., department, room, Global Positioning System (GPS), etc.), logical location (e.g., what network infrastructure endpoints (e.g. switches, wireless access point, etc.) an endpoint is connected to, etc.
- o Purpose - describes how the endpoint is used within the enterprise (e.g., end-user system, database server, public web server, etc.)
- o Criticality - enterprise defined rating (possibly a score) that helps determine the criticality of the endpoint. If this endpoint is attacked or lost, what is the impact to the overall enterprise?

It is important to note that some of these attributes may exist natively on the endpoint whereas other attributes may be assigned by a human, computed, or derived from other data and may or may not be available for collection on the endpoint.

Furthermore, the possibility should be left open for enterprises to define their own custom queries and algorithms to gather and derive enterprise-specific attributes that are deemed of interest to regular enterprise operations.

In addition to collecting these attributes, metadata about the attributes should also be collected which could include:

Data origin - where the data originated from

Data source - what provided the data

Date and time of collection - when the data was collected

3.3.1. SACM Use Case Alignment

This sub-step aligns with the Data Publication building block capability because this section involves storage of endpoint attributes within an enterprise Repository. This sub-step also aligns with the Endpoint Characterization and Endpoint Target Identification building block capabilities because it further characterizes the endpoint through automated and possibly manual means. There is direct alignment with the Endpoint Component Inventory, Posture Attribute Identification, and Posture Attribute Value Collection building block capabilities since the purpose of this sub-step is to perform an initial inventory of the endpoint and collect basic attributes and their values. Last, there is alignment with the Collection Guidance Acquisition building block capabilities as the inventory and collection of endpoint attributes would be directed by some type of enterprise or third-party guidance.

3.4. Implementation Examples

Within the SACM Architecture, the Internal and External Collector components could be used to allow enterprises to collect posture attributes that demonstrate compliance with enterprise policy. Endpoints can be required to provide posture attributes, which may include identification attributes to enable persistent communications.

The SWID Message and Attributes for IF-M standard defines collection and validation of software identities using the ISO Software Identification Tag Standard. Using this standard, the identity of all installed software including the endpoint operating system, could be collected and used for later assessment.

The OVAL Definitions Model provides a data model that can be used to specify what posture attributes to collect as well as their expected values which can be used to drive an assessment.

The OVAL System Characteristics Model can be used to capture information about an endpoint. The model is specifically suited to expressing OS information, endpoint identification information (such as IP and MAC addresses), and other endpoint metadata.

4. Vulnerability Description Data

The next step in the Vulnerability Assessment scenario begins after vulnerability description data has been received and processed into a form that can be used in the assessment of the enterprise. As a part of the enterprise process for managing vulnerability description data, the enterprise should store all received and processed vulnerability description data in a Repository. The stored vulnerability description data can be used and compared with later vulnerability description data for the purpose of duplicate detection and in some cases, guidance on how to handle similar issues.

All vulnerability description data should be assigned an internal tracking ID by the enterprise as a first step as this helps compensate for the fact that incoming vulnerability description data might not have a global identifier when it is received, and might never be assigned one.

High-level vulnerability description data metadata to store would include:

- o Ingest date and time - the date and time that the vulnerability description data was received by the enterprise.

- o Date and time of vulnerability description data release (i.e., publication or disclosure date and time) - Some older vulnerability description data may be ingested long after publication. This can be useful when reviewing historical enterprise information to (potentially) identify the period when a particular endpoint was first assessed as vulnerable. Sometimes this information will help to differentiate between similar vulnerability description data.
- o Version - the version or iteration of the vulnerability description data according to the author, if applicable.
- o External Vulnerability Description Data ID(s) (if applicable) - any external or third-party IDs assigned to the vulnerability description data should be tracked. There could be multiple IDs in some cases (e.g., vendor bug id, global ID, discoverer's local ID, third-party vulnerability database ID, etc.).
- o Severity Score (if available) - these may be useful for later mitigation prioritization.

In addition to the described metadata, the raw or original vulnerability description data would be stored along with the specific information extracted from it that is to be used in the applicability and assessment process.

4.1. SACM Use Case Alignment

This step aligns with the Data Publication and Data Retrieval building block capabilities because this section details storage of vulnerability description data within an enterprise Repository and later retrieval of the same.

4.2. Implementation Examples

The Common Vulnerability Reporting Framework (CVRF) is an XML-based language that attempts to standardize the creation of vulnerability report documentation. Using CVRF, the enterprise could create automated tools based on the standardized schema which would obtain the needed and relevant information useful for later assessments and assessment results.

5. Endpoint Applicability and Assessment

When new vulnerability description data is received by the enterprise, applicable enterprise endpoints must be identified and assessed. Endpoints are first examined using the already obtained pre-assessment data. If this is not sufficient to determine endpoint

applicability, a secondary data collection for additional data and attributes may be performed to determine status with regard to the vulnerability description data.

5.1. Applicability

The applicability of an endpoint and its vulnerability status can, in many cases, be determined entirely by the existence of a particular version of installed software on the endpoint. This data may have been collected in the pre-assessment data collection. If the applicability and vulnerability status of an endpoint can be determined entirely by the pre-collected data attribute set, no further data collection is required.

Other cases may require specific data (i.e., file system attributes, specific configuration parameters, etc.) to be collected for the assessment of a particular vulnerability description data. In these cases, a secondary, targeted vulnerability assessment is required. Administrators may want to evaluate applicability to the vulnerability description data iteratively. Specifically, the process would compare against pre-collected data first (easy to do and the data is stored in a Repository), and then if needed, query endpoints that are not already excluded from applicability for additional required data. (I.e., A "fast-fail" model). To do this, the criteria for determining applicability must be separable, so that some conclusions can be drawn based on the possession of partial data.

5.1.1. SACM Use Case Alignment

This sub-step aligns with the Data Retrieval, Data Query, and Posture Attribute Value Query building block capabilities because, in this sub-step, the process is attempting to determine the vulnerability status of the endpoint using the data that has previously been collected.

5.2. Secondary Assessment

If the applicability and vulnerability status of an endpoint cannot be determined by the pre-assessment data collection, a secondary and targeted assessment of the endpoint will be required. A secondary assessment may also be required in the case that data on-hand (either from pre-assessment or from prior secondary assessments) is stale or out-of-date.

The following data types and attributes are examples of what might be required in the case of a secondary and targeted assessment:

- o Specific files and attributes - i.e., file name, versions, size, write date, modified date, checksum, etc. Some vulnerabilities may only be distinguishable through the presence or absence of specific files or their attributes.
- o Shared libraries - Some vulnerabilities will affect many products across multiple vendors. In these cases the vulnerability may apply to a shared library. Under these circumstances, product versions may be less helpful than looking for the presence of one or more specific files and their attributes.
- o Other software configuration information (if applicable) - e.g., Microsoft Windows registry queries, Apple configuration profiles, GConf, Proc filesystem, text configuration files and their parameters, and the installation paths. Sometimes vulnerabilities only affect certain software configurations and in some cases these are not the default configurations. Certain configuration attributes can be used to determine the current configuration state.

Note that the secondary assessment described here does not need to be a pull assessment that is initiated by the server. The secondary assessment could also be part of a push to the server when the endpoint detects a change to a vulnerability assessment baseline.

5.2.1. SACM Use Case Alignment

This sub-step aligns with the Data Publication building block capability because this section details storage of endpoint attributes within an enterprise Repository. The sub-step also aligns with the Collection Guidance Acquisition building block capability since the vulnerability description data (guidance) drives the collection of additional endpoint attributes.

This sub-step aligns with the Endpoint Characterization (both manual and automated) and Endpoint Target Identification building block capabilities because it could further characterize the endpoint through automated and possibly manual means. There is direct alignment with the Endpoint Component Inventory, Posture Attribute Identification, and Posture Attribute Value Collection building block capabilities since the purpose of this sub-step is to perform additional and more specific component inventories and collections of endpoint attributes and their values.

5.3. Implementation Examples

Within the SACM Architecture, the assessment task would be handled by the Evaluator component. If pre-assessment data is used, this would be stored on and obtained from a Data Store component.

Within the SACM Architecture, the Internal and External Collector components could be used to allow enterprises to collect posture attributes that demonstrate compliance with enterprise policy. Endpoints can be required to provide posture attributes, which may include identification attributes to enable persistent communications.

The SWID Message and Attributes for IF-M standard defines collection and validation of software identities using the ISO Software Identification Tag Standard. Using this standard, all installed software including the endpoint operating system could be collected and stored for later assessment.

The OVAL Definitions Model provides a data model that can be used to specify what posture attributes to collect as well as their expected values which can be used to drive an assessment.

The OVAL System Characteristics Model can be used to capture information about an endpoint. The model is specifically suited to expressing OS information, endpoint identification information (such as IP and MAC addresses), and other endpoint metadata.

The SACM Internal and External Attribute Collector components can be used to allow enterprises to collect posture attributes that demonstrate compliance with enterprise policy. Endpoints can be required to provide posture attributes, which may include identification attributes to enable persistent communications.

6. Assessment Results

Assessment results present the results of an assessment, along with sufficient context so a human or machine can make the appropriate response. This context might include a description of the issue provided by the vulnerability description data, the endpoint attributes that indicate applicability, or other information needed to respond to the results of the assessment. Data in this step is stored for auditing and forensic purposes.

The following details are important to track in assessment results. Note that information may be "included" by providing pointers to other records stored in a Repository (e.g., vulnerability description data, endpoint data, etc.).

- o Date and time of assessment - The date and time that the assessment was performed. To understand when the data was compared against the vulnerability description data and what conclusions were drawn.
- o Data collection/attribute age - The age of the data used in the assessment to make the endpoint status determination.
- o Endpoint ID - The endpoint itself must be identified for tracking results over time.
- o Vulnerability description data ID(s) - May include both the internally defined ID as well as one or more externally defined IDs if they exist. The internally assigned ID allows linkage to the correct vulnerability description data. If available, external IDs provide a "pivot point" to additional external information.
- o Vulnerable software product(s) - Identifies the software products on the endpoint that resulted in the endpoint being declared applicable. Since some vulnerability description data identify vulnerabilities in multiple products, this will help identify the specific product (or products) found to be vulnerable in the endpoint assessment.
- o Endpoint vulnerability status - The endpoint status based on the vulnerability description data. Does the vulnerability exist on the endpoint?
- o Vulnerability description - Not needed for automated assessment but probably should be included for human review. The reason for inclusion is to support the human user understanding of the vulnerability assessment results within the application front-end or interface.
- o Vulnerability remediation - Similar to the above, remediation or vendor patch information would be useful for a human response. In many cases, this information may be a part of the description information described above. Note that patch information may change over time due to supersession of the vendor patches.

6.1. SACM Use Case Alignment

This step aligns with the Data Publication and Data Retrieval building block capabilities because this section details storage of vulnerability assessment results within an enterprise Repository and later retrieval of the same.

6.2. Implementation Examples

The OVAL Results Model provides a data model to encode the results of the assessment, which could then be stored in a Repository and later accessed. The assessment results described in this scenario could be stored and later accessed using the OVAL Results Model. Note that the use of the OVAL Results Model for sharing results is not recommended per section 7.3 of the OVAL and the SACM Information Model [draft-hansbury-sacm-oval-info-model-mapping-01].

Within the SACM Architecture, the generation of the assessment results would occur in the Report Generator component. Those results might then be moved to a Data Store component for later sharing and retrieval as defined by SACM.

7. IANA Considerations

This memo includes no request to IANA.

8. Security Considerations

This document provides a core narrative that walks through an automated enterprise vulnerability assessment scenario and is aligned with SACM "Endpoint Security Posture Assessment: Enterprise Use Cases" [RFC7632]. As a result, the security considerations for [RFC7632] apply to this document. Furthermore, the vulnerability description data may provide attackers with useful information such as what software an enterprise is running on their endpoints. As a result, organizations should properly protect the vulnerability description data it ingests.***TODO IS THIS COVERED BY RFC7632????***

9. Informative References

[charter-ietf-sacm-01]

Security Automation and Continuous Monitoring, "Charter, Version 1.0", July 2013.

[critical-controls]

Council on CyberSecurity, "Critical Security Controls, Version 5.1".

[draft-hansbury-sacm-oval-info-model-mapping-01]

Security Automation and Continuous Monitoring, "OVAL and the SACM Information Model", November 2015.

[I-D.ietf-sacm-requirements]

Cam-Winget, N. and L. Lorenzin, "Secure Automation and Continuous Monitoring (SACM) Requirements", draft-ietf-sacm-requirements-11 (work in progress), November 2015.

[RFC7632] Waltermire, D. and D. Harrington, "Endpoint Security Posture Assessment: Enterprise Use Cases", RFC 7632, DOI 10.17487/RFC7632, September 2015, <<http://www.rfc-editor.org/info/rfc7632>>.

Appendix A. Change Log

A.1. Changes in Revision 01

Clarification of the vulnerability description data IDs in sections 4 and 6.

Added "vulnerability remediation" to the Assessment Results and Data Attribute Table and Definitions sections.

Added Implementation Examples to Endpoint Identification and Initial (Pre-Assessment) Data Collection, Vulnerability Description Data, Endpoint Applicability and Assessment, and Assessment Results sections.

Added an example to vulnerability description data in the scope section.

Added a sentence to clarify vulnerability description data definition in the scope section.

Added data repository example for long-term storage scope item.

Added sentence to direct reader to examples of basic system information in endpoint identification section.

Split the examples of information to collect in the pre-assessment collection section into a basic and advanced list.

Added examples of data stored in the repository in the Assessment Results section.

Added sentence for human-assigned attributes in the Future Work section.

Replaced "vulnerability report" to "vulnerability description data" because the term report was causing confusion. Similarly, replaced "assessment report" with "assessment results".

Replaced "Configuration Management Database (CMDB)" with "Repository" which is SACM's term for a data store.

Replaced endpoint "Role" with "Purpose" because "Role" is already defined in SACM. Also, removed "Function" because it too is already defined in SACM.

Clarified that the document does not try to define a normalized data format for vulnerability description data although it does not preclude the creation of such a format.

Included additional examples of software configuration information.

Clarified the section around endpoint identification to make it clear designation attributes used to correlate and identify endpoints are both persistent and unique. Furthermore, text was added to explain how the persistency of attributes may vary. This was based on knowledge gained from the Endpoint ID Design Team.

Updated the Security Considerations section to mention those described in [RFC7632].

Removed text around Bring Your Own Device (BYOD). While important, BYOD just adds complexity to this initial draft. BYOD should be addressed in a later revision.

Merged the list of "basic endpoint information" and the list of "human-assigned endpoint attributes" as both represent data we want to collect about an endpoint. Whether or not that data is natively available on the endpoint for collection or assigned by a human, computed, or derived from other data which may or may not be available on the endpoint for collection seems arbitrary. With this scenario, we primarily care about expressing information needs rather than how the information is collected or from where.

Appendix B. Continuous Vulnerability Assessment

It is not sufficient to perform a single assessment when vulnerability description data is published without any further checking. Doing so does not address the possibility that the reported vulnerability might be introduced to the enterprise environment after the initial assessment completes. For example, new endpoints can be introduced to the environment which have old software or are not up-to-date with patches. Another example is where unauthorized or obsolete software is installed on an existing endpoint by enterprise users after vulnerability description data and initial assessment has taken place. Moreover, enterprises might not wish to, or be able to, assess all vulnerability description data

immediately when they come in. Conflicts with other critical activities or limited resources might mean that some alerts, especially those that the enterprise deems as "low priority", are not used to guide enterprise assessments until sometime after the initial receipt.

The scenario above describes a single assessment of endpoints. However, it does not make any assumptions as to when this assessment occurs relative to the original receipt of the vulnerability description data that led to this assessment. The assessment could immediately follow ingest of the vulnerability description data, could be delayed, or the assessment might represent a reassessment of some vulnerability description data against which endpoints had previously been assessed. Moreover, the scenario incorporates long-term storage of collected data, vulnerability description data, and assessment results in order to facilitate meaningful and ongoing reassessment.

Appendix C. Priority

Priorities associated with the vulnerability description data, assessment results, and any remedy is important, but is treated as a separate challenge and, as such, has not been integrated into the description of this scenario. Nevertheless, it is important to point out and describe the use of priorities in the overall vulnerability description data scenario as they separable issues with their own sets of requirements.

Priority in regard to vulnerability description data, can be viewed in a couple of different ways within an enterprise. The assessment prioritization involves prioritization of the vulnerability description data assessment process. This determines what vulnerability description data is assessed, and in what order it is assessed in. For instance, a vulnerability affecting an operating system or application used throughout the enterprise would likely be prioritized higher than a vulnerability in an application which is used only on a few, low-criticality endpoints.

The prioritization of remedies relates to the enterprise remediation and mitigation process based on the discovered vulnerabilities. Once an assessment has been performed and applicable endpoints identified, enterprise vulnerability managers must determine where to focus their efforts to apply appropriate remedies. For example, a vulnerability that is easily exploitable and which can allow arbitrary code execution might be remedied before a vulnerability that is more difficult to exploit or which just degrades performance.

Some vulnerability description data include severities and/or other information that places the vulnerability in context. This information can be used in both of the priority types discussed above. In other cases, enterprise administrators may need to prioritize based only on what they know about their enterprise and the description provided in the vulnerability description data.

Examples of data attributes specific to priority of assessments and/or remedies include (but not limited to) the following:

- o Enterprise - defined purpose of the device, criticality of the device, exposure of the device, etc.
- o Severity attributes - A rating or score that attempts to provide the level of severity or criticality associated with a given vulnerability.
- o Cyber threat intelligence - information such as tactics, techniques, and procedures of threat actors, indicators of compromise, incidents, courses of action, etc. that help the enterprise understand relevant threats and how to detect, mitigate, or respond to them.

Appendix D. Data Attribute Table and Definitions

D.1. Table

The following table maps all major data attributes against each major process where they are used.

	vulnerability description data	Endpoint Identification and Initial (Pre-Assessment) Data Collection	Endpoint Applicability and Assessment	Assessment Results
Endpoint				
Collection date/time		X	X	
Endpoint type		X	X	
Hardware ver	X	X	X	

tion/firmware				
Operating system	X	X	X	
Operating system attributes (e.g., version, service pack level, edition, etc.)	X	X	X	
Installed software name	X	X	X	X
Installed software attributes (e.g., version, patch level, install path, etc.)	X	X	X	X
Open ports/services	X	X	X	
Operating system optional component inventory	X	X	X	
Location		X		X
Purpose		X		X
Criticality		X		X
File system attributes (e.g., versions,	X		X	

size, write date, modified date, checksum, etc.)				
Shared libraries	X		X	
Other software configuration information	X		X	
External vulnerability description data				
Ingest Date	X		X	
Date of Release	X		X	
Version	X		X	
External vuln ID	X		X	X
Severity Score				X
Assessment Results				
Date of assessment			X	X
Date of data collection		X	X	X
Endpoint identification and/or locally assigned ID		X	X	X

Vulnerable software product(s)	X	X	X	X
Endpoint vulnerability status			X	X
Vulnerability description	X			X
Vulnerability remediation	X			X

Table 1: Vulnerability Assessment Attributes

D.2. Definitions

Endpoint

- o Collection date/time - the date and time of data collection
- o Endpoint type - the device type of the endpoint (e.g., standard computer, printer, router, mobile device, tablet, etc.)
- o Hardware version/firmware - the hardware or firmware version if applicable (e.g., BIOS version, firmware revision, etc.)
- o Operating system - Operating system name
- o Operating system attributes - Operating system high-level attributes (e.g., version, service pack level, edition, etc.). Would not include configuration details.
- o Installed software name - List of all installed software packages (i.e., software inventory). May or may not include software installed by the operating system.
- o Installed software attributes - Software high-level attributes (e.g., version, patch level, install path, etc.). Would not include configuration details.
- o Open ports/enabled services - Listening network ports (e.g., TCP, UDP, etc.) as well as services that are starting, running, suspended, or enabled to run pending some event.

- o Operating system optional component inventory - Operating system specific components and software (when NOT already included in the general software inventory)
- o Location - The physical location of an enterprise endpoint (e.g., department, room, etc.)
- o Purpose - describes how the endpoint is used within the enterprise (e.g., end user system, database server, public web server, etc.)
- o Criticality - An enterprise-defined rating (possibly a score) that helps determine the criticality of the endpoint. If this endpoint is attacked or lost, what is the impact to the overall enterprise?
- o File system attributes - Attributes that describe the file or directory (e.g., versions, size, write date, modified date, checksum, etc.)
- o Shared libraries - libraries that can be used by and installed with many different software applications. A shared library vulnerability could affect multiple software applications in the same way.
- o Other software configuration information - operating system or software application configuration attributes that go beyond that basic information already captured (e.g., Microsoft Windows registry, Apple configuration profiles, GConf, Proc filesystem, text configuration files and their parameters, and the installation paths.)

External vulnerability description data

- o Ingest Date - the date that the vulnerability description data was received by the enterprise.
- o Date of Release - publication or disclosure date of the vulnerability description data
- o Version - the version or iteration of the vulnerability description data according to the author, if applicable.
- o External vuln ID - external or third-party IDs assigned to the vulnerability description data. Could be multiple IDs in some cases (e.g., vendor bug id, global ID, discoverer's local ID, third-party vulnerability database ID, etc.).

- o Severity Score - the severity of the vulnerability description data according to the vulnerability description data author, if applicable.

Assessment Results

- o Date of assessment - The date that the assessment was performed against an endpoint.
- o Date of data collection - The age of the data used in the assessment to make the endpoint status determination.
- o Endpoint identification and/or locally assigned ID - The ID assigned to the enterprise endpoint. Must be assigned for tracking results over time.
- o Vulnerable software product(s) - The vulnerable software products identified as being installed on the endpoint.
- o Endpoint vulnerability status - Overall vulnerability status of the enterprise endpoint (i.e., Pass or Fail)
- o Vulnerability description - A human-consumable description of a vulnerability. Supports the human user understanding of the vulnerability assessment results within an application front-end or user interface.
- o Vulnerability remediation - The fix, workaround, or patch information for a vulnerability. This information may be a part of the vulnerability description described previously. Note that this information can change over time due to vendor patch supersession.

Appendix E. Alignment with Other Existing Works

E.1. Critical Security Controls

The Council on CyberSecurity's Critical Security Controls [critical-controls] includes security controls for a number of use scenarios, some of which are covered in this document. This section documents the alignment between the Council's controls and the relevant elements of the scenario.

E.1.1. Continuous Vulnerability Assessment

"CSC 4: Continuous Vulnerability Assessment and Remediation," which is described by the Council on CyberSecurity as "Continuously acquire, assess, and take action on new information in order to

identify vulnerabilities, remediate, and minimize the window of opportunity for attackers." The scenario described in this document is aligned with CSC 4 in multiple ways:

CSC 4-1 applies to this scenario in that it calls for running regular, automated scanning to deliver prioritized lists of vulnerabilities with which to respond. The scenario described in this document is intended to be executed on a continuous basis, and the priorities of both vulnerability description data and the remedy of vulnerabilities are discussed in the Priority section earlier in this document.

This scenario assumes that the enterprise already has a source for vulnerability description data as described in CSC 4-4.

Both CSC 4-2 and 4-7 are made possible by writing information to a Repository since this makes previously collected data available for later analysis.

While this scenario does not go into the details of how prioritization would be calculated or applied, it does touch on some of the important ways in which prioritization would impact the endpoint assessment process in the Priority section. As such, the Priority section aligns with CSC 4-10, which deals with vulnerability priority. Vulnerability priority in this scenario is discussed in terms of the vulnerability description data priority during receipt, as well as the vulnerability priority with regards to remedies.

The described scenario does not address the details of applying a remedy based on assessment results. As such, CSC 4-5, 4-8, and 4-9, which all deal with mitigations and patching, are out of scope for this work. Similarly, CSC 4-3 prescribes performing scans in authenticated mode and CSC 4-6 prescribes monitoring logs. This scenario does not get into the means by which data is collected, focusing on "what" to collect rather than "how", and as such does not have corresponding sections, although the procedures described are not incompatible with either of these controls.

The CSC 4 System Entity Relationship diagram and numbered steps directly align with the scenario described in this document with the exception of step 7 (patch response). Steps 1 -6 in CSC 4 describe the overall process for vulnerability management starting with obtaining the vulnerability description data from the source in Step 1, to producing assessment results in step 6.

E.1.2. Hardware and Software Inventories

This scenario is also aligned with, and describes a process for, collecting and maintaining hardware and software inventories, which are covered by the Council on CyberSecurity CSC 1 "Inventory of Authorized and Unauthorized Devices" and CSC 2 "Inventory of Authorized and Unauthorized Software." This scenario documents a process that is specific to collecting and maintaining hardware and software data attributes for vulnerability assessment purposes, but the collection of the hardware attributes and software inventory documented in the Endpoint Data Collection section that follows can also be used for the purpose of implementing authorized and unauthorized hardware and software management processes (e.g., scanning tools looking for unauthorized software). Moreover, the ability to accurately identify endpoints and, to a lesser degree, applications is integral to effective endpoint data collection and vulnerability management.

The Endpoint Data Collection section does not have coverage for the specific details described in CSC 1 and 2 as they are different processes and would be out-of-scope of this scenario, but the section does provide the data necessary to support the controls.

The Endpoint Identification and Endpoint Data Collection sections within this scenario align with CSC 1-1 and 1-4 by identifying enterprise endpoints and collecting their hardware and network attributes. The Endpoint Data Collection section aligns with and supports CSC 2-3 and 2-4 by defining a software inventory process and a method of obtaining operating system and file system attributes. The rest of the items from CSC 1 and 2 deal with implementation details and would be out-of-scope for this document.

CSC 2-9 describes the use of a software ID tag in XML format. SWID tags (https://en.wikipedia.org/wiki/ISO/IEC_19770) would also be a possible implementation for the Endpoint Data Collection section described in this scenario.

Appendix F. SACM Usage Scenarios

The SACM "Endpoint Security Posture Assessment: Enterprise Use Cases" document ([RFC7632]) defines multiple usage scenarios that are meant to provide examples of implementing the use cases and building block capabilities. Below is a brief summary of some of these usage scenarios and how this document aligns and/or adds additional value to the identified usage scenarios.

- o Automated Checklist Verification (2.2.2) - "An enterprise operates a heterogeneous IT environment. They utilize vendor-provided

automatable security configuration checklists for each operating system and application used within their IT environment. Multiple checklists are used from different vendors to ensure adequate coverage of all IT assets." The usage scenario, as defined in the RFC, is targeted at the checklist level and can be interpreted as being specific to endpoint configuration. There is mention of patch assessment and vulnerability mitigation, but the usage scenario could be expanded upon by including vulnerability verification. Replacing the idea of a checklist in the SACM usage scenario with vulnerability would allow the usage scenario to align almost exactly with the scenario described in this document. Instead of collecting automatable security configuration checklists, the enterprise would collect automatable vulnerability description data available from the vendor as described or possibly from other interested third-parties.

- o Detection of Posture Deviations (2.2.3) - "An enterprise has established secure configuration baselines for each different type of endpoint within their IT environment. When an endpoint connects to the network, the appropriate baseline configuration is communicated to the endpoint. Once the baseline has been established, the endpoint is monitored for any change events pertaining to the baseline on an ongoing basis. When a change occurs to posture defined in the baseline, updated posture information is exchanged. When the endpoint detects a posture change, an alert is generated identifying the specific changes in posture." This usage scenario would support the concept of endpoints signaling or alerting the enterprise to changes in the posture relates to endpoint vulnerabilities in the same way that it would for configurations. Replacing the idea of a checklist with vulnerability description data allows the SACM usage scenario and the scenario described in this document to align in their objectives.
- o Asynchronous Compliance/Vulnerability Assessment at Ice Station Zebra (2.2.5) - "An isolated arctic IT environment that is separated from the main university network. The only network communications are via an intermittent, low-speed, high-latency, high-cost satellite link. Remote network admins will need to show continued compliance with the security policies of the university, the government, and the provider of the satellite network, as well as keep current on vulnerability testing." This SACM usage scenario describes vulnerability assessment and aligns well with the vulnerability scenario described in this document. The endpoint assets are identified and associated data is published in a Repository. Vulnerability description data is collected and saved in a Repository as it is released. The vulnerability description data is queued for later assessment, then the

assessment results and vulnerability description data are stored after assessment. The only real difference in this SACM usage scenario is the timing of the assessments. The scenario described within this document would have no problems adjusting to the timing of this SACM usage scenario or anything similar.

Appendix G. SACM Requirements and Charter - Future Work

In the course authoring this document, some additional considerations for possible future work were noted. The following points were taken from the SACM Requirements [I-D.ietf-sacm-requirements], SACM Charter [charter-ietf-sacm-01], and SACM Use Cases ([RFC7632]) documents and represent work that may be necessary to support the tasks or goals of SACM going forward.

- o The SACM requirements mentions "Result Reporting" with applications but no detail around what the assessment results data set should include. In the case of vulnerability assessment results, context is important and details beyond just a Pass or Fail result are needed in order to take action. A good example of this might be the Priority of the vulnerability itself and how many systems it affects within the enterprise. With this in mind, it might be worthwhile to investigate a minimum data set or schema for assessment results. The concern here is with vulnerability description data, but this could apply to other enterprise processes as well.
- o The "Human-assigned endpoint attributes" mentioned previously in this scenario are touched on in the SACM use cases, but the topic could probably be explored in much more depth. Enterprise policy and behaviors could be greatly influenced by endpoint attributes such as locations, how the endpoint is used, and criticality. When and how these data attributes are collected, as well as what the minimum or common set might look like, would be good topics for future related SACM work. In addition, the storage of these attributes could be central (stored in a data repository) or they could be assigned and stored on the endpoints themselves.

Authors' Addresses

Christopher Coffin
The MITRE Corporation
202 Burlington Road
Bedford, MA 01730
USA

Email: ccoffin@mitre.org

Brant Cheikes
The MITRE Corporation
202 Burlington Road
Bedford, MA 01730
USA

Email: bcheikes@mitre.org

Charles Schmidt
The MITRE Corporation
202 Burlington Road
Bedford, MA 01730
USA

Email: cmschmidt@mitre.org

Daniel Haynes
The MITRE Corporation
202 Burlington Road
Bedford, MA 01730
USA

Email: dhaynes@mitre.org

Jessica Fitzgerald-McKay
Department of Defense
9800 Savage Road
Ft. Meade, Maryland
USA

Email: jmfitz2@nsa.gov

David Waltermire
National Institute of Standards and Technology
100 Bureau Drive
Gaithersburg, Maryland 20877
USA

Email: david.waltermire@nist.gov

Internet Engineering Task Force
Internet-Draft
Intended status: Informational
Expires: January 1, 2017

M. Georgescu, Ed.
NAIST
June 30, 2016

The STRIDE towards IPv6: A Threat Model for IPv6 Transition Technologies
draft-georgescu-opsec-ipv6-trans-tech-threat-model-01

Abstract

This document provides a structured approach for analyzing the threats associated with the various IPv6 transition technologies specified by the IETF. The threat model is built around the established STRIDE threat classification and is aimed at existing IPv6 transition technologies, as well as their future developments.

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 1, 2017.

Copyright Notice

Copyright (c) 2016 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.	Requirements Language	3
3.	The Generic Categories of IPv6 Transition Technologies	3
4.	Building The Threat Model	4
4.1.	Establish the function	4
4.2.	Identify the generic category	4
4.3.	Decompose the technology	4
4.4.	Identify the threats	5
4.4.1.	STRIDE-DFD Association	5
4.4.2.	Level of Trust	6
4.4.3.	Documenting the Threats	6
4.4.4.	Complex Threats	7
4.5.	Review, Repeat and Validate	7
5.	Dual Stack Threat Model	8
5.1.	Establish the Function	8
5.2.	Identify the Generic Category	8
5.3.	Decompose the Technology	8
5.4.	Identify the threats	9
5.4.1.	STRIDE-DFD Association	9
5.4.2.	From Trust to Likelihood	10
5.4.3.	Documenting the Threats	10
5.4.4.	Complex Threats	11
5.5.	Review, Repeat and Validate	12
6.	Single Translation Threat Model	13
6.1.	Decompose the Technology	13
6.2.	Identify the threats	14
7.	Double Translation Threat Model	15
7.1.	Decompose the Technology	15
7.2.	Identify the threats	15
8.	Encapsulation Threat Model	16
8.1.	Decompose the Technology	16
8.2.	Identify the threats	17
9.	Acknowledgments	17
10.	IANA Considerations	17
11.	Security Considerations	17
12.	References	18
12.1.	Normative References	18
12.2.	Informative References	18
Appendix A.	Appendix A	21
Author's Address	30

1. Introduction

When building an IPv6 transition plan, security is arguably one of the biggest concerns for network operators, as a heterogeneous IPv4 and IPv6 environment greatly increases the attack surface. To that

end, building a threat model for IPv6 transition technologies can help clarify and categorize the associated security threats. In turn, this should facilitate the search for mitigation solutions.

The security considerations of IPv6 transition technologies has generally been analyzed in each of the corresponding specifications, and some documents have discussed the general threats associated with transition technologies (see e.g. [RFC4942]).

However, more structured threat modeling has proved useful for understanding the security of intricate systems. Structured approaches allows one to discover, categorize and classify the threats according to their potential impact on the system. Considering the complicated nature of IPv6 transition technologies, threat modeling makes a good candidate for better understanding their security implications. This document follows a structured approach for analyzing the threats associated with transition technologies, that considers the functions of a transition technology as well as the context in which the technology is used.

The threat model uses the established STRIDE mnemonic and threat classification. STRIDE stands for Spoofing, Tampering, Repudiation, Information Disclosure, Denial of service and Elevation of Privilege, a generic list of threats which can be used to classify various threats and provides some basic mitigation directions. Since similar transition technologies can be associated with a similar list of threats, the document considers the generic classification of IPv6 transition technologies described in [draft-bmwg-v6trans].

2. 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].

3. The Generic Categories of IPv6 Transition Technologies

Table 1 presents the generic categories described in [draft-bmwg-v6trans] and some sample IPv6 transition technologies specified by the IETF.

Table 1. IPv6 Transition Technologies Categories

	Generic category	IPv6 Transition Technology
1	Dual-stack	Dual IP Layer Operations [RFC4213]
2	Single translation	NAT64 [RFC6146], IVI [RFC6219]
3	Double translation	464XLAT [RFC6877], MAP-T [RFC7599]
4	Encapsulation	DSLite [RFC6333], MAP-E [RFC7597] Lightweight 4over6 [RFC7596] 6RD [RFC5569]

4. Building The Threat Model

To build a threat model for IPv6 transition technologies a series of steps is recommended. The steps were inspired by the threat modelling approach described in [stride-shostack]. These steps are detailed in the following subsections.

4.1. Establish the function

The function of the IPv6 transition technology needs to be clearly documented. Depending on the context, the technology can incorporate multiple services, which need to be clearly identified in order to perform an effective threat analysis.

4.2. Identify the generic category

The category should be identified considering the generic classification defined in Section 3. This step can help reuse the threat analysis data for technologies which are part of the same category.

4.3. Decompose the technology

Build a data flow diagram (DFD) and highlight the entry points, protected resources and trust boundaries. The entry points should be assigned a level of trust considering the trust boundaries.

The external entities, process, data store and data flow elements should be depicted in the same diagram. The IP protocol suite and the protocols used for the designated function should be identified as well. This can narrow down the attack surface.

Figure 1 presents the basic elements of a data flow diagram as well as general rules for their association with network elements.

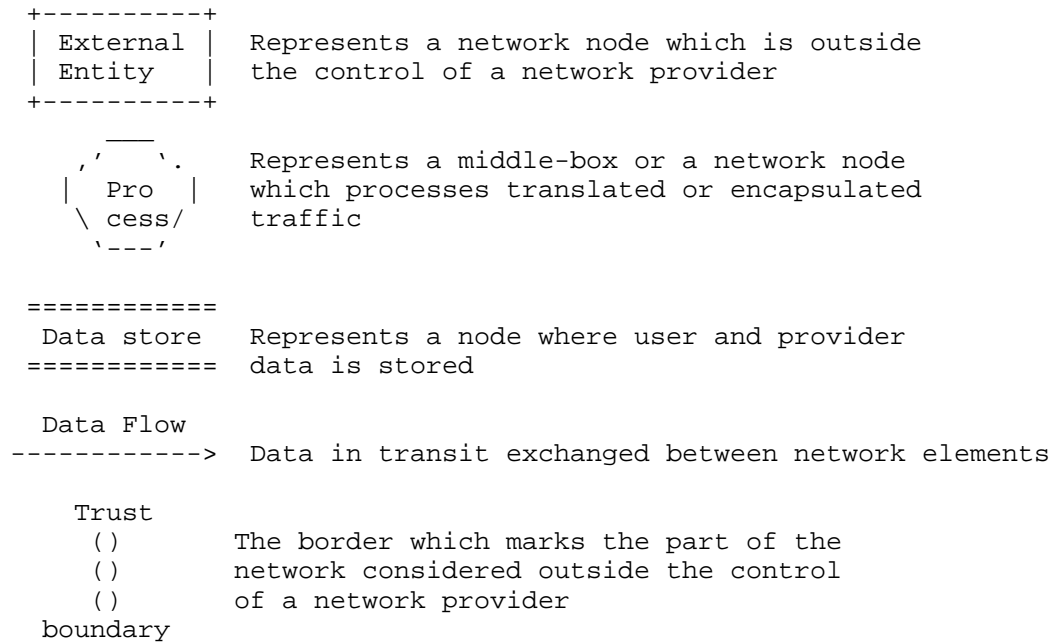


Figure 1. DFD Elements

4.4. Identify the threats

4.4.1. STRIDE-DFD Association

The STRIDE model associates the six categories of threats to each of the elements described in the DFD. Based on this association, we get an initial assessment of the threats as shown in Table 2. To clarify, a data flow, for example, is susceptible to tampering, information disclosure and denial of service threats. The initial threat assessment must be followed by a detailed analysis which should consider the protocols used in conjunction with the transition technology.

Table2. DFD-STRIDE Associations

S	T	R	I	D	E
#		#			
O	O	O	O	O	O
	=	=	=	=	
	>		>	>	
#	External entity				
O	Process				
=	Data store				
>	Data flow				

4.4.2. Level of Trust

We associate a level of trust with each entry point. Entry points that are trusted are assumed to behave as expected. That is, if the entry point is considered trusted, we can assume the likelihood of an attack is low. Furthermore, the six categories of STRIDE attacks could be assigned a likelihood by considering their association with the DFD elements that are entry points.

For instance, let's suppose we have an untrusted entry point (High likelihood of exploitation) which is also an external entity. Spoofing and repudiation are potential threats for an external entity. By association, the two types of attacks can be considered to have a high likelihood of being exploited. Using this logic, we can assign a likelihood value to each found threat. This can represent a base for prioritizing mitigation solutions. The likelihood levels can be defined in accordance with the levels of trust assigned to the the entry points.

4.4.3. Documenting the Threats

Each discovered threat should be documented using the format presented in Table 3.

Table2. Threat Info Format

Field Name	Description
Threat-ID	A code associated with each identified threat
Description	A summarized description of the threat
STRIDE	The association with the STRIDE categories
Mitigation	Details about possible mitigation solutions
Likelihood	Likelihood of the threat being exploited
Validation	Empirical validation data

The Threat-ID is supposed to be an easy way to refer and identify the threat within the IETF. The tentative format is IETF-TDB-[associated protocol/technology]-[serial number]. IETF-TDB stands for IETF Threat Database in the hope that in the future a threat database will be maintained within the IETF. The serial number is incremented with each threat found for a particular protocol or technology.

4.4.4. Complex Threats

As the subcomponents and subprotocols interact, the threats can fuse and result in convoluted threats with a higher likelihood of exploitation. Depending on the list of discovered threats, the possibility of a fusion between threats should be analyzed.

4.5. Review, Repeat and Validate

Steps 1 and 3 have to be reviewed in the context of potential changes in the technology function and associated protocols. Step 4 should be repeated periodically, as threats may have been overlooked, or the context set by steps 1 and 3 may have changed. If the transition technologies have existing implementations, the analysis should be confirmed with empirical data.

The next sections applied the proposed threat modeling approach to the IPv6 transition technologies identified in Section 3.

5. Dual Stack Threat Model

5.1. Establish the Function

The function for dual-stack transition technologies is to ensure a safe data exchange over a dual-stack infrastructure. In other words, the data can be transferred over both IPv4 and IPv6. From a network service perspective, the main function is data forwarding. This includes interior gateway routing solutions. We start with the assumption that services such as address provision, DNS resolution or exterior gateway routing are performed by other nodes within the core network. This assumption is common for all the four generic categories of IPv6 transition technologies.

5.2. Identify the Generic Category

Since we are targeting the generic category itself, the step is unnecessary here. This stands for the other three categories as well.

5.3. Decompose the Technology

A DFD for dual-stack transition technologies is presented in Figure 2. The diagram represents a basic use case and includes a minimal set of elements.

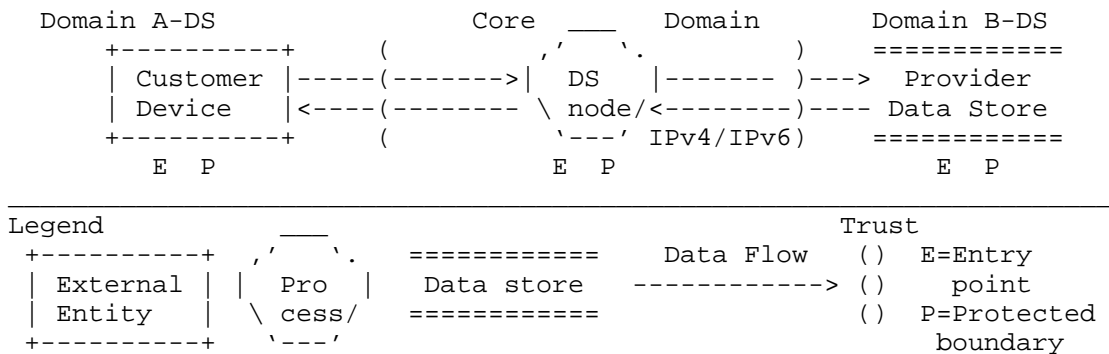


Figure 2. Data Flow Diagram (DFD) for Dual Stack (DS) technologies

In Domain A, which is assumed to be on the customer side we have a Customer Device which initiates the data exchange. It represents one of the entry points of the system and contains important data, which should be regarded as an asset and protected. The Customer Device is regarded as an external element because it is outside the control zone of the assumed network provider. The data request is transmitted over IPv4 or IPv6 to a Dual-stack node.

The Dual-stack node is another entry point and contains valuable topology information which should to protected as well. The Dual-stack node forwards in turn the data request to the provider data store. The Data store is the last entry point in the system and it is assumed to contain valuable data. The data reply is forwarded back to the customer device.

The only trusted entry point in the system is the Dual-stack node. The other two entry points are considered untrusted, since they are outside the control of the production network. That means they can be exploited with a higher likelihood by an attacker.

Considering the data can be transferred over both IPv4 and IPv6, we need to consider both IP protocol suites. Furthermore, the possibility of using security and routing protocols should be considered.

5.4. Identify the threats

5.4.1. STRIDE-DFD Association

By analyzing the DFD in association with the STRIDE threats per element chart, we can make the associations depicted in Table 3.

Table3. DFD-STRIDE Associations DS

```

+-----+-----+-----+-----+-----+-----+
| S | T | R | I | D | E |
+-----+-----+-----+-----+-----+-----+
| #-H |   | #-H |   |   |   |
+-----+-----+-----+-----+-----+-----+
| O-L|O-L|O-L|O-L|O-L|O-L|
+-----+-----+-----+-----+-----+-----+
|   |=-H|=-H|=-H|=-H|   |
+-----+-----+-----+-----+-----+-----+
|   |>-H|   |>-H|>-H|   |
+-----+-----+-----+-----+-----+-----+
| # | Customer device |
+-----+-----+-----+-----+-----+-----+
| O | DS node |
+-----+-----+-----+-----+-----+-----+
| = | Provider data store|
+-----+-----+-----+-----+-----+-----+
| > | Data flow |
+-----+-----+-----+-----+-----+-----+

```

5.4.2. From Trust to Likelihood

Looking at the associations in Table 3, The Customer Device can be subject to spoofing and repudiation attacks. It being an untrusted entry point, that means there is a high likelihood of an attack. This is marked in Table 3 with H.

The Dual-stack node can be subject to all six types of attacks. However, the likelihood of that happening is low, considering it is a trusted entry point.

The Data flow is vulnerable to tampering, information disclosure and denial of service. Considering it traverses untrusted parts of the system, the level of likelihood of an attack on the data flow is high.

Lastly, the Data store could potentially be targeted by tampering, repudiation, information disclosure and denial of service attacks. The likelihood for these to happen is high as well, the data store being an untrusted entry point.

5.4.3. Documenting the Threats

The Tables 5-10 of the Appendix contain a non-exhaustive collection of existing threats, which have been collected by surveying a part of existing literature on this subject. For further documentation, each threat has been provided with a reference in the first column. For reuse purposes, the threats are organized according to the categories of protocols which would be necessary for accomplishing the function of the IPv6 transition technologies.

For dual-stack transition technologies the protocol threats associated with the IPv4 suite (Table 6), IPv6 suite (Table 7), routing (Table 10) and switching (Table 5) could potentially be exploited from the 3 entries of the system: the untrusted (High likelihood of exploitation) Customer device, the trusted (Low likelihood of exploitation) Dual-stack node (Process) and untrusted (High likelihood of exploitation) Provider Data store.

The IPv4 suite, transport layer and most of the IPv6 suite protocols are associated with all the elements of the DFD. By extrapolation, their threats have a high likelihood of occurrence. Some of the IPv6 protocol threats (Table 7), namely IETF-TDB-ND-3 to IETF-TDB-ND-6 and the Layer 2 technologies' threats (Table 5) can only be associated with routers or switches. In the context of the DFD, they could only be associated with the Dual-stack node. That means they have a low likelihood of occurrence. Similarly, the routing protocols

(Table 10) can only be associated with the Dual-stack node. By association, they also have a low likelihood of being exploited.

5.4.4. Complex Threats

By analyzing the interaction between the three elements of the DFD and the protocols used by Dual stack transition technologies, we can uncover other threats. For example, if the IETF-TDB-ARP-1(ARP cache poisoning) is used to perform a Denial of Service attack on the Dual-stack node from the Customer device, the likelihood of exploitation rises for the IETF-TDB-ND-10 (ND Replay Attacks) threats. IETF-TDB-ARP-1 could be replaced by any other DoS threat associated with the IPv4 protocol suite. This complex threat could be prevented by ensuring that the IPv4 suite DoS threats are properly mitigated. Examples of convoluted threats for the four generic IPv6 transition technologies are presented in Table 4.

Table4. Complex Threats

	ThreatID	Description	S	T	R	I	D	E	Mitigation
1 V	IETF-TDB -DS-1	IETF-TDB -ARP-1 + IETF-TDB -ND-4	H	H	H	H	H		DoS Mitigation for IPv4 suite
2 V	IETF-TDB -DS-2	IETF-TDB -ARP-1 + IETF-TDB -OSPFv3-1	H	H	H	H	H	H	Crypto authen
3 X	IETF-TDB -ltransl-1	IETF-TDB IP/ICMP-3 + IETF-TDB -ICMPv6-1	H		H	H	H		No widely accepted mitigation
4 V	IETF-TDB -ltransl-2	IETF-TDB -TCP-1 + IETF-TDB -ND-4	H	H	H	H	H		Block non- internal traffic
5 X	IETF-TDB	IETF-TDB -IP/ICMP-4	L	L	L	L	L		No widely accepted

	-2transl-1	+ IETF-TDB -ND-4								mitigation	
6 V	IETF-TDB -2transl_2	IETF-TDB -IP/ICMP-1 + IETF-TDB -OSPFv3-1	L	L	L	L	L	L	L	reverse path checks	
7	IETF-TDB -encaps-1	IETF-TDB -IPv6-1 + IETF-TDB -4encaps_1					H	H		IPv4 firewall before decaps	
Legend											
H	associated with High likelihood						L	associated with Low likelihood			

Another convoluted threat can result from exploiting IPv4 or IPv6 spoofing threats to increase the likelihood of an attack on routing protocols with simple authentication, such as or IETF-TDB-OSPFv3-1, IETF-TDB-OSPFv2-1 or IETF-TDB-RIPv2-1. Since the attack could be performed from an untrusted entry point (Customer device or Data store), the likelihood of the threat being exploited rises to High. This type of attack can be mitigated by using cryptographic authentication for the routing protocols.

The list of threats can help technology implementors and network operators alike prioritize the threats and mitigate accordingly.

5.5. Review, Repeat and Validate

This step is necessary if the technology analyzed or associated protocols change. For example if the routing system were to be only OSPFv3, then the threats associated with other routing protocols could be ignored. Also, the detailed analysis of threats is far from exhaustive. In terms of convoluted new threats, only a few are presented as an example. If this was to be an updated database of threats, it would need constant update.

To further validate the presented threats, a simple penetration testbed was built. The details of the testbed are presented in Figure 3. MAP-T [RFC7599] was used as transition technology. Asamap [asamap2014], a transition implementation developed in Japan, was

used as the base for MAP-T. The threats which were successfully emulated, have been marked accordingly in the first column of Table 4. In the case of the convoluted threats identified for Dual-stack transition technologies, both threats were emulated successfully by performing ARP Cache Spoofing, Neighbor Advertisement (NA) flooding and simple traffic analysis.

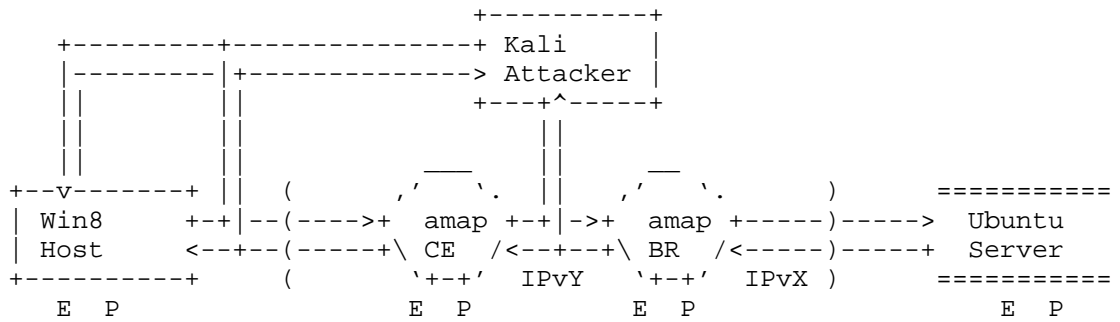


Figure 3. Pentestbed Setup

6. Single Translation Threat Model

To avoid redundant information, the following three subsections will only mark the differences with the threat modeling process presented for Dual-stack transition technologies.

One of the fundamental differences is that the single translation technologies would require a node to algorithmically translate the IPvX packets to IPvY, as shown in Figure 4.

6.1. Decompose the Technology

A DFD for single translation transition technologies is presented in Figure 4. The diagram represents a basic use case and includes a minimal set of elements.

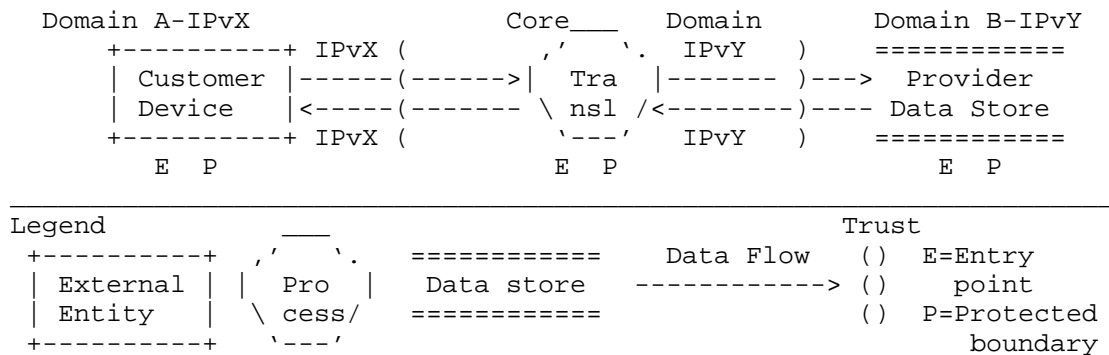


Figure 4. DFD for ltransl technologies

6.2. Identify the threats

For both translation directions 4->6 and 6->4, the threats for the IPv4 suite (Table 6), IPv6 suite (Table 7), routing (Table 10) and switching (Table 5) should be considered. There are technologies that use stateful mapping algorithms e.g. Stateful NAT64 [RFC6146], which create dynamic correlations between IP addresses or {IP address, transport protocol, transport port number} tuples. Consequently, we need to consider the protocols used at the transport layer (Table 9) as part of the attack surface. The threats presented in Table X, associated with the IP/ICMP translation algorithm (IP/ICMP) should be considered as well.

In terms of convoluted threats, one example could be exploiting the IETF-TDB-IP/ICMP-3 threat (IPAuth does not work with IP/ICMP) which would increase the likelihood of IETF-TDB-ND-4 (Default router is killed) or IETF-TDB-ND-5 (Good router goes bad) threats being exploited. Since there is no widely-accepted mitigation for any of the three threats, this convoluted threat is lacking a mitigation solution as well. Fortunately, both complex threats could not be validated empirically. An IPsec VPN connection was successfully established using UDP encapsulation between the Windows Host and the Ubuntu Server. Moreover, the IETF-TDB-ND-4 and IETF-TDB-ND-5 could not be validated empirically, as Asamap [asamap2014] does not accept RA messages when IPv6 forwarding is enabled.

If the IETF-TDB-TCP-1 threat (SYN flood) is exploited from an untrusted entry point, it increases the likelihood of a IETF-TDB-ND-10 (ND Replay attacks) threat. This threat can be mitigated by blocking packets with non-internal addresses from leaving the network. Both the SYN flood attack and the Neighbor Advertisement (NA) flooding attacks were staged successfully.

7. Double Translation Threat Model

The main difference between the Single translation case and the double translation case is the need for an extra translation device as part of the core network (Figure 5). Another important difference would be that in the untrusted zone, the Customer device and Data store would employ the same IP suite.

7.1. Decompose the Technology

A DFD for double translation transition technologies is presented in Figure 5. The diagram represents a basic use case and includes a minimal set of elements.

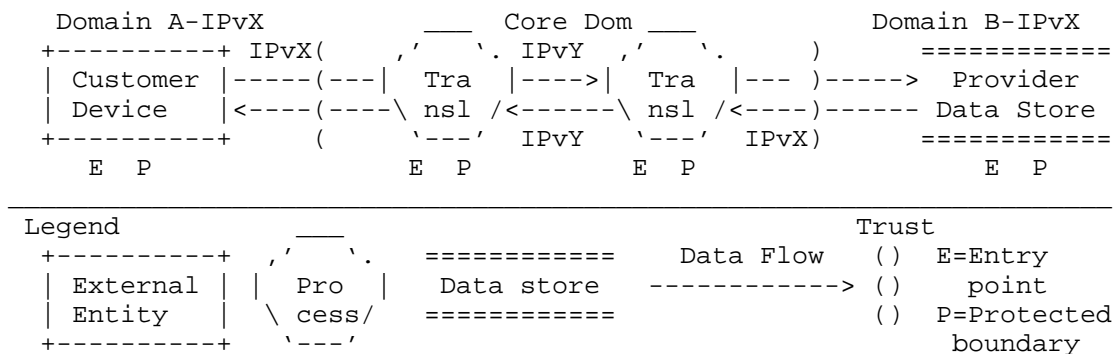


Figure 5. DFD for 2transl technologies

7.2. Identify the threats

The considered threats for the untrusted elements would be either the IPv4 suite (Table 6) or the IPv6 suite (Table 7) protocol threats. Similar to the single translation technologies, the routing (Table 10), switching (Table 5), transport layer (Table 9) and IP/ICMP (Table 8) threats should be analyzed as well.

The use of stateful translation mechanisms can expose a double translation technology to the IETF-TDB-IP/ICMP-4 threat (DoS by exhaustion of resources). A convoluted threat can result by exploiting this threat on one of the translators and the IETF-TDB-ND-4 or IETF-TDB-ND-5 threats on the other translator. This threat would have a higher likelihood of exploitation since it is associated with an untrusted entry point. In terms of mitigation, further investigation is needed, as there are no widely accepted mitigation techniques. Although the IETF-TDB-IP/ICMP-4 threat was replicated with success, the IETF-TDB-ND-10 or IETF-TDB-ND-5 could not be emulated because of a simple built-in mitigation mechanism

implemented by Asamap [asamap2014]. Router advertisement (RA) messages are not accepted while in IPv6 forwarding mode.

The IETF-TDB-IP/ICMP-4 threat can also fuse with the simple authentication threats such as IETF-TDB-OSPFv3-1 , IETF-TDB-OSPFv2-1 or IETF-TDB-RIPv2-1 to affect both translating nodes. The likelihood of the threats become higher by fusing them, since the flooding attack can be performed from an untrusted entry point, the customer network. This threat could be mitigated by using cryptographic authentication or implementing reverse path checks. The convoluted threat was validated by flooding the translation table of the first translator and forcing it to crash. OSPFv3 information disclosure was emulated with simple traffic analysis. To validate the other types of threats, a rogue router instance was created using Asamap [asamap2014].

8. Encapsulation Threat Model

Similar to double translation IPv6 transition technologies, encapsulation technologies, the core network traffic is forwarded through at least two devices, an Encapsulator and a Decapsulator (Figure 6). As the main difference, the traffic is encapsulated. This means more overhead but also more support for end-to-end security protocols. Packets are encapsulated either over IPv4 or IPv6.

8.1. Decompose the Technology

A DFD for encapsulation transition technologies is presented in Figure 6. The diagram represents a basic use case and includes a minimal set of elements.

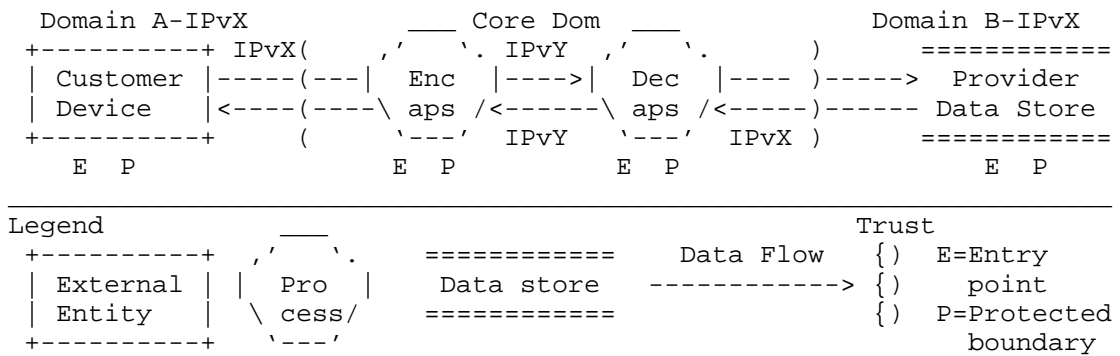


Figure 6. DFD for encaps technologies

8.2. Identify the threats

For the untrusted domain devices we would consider either the IPv4 suite (Table 6) or the IPv6 suite (Table 7) threats. In addition the routing (Table 10), switching (Table 5), transport layer (Table 9) and encapsulation-related (Table 8) threats should be considered.

Convolutated threats can arise by exploiting the IETF-TDB-4encaps-1 threat (avoiding IPv4 network security measures with encapsulation). This threat can facilitate IPv6 suite DoS threats on the Decapsulator device. This convolutated threat would increase the likelihood of a successful DoS attack from the Customer Device. The threat could be mitigated by making use of an IPv4 firewall before decapsulating the packets.

9. Acknowledgments

The author would like to thank Fernando Gont for his review and useful suggestions.

This document was derived from a template contributed by the xml2rfc project.

10. IANA Considerations

This memo includes no request to IANA.

All drafts are required to have an IANA considerations section (see Guidelines for Writing an IANA Considerations Section in RFCs [RFC5226] for a guide). If the draft does not require IANA to do anything, the section contains an explicit statement that this is the case (as above). If there are no requirements for IANA, the section will be removed during conversion into an RFC by the RFC Editor.

11. Security Considerations

This memo attempts to build a threat model for IPv6 transition technologies. The author would like to encourage the use of a similar threat modeling approach when writing the security considerations of standards developed in the IETF. To be more concrete the following steps could be reused:

R1 Identify the function

R2 Associate the technology with a generic category (if any)

R3 Decompose the technology

R4 Identify the threats

R5 Review, repeat and validate

12. References

12.1. Normative References

[draft-bmwg-v6trans]

Georgescu, M. and G. Lencse, "Benchmarking Methodology for IPv6 Transition Technologies", 2015, <<https://tools.ietf.org/html/draft-ietf-bmwg-ipv6-trans-tech-benchmarking-01>>.

[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>>.

[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>>.

12.2. Informative References

[arps] Abad, C. and R. Bonilla, "An analysis on the schemes for detecting and preventing ARP cache poisoning attacks", 2007.

[asamap2014]

Asama, M., "MAP supported Vyatta", 2014, <<http://enog.jp/~masakazu/vyatta/map/>>.

[bellovin89]

Bellovin, S., "Security Problems in the TCP/IP Protocol Suite", 1989.

[harris99]

Harris, B. and R. Hunt, "TCP/IP security threats and attack methods", 1999.

[icmps]

Low, C., "ICMP attacks illustrated", 2001.

[RFC2328]

Moy, J., "OSPF Version 2", STD 54, RFC 2328, DOI 10.17487/RFC2328, April 1998, <<http://www.rfc-editor.org/info/rfc2328>>.

- [RFC2629] Rose, M., "Writing I-Ds and RFCs using XML", RFC 2629, DOI 10.17487/RFC2629, June 1999, <<http://www.rfc-editor.org/info/rfc2629>>.
- [RFC3552] Rescorla, E. and B. Korver, "Guidelines for Writing RFC Text on Security Considerations", BCP 72, RFC 3552, DOI 10.17487/RFC3552, July 2003, <<http://www.rfc-editor.org/info/rfc3552>>.
- [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>>.
- [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>>.
- [RFC4213] Nordmark, E. and R. Gilligan, "Basic Transition Mechanisms for IPv6 Hosts and Routers", RFC 4213, DOI 10.17487/RFC4213, October 2005, <<http://www.rfc-editor.org/info/rfc4213>>.
- [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>>.
- [RFC4822] Atkinson, R. and M. Fanto, "RIPv2 Cryptographic Authentication", RFC 4822, DOI 10.17487/RFC4822, February 2007, <<http://www.rfc-editor.org/info/rfc4822>>.
- [RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, DOI 10.17487/RFC5226, May 2008, <<http://www.rfc-editor.org/info/rfc5226>>.
- [RFC5569] Despres, R., "IPv6 Rapid Deployment on IPv4 Infrastructures (6rd)", RFC 5569, DOI 10.17487/RFC5569, January 2010, <<http://www.rfc-editor.org/info/rfc5569>>.

- [RFC6052] Bao, C., Huitema, C., Bagnulo, M., Boucadair, M., and X. Li, "IPv6 Addressing of IPv4/IPv6 Translators", RFC 6052, DOI 10.17487/RFC6052, October 2010, <<http://www.rfc-editor.org/info/rfc6052>>.
- [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>>.
- [RFC6219] Li, X., Bao, C., Chen, M., Zhang, H., and J. Wu, "The China Education and Research Network (CERNET) IVI Translation Design and Deployment for the IPv4/IPv6 Coexistence and Transition", RFC 6219, DOI 10.17487/RFC6219, May 2011, <<http://www.rfc-editor.org/info/rfc6219>>.
- [RFC6274] Gont, F., "Security Assessment of the Internet Protocol Version 4", RFC 6274, DOI 10.17487/RFC6274, July 2011, <<http://www.rfc-editor.org/info/rfc6274>>.
- [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>>.
- [RFC6877] Mawatari, M., Kawashima, M., and C. Byrne, "464XLAT: Combination of Stateful and Stateless Translation", RFC 6877, DOI 10.17487/RFC6877, April 2013, <<http://www.rfc-editor.org/info/rfc6877>>.
- [RFC7596] Cui, Y., Sun, Q., Boucadair, M., Tsou, T., Lee, Y., and I. Farrer, "Lightweight 4over6: An Extension to the Dual-Stack Lite Architecture", RFC 7596, DOI 10.17487/RFC7596, July 2015, <<http://www.rfc-editor.org/info/rfc7596>>.
- [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>>.

[stride-shostack]

Shostack, A., "Experiences threat modeling at microsoft", 2008, <<http://mail.homeport.org/~adam/modsec08/Shostack-ModSec08-Experiences-Threat-Modeling-At-Microsoft.pdf>>.

[sws] Rouiller, S., "Virtual LAN Security: weaknesses and countermeasures", 2003.

[udps] Garg, A. and A. Reddy, "Mitigation of DoS attacks through QoS regulation", 2004.

[x1037] ITU-T, "IPv6 technical security guidelines. Recommendation X.1037", 2013, <<https://www.itu.int/rec/T-REC-X.1037/en>>.

Appendix A. Appendix A

Table5. L2 Technologies Threats

	ThreatID	Description	S	T	R	I	D	E	Mitigation
1	IETF-TDB -VLAN-1 [x1037]	Exhaust a the FIB of an L2switch					L		IEEE 802.1x authen tication
2	IETF-TDB -VLAN-2 [sws]	CAM Overflow					L		port -security features
3	IETF-TDB -VLAN-3 [sws]	Basic VLAN Hopping	L						Software update
4	IETF-TDB -VLAN-4 [sws]	Double encapsulation VLAN Hopping	L					L	Disable Auto -trunking
5	IETF-TDB -VLAN-5 [sws]	Spanning Tree Attack				L	L		Disable STP; BPDU Guard
Legend									
H	associated with High likelihood					L	associated with Low likelihood		

Table6. IPv4 Protocol Suite Threats

	ThreatID	Description	S	T	R	I	D	E	Mitigation
1	IETF-TDB -IPv4-1 [harris99]	IP source address spoofing	H	H	H	H			Apply ACLs filter source address traffic
2	IETF-TDB	Mal		H					Version

	-IPv4-2 [RFC6274]	formed version field							checked to be 4
3	IETF-TDB -IPv4-3 [RFC6274]	forged DSCP field	H					H	Filter unrecogn ized DSCP
4	IETF-TDB -IPv4-4 [RFC6274]	Buffer overflow IP frag mentation						H	avoid illegit imate re assembly
5	IETF-TDB -ICMP-1 [harris99]	Ping o'death						H	do not accept oversized ICMP
6	IETF-TDB -ICMP-2 [bellovin89]	ICMP redirects	H	H	H	H	H		don't update routing tables with ICMP Redirects
7	IETF-TDB -ICMP-3 [icmps]	ICMP sweep for recon						H	Selective filtering of ICMP
8	IETF-TDB -ICMP-6 [icmps]	ICMP flooding						H	Selective filtering of ICMP
9	IETF-TDB -ARP-1 [arps]	ARP cache poisoning	H	H	H	H	H		Static ARP entries, arpwatch
10	IETF-TDB -ARP-2 [RFC6274]	ARP cache overrun						H	Selective drop of packets

Table7. IPv6 Protocol Suite Threats

	ThreatID	Description	S	T	R	I	D	E	Mitigation
1	IETF-TDB -IPv6-1 [RFC4942]	Routing header to evade access controls	H				H		Access controls based on dest addresses
2	IETF-TDB -IPv6-2 [RFC4942]	Site-scope multicast addresses reconnaissance				H			Drop packets with site-scope dest addresses
3	IETF-TDB -IPv6-3 [RFC4942]	Anycast traffic identification				H			Restrict outside anycast services
4	IETF-TDB -IPv6-4 [RFC4942]	Extension headers excessive hop-by-hop options					H		Drop packets with unknown options
5	IETF-TDB -IPv6-5 [RFC4942]	Overuse of IPv6 router alert Option					H		Filter externally generated Router Alert packets
6	IETF-TDB -IPv6-6 [RFC4942]	IPv6 fragmentation overload of reconstruct buffers					H		Mandating the size of packet fragments
7	IETF-TDB -IPv6-7 [RFC4942]	IPv4-Mapped IPv6 Addresses	H				H		Avoid IPv4-mapped IPv6 addresses

8	IETF-TDB -ICMPv6-1 [RFC4443]	ICMPv6 spoofing	H				H		IPAuth
9	IETF-TDB -ICMPv6-2 [RFC4443]	ICMPv6 Redirects	H		H	H			IPAuth or ESP
10	IETF-TDB -ICMPv6-3 [RFC4443]	Back-to -back erroneous IP packets					H		ICMP error rate limiting
11	IETF-TDB -ICMPv6-4 [RFC4443]	Send ICMP Parameter Problem to multicast source				H	H		Secure multicast traffic
12	IETF-TDB -ICMPv6-5 [RFC4443]	ICMP passed to upper-layers					H		IPSec
14	IETF-TDB -SLAAC-1 [RFC4942]	Address Privacy Extensions Interaction with DDoS Defenses					H		Tune the change rate of the node address
15	IETF-TDB -ND-1 [RFC3756]	NS/NA Spoofing	H				H		SEND
16	IETF-TDB -ND-2 [RFC3756]	NUD failure					H		SEND
17	IETF-TDB -ND-3 [RFC3756]	Malicious Last Hop Router			L	L	L		SEND
18	IETF-TDB -ND-4 [RFC3756]	Default router is 'killed'			L	L	L		No widely accepted mitigation technique

19	IETF-TDB -ND-5 [RFC3756]	Good Router Goes Bad			L	L	L		No widely accepted mitigation technique
20	IETF-TDB -ND-6 [RFC3756]	Spoofed Redirect Message			L	L	L		SEND; Still an issue for ad-hoc cases
21	IETF-TDB -ND-7 [RFC3756]	Bogus On-Link Prefix					L		SEND
22	IETF-TDB -ND-8 [RFC3756]	Bogus Address Config Prefix					L		SEND; Still an issue for ad-hoc cases
23	IETF-TDB -ND-9 [RFC3756]	Parameter Spoofing	L		L	L			SEND; Still an issue for ad-hoc cases
24	IETF-TDB -ND-10 [RFC3756]	ND Replay attacks	H				H		SEND
25	IETF-TDB -ND-11 [RFC3756]	Neighbor Discovery DoS					H		Rate limit NS messages
26	IETF-TDB DAD_1 [RFC3756]	DAD DoS					H		SEND
27	IETF-TDB -SEND-1 [RFC3971]	Authorization Delegation Discovery DoS					H		Cache discovered info and limit the number of discovery processes

28	IETF-TDB -MIPv6-1 [RFC4942]	Obsolete Home Address Option Mobile IPv6	H							Secure Binding Update messages
----	-----------------------------------	---	---	--	--	--	--	--	--	---

Table8. Basic Transition Technologies Threats

	ThreatID	Description	S	T	R	I	D	E	Mitigation
1	IETF-TDB-IP/ICPM-1 [RFC6052]	IPv4 spoofing with IPv4-embedded IPv6	L						Implement reverse path checks
2	IETF-TDB-IP/ICMP-2 [RFC6145]	ESP fails with IPv6-to-IPv4 translation				L			Use checksum-neutral addresses
3	IETF-TDB-IP/ICMP-3 [rfc6145]	Auth Headers cannot be used across IPv6-to-IPv4				L			No widely accepted mitigation
4	IETF-TDB-IP/ICMP-4 [RFC6145]	Stateful translators resources exhaustion					L		No widely accepted mitigation
5	4encaps_1 [RFC4942]	Tunneling IPv6 over IPv4 breaks IPv4 Network's security assumptions				L			route encaps traffic through IPv4 firewall before decaps

Table9. L4 Technologies Threats

	ThreatID	Description	S	T	R	I	D	E	Mitigation
--	----------	-------------	---	---	---	---	---	---	------------

1	IETF-TDB -TCP-1 [harris99]	SYN flood						H		Block non- internal addresses from leaving
2	IETF-TDB -TCP-2 [harris99]	SYN /ACK flood	H		H			H		L3/L4 Packet Filtering
3	IETF-TDB -TCP-3 [harris99]	ACK or ACK -PUSH Flood	H		H			H		L3/L4 Packet Filtering
4	IETF-TDB -TCP-4 [harris99]	Frag mented ACK Flood						H		L3/L4 Packet Filtering
5	IETF-TDB -TCP-5 [harris99]	TCP Spoofing sequence number prediction	H							Block non -internal traffic from leaving
6	IETF-TDB -TCP-6 [harris99]	TCP session hijacking sequence number prediction	H	H	H	H	H	H	H	Block non -internal traffic from leaving
7	IETF-TDB -TCP-7 [harris99]	RST and FIN DoS						H		L3/L4 Packet Filtering Stateful Flow Awareness
8	IETF-TDB -UDP-8 [udps]	UDP flood						H		QoS regulation ;L3/L4 Packet Filtering

9	IETF-TDB -NAT44-9 [rfc7957]	Port set exhaustion						H		Address Dependent Filtering
---	-----------------------------------	------------------------	--	--	--	--	--	---	--	-----------------------------------

Table10. Routing Technologies Threats

	ThreatID	Description	S	T	R	I	D	E	Mitigation	
1 x	IETF-TDB -RIPv2-1 [RFC4822]	simple password authen	L	L	L	L	L	L	crypto authen	
2 x	IETF-TDB -OSPFv2-1 [RFC2328]	simple password authen	L	L	L	L	L	L	crypto authen	
3 x	IETF-TDB -OSPFv2-2 [RFC2328]	OSPFv2 authen sequence number prediction	L	L	L	L	L	L	crypto sequence number	
4	IETF-TDB -OSPFv3-1 [RFC4552]	OSPFv3 using the same manual key	L	L	L	L	L	L	no manual keys	
Legend										
H	associated with High likelihood				L	associated with Low likelihood				

Author's Address

Marius Georgescu (editor)
 NAIST
 Takayama 8916-5
 Nara 630-0192
 Japan

Phone: +81 743 72 5216
 Email: liviumarius-g@is.naist.jp
 URI: http://www.ipv6net.ro

Operations Area Working Group
Internet-Draft
Intended status: Best Current Practice
Expires: August 7, 2016

F. Gont
SI6 Networks / UTN-FRH
F. Baker
Cisco Systems
February 4, 2016

On Firewalls in Network Security
draft-gont-opsawg-firewalls-analysis-02

Abstract

This document analyzes the role of firewalls in network security, and recognizes their role in the internet architecture. It suggests a line of reasoning about their usage, and analyzes common kinds of firewalls and the claims made for them.

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 August 7, 2016.

Copyright Notice

Copyright (c) 2016 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.	Reasoning about Firewalls	4
3.1.	A Simple Model of Communication	4
3.2.	The Role of Firewalls in Internet Security	5
3.3.	Firewalls and The End-to-End Principle	5
4.	Common kinds of firewalls	6
4.1.	Perimeter security: Protection from aliens and intruders	7
4.2.	Pervasive access control	8
4.3.	Intrusion Management: Contract and Reputation filters . .	9
5.	Firewalling Strategies	10
5.1.	Blocking Traffic Unless It Is Explicitly Allowed (default deny)	11
5.2.	Allow Traffic Unless It Is Explicitly Blocked (default allow)	11
6.	Assumptions on IP addresses and Transport Protocol Port Numbers	12
7.	State Associated with Filtering Rules	13
8.	Enforcing Protocol Syntax at the Firewall	14
9.	Performing Deep Packet Inspection	14
10.	IANA Considerations	15
11.	Security Considerations	15
12.	Acknowledgements	15
13.	References	16
13.1.	Normative References	16
13.2.	Informative References	16
	Authors' Addresses	18

1. Introduction

Prophylactic perimeter security in the form of firewalls, and the proper use of them, have been a fractious sub-topic in the area of internet security. Firewalls have been largely seen by many in the IETF as a poor approach to security, and often as unnecessary and rather "evil" devices that hinder innovation and the deployment of new protocols and applications. Operationally, they are also seen by some as attack vectors, with state exhaustion attacks, side-effects of the imposition of symmetry requirements and single points of failure. This document analyzes the role of firewalls in network security, and recognizes their role in the internet architecture. It suggests a line of reasoning about their usage, and analyzes common kinds of firewalls and the claims made for them.

This document has, among others, the following goals:

- o Recognize the important role of firewalls in enterprise security architecture for providing "prophylactic" security, rather than as "evil" ad-hoc functionality/devices (see Section 3.2).
- o Analyze common kinds of firewalls and claims made for them (see Section 4).
- o Analyze implicit assumptions made by firewalls, identifying where/when some of those assumptions may not apply (see e.g. Section 6).
- o Discuss trade-offs in the possible firewalling paradigms (see Section 5).
- o Provide conceptual guidance regarding the use and deployment of .
- o Identify harmful behavior/policies commonly implemented and applied by firewalls, in the hopes of improving the state of affairs in that area.
- o Possibly trigger other work in the area of firewalls, as a result of the previous items.

2. Terminology

Firewall:

A device or software that imposes a policy whose effect is "a stated type of network traffic may or may not be allowed from A to B". The firewall may reside in the destination itself (a "host firewall"), or in any intermediate system (a "network firewall"). The firewalling functionality may be implemented in a general purpose system (e.g. an ACL in a router), or in a special purpose middleware device (e.g., a "firewall product"). The details of the policy, the granularity with which a policy can be applied, how such policy is configured, or of the firewall's implementation are just that - implementation details.

We also note that a firewall may enforce policies at different layers. Typically, the layer at which a firewall operates will impact the type of policies that a firewall will be able to apply: for example, a layer-3 firewall may be able to enforce simple policies based on layer-3 addresses and some simple layer-4 parameters such as transport protocol port numbers, while an "application firewall" may be able to enforce policies on higher-level entities such as application-request types. We note that all such firewall types essentially enforce the same role of enforcing a policy of some sort on network traffic, and hence are

referred to with the generic term "firewall" (or "firewall device" in some cases) throughout this document.

Perimeter:

The position in which the specific security policy applies. In typical deployed networks, there are usually some easy-to-define perimeters. A network connected with another network has a perimeter where the two meet, which is defined by what equipment is operated by each network. It invariably imposes a security policy at that boundary, which may be as simple as "all traffic is welcome" and as complex as matching arriving and departing traffic to ensure specific behaviors, or inspecting traffic according to various algorithms. Firewall functionality is usually implemented at or close to such network perimeters.

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].

3. Reasoning about Firewalls

3.1. A Simple Model of Communication

Any communication requires at least three components:

- o a sender, someone or some thing that sends a message,
- o a receiver, someone or some thing that receives the message, and
- o a channel, which is a medium by which the message is communicated.

In the Internet, the IP network is the channel; it may traverse something as simple as a directly connected cable or as complex as a sequence of ISPs, but it is the means of communication. In normal communications, a sender sends a message via the channel to the receiver, who is willing to receive and operate on it. In contrast, attacks are a form of harassment. A receiver exists, but is unwilling to receive the message, has no application to operate on it, or is by policy unwilling to. Attacks on infrastructure occur when message volume overwhelms infrastructure or uses infrastructure but has no obvious receiver.

By that line of reasoning, a firewall operating at layer-3 primarily protects infrastructure, by preventing traffic that would attack it from it. The best prophylactic might use a procedure for the dissemination of Flow Specification Rules [RFC5575] to drop traffic sent by an unauthorized or inappropriate sender or which has no host

or application willing to receive it as close as possible to the sender.

In other words, a firewall is comparable to the human skin, and has as its primary purpose the prophylactic defense of a network. By extension, the firewall also protects a set of hosts and applications, and the bandwidth that serves them, as part of a strategy of defense in depth. Since there is no one way to prevent attacks, a firewall is not itself a security strategy; the analogy to the skin would say that a body protected only by the skin has an immune system deficiency and cannot be expected to long survive. That said, every security solution has a set of vulnerabilities; the vulnerabilities of a layered defense is the intersection of the vulnerabilities of the various layers (e.g., a successful attack has to thread each layer of defense).

3.2. The Role of Firewalls in Internet Security

One could compare the role of firewalls in prophylactic perimeter security to that of the human skin: the service that the skin performs for the rest of the body is to keep common crud out, and as a result prevent much damage and infection that could otherwise occur. The body supplies prophylactic perimeter security for itself and then presumes that the security perimeter has been breached; real defenses against attacks on the body include powerful systems that detect changes (anomalies) counterproductive to human health, and recognizable attack syndromes such as common or recently-seen diseases. One might well ask, in view of those superior defenses, whether there is any value in the skin at all; the value is easily stated, however. It is not in preventing the need for the stronger solutions, but in making their expensive invocation less needful and more focused.

3.3. Firewalls and The End-to-End Principle

One common complaint about firewalls in general is that they violate the End-to-End Principle [Saltzer]. The End-to-End Principle is often incorrectly stated as requiring that "application specific functions ought to reside in the end nodes of a network rather than in intermediary nodes, provided they can be implemented 'completely and correctly' in the end nodes" or that "there should be no state in the network." What it actually says is heavily nuanced, and is a line of reasoning applicable when considering any two communication layers.

[Saltzer] "presents a design principle that helps guide placement of functions among the modules of a distributed computer system. The principle, called the end-to-end argument, suggests that

functions placed at low levels of a system may be redundant or of little value when compared with the cost of providing them at that low level."

In other words, the End-to-End Argument is not a prohibition against lower layer retries of transmissions, which can be important in certain LAN technologies, nor of the maintenance of state, nor of consistent policies imposed for security reasons. It is, however, a plea for simplicity. Any behavior of a lower communication layer, whether found in the same system as the higher layer (and especially application) functionality or in a different one, that from the perspective of a higher layer introduces inconsistency, complexity, or coupling, extracts a cost. That cost may be in user satisfaction, difficulty of management or fault diagnosis, difficulty of future innovation, reduced performance, or something else. Such costs need to be clearly and honestly weighed against the benefits expected, and used only if the benefit outweighs the cost.

From that perspective, introduction of a policy that prevents communication under an understood set of circumstances, whether it is to prevent access to pornographic sites or to prevent traffic that can be characterized as an attack, does not fail the End-to-End Argument; there are any number of possible sites on the network that are inaccessible at any given time, and the presence of such a policy is easily explained and understood.

What does fail the End-to-End Argument is behavior that is intermittent, difficult to explain, or unpredictable. If a site can be reached sometimes and not at other times, or can be reached using this host or application but not another, one will wonder why that is the case, and may not even know where to look for the issue.

4. Common kinds of firewalls

There are at least three common kinds of firewalls:

- o Context or Zone-based firewalls, that protect systems within a perimeter from systems outside it,
- o Pervasive routing-based measures, which protect intermingled systems from each other by enforcing role-based policies, and
- o Systems that analyze network traffic behavior and trigger on events that are unusual, match a signature, or involve an untrusted peer.

Each kind of firewall addresses a different view of the network. A zone-based firewall (Section 4.1) views the network as containing

zones of trust, and deems applications inside its zone of protection to be trustworthy. A role-based firewall (Section 4.2) identifies parties on the basis of membership in groups, and prevents unauthorized communication between groups. A reputation, anomaly, or signature-based intrusion management system (Section 4.3) depends on active administration, and permits known applications to communicate while excluding unknown or known-evil applications. In each case, the host or application is its own final bastion of defense, but having a host blocking incoming traffic (so-called "host firewalls") does not defend infrastructure. That is, each type of prophylactic has a purpose, and none of them is a complete prophylactic defense.

Each type of defense, however, can be assisted by enabling an application running in a host to inform the network of what it is willing to receive. As noted in Section 4.1, a zone-based firewall, generally denies all incoming sessions and permits responses to sessions initiated outbound from the zone, but can in some cases be configured to also permit specific classes of incoming session requests, such as WWW or SMTP to an appropriate server. A simple way to enable a zone-based firewall to prevent attacks on infrastructure (traffic to an un-instantiated address or to an application that is off) while not impeding traffic that has a willing host and application would be for the application to inform the firewall of that willingness to receive incoming sessions. The Port Control Protocol [RFC6887], or PCP, is an example of a protocol designed for that purpose.

4.1. Perimeter security: Protection from aliens and intruders

As discussed in [RFC6092], the most common kind of firewall is used at the perimeter of a network. Perimeter security assumes two things: that applications and equipment inside the perimeter are under the control of the local administration and are therefore probably doing reasonable things, and that applications and equipment outside the perimeter are unknown.

For example, it may enforce simple permission rules, such as that external web clients are permitted to access a specific web server or that external SMTP MTAs are permitted to access internal SMTP MTAs. Apart from those rules, a session may be initiated from inside the perimeter, and responses from outside will be allowed through the firewall, but sessions may never be initiated from outside.

In addition, perimeter firewalls often perform some level of inspection/analysis, either as application proxies or through deep packet inspection, to verify that the protocol claimed to be being passed is in fact the protocol being passed.

In many scenarios the existence and definition of zone-based perimeter defenses is arguably a side-effect of the deployment of Network Address Translation [RFC2993]. Since e.g. a single address is shared among multiple systems, the NAT device needs to translate both the IP addresses and the transport protocol ports in order to multiplex multiple communication instances from different nodes into the same external address. Thus, the NAT device must keep a state table to know how to translate the IP addresses and transport protocol ports of incoming packets. Packets originating from the internal network will either match an existing entry in the state table, or create a new one. On the other hand, packets originating in the external network will either match an existing entry in the state table, or be dropped. Thus, as a side effect, NATs implicitly require that communication be initiated from the internal network, and only allow return traffic from the external network. We note that this is a side-effect of multiplexing traffic from multiple nodes on a single IP address, rather than a design goal of NAT devices or their associated network translation function.

Some applications make the mistake of coupling application identities to network layer addresses, and hence employ such addresses in the application protocol. Thus, Network Address Translation forces the translator to interpret packet payloads and change addresses where used by applications.

As a result, if the transport or application headers are not understood by the translator, this has the effect of damaging or preventing communication. Detection of such issues can be sold as a security feature, although it is really a side-effect of a failure. While this can have useful side-effects, such as preventing the passage of attack traffic that masquerades as some well-known protocol, it also has the nasty side-effect of making innovation difficult. This has slowed the deployment of SCTP [RFC4960], since a firewall will often not permit a protocol it does not know even if a user behind it opens the session. When a new protocol or feature is defined, the firewall needs to stop applying that rule, and that can be difficult to make happen.

4.2. Pervasive access control

Another access control model, often called "Role-based", tries to control traffic in flight regardless of the perimeter. Given a rule that equipment located in a given routing domain or with a specific characteristic (such as "student dorms") should not be able to access equipment in another domain or with a specific characteristic (such as "academic records"), it might prevent routing from announcing the second route in the domain of the first, or it might tag individual packets ("I'm from the student dorm") and filter on those tags at

enforcement points throughout network. Such rules can be applied to individuals as well as equipment; in that case, the host needs to tag the traffic, or there must be a reliable correlation between equipment and its user.

One common use of this model is in data centers, in which physical or virtual machines from one tenant (which is not necessarily an "owner" as much as it is a context in which the system is used) might be co-resident with physical or virtual machines from another. Inter-tenant attacks, espionage, and fraud are prevented by enforcing a rule that traffic from systems used by any given tenant is only delivered to other systems used by the same tenant. This might, of course have nuances; under stated circumstances, identified systems or identified users might be able to cross such a boundary.

The major impediment in deployment is complexity. The administration has the option to assign policies for individuals on the basis of their current location (e.g. as the cross-product of people, equipment, and topology), meaning that policies can multiply wildly. The administrator that applies a complex role-based access policy is probably most justly condemned to live in the world he or she has created.

4.3. Intrusion Management: Contract and Reputation filters

The model proposed in Advanced Security for IPv6 CPE [I-D.vyncke-advanced-ipv6-security] could be compared to purchasing an anti-virus software package for one's computer. The proposal is to install a set of filters, perhaps automatically updated, that identify "bad stuff" and make it inaccessible, while not impeding anything else.

It depends on four basic features:

- o A frequently-updated signature-based Intrusion Prevention System which inspects a pre-defined set of protocols at all layers (from layer-3 to layer-7) and uses a vast set of heuristics to detect attacks within one or several flows. Upon detection, the flow is terminated and an event is logged for further optional auditing.
- o A centralized reputation database that scores prefixes for degree of trust. This is unlikely to be on addresses per se, since e.g. temporary addresses [RFC4941] change regularly and frequently.
- o Local correlation of attack-related information, and
- o Global correlation of attacks seen, in a reputation database.

The proposal does not mention anomaly-based intrusion detection, which could be used to detect zero-day attacks and new applications or attacks. This would be an obvious extension.

The comparison to anti-virus software is real; anti-virus software uses similar algorithms, but on API calls or on data exchanged rather than on network traffic, and for identified threats is often effective.

The proposal also has weaknesses:

- o People do not generally maintain anti-virus packages very well, letting contracts expire,
- o Reputation databases have a bad reputation for distributing information which is incorrect, out of date, or compromised by attackers,
- o Anomaly-based analysis identifies changes but is often ineffective in determining whether new application or application behaviors are pernicious (false positives). Someone therefore has to actively decide - a workload the average homeowner might have little patience for, and
- o Signature-based analysis applies to attacks that have been previously identified, and must be updated as new attacks develop. As a result, in a world in which new attacks literally arise daily, the administrative workload can be intense, and reflexive responses like accepting https certificates that are out of date or the download and installation of unsigned software on the assumption that the site administrator is behind are themselves vectors for attack.

Security has to be maintained to be useful, because attacks are maintained.

5. Firewalling Strategies

There is a great deal of tension in firewall policies between two primary goals of networking: the security goal of "block traffic unless it is explicitly allowed" and the networking goal of "trust hosts with new protocols". The two inherently cannot coexist easily in a set of policies for a firewall.

The following subsections discuss the "default deny" and "default allow" security paradigms.

5.1. Blocking Traffic Unless It Is Explicitly Allowed (default deny)

Many networks enforce the so-called "default deny" policy, in which traffic is blocked unless it is explicitly allowed. The rationale for such policy is that it is easier to open "holes" in a firewall to allow specific protocols, than trying to block all protocols that might be employed as an attack vectors; and that a network should only support the protocols it has been explicitly meant to support.

The drawback of this approach is that the security goal of "block traffic unless it is explicitly allowed" prevents useful new applications. This problem has been seen repeatedly over the past decade: a new and useful application protocol is specified, but it cannot get wide adoption because it is blocked by firewalls. The result has been a tendency to try to run new protocols over established applications, particularly over HTTP [RFC3205]. The result is protocols that do not work as well they might if they were designed from scratch.

Worse, the same goal prevents the deployment of useful transports other than TCP, UDP, and ICMP. A conservative firewall that only knows those three transports will block new transports such as SCTP [RFC4960]; this in turn causes the Internet to not be able to grow in a healthy fashion. Many firewalls will also block TCP and UDP options they don't understand, and this has the same unfortunate result.

5.2. Allow Traffic Unless It Is Explicitly Blocked (default allow)

Some networks enforce the so-called "default allow" policy, in which traffic is allowed unless it is explicitly blocked. This policy is usually enforced at perimeters where a comprehensive security policy is not really desirable or possible, but some level of packet filtering is considered appropriate. One common example of such policy could be an ISP blocking TCP port 25 (SMTP), but allowing all other traffic.

When a strict security policy is to be enforced (e.g., at an organizational network's edge), the "default allow" policy tends to be rather inappropriate, since it is usually easier and more effective to identify the traffic that must be allowed through the firewall (and open the necessary "holes" in the firewall) than to identify and block all traffic that may be considered undesirable/inappropriate.

6. Assumptions on IP addresses and Transport Protocol Port Numbers

In a number of scenarios, simple firewall rules have traditionally been specified in terms of the associated IP addresses and transport protocol port numbers. In general, this assumes that the associated IP addresses are stable, and that there is a "well known" transport protocol port number associated with each application.

In the IPv4 world, IP addresses may be considered rather stable. However, IPv6 introduces the concept of "temporary addresses" [RFC4941] which, by definition, change over time. This may prevent the enforcement of filtering policies based on specific IPv6 addresses, or may lead to filtering based on a more coarse granularity (e.g. specific address prefixes, as opposed to specific IPv6 addresses). In some scenarios, from the point of view of enforcing filtering policies, it might be desirable to disable temporary addresses altogether.

For example, an administrator might prefer that a caching DNS server, a secondary DNS server doing zone transfers, or an SMTP MTA, always employ the same source IPv6 address, as opposed to the temporary addresses that change over time.

The server-side transport protocol port is generally the so-called "well-known port" corresponding to the associated application. While widespread, this practice should probably be considered a kludge/short-cut rather than a "design principle" that can be relied upon for the general case. For example, use of DNS SRV records [RFC2782], or applications such as "portmapper" [Portmap] [RFC1833] might mean that the associated transport protocol port number cannot be assumed to be well-known, but rather needs to be dynamically learned. In other cases, applications may employ (by design) ephemeral port numbers, and there may be no obvious way to dynamically learn the port number being employed. FTP [RFC0959] and SIP [RFC3261] are examples of such applications.

Finally, as a result of widespread packet filtering, many protocols tend to be tunneled employing specific transport-protocol port numbers that are known to be more generally allowed by firewalls, such as TCP port 80 (HTTP). This essentially breaks the assumption that port numbers actually identify the actual application protocol using them.

Some of the so called "next generation" firewalls make fewer assumptions about port numbers, and tend to analyze the application data stream in order to infer the application protocol type, regardless of the well-known port being used. While this may prevent the circumvention of some security controls, it also implies Deep

Packet Inspection (DPI), and therefore there are a number of associated considerations, both in terms of introduced attack vectors and other possibilities for evasion of security controls (please see Section 9 for further discussion).

7. State Associated with Filtering Rules

There are two main paradigms for packet filtering:

- o Stateless filtering
- o Stateful filtering

Stateless filtering implies that the decision on whether to allow or block a specific traffic entity is based solely on the contents of such entity. One common example of such paradigm is the enforcement of network ingress filtering [RFC2827], in which packets may be blocked based on their IP addresses. Stateless filtering scales well, since there are no state requirements on the filtering device other than that associated with maintaining the filtering rules to be applied to the incoming traffic entities (e.g., packets).

On the other hand, stateful filtering implies that the decision on whether to allow or block a traffic entity is not only based on the contents of such entity, but also on the existence (or lack of) previous state associated with such entity. A common example of such paradigm is a firewall that "allows outbound connection requests and only allows return traffic from the external network" (such as the policy implicitly enforced by most NAT devices). For obvious reasons, the firewall needs to maintain state in order to be able to enforce such policies; that is, the firewall may need to keep track of all on-going communication instances, possibly applying timeouts and garbage collection on the associated state table.

Stateful filtering tends to allow more powerful packet filtering, at the expense of increased state. Thus, stateful filtering may be desirable when trying to perform deep packet inspection, but may be undesirable when the firewall is meant to block some Denial of Service attacks, since the firewall itself may become "the weakest link in the chain". Typically, the higher the firewall operates in the network stack, the more state will be required associated. For example, in order for a firewall to enforce a filtering policy based on application-layer request types, the firewall will need to enforce its filtering policy on the application-layer protocol stream, thus implying the need to perform layer-3 and layer-4 reassembly, etc.

When stateful packet filtering is warranted, its associated security implications should be considered. For example, an administrator may

want to enforce traffic filtering to mitigate denial of service attacks; however, when enforcement of such filtering implies increased state at the firewall, the firewall itself may become the easiest target for performing a denial of service attack.

8. Enforcing Protocol Syntax at the Firewall

Some firewalls try to enforce the protocol syntax by checking that only traffic complying with existing protocol definitions is allowed. While this can have useful side-effects, such as preventing the aforementioned traffic from triggering pathological behavior at the target system, it also has the nasty side-effect of making innovation difficult. For example, one of the issues in the deployment of Explicit Congestion Notification [RFC3168] has been that common firewalls often inspect reserved/unused bits and require them to be set to zero to close covert channels. Another example is the plethora of filtering rules applied to DNS traffic [DNS-FILTERING]. When a new protocol or feature is defined, the firewall needs to stop applying that rule, and that can be difficult to make happen.

NOTE:

A somewhat related concept is that of traffic normalization (or "scrubbing"), in which the filtering device can "normalize" traffic by e.g. clearing bits that are expected to be cleared, changing some protocol fields such that they are within "normal" ranges, etc. (see e.g. the discussion of "traffic normalization" in [OpenBSD-PF]). While this can have the useful effect of blocking DoS attacks to sloppy implementations that do not enforce sanity checks on the received packets, it also has the nasty side-effect of making innovation difficult, or even breaking deployed protocols. For example, some firewalls are known to enforce a default packet normalization policy that clears the TCP URG bit, as a result of the TCP urgent mechanism being associated with some popular DoS attacks. Widespread deployment of such firewalls has essentially rendered the TCP urgent mechanism unusable, leading to its eventual formal deprecation in [RFC6093].

We note that, as per our definition of "firewall" in Section 2, "traffic normalization" is not considered a firewall function.

9. Performing Deep Packet Inspection

While filtering packets based on the layer-3 protocol header fields is rather simple and straight-forward, performing enforcing a filtering policy at upper layer protocols can be a challenging task.

For example, IP fragmentation may make this task quite challenging, since even the very layer-4 protocol header could be present in a

non-first fragment. In a similar vein, IPv6 extension headers may represent a challenge for a filtering device, since they can result in long IPv6 extension header chains [RFC7112] [I-D.gont-v6ops-ipv6-ehs-packet-drops].

This problem is exacerbated as one tries to filter packets based on upper layer protocol contents, since many of such protocols implement some form of fragmentation/segmentation and reassembly. In many cases, the reassembly process could possibly lead to different results, and this may be exploited by attackers for circumventing security controls [Ptacek1998] [RFC6274].

In general, the upper in the protocol stack that a filtering policy is to be enforced, the more complex the task becomes: an attacker has more opportunities for obfuscation, ranging from e.g. ambiguities in IP and/or TCP reassembly, to e.g. application-layer obfuscation (such as HTTP URL obfuscation or JavaScript bytecode obfuscation). This usually implies that, in order to reliably enforce a filtering policy, more state is required on the firewall; and the considerations in Section 7 should be evaluated.

10. IANA Considerations

This memo asks the IANA for no new parameters. It can be published as an RFC by the RFC Editor.

11. Security Considerations

This document recognizes the role of firewalls in network security, and discusses a number of considerations associated with firewalls which may be of use when designing or deploying firewalls. This document, by itself, does not introduce any security implications.

12. Acknowledgements

The authors would like to thank (in alphabetical order) Fleming Andraeson, Mark Andrews, Lee Howard, Joel Jaeggli, Al Morton, Eric Vyncke and James Woodyatt, for providing valuable comments on earlier versions of this document.

This document is based on [I-D.ietf-opsawg-firewalls-00] authored by Fred Baker, and [I-D.ietf-opsawg-firewalls-01] authored by Paul Hoffman.

13. References

13.1. Normative References

- [RFC1833] Srinivasan, R., "Binding Protocols for ONC RPC Version 2", RFC 1833, DOI 10.17487/RFC1833, August 1995, <<http://www.rfc-editor.org/info/rfc1833>>.
- [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>>.
- [RFC3205] Moore, K., "On the use of HTTP as a Substrate", BCP 56, RFC 3205, DOI 10.17487/RFC3205, February 2002, <<http://www.rfc-editor.org/info/rfc3205>>.
- [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>>.
- [RFC6887] Wing, D., Ed., Cheshire, S., Boucadair, M., Penno, R., and P. Selkirk, "Port Control Protocol (PCP)", RFC 6887, DOI 10.17487/RFC6887, April 2013, <<http://www.rfc-editor.org/info/rfc6887>>.
- [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>>.

13.2. Informative References

- [DNS-FILTERING]
Andrews, M., "On Firewalls in Internet Security (Fwd: New Version Notification for draft-gont-opsawg-firewalls-analysis-00.txt)", post to the OPSAWG mailing-list, Message-Id: <20151012002551.8F7CD3A2FFD8@rock.dv.isc.org>, 2015, <<https://mailarchive.ietf.org/arch/msg/opsawg/2YQl6xBz6jtMyIkyAx59U-oPmPQ>>.
- [I-D.gont-v6ops-ipv6-ehs-packet-drops]
Gont, F., Hilliard, N., Doering, G., LIU, S., and W. Kumari, "Operational Implications of IPv6 Packets with Extension Headers", draft-gont-v6ops-ipv6-ehs-packet-drops-02 (work in progress), February 2016.

- [I-D.ietf-opsawg-firewalls-00]
Baker, F., "On Firewalls in Internet Security", draft-ietf-opsawg-firewalls-00 (work in progress), June 2012.
- [I-D.ietf-opsawg-firewalls-01]
Baker, F. and P. Hoffman, "On Firewalls in Internet Security", draft-ietf-opsawg-firewalls-01 (work in progress), October 2012.
- [I-D.vyncke-advanced-ipv6-security]
Vyncke, E., Yourtchenko, A., and M. Townsley, "Advanced Security for IPv6 CPE", draft-vyncke-advanced-ipv6-security-03 (work in progress), October 2011.
- [OpenBSD-PF]
OpenBSD, , "pf(4) manual page: pf -- packet filter", 2015, <<http://www.openbsd.org/cgi-bin/man.cgi/OpenBSD-current/man4/pf.4&query=pf>>.
- [Portmap] Wikipedia, , "Portmap", 2014, <<https://en.wikipedia.org/wiki/Portmap>>.
- [Ptacek1998]
Ptacek, T. and T. Newsham, "Insertion, Evasion and Denial of Service: Eluding Network Intrusion Detection", 1998, <<http://www.aciri.org/vern/Ptacek-Newsham-Evasion-98.ps>>.
- [RFC0959] Postel, J. and J. Reynolds, "File Transfer Protocol", STD 9, RFC 959, DOI 10.17487/RFC0959, October 1985, <<http://www.rfc-editor.org/info/rfc959>>.
- [RFC2782] Gulbrandsen, A., Vixie, P., and L. Esibov, "A DNS RR for specifying the location of services (DNS SRV)", RFC 2782, DOI 10.17487/RFC2782, February 2000, <<http://www.rfc-editor.org/info/rfc2782>>.
- [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>>.
- [RFC2993] Hain, T., "Architectural Implications of NAT", RFC 2993, DOI 10.17487/RFC2993, November 2000, <<http://www.rfc-editor.org/info/rfc2993>>.

- [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>>.
- [RFC3261] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, DOI 10.17487/RFC3261, June 2002, <<http://www.rfc-editor.org/info/rfc3261>>.
- [RFC4960] Stewart, R., Ed., "Stream Control Transmission Protocol", RFC 4960, DOI 10.17487/RFC4960, September 2007, <<http://www.rfc-editor.org/info/rfc4960>>.
- [RFC5575] Marques, P., Sheth, N., Raszuk, R., Greene, B., Mauch, J., and D. McPherson, "Dissemination of Flow Specification Rules", RFC 5575, DOI 10.17487/RFC5575, August 2009, <<http://www.rfc-editor.org/info/rfc5575>>.
- [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>>.
- [RFC6093] Gont, F. and A. Yourtchenko, "On the Implementation of the TCP Urgent Mechanism", RFC 6093, DOI 10.17487/RFC6093, January 2011, <<http://www.rfc-editor.org/info/rfc6093>>.
- [RFC6274] Gont, F., "Security Assessment of the Internet Protocol Version 4", RFC 6274, DOI 10.17487/RFC6274, July 2011, <<http://www.rfc-editor.org/info/rfc6274>>.
- [Saltzer] Saltzer, J., Reed, D., and D. Clark, "End-to-end arguments in system design", ACM Transactions on Computer Systems (TOCS) v.2 n.4, p277-288, Nov 1984.

Authors' Addresses

Fernando Gont
SI6 Networks / UTN-FRH
Evaristo Carriego 2644
Haedo, Provincia de Buenos Aires 1706
Argentina

Phone: +54 11 4650 8472
Email: fgont@si6networks.com
URI: <http://www.si6networks.com>

Fred Baker
Cisco Systems
Santa Barbara, California 93117
USA

Email: fred@cisco.com

opsec
Internet-Draft
Intended status: Best Current Practice
Expires: January 4, 2018

F. Gont
UTN-FRH / SI6 Networks
R. Hunter
Globis Consulting BV
J. Massar
Massar Networking
W. Liu
Huawei Technologies
July 3, 2017

Defeating Attacks which employ Forged ICMPv4/ICMPv6 Error Messages
draft-gont-opsec-icmp-ingress-filtering-03.txt

Abstract

Over the years, a number of attack vectors that employ forged ICMPv4/ICMPv6 error messages have been disclosed and exploited in the wild. The aforementioned attack vectors do not require that the source address of the packets be forged, but do require that the addresses of the IPv4/IPv6 packet embedded in the ICMPv4/ICMPv6 payload be forged. This document discusses a simple, effective, and straightforward method for using ingress traffic filtering to mitigate attacks that use forged addresses in the IPv4/IPv6 packet embedded in an ICMPv4/ICMPv6 payload.

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. Applicability Statement	3
4. Overview	3
4.1. Generation of ICMP Error Messages in Legitimate Scenarios	4
4.2. Attack Scenario	5
5. ICMPv4/ICMPv6 Network Ingress Filtering	7
6. IANA Considerations	7
7. Security Considerations	7
8. Acknowledgements	8
9. References	8
9.1. Normative References	8
9.2. Informative References	9
Authors' Addresses	9

1. Introduction

Over the years, a number of attack vectors that employ forged ICMPv4/ICMPv6 error messages have been disclosed and exploited in the wild. The effects of these attack vectors have ranged from Denial of Service (DoS) to performance degradation [US-CERT] [RFC5927] [I-D.gont-v6ops-ipv6-ehs-packet-drops].

The aforementioned attack vectors do not require that the Source Address of the ICMP [RFC0792] or ICMPv6 [RFC4443] attack packets to be forged, but do require that the Destination Address of the IPv4 [RFC0791] (in the case of ICMPv4) or IPv6 (in the case of ICMPv6) packet embedded in the ICMPv4/ICMPv6 payload be forged. Thus, performing ingress filtering (ala BCP38 [RFC2827]) on the Destination Address of the embedded IPv4/IPv6 packet results in a simple, effective, and straightforward mitigation for any attack vectors based on ICMPv4/ICMPv6 error messages.

Section 4 provides an overview of how ICMP/ICMPv6 error messages are generated, and how packets are crafted to perform attacks based on

ICMPv4/ICMPv6 error messages. Section 5 specifies network ingress filtering based on the ICMP/ICMPv6 payload.

2. Terminology

Throughout this document the term "IP" is employed to refer to both the IPv4 [RFC0791] and IPv6 [RFC2460] protocols. That is, the term "IP" is employed when we do not mean to make a distinction between both versions of the protocol. In a similar vein, the term "ICMP" is employed to refer to both the ICMPv4 [RFC0792] and ICMPv6 [RFC4443] protocols. That is, the term "ICMP" is employed when we do not mean to make a distinction between both versions of the protocol.

For obvious reasons, ICMPv4 will only be employed in conjunction with IPv4, and ICMPv6 will always be employed in conjunction with IPv6. That is, the phrase "the IP packet embedded in the ICMP payload" means "the IPv4 packet embedded in the ICMPv4 payload" payload or "the IPv6 packet embedded in the ICMPv6 payload" (but NOT e.g. "the IPv4 packet embedded in the ICMPv6 payload").

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].

3. Applicability Statement

The filtering policy specified in this document could be enforced at the border firewall of a non-multihomed network or at a CPE router, such that users of that network are prevented from performing ICMP-based attacks against other parties.

The filtering policy specified in this document SHOULD NOT be enforced in multihoming scenarios, or other scenarios where this policy could lead to false positives and therefore incorrect packet drops.

4. Overview

Attack vectors based on ICMP error messages have been known for a long time, and have been described in detail in [RFC5927]. The following subsections provide an overview of how ICMP error messages are generated in legitimate scenarios, and how an attacker would forge an ICMP error message in order to perform an attack based on ICMP error messages.

4.1. Generation of ICMP Error Messages in Legitimate Scenarios

The following figure illustrates a very simple network scenario in which two hosts (H1 and H2) are connected to each other by means of the router R1:

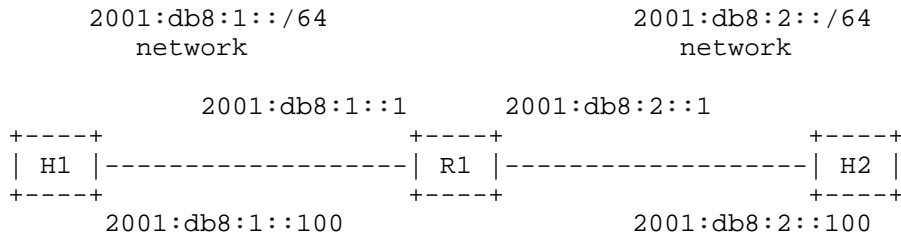


Figure 1: Sample Scenario for ICMP/ICMPv6 Error Generation

The aforementioned figure illustrates the IPv6 addresses assigned to each of the involved network interfaces. For simplicity sake, this figure employs only IPv6 addresses, but the same logic applies to the IPv4 case.

Let us assume that H1 sends a packet towards H2, and that R1 encounters an error condition while processing such a packet. Typically, the error condition will be reported to H1 by means of an ICMPv6 error message. The error message will have the following structure:

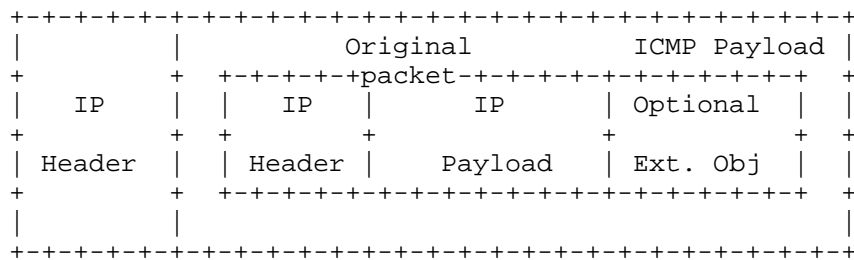


Figure 2: Structure of ICMPv4/ICMPv6 Error Messages

NOTES:

For completeness-sake, the figure above depicts the structure of ICMP error messages including ICMP extension objects (see [RFC4884]). Use of such extension objects does not affect the discussion in this document.

In the IPv6 case, the "IP header" corresponds to the entire IPv6 header chain. Additionally, in the IPv4 scenarios in which

Network Address Translation (NAT) is in place, the NAT device could fail to translate the IPv4 addresses of the embedded packet.

where the ICMPv6 error message embeds the whole (or part of) the original packet that elicited the error message.

In our scenario, the relevant header fields would have the following values:

- o Source Address: 2001:db8:1::1
- o Destination Address: 2001:db8:1::100
- o Source Address (embedded packet): 2001:db8:1::100
- o Destination Address (embedded packet): 2001:db8:2::100

It should be clear that the Source Address of the packet could be virtually any address (since it corresponds to the IP address of a router reporting the error), while the Destination Address of the packet will be that of the target/destination of the ICMP error message. On the other hand, the IP addresses of the embedded packet will be those of the packet that elicited the ICMP error message.

The embedded IP packet is typically employed by the receiving system to demultiplex the ICMP error message.

4.2. Attack Scenario

The following figure illustrates a very simple attack scenario in which an attacker (H3) tries to perform an attack against H1, while H1 is communicating with H2:

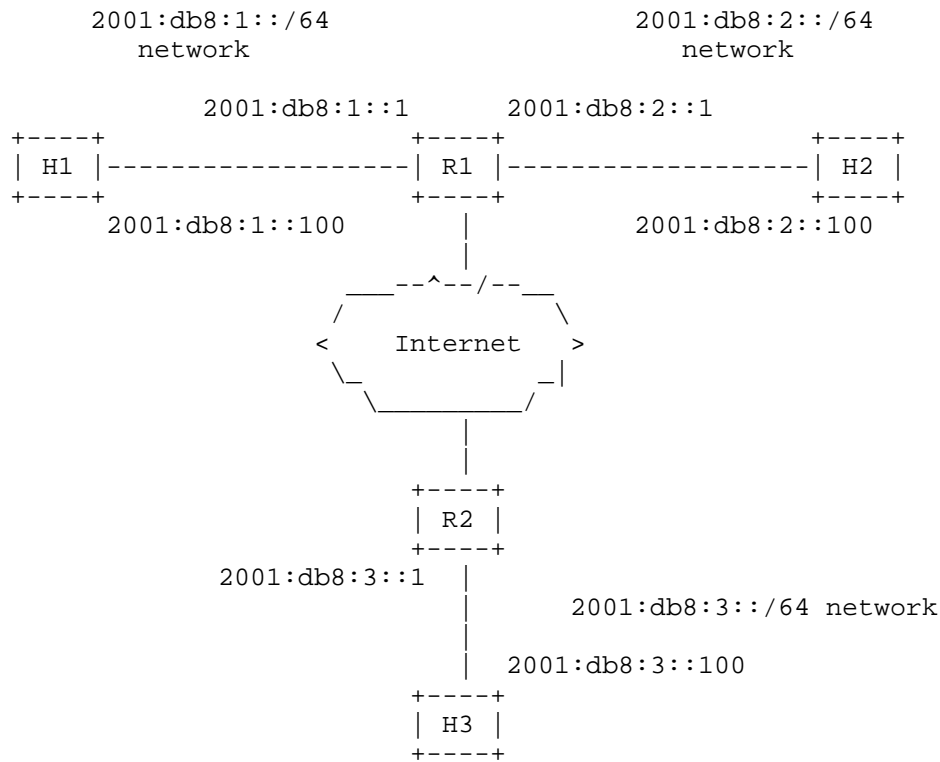


Figure 3: Hypothetical Attack Scenario

In our scenario, the attack packet sent by the attacker would have the same structure as that of Figure 2, with the following values:

- o Source Address: 2001:db8:3::100 (or forged address)
- o Destination Address: 2001:db8:1::100
- o Source Address (embedded packet): 2001:db8:1::100
- o Destination Address (embedded packet): 2001:db8:2::100

The Source Address of the packet is rather irrelevant and need not be forged. The Destination Address of the packet will be that of the attack target (H1 in our case). The Source Address of the embedded packet will be that of the attack target (H1 in our case). Finally, the Destination Address of the embedded packet will be that of the peer with which the attack target is communicating (H2 in our case).

If router R2 were to inspect the payload of the ICMP attack packet, it would conclude that the attack packet cannot be possibly valid, since packets destined to 2001:db8:2::100 would never be forwarded to the network from which the error message is originating. In a similar vein, if R1 were to examine the payload of the aforementioned ICMP error message, it would also conclude that the ICMP error message cannot be possibly valid, for the same reason stated before. Thus, filtering ICMP messages based on the ICMP payload could be employed as a countermeasure for attacks based on ICMP error messages.

5. ICMPv4/ICMPv6 Network Ingress Filtering

A node (e.g. firewall) meaning to enforce the filtering policy specified in this document SHOULD check:

```
IF    embedded packet's Destination Address is from within my network
THEN  forward as appropriate
```

```
IF    embedded packet's Destination Address is anything else
THEN  drop packet
```

NOTE: The destination match is due to a learned route (which assumes some minimal level of path or routing symmetry which firewalls tend to require anyway); or an access list.

We note, however, that the techniques described in [RFC3704] should be evaluated when the aforementioned network ingress filtering is to be implemented in more complex network scenarios, such as that of a multihomed networks. In multihomed scenarios, this filtering policy tends to be undesirable since it is likely to lead to false positives.

Finally, we note that packet drops SHOULD be logged, since this then provides a basis for monitoring any suspicious activity.

6. IANA Considerations

This document has no actions for IANA.

7. Security Considerations

This document provides advice on performing network ingress filtering on ICMPv4 and ICMPv6 error messages, such that attacks based on such messages can be mitigated by means of network packet filtering. Implementation of this filtering technique may depend on the ability of the filtering device to inspect the payload of ICMP messages.

We note that a given platform may or may not be able to filter ICMP error messages based on the ICMP payload. Thus, the aforementioned filter SHOULD only be performed where applicable. Additionally, enforcing the aforementioned filtering method might impact the performance of the filtering device (see e.g., [I-D.gont-v6ops-ipv6-ehs-packet-drops] and [Zack-FW-Benchmark] for a discussion of the IPv6 case). This should be considered before enabling the aforementioned filtering method.

8. Acknowledgements

The authors of this document would like to thank (in alphabetical order) Ron Bonica, Igor Gashinsky, Joel Jaeggli, Merike Kaeo, Jen Linkova, Vic Liu, Carlos Pignataro, and Eric Vyncke, for providing valuable comments on earlier versions of this document.

9. References

9.1. Normative References

- [RFC0791] Postel, J., "Internet Protocol", STD 5, RFC 791, DOI 10.17487/RFC0791, September 1981, <<http://www.rfc-editor.org/info/rfc791>>.
- [RFC0792] Postel, J., "Internet Control Message Protocol", STD 5, RFC 792, DOI 10.17487/RFC0792, September 1981, <<http://www.rfc-editor.org/info/rfc792>>.
- [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>>.
- [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>>.
- [RFC4884] Bonica, R., Gan, D., Tappan, D., and C. Pignataro, "Extended ICMP to Support Multi-Part Messages", RFC 4884, DOI 10.17487/RFC4884, April 2007, <<http://www.rfc-editor.org/info/rfc4884>>.

9.2. Informative References

- [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.
- [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>>.
- [RFC3704] Baker, F. and P. Savola, "Ingress Filtering for Multihomed Networks", BCP 84, RFC 3704, DOI 10.17487/RFC3704, March 2004, <<http://www.rfc-editor.org/info/rfc3704>>.
- [RFC5927] Gont, F., "ICMP Attacks against TCP", RFC 5927, DOI 10.17487/RFC5927, July 2010, <<http://www.rfc-editor.org/info/rfc5927>>.
- [US-CERT] US-CERT, "US-CERT Vulnerability Note VU#222750: TCP/IP Implementations do not adequately validate ICMP error messages", <http://www.kb.cert.org/vuls/id/222750>, 2005.
- [Zack-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>>.

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: <https://www.si6networks.com>

Ray Hunter
Globis Consulting BV
Weegschaalstraat 3
Eindhoven 5632CW
NL

Email: v6ops@globis.net

Jeroen Massar
Massar Networking
Swiss Post Box 101811
Zuercherstrasse 161
Zuerich CH-8010
CH

Email: jeroen@massar.ch
URI: <http://jeroen.massar.ch>

Will(Shucheng) Liu
Huawei Technologies
Bantian, Longgang District
Shenzhen 518129
P.R. China

Email: liushucheng@huawei.com

opsec
Internet-Draft
Intended status: Informational
Expires: September 22, 2016

F. Gont
SI6 Networks / UTN-FRH
M. Ermini
ResMed
W. Liu
Huawei Technologies
March 21, 2016

Requirements for IPv6 Enterprise Firewalls
draft-gont-opsec-ipv6-firewall-reqs-03

Abstract

While there has been some work in the area of firewalls, concrete requirements for IPv6 firewalls have never been specified in the RFC series. The more limited experience with the IPv6 protocols and the more reduced number of firewalls that support IPv6 has made it rather difficult to infer what are reasonable features to expect in an IPv6 firewall. This has typically been a problem for network operators, who typically have to produce a "Request for Proposal" from scratch that describes such features. This document specifies a set of requirements for IPv6 firewalls, in order to establish some common-ground in terms of what features can be expected in them.

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 September 22, 2016.

Copyright Notice

Copyright (c) 2016 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. DISCLAIMER	2
2. Introduction	3
3. Conventions	3
3.1. Requirements Language	3
3.2. Terminology	4
3.3. Numbering Conventions	5
4. General Security Features	5
5. IPv6-Specific Features	7
6. VPN Security Requirements	8
7. Denial of Service (DoS) Protection	9
8. Application Layer Firewall	11
9. Logging, Auditing and Security Operation Centre (SOC) requirements	11
10. Console and Events Visualization requirements	13
11. Reporting requirements	14
12. IANA Considerations	14
13. Security Considerations	14
14. Acknowledgements	14
15. References	14
15.1. Normative References	14
15.2. Informative References	16
Authors' Addresses	18

1. DISCLAIMER

This initial version of the document is based on a typical IPv6 firewall "Request for Proposal" (RFP), and is mostly meant to trigger discussion in the community, and define a direction for the document. Future versions of this document may contain all, more, or a subset of the requirements present in the current version of this document. Additionally, the current version DOES NOT yet properly separate requirements among MUST/REQUIRED, SHOULD/RECOMMENDED, and MAY/OPTIONAL.

Please DO read Section 3 of this document, since it provides important information about the conventions used throughout this document that is mandatory to be able to understand it.

Finally, please note this version is meant to provide requirements, rather than implementation guidelines.

2. Introduction

While the IETF has published a large number of documents discussing IP and IPv6 packet filtering (see e.g. [RFC7126] and some documents on the topic of IP firewalls (see e.g. [RFC2979] and [RFC3511]), concrete requirements for IP firewalls have never been specified in the RFC series. When it comes to IPv4, a number of features have become common over the years, and firewall requirements have somehow become operational wisdom. When it comes to IPv6 [RFC2460], the more limited experience with the protocols, and the reduced variety of IPv6 firewalls has made it rather difficult to specify what are reasonable features to be expected of an IPv6 firewall. This has proven to be a problem for network operators (who have typically had to produce a "Request for Proposal" from scratch), but also for vendors (who lack a well defined set of requirements that can serve as a roadmap for implementation).

This situation has not only made the process of purchasing an IPv6 firewall harder, but at times has also meant that a number of important/basic features have remained unimplemented by major firewall vendors, or that aforementioned features have not behaved as expected.

This document aims to provide a set of requirements for firewall vendors, which are specified as "MUST", "SHOULD", or "MAY". An IPv6 firewall product is said to be "fully-compliant" with this specification provided it implements all requirements marked as "MUST" and "SHOULD". An IPv6 firewall product is said to be "conditionally-compliant" with this specification provided it implements all requirements marked as "MUST", but fails to implement one or more of the requirements marked as "SHOULD".

3. Conventions

3.1. Requirements Language

Take careful note: Unlike other IETF documents, the key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are not used as described in [RFC2119]. This document uses these keywords not

strictly for the purpose of interoperability, but rather for the purpose of establishing industry-common baseline functionality.

In this document, the words that are used to define the significance of each particular requirement are capitalized. These words are:

- o "MUST" This word, or the words "REQUIRED" and "SHALL" mean that the item is an absolute requirement of the specification.
- o "SHOULD" This word or the adjective "RECOMMENDED" means that there may exist valid reasons in particular circumstances to ignore this item, but the full implications should be understood and the case carefully weighed before choosing a different course.
- o "MAY" This word or the adjective "OPTIONAL" means that this item is truly optional. One vendor may choose to include the item because a particular marketplace requires it or because it enhances the product, for example; another vendor may omit the same item.

A firewall implementation is a module that supports at least one of the feature types defined in this document. Firewall implementations may support multiple feature types, but conformance is considered individually for each type.

A firewall implementation is not compliant with a specific feature type if it fails to satisfy one or more of the MUST requirements of such specific feature type. An implementation that satisfies all the MUST and all the SHOULD requirements of a specific feature is said to be "unconditionally compliant" with such feature type; one that satisfies all the MUST requirements but not all the SHOULD requirements is said to be "conditionally compliant" with such feature type.

3.2. Terminology

Where possible, this document employs the terminology defined in [RFC2647]. Other additional terms are defined below:

session:

The term session refers to any protocol instance that involves some sort of stateful exchange. Examples of "sessions" could be TCP connections, UDP query/response pairs, ICMPv6 echo/response pairs, etc. Our definition of session corresponds to the definition of "connection" in Section 3.7 of [RFC2647], but we rather employ "session" to avoid possible confusion.

XXX: Should we just get rid of the term "session" and use "connection" throughout this document, with a big reference to the definition in RFC2647?

3.3. Numbering Conventions

The items for each feature type will follow a monotonically-increasing order -- typically in increments to 10. This is to prevent the insertion of an item in the list of requirements to change the numbering of all the following requirements. Prior to the final publication of this document, each of items of each the feature types will be numbered starting from 1, with increments of 1 (1, 2, 3, 4, etc.).

NOTE: Those with BASIC language programming experience may find the idea familiar.

4. General Security Features

REQ GEN-5:

The firewall MUST include performance benchmarking documentation. Such documentation MUST include information that reflects firewall performance with respect to IPv6 packet, but also regarding how IPv6 traffic may affect the performance of IPv4 traffic. The aforementioned documentation MUST be, at the very least, conditionally-compliant with both [RFC3511] and [RFC5180] (that is, it MUST support all "MUST" requirements in such documents, and may also support the "SHOULD" requirements in such documents).

NOTE: This is for operators to spot be able to identify cases where a devices may under-perform in the presence of IPv6 traffic (see e.g. [FW-Benchmark]). XXX: This note may be removed before publication if deemed appropriate.

REQ GEN-10:

All features of the firewall MUST be able to be individually configured (at least ON or OFF, with other configurable parameters as applicable). A well-documented default initial setting must be provided for each feature.

REQ GEN-20:

MUST support basic Access Control Lists (ACLs).

REQ GEN-30:

MUST support stateless packet inspection and filtering at transport layer.

REQ GEN-40:

MUST support stateful packet inspection and filtering at transport layer.

REQ GEN-50:

SHOULD support full-proxy for the TCP [RFC0793] connections (the handshake is validated on the firewall before reaching the target system).

Some products identify this feature with terms such as "TCP Intercept and Limiting Embryonic Connections".

REQ GEN-60:

MUST be able to enforce timeouts on protocol sessions based on the upper-layer protocol (e.g. enforce a timeout on the FIN-WAIT state for TCP connections, a timeout for DNS query/response pair, etc.). In general, it MUST have different timeout parameters and thresholds to be used to prevent idle sessions from exhausting resources on the device and/or the service that is defended. For sessions composed of multiple packets (such as TCP connections), the exchange of valid packets MUST refresh the timers employed for enforcing the aforementioned timeouts.

NOTE: This is to avoid the known and buggy behavior where firewalls enforce a maximum lifetime for the protocol session (e.g. TCP connection) regardless of whether there is ongoing exchange of legitimate packets for such session.

REQ GEN-70:

MUST be able to provide anti-spoofing features (e.g. uRPF).

REQ GEN-80:

MUST be able to redirect specific traffic to a proxy server e.g. for HTTP/S protocols.

NOTE: "Redirection means that the firewall should be able to divert the traffic to a proxy - i.e., take the traffic, send it to an inspection engine, receive it back and forward it (all this completely transparent to the users).

REQ GEN-90:

MUST be able to detect and reject invalid source or destination addresses (e.g. local-link addresses that try to traverse the firewall) with a single policy.

5. IPv6-Specific Features

REQ SPC-10:

MUST be able to filter ICMPv6 [RFC4443] traffic at a message type/code granularity. [RFC4890] MUST be employed for the default filtering configuration.

REQ SPC-20:

MUST be able to block packets containing any specified extension header type (based on its Next Header value), on a specified number of instances of a specified extension header type, and on a specified overall number of IPv6 extension headers.

REQ SPC-30:

MUST be able to block IPv6 packets that employ a Routing Header both at the granularity of Extension Header Type (as required in SPC-20) and Routing Header Type.

REQ SPC-40:

SHOULD be able to drop packets based on IPv6 option types.

REQ SPC-50:

MUST be able to detect IPv6 tunnels such as SIIT [RFC6145], 6to4 [RFC3056], 6in4 [RFC4213], ISATAP [RFC5214] and Teredo [RFC4380] (please see [RFC7123]), and MUST be able to selectively block or allow them for specific sources, destinations, routes or interfaces.

REQ SPC-60:

MUST be able to validate IPv6 Neighbor Discovery [RFC4861] packets (RS, RA, NS, NA, Redirect) according to [I-D.ietf-opsec-ipv6-nd-security].

REQ SPC-70:

MUST be able to statefully match ICMPv6 errors to TCP [RFC0793], UDP [RFC0768], and ICMPv6 [RFC4443] communication instances (see [RFC5927]).

REQ SPC-80:

MUST be able to parse all defined extension headers according to [RFC7045], and SHOULD filter packets containing IPv6 Extension Headers as recommended in [draft-gont-opsec-ipv6-eh-filtering].

REQ SPC-90:

MUST be able to find the upper-layer protocol in an IPv6 header chain (see [RFC7112]).

REQ SPC-100:

SHOULD be able to normalize (rewrite) the following IPv6 header fields on a per-interface basis:

- * Hop Limit

6. VPN Security Requirements

REQ VPN-10:

MUST implement IPsec-based [RFC4301] VPN technology.

REQ VPN-20:

MUST implement "hub-and-spoke" Dynamic Multipoint VPN-like technology, allowing creation of dynamic-meshed VPN without having to pre-configure all of possible tunnels.

REQ VPN-30:

MUST implement SSL/TLS-based [SSL-VPNs] VPN technology.

REQ VPN-40:

MUST be able to use digital certificates, including CRL and OCSP revocation checking methods, to mutually authenticate VPN peers.

REQ VPN-50:

MUST be able to disable or enable split-tunnelling feature on VPN as required.

REQ VPN-60:

MUST support the enrollment of the system in a PKI infrastructure for the regular renewal of certificates.

REQ VPN-70:

MUST be able to transit IPv4 and IPv6 packets providing full parity for services, and also offer both protocols in dual-stack in the same VPN connection.

REQ VPN-80:

MUST be able to apply to the tunnelled content that is terminated on the device, the same inspection policies that are possible in the non tunnelled traffic.

REQ VPN-90:

MUST perform a full validation of the certificates' chains when verifying the validity of the OCSP/CLR responses. Caching of responses SHOULD be configurable by end users, and the default response SHOULD be not to accept a non-valid certificate. The default response MAY be overridden by the administrators, but it MUST be configurable on a per-domain basis (e.g. accept incomplete

certificate chains for "intranet_of_internal_corp.example.org", but refuse it for all of the other domains).

7. Denial of Service (DoS) Protection

REQ DoS-10:

MUST be able to protect against implementation-specific attacks, including:

- * Winnuke [Myst1997]
- * ping-of-death [Kenney1996]
- * Smurf [CERT1998a]
- * LAND Attack [Meltman1997]
- * Teardrop Attack [CERT1997] [Junos-Teardrop]

REQ DoS-20:

MUST be able to protect against IPv6 resource exhaustion attacks, including:

- * fragment flooding attacks
- * Neighbor Cache Exhaustion attacks, whether launched from a local network (see [I-D.ietf-opsec-ipv6-nd-security] or from remote networks (see [RFC6583])

REQ DoS-30:

MUST be able to protect against TCP flooding attacks: connection-flooding, FIN-WAIT-1 flooding, etc. (see e.g. [CPNI-TCP])

REQ DoS-40:

MUST be able to protect against TCP resource exhaustion attacks: zero-window attacks, SYN-floods, etc. (see e.g. [CPNI-TCP])

REQ DoS-50:

MUST be able to detect and drop malformed IPv6 packets (incorrect header/option lengths, etc.).

REQ DoS-60:

MUST be able to detect and drop malformed TCP packets (incorrect segment/options lengths, etc.).

REQ DoS-70:

MUST be able to provide bandwidth management (QoS or anti-flooding) policies customizable for specific source and destination networks, or by VLAN or MPLS ID.

REQ DoS-80:

MUST be able to participate to a blackhole/synkhole routing infrastructure as per [RFC5635].

REQ DoS-85:

MUST be able to fetch and use third party "reputational" IP white- and black-lists (e.g. download them via RSS feeds or query via them DNS record) and use them in policy constructs/ACLs. In general, it MUST be able to provide some form of reputational service for IP addresses which must include IPv6 networks.

REQ DoS-90:

MUST be able to set up a maximum session setup rate, and detect hosts or networks exceeding it.

REQ DoS-100:

MUST be able to set up a maximum IPv6 source and/or destination session limit, and detect when they are exceeded.

REQ DoS-110:

For each of the previous detection controls, different configurable reactions SHOULD be possible by IPv6 address and network sources and/or destinations. The minimum actions required are the following:

1. allow the traffic ("ignore" or "whitelist")
2. allow the traffic but log ("bypass" or "detection only" mode)
3. drop the packet (only the offending packet but do not reset the connection)
4. drop session (drop the entire connection, but do not send a reset back)
5. "greylist" - put it in a list of blocked addresses, but remove it from the list after a configurable amount of time
6. send an email/SMS/pager text to the firewall administrator
7. send TCP reset to source only
8. send TCP reset to destination only

9. send TCP reset to both source and destination
 10. perform a specific, preconfigured change on the firewall policy
 11. feed a third party source such as a switch/NAC/NAP or RADIUS system, to isolate/quarantine the offending port/MAC address/user
 12. quarantine the specific traffic or source (block them for a configurable amount of time, e.g. 5 minutes, and then allow them again; eventually, the quarantine time may get longer if the offense is repeated)
8. Application Layer Firewall

REQ APP-10:

MUST be able to provide web filtering features, such as enforcing access to allowed web content and filtering high risk URLs such as anonymizers and known hostile addresses.

REQ APP-20:

MUST be able to provide email filtering features, such as mitigating spam, phishing and email harvesting, and enforce email policies.

9. Logging, Auditing and Security Operation Centre (SOC) requirements

REQ SOC-10:

MUST generate log for all the changes performed to the system, including change of group membership for a device, new or removed devices in a group, new or removed administrators.

REQ SOC-20:

MUST provide the following features:

1. Connection logs
2. Local log storage
3. Network logging
4. Real time log viewer
5. Attack detected
6. Per rule logging

7. Automatic log file compression
8. Log file rotation

REQ SOC-30:

MUST be able to generate a log for:

1. all the logins, logouts and failed login attempts from firewall administrators
2. any modifications or disabling of the firewall rules

REQ SOC-40:

Any security event detected - malicious traffic, hit of a policy, policy violation, termination of a session and so on - MUST be able to generate a log, and be configurable to do that or not by administrators.

REQ SOC-50:

There MUST be a mechanism to prevent log flooding from the device against the management system, such as aggregation of like events.

REQ SOC-60:

The amount of information in the alerts MUST be configurable; it SHOULD be possible to have the date/time and type of event and the full payload of the traffic that has triggered the signature/event.

REQ SOC-70:

The firewall MUST minimize the number of log entries generated for a single event - e.g. when repeated similar events for a short period of time are detected, they are aggregated and the cumulative number of events is reported.

REQ SOC-80:

The firewall MUST be able to send logs in multiple ways and formats, for instance UDP syslog, TCP syslog, SMTP, SNMP and so on. It must be possible to configure different ways and formats for different policies and configure some ways and formats as a "backup" in the case that the main way fails. Please describe the different possibilities.

REQ SOC-90:

The firewall SHOULD alert the firewall administrator when the policy to be enforced does not follow the advice in [RFC4890] -- particularly if the filtering policy would block/drop ICMPv6 Packet Too Big error messages.

10. Console and Events Visualization requirements

REQ CON-10:

MUST provide a dashboard view, which must be customizable by end-user and end-users' group (e.g. their Microsoft Active Directory or LDAP group).

REQ CON-20:

The dashboard must be able to include system health monitoring information, such as the following:

1. CPU idle
2. Real and Swap memory usage
3. Disk usage
4. Number of accepted and dropped packets
5. Operating status for all supported facilities (HA, QoS, VPN)
6. VPN tunnels status
7. NIC link state

REQ CON-30:

MUST have the possibility to select a particular piece of data or individual alert, and visualize the policy that has triggered the event.

REQ CON-40:

MUST be able to create exception filters that will suppress visualization of a specific alert (e.g. from specific sources, or specific events), without actually affecting the detection and log retention.

REQ CON-50:

MUST provide a remote access method to obtain all current operational data on demand, in a documented format, covering items such as those listed in REQ CON-20.

Note: This is to be able to integrate firewall operations in an existing NMS.

11. Reporting requirements

REQ REP-10:

Built in reports MUST be provided by default, such as protocol distribution, policy and rule matched, top attacks, top sources/destinations, top targets, top geographical sources, device status including utilizations, and so on.

REQ REP-20:

SHOULD allow to run reporting over historical and archived logs, automatically restoring and re-archiving them.

12. IANA Considerations

There are no IANA registries within this document. The RFC-Editor can remove this section before publication of this document as an RFC.

13. Security Considerations

[TBD]

14. Acknowledgements

The initial version was based on an (unpublished) document authored by Marco Ermini.

The authors would like to thank (in alphabetical order), Mikael Abrahamsson, Cameron Byrne, Brian Carpenter, Tim Chown, Jakub (Jake) Czyz, Marc Heuse, Simon Perreault, Carsten Schmoll, Robert Sleigh, Donald Smith, Qiong Sun, Gunter Van de Velde, and Scott Weeks, for providing valuable comments on earlier versions of this document.

15. References

15.1. 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>>.

- [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>>.
- [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>>.
- [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>>.
- [RFC0768] Postel, J., "User Datagram Protocol", STD 6, RFC 768, DOI 10.17487/RFC0768, August 1980, <<http://www.rfc-editor.org/info/rfc768>>.
- [RFC0793] Postel, J., "Transmission Control Protocol", STD 7, RFC 793, DOI 10.17487/RFC0793, September 1981, <<http://www.rfc-editor.org/info/rfc793>>.
- [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>>.
- [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>>.
- [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>>.
- [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>>.

- [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>>.

15.2. Informative References

- [RFC2647] Newman, D., "Benchmarking Terminology for Firewall Performance", RFC 2647, DOI 10.17487/RFC2647, August 1999, <<http://www.rfc-editor.org/info/rfc2647>>.
- [RFC4213] Nordmark, E. and R. Gilligan, "Basic Transition Mechanisms for IPv6 Hosts and Routers", RFC 4213, DOI 10.17487/RFC4213, October 2005, <<http://www.rfc-editor.org/info/rfc4213>>.
- [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>>.
- [RFC5180] Popoviciu, C., Hamza, A., Van de Velde, G., and D. Dugatkin, "IPv6 Benchmarking Methodology for Network Interconnect Devices", RFC 5180, DOI 10.17487/RFC5180, May 2008, <<http://www.rfc-editor.org/info/rfc5180>>.
- [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>>.
- [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>>.
- [RFC7123] Gont, F. and W. Liu, "Security Implications of IPv6 on IPv4 Networks", RFC 7123, DOI 10.17487/RFC7123, February 2014, <<http://www.rfc-editor.org/info/rfc7123>>.
- [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>>.
- [RFC2979] Freed, N., "Behavior of and Requirements for Internet Firewalls", RFC 2979, DOI 10.17487/RFC2979, October 2000, <<http://www.rfc-editor.org/info/rfc2979>>.

- [RFC3511] Hickman, B., Newman, D., Tadjudin, S., and T. Martin, "Benchmarking Methodology for Firewall Performance", RFC 3511, DOI 10.17487/RFC3511, April 2003, <<http://www.rfc-editor.org/info/rfc3511>>.
- [RFC5927] Gont, F., "ICMP Attacks against TCP", RFC 5927, DOI 10.17487/RFC5927, July 2010, <<http://www.rfc-editor.org/info/rfc5927>>.
- [I-D.ietf-opsec-ipv6-nd-security]
Gont, F., Bonica, R., and W. Will, "Security Assessment of Neighbor Discovery (ND) for IPv6", draft-ietf-opsec-ipv6-nd-security-00 (work in progress), October 2013.
- [RFC6274] Gont, F., "Security Assessment of the Internet Protocol Version 4", RFC 6274, DOI 10.17487/RFC6274, July 2011, <<http://www.rfc-editor.org/info/rfc6274>>.
- [CPNI-TCP]
CPNI, , "Security Assessment of the Transmission Control Protocol (TCP)", <http://www.gont.com.ar/papers/tn-03-09-security-assessment-TCP.pdf>, 2009.
- [SSL-VPNs]
Hoffman, P., "SSL VPNs: An IETF Perspective", IETF 72, SAAG Meeting., 2008, <<http://www.ietf.org/proceedings/72/slides/saag-4.pdf>>.
- [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>>.
- [Junos-Teardrop]
Juniper, j., "Understanding Teardrop Attacks", Junos OS Security Configuration Guide, 2010, <<http://www.juniper.net/techpubs/software/junos-es/junos-es93/junos-es-swconfig-security/understanding-teardrop-attacks.html>>.
- [draft-gont-opsec-ipv6-eh-filtering]
Gont, F., Ermini, M., and W. Liu, "Recommendations on Filtering of IPv6 Packets Containing IPv6 Extension Headers", draft-gont-opsec-ipv6-filtering-00, Work in Progress, April 2014.

[Kenney1996]

Kenney, M., "The Ping of Death Page", 1996,
<<http://www.insecure.org/splloits/ping-o-death.html>>.

[Meltman1997]

Meltman, M., "new TCP/IP bug in win95", 1997,
<<http://insecure.org/splloits/land.ip.DOS.html>>.

[Myst1997]

Myst, M., "Windows 95/NT DoS", 1997,
<<http://insecure.org/splloits/land.ip.DOS.html>>.

[CERT1997]

CERT, , "CERT Advisory CA-1997-28: IP Denial-of-Service
Attacks", 1997,
<<http://www.cert.org/advisories/CA-1997-28.html>>.

[CERT1998a]

CERT, , "CERT Advisory CA-1998-01: Smurf IP Denial-of-
Service Attacks", 1998,
<<http://www.cert.org/advisories/CA-1998-01.html>>.

Authors' Addresses

Fernando Gont
SI6 Networks / UTN-FRH
Evaristo Carriego 2644
Haedo, Provincia de Buenos Aires 1706
Argentina

Phone: +54 11 4650 8472
Email: fgont@si6networks.com
URI: <http://www.si6networks.com>

Marco Ermini
ResMed
Fraunhoferstrasse 16
Munich, Bayern 82152
Deutschland

Phone: +49 175 4395642
Email: marco.ermini@resmed.com
URI: <http://www.resmed.com>

Will Liu
Huawei Technologies
Bantian, Longgang District
Shenzhen 518129
P.R. China

Email: liushucheng@huawei.com

OPSEC
Internet-Draft
Intended status: Informational
Expires: September 12, 2019

E. Vyncke, Ed.
Cisco
K. Chittimaneni
WeWork
M. Kaeo
Double Shot Security
E. Rey
ERNW
March 11, 2019

Operational Security Considerations for IPv6 Networks
draft-ietf-opsec-v6-16

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 several places of a network (enterprises, service providers and residential users) and proposes technical and procedural mitigations techniques. Some very specific place of a network such as Internet of Things are not discussed in 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 September 12, 2019.

Copyright Notice

Copyright (c) 2019 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.	Requirements Language	4
2.	Generic Security Considerations	4
2.1.	Addressing Architecture	4
2.1.1.	Statically Configured Addresses	5
2.1.2.	Use of ULAs	5
2.1.3.	Point-to-Point Links	5
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
2.1.7.	Using a /64 per host	7
2.2.	Extension Headers	7
2.2.1.	Order and Repetition of Extension Headers	8
2.2.2.	Hop-by-Hop Options Header	8
2.2.3.	Fragment Header	8
2.2.4.	IP Security Extension Header	9
2.3.	Link-Layer Security	9
2.3.1.	ND/RA Rate Limiting	9
2.3.2.	RA/NA Filtering	10
2.3.3.	Securing DHCP	11
2.3.4.	3GPP Link-Layer Security	12
2.3.5.	SeND and CGA	12
2.4.	Control Plane Security	13
2.4.1.	Control Protocols	14
2.4.2.	Management Protocols	15
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	18

2.6.1.	Data Sources	19
2.6.2.	Use of Collected Data	23
2.6.3.	Summary	25
2.7.	Transition/Coexistence Technologies	25
2.7.1.	Dual Stack	26
2.7.2.	Transition Mechanisms	26
2.7.3.	Translation Mechanisms	30
2.8.	General Device Hardening	32
3.	Enterprises Specific Security Considerations	32
3.1.	External Security Considerations:	33
3.2.	Internal Security Considerations:	33
4.	Service Providers Security Considerations	34
4.1.	BGP	34
4.1.1.	Remote Triggered Black Hole Filtering	34
4.2.	Transition Mechanism	34
4.3.	Lawful Intercept	34
5.	Residential Users Security Considerations	35
6.	Further Reading	36
7.	Acknowledgements	36
8.	IANA Considerations	36
9.	Security Considerations	36
10.	References	37
10.1.	Normative References	37
10.2.	Informative References	37
	Authors' Addresses	49

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", "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. 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 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 due to malicious criminal activity. Using PA space exposes the organization to a renumbering of the complete network including security policies (based on ACL), audit system, ... in short a complex task which could lead to some temporary security risk if done for a large network and without automation; hence, for large networks, PI space should be preferred even if it comes with additional complexities (for example BGP routing) and duties (adding a route6 object in the Regional Internet Registry database).

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 to limit 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).

2.1.1.1. 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 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 realizable than expected; see also [RFC7707]. The use of well-known (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 and more to come possible, hence, operators should never rely on the 'impossible to find because my address is random' paradigm.

While in some environments obfuscating addresses could be considered an added benefit; it does not preclude that perimeter rules are actively enforced and that statically configured addresses follow some logical allocation scheme for ease of operation (as simplicity always helps security). Typical deployments will have a mix of static and non static addresses.

2.1.1.2. Use of ULAs

Unique Local Addresses (ULAs) [RFC4193] are intended for scenarios where systems are not globally reachable, despite formally having global scope. ULA looks similar to [RFC1918] addresses but have different use cases. One use of ULA is described in [RFC4864] and some considerations on using ULA is described in the draft document [I-D.ietf-v6ops-ula-usage-considerations].

2.1.1.3. Point-to-Point Links

[RFC6164] in section 5.1 documents the reasons why to use a /127 for inter-router point-to-point links; notably, a /127 prevents the ping-pong attack between routers not implementing correctly [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 against infrastructure devices, the operational disadvantages need also to be carefully considered; see also [RFC7404].

2.1.4. Temporary Addresses - Privacy Extensions for SLAAC

Historically stateless address autoconfiguration (SLAAC) relied on an automatically generated 64 bit interface identifier (IID) based on the EUI-64 MAC address, which together with the /64 prefix makes up the globally unique IPv6 address. The EUI-64 address is generated from the 48-bit stable MAC address. [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 [RFC4941] or [RFC8064].

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 somehow reduce the attack exposure window.

Using [RFC4941] privacy extension addresses prevents the operator from building host specific access control lists (ACLs).

[RFC8064] specifies another way to generate 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 address on different networks.

As [RFC4941] privacy extension addresses could also be used to obfuscate some malevolent activities (whether on purpose or not), specific user attribution/accountability procedures should be put in place as described in section Section 2.6.

In some extreme use cases where user accountability is more important than user privacy, network operators may consider to disable SLAAC and rely only on DHCPv6; but, not all operating systems support DHCPv6 so some hosts will not get any IPv6 connectivity. Disabling SLAAC and privacy extensions addresses can be done for most OS and for non-hacker users by sending RA messages 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. Hackers will find a way 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 [RFC8064] is available, the stable privacy address is probably a good balance between privacy (among multiple networks) and security/user attribution (within a network).

2.1.5. Privacy consideration of Addresses

The reader can learn more about privacy considerations for IPv6 addresses 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 2.3.3).

While there are no fundamental differences with IPv4 and IPv6 security concerns about DNS, there are specific consideration in DNS64 [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.

2.1.7. Using a /64 per host

An interesting approach is using a /64 per host as proposed in [RFC8273]. This allows an easier user attribution (typically based on the host MAC address) as its /64 prefix is stable even if applications, containers within the host can change of IPv6 address within this /64.

2.2. Extension Headers

The extension headers are an important difference between IPv4 and IPv6. The packet structure does make a big difference. For instance, it's trivial to find (in IPv4-based packets) the upper layer protocol type and protocol header, while in IPv6 it actually isn't as the extension header chain must be parsed completely. The 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].

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 [RFC7872]. Understanding the role of varying extension headers is important and this section enumerates the ones that need careful consideration.

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]. The uniform TLV format

to be used for defining future extension headers is described in [RFC6564].

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.

There is work in progress at the IETF about 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 this document 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 lead to nodes crashing when receiving or forwarding wrongly formatted packets.

A firewall or any edge device should be used to enforce the recommended order and number of occurrences of extension headers.

2.2.2. Hop-by-Hop Options Header

The hop-by-hop options header, when present in an IPv6 packet, forces all nodes in the path to inspect this header in the original IPv6 specification [RFC2460]. This was 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. Section 4.3 of the current Internet Standard for IPv6, [RFC8200], is more sensible to this respect as the processing of hop-by-hop options header by intermediate routers is optional.

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 to:

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).

Else, stateless filtering could be bypassed by an hostile party. [RFC6980] applies a stricter rule to NDP by enforcing the drop of fragmented NDP packets. [RFC7113] describes how RA-guard function described in [RFC6105] should behave in presence of fragmented RA packets.

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] 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].

2.3.1. 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] 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).
- o Using /127 on point-to-point link per [RFC6164].

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]

2.3.2. RA/NA 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] 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. 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] 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] 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 Neighbor Discovery attacks.

The Source Address Validation Improvements (SAVI) working group has worked on other ways to mitigate the effects of such attacks. [RFC7513] would help in creating bindings between a DHCPv4 [RFC2131] /DHCPv6 [RFC3315] 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 address.

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 line of defense is fully effective when weird fragments are dropped by routers and switches as described in Section 2.2.3. The generated log should also be analyzed to act on violations.

2.3.3. Securing DHCP

Dynamic Host Configuration Protocol for IPv6 (DHCPv6), as originally detailed in [RFC3315] now obsoleted by [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 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 unintentional, a misconfigured DHCP server can have the same impact. Additional threats against DHCP are discussed in the security considerations section of [RFC8415].

[RFC7610], DHCPv6-Shield, 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. Of course, extension headers could be leveraged to bypass DHCPv6-Shield unless [RFC7112] is enforced. Another way to secure DHCPv6 would be to use the secure DHCPv6 protocol which is currently work in progress per [I-D.ietf-dhc-sedhcpv6] , but, with no real deployment known by the authors of this document.

It is recommended to use DHCP-shield and to analyze the log generated by this security feature.

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 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] for a more detailed discussion on the 3GPP link model, NDP on it and the address configuration detail.

2.3.5. 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 after many years after their specifications, CGA and SeND do not have wide support from generic operating systems; hence, their usefulness is limited and should not be relied upon.

2.4. Control Plane Security

[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 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 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 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 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] and more generally [RFC7112] highlights 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 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 may cause Path MTU holes.

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 the 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] or [RFC2740] that was obsoleted by the former. 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 wordings. OSPFv3 can also use normal ESP to encrypt the OSPFv3 payload to hide the routing information.

[RFC7166] (which obsoletes [RFC6506] changes OSPFv3's 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 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 [RFC8504] (that obsoletes [RFC6434]) is a 'SHOULD' and no more a '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.

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 (see [RFC8190])

- 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]

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 operators 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 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 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) 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 [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 including the reverse DNS mapping into a FQDN (which should not be trusted).

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 using NETCONF [RFC6241] to collect the 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 or any other monitoring mechanism.

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.

Using the approach of one /64 per host (Section 2.1.7) replaces the neighbor cache dumps by a mere caching of the allocated /64 prefix when combined with strict enforcement rule on the router and switches to prevent IPv6 spoofing.

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. 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 in the IPv4 era. DHCPv6 servers that keep the relayed data-link layer address in addition to the DUID in the lease file do not suffer from this limitation.

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. 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 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 is(are) used to reach this prefix (assuming that anti-spoofing mechanisms are used);
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 SAVI events,
3. based on the incident time and on the IPv6 address, retrieve the data-link address from the DHCP lease file (Section 2.6.1.5);
4. 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. If a Configuration Management Data Base (CMDB) is used, the mapping between the data-link layer address and a switch port.

At the end of the process, the interface the host originating malicious activity or the username which was abused for malicious activity has been determined.

2.6.2.2. Inventory

RFC 7707 [RFC7707] (which obsoletes RFC 5157 [RFC5157]) is about the difficulties for an attacker to scan an IPv6 network due to the vast number of IPv6 addresses per link (and why in some case it can still be done). While the huge addressing space can sometime be perceived as a 'protection', it also make 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 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.

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.

Moreover, [RFC7934] recommends to use 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 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

As it is expected that some networks will not run in a pure IPv6-only way, 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.

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 traffics are native (easier to observe and secure) and should have the same network processing (path, quality of service, ...). Dual stack allows you to gradually turn IPv4 operations down when your IPv6 network is ready for prime time. On the other hand, the operators have to manage two networks with the added complexities.

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

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 over IPv6 link-local address or over a global IPv6 address if rogue RA or rogue DHCPv6 addresses are provided by an attacker.

2.7.2. Transition Mechanisms

There are many tunnels used for specific use cases. Except when protected by IPsec [RFC4301], all those tunnels have a couple of security issues (most of them because they are tunnels and are 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; combined with the absence of authentication, a man in the middle 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 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 an malevolent IPv6 traffic contained in the tunnel.

To mitigate the bypassing of security policies, it is recommended 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 2.7.2.7), 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 protocol 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.

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].

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 than 6to4 tunnels and have little security issues except lack of confidentiality. The security considerations (Section 12) of [RFC5969] describes how to secure the 6rd tunnels.

IPsec [RFC4301] for the transported IPv6 traffic can be used if confidentiality is important.

2.7.2.4. 6PE and 6VPE

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 VPN described in [RFC4364], the security of these networks is also similar to the one described in [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.

2.7.2.5. DS-Lite

DS-lite is more a translation mechanism and is therefore analyzed further (Section 2.7.3.3) in this document.

2.7.2.6. Mapping of Address and Port

With the encapsulation and translation 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 manage.

2.7.2.7. 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] which has been deprecated by [RFC7526], and is no more 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 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).

Teredo is now mostly never used and it is no more automated in most environment, so, it is less of a threat.

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. 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 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 [RFC7915] (that obsoletes [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-IP4 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::

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] (that 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: 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, Mustafa Suha Botsali, Brian Carpenter, Tim Chown, Lorenzo Colitti, Markus deBruen, Tobias Fiebig, Fernando Gont, Jeffry Handal, Lee Howard, Panos Kampanakis, Erik Kline, Jouni Korhonen, Mark Lentczner, Bob Sleight, Tarko Tikan, Ole Troan, Bernie Volz (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, <<https://www.rfc-editor.org/info/rfc2119>>.
- [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>>.

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>>.
- [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>>.
- [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-dhc-sedhcpv6] Li, L., Jiang, S., Cui, Y., Jinmei, T., Lemon, T., and D. Zhang, "Secure DHCPv6", draft-ietf-dhc-sedhcpv6-21 (work in progress), February 2017.

- [I-D.ietf-opsec-ipv6-eh-filtering]
Gont, F. and W. LIU, "Recommendations on the Filtering of IPv6 Packets Containing IPv6 Extension Headers", draft-ietf-opsec-ipv6-eh-filtering-06 (work in progress), July 2018.
- [I-D.ietf-v6ops-ula-usage-considerations]
Liu, B. and S. Jiang, "Considerations For Using Unique Local Addresses", draft-ietf-v6ops-ula-usage-considerations-02 (work in progress), March 2017.
- [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, S. and E. Vyncke, "IPv6 Security", ISBN 1-58705-594-5, Publisher CiscoPress, December 2008.
- [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. Rooks, "Guidelines for the Secure Deployment of IPv6", 2010,
<<http://csrc.nist.gov/publications/nistpubs/800-119/sp800-119.pdf>>.
- [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>>.
- [RFC2740] Coltun, R., Ferguson, D., and J. Moy, "OSPF for IPv6", RFC 2740, DOI 10.17487/RFC2740, December 1999, <<https://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, <<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>>.
- [RFC2993] Hain, T., "Architectural Implications of NAT", RFC 2993, DOI 10.17487/RFC2993, November 2000, <<https://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, <<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>>.

- [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>>.
- [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>>.
- [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>>.
- [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>>.
- [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>>.
- [RFC4941] Narten, T., Draves, R., and S. Krishnan, "Privacy Extensions for Stateless Address Autoconfiguration in IPv6", RFC 4941, DOI 10.17487/RFC4941, September 2007, <<https://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, <<https://www.rfc-editor.org/info/rfc4942>>.
- [RFC5157] Chown, T., "IPv6 Implications for Network Scanning", RFC 5157, DOI 10.17487/RFC5157, March 2008, <<https://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, <<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>>.
- [RFC6145] Li, X., Bao, C., and F. Baker, "IP/ICMP Translation Algorithm", RFC 6145, DOI 10.17487/RFC6145, April 2011, <<https://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, <<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>>.
- [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>>.

- [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, <<https://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, <<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., Soinen, 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>>.
- [RFC6506] Bhatia, M., Manral, V., and A. Lindem, "Supporting Authentication Trailer for OSPFv3", RFC 6506, DOI 10.17487/RFC6506, February 2012, <<https://www.rfc-editor.org/info/rfc6506>>.
- [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>>.
- [RFC6810] Bush, R. and R. Austein, "The Resource Public Key Infrastructure (RPKI) to Router Protocol", RFC 6810, DOI 10.17487/RFC6810, January 2013, <<https://www.rfc-editor.org/info/rfc6810>>.
- [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>>.
- [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>>.
- [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>>.
- [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>>.
- [RFC7381] Chittimani, 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>>.
- [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>>.
- [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>>.
- [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>>.
- [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>>.
- [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>>.
- [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>>.
- [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>>.
- [SCANNING] "Mapping the Great Void - Smarter scanning for IPv6", <http://www.caida.org/workshops/isma/1202/slides/aims1202_rbarnes.pdf>.

Authors' Addresses

Eric Vyncke (editor)
Cisco
De Kleetlaan 6a
Diegem 1831
Belgium

Phone: +32 2 778 4677
Email: evyncke@cisco.com

Kiran K. Chittimaneni
WeWork

Email: kk.chittimaneni@gmail.com

Merike Kaeo
Double Shot Security
3518 Fremont Ave N 363
Seattle 98103
USA

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

IP Multicast
Internet-Draft
Intended status: Informational
Expires: June 26, 2016

E. Vyncke
Cisco
E. Rey
ERNW
A. Atlasis
NCI Agency
December 24, 2015

MLD Security
draft-vyncke-pim-mld-security-01

Abstract

The latest version of Multicast Listener Discovery protocol is defined in RFC 3810, dated back in 2004, while the first version of MLD, which is still in use and has not been deprecated, is defined in RFC 2710 and is dated back in 1999. New security research has exhibited new vulnerabilities in MLD, both remote and local attack vectors. This document describes those vulnerabilities and proposes specific mitigation 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 June 26, 2016.

Copyright Notice

Copyright (c) 2015 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. Local Vulnerabilities	3
2.1. Downgrading to MLDv1	3
2.2. Queries sent to unicast address	4
2.3. Win the election	4
2.4. Host enumeration and OS fingerprinting	4
2.5. Flooding of MLD messages	4
2.6. Amplification	4
3. Remote Vulnerabilities	5
4. Mitigations	5
5. IANA Considerations	5
6. Security Considerations	6
7. Acknowledgements	6
8. References	6
8.1. Normative References	6
8.2. Informative References	6
Authors' Addresses	7

1. Introduction

The Multicast Listener Discovery protocol version 2 (MLDv2) RFC3810 [RFC3810] has a security section but it was not exhaustive and the focus was only on local forged MLD packets. The same is also true for the first version of MLD (now called MLDv1), which is still in use, defined in RFC 2710. This document goes beyond those attacks.

For the reader who is not familiar with MLDv2, here are the main points:

Multicast routers send MLD queries which are either generic (query about all multicast group) sent to ff02::1 (link-scope all nodes) or specific (query about a specific group) sent to this multicast group. Query messages can also be sent to a unicast address.

Multicast members reply to MLDv2 queries with reports sent to ff02::16 (link-scope all MLDv2 routers). In version 1 of MLD RFC2710 [RFC2710], the reports are sent to the multicast group being reported. Reports can be transmitted twice or more in order to ensure that the MLD router gets at least one report.

When a node ceases to listen to a multicast address on an interface, it sends an MLDv1 Done message or a specially crafted MLDv2 Report message.

All MLD packets are ICMPv6 RFC4443 [RFC4443] messages sent with a hop-limit of 1, from a link-local address and there is no authentication.

MLD messages received with a hop-limit greater than 1 should be discarded.

Neighbor Discovery Protocol RFC4861 [RFC4861] requires nodes to become member of the respective solicited-node multicast groups for all their link-scope and global-scope addresses.

Switches are assumed to implement MLD snooping RFC4541 [RFC4541] to learn where to forward multicast packets. It must be noted though that implementations of MLD snooping do not act on link-local multicast groups such as solicited-node multicast group: they simply forward all packets destined to a link-local multicast group to all port in the same layer-2 network.

MLDv2 was designed to be interoperable with MLDv1.

The main difference between MLDv1 and MLDv2 from a functionality perspective is that MLDv1 does not support "source filtering" (in MLDv2 nodes can report interest in traffic only from a set of source addresses or from all except a set source addresses).

Every IPv6 node must support MLD.

This document is heavily based on previous research: [Troopers2015].

2. Local Vulnerabilities

2.1. Downgrading to MLDv1

A single MLDv1 report message is enough to downgrade all MLD nodes (hosts and routers) to the version 1 protocol. This could be used to force a MLD host to reply with MLDv1 reports sent to the multicast group rather than to ff02::16. This downgrade to MLDv1 could also be used to transmit the MLDv1 report with a 'done' operation to remove a listener (stopping the multicast traffic on the subnet). Another consequence of downgrading to MLDv1 can be the fact that an attacker can also used "Host Suppression" feature as part of a DoS attack, make the launch of such an attack easier.

2.2. Queries sent to unicast address

Section 5.1.15 of RFC3810 [RFC3810], specifies that for debugging purposes, nodes must accept and process queries sent to any of their addresses (including unicast). Lab testing, described in [Troopers2015], clearly shows that all implementations except FreeBSD accept and process MLD queries sent to a unicast global address. This can be exploited to completely bypass the legitimate MLD router and interact directly (for whatever purpose) with the targets (including legitimate routers and clients).

2.3. Win the election

When there are multiple MLD routers in a layer-2 domain, the one with the lowest IPv6 address wins the election and becomes the designated MLD router. A hostile node can then send from a lower link-local address an MLD message and become the MLD router. This fact in combination with the direct interaction with the targets could be leveraged to mount a denial of service attack.

2.4. Host enumeration and OS fingerprinting

Some hosts try to prevent host enumeration by not responding to ICMPv6 echo request messages sent to any multicast group. But, the same hosts must reply to any MLD queries including the generic one sent to ff02::1, this allows for MLD host enumeration. As hosts join different groups based on their operating system (specific groups for Microsoft Windows for example), the MLD report can also help for Operating System (OS) fingerprinting.

2.5. Flooding of MLD messages

If an implementation does not rate limit in hardware the rate of processed MLD messages, then they are vulnerable to a CPU exhaustion denial of services. If a node does not limit the number of states associated to MLD, then this node is vulnerable to a memory exhaustion denial of services.

2.6. Amplification

Nodes usually join multiple groups (for example, Microsoft Windows 8.1 joins 4 groups). Therefore a forged generic MLDv1 query will force those nodes to transmit MLDv1 reports for each of their groups (in our example 4); furthermore, many implementations send MLD reports twice (in our example 8 in total). MLDv2 is a little better because reports are sent to ff02::16 and not to the multicast group.

3. Remote Vulnerabilities

MLD messages with hop-limit different than 1 should be discarded but nothing prevents a hostile party located n hops away from the victim to send any MLD messages with a hop-limit set to $n+1$. Therefore, a remote hostile party can mount attacks against MLD (especially because implementations process MLD queries sent to a global unicast address).

4. Mitigations

This section proposes some mitigation techniques that could be used to prevent the above attacks. This section is not a specification of any kind, the words 'should' is plain English and is not related to RFC2119 [RFC2119].

Mitigation by specific implementations:

Similar to RA-guard RFC6105 [RFC6105], there should be a MLD-guard function in layer-2 switches; MLD queries (either version 1 or version 2) received on ports attached to non multicast routers should be discarded. Switches could also block all MLDv1 packets in order to prevent the downgrading of MLD version. Of course, this requires all nodes to support MLDv2.

All nodes should be able to disable MLDv1.

Control plane policing should also be implemented in order to avoid denial of services attacks.

Mitigation by a protocol update of RFC2710 [RFC2710] and RFC3810 [RFC3810]:

MLD queries should not be accepted and processed when sent to a unicast address (either link-local or global scope). This requires update of RFC 3810 and RFC 2710.

To mitigate the remote attacks, the hop-limit should have been set to 255 and MLD nodes should discard packets with a hop-limit different than 255.

5. IANA Considerations

This document contains no IANA considerations.

6. Security Considerations

This document describes multiple vulnerabilities that have been described above and tries to mitigate them or even eliminate some of them by making specific suggestions for update of the protocol as well as by suggesting the implementation of related security mechanisms to layer-2 devices.

7. Acknowledgements

The authors would like to thank Stig Venaas for some discussions on this topic.

8. References

8.1. Normative References

- [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>>.
- [RFC3810] Vida, R., Ed. and L. Costa, Ed., "Multicast Listener Discovery Version 2 (MLDv2) for IPv6", RFC 3810, DOI 10.17487/RFC3810, June 2004, <<http://www.rfc-editor.org/info/rfc3810>>.

8.2. Informative 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>>.
- [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>>.
- [RFC4541] Christensen, M., Kimball, K., and F. Solensky, "Considerations for Internet Group Management Protocol (IGMP) and Multicast Listener Discovery (MLD) Snooping Switches", RFC 4541, DOI 10.17487/RFC4541, May 2006, <<http://www.rfc-editor.org/info/rfc4541>>.

- [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>>.
- [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>>.
- [Troopers2015] Rey, E., Atlasis, A., and J. Salazar, "MLD Considered Harmful", 2015, <https://www.troopers.de/media/filer_public/7c/35/7c35967a-d0d4-46fb-8a3b-4c16df37ce59/troopers15_ipv6secsummit_atlasis_rey_salazar_mld_considered_harmful_final.pdf>.

Authors' Addresses

Eric Vyncke
Cisco
De Kleetlaan 6a
Diegem 1831
Belgium

Phone: +32 2 778 4677
Email: evyncke@cisco.com

Enno Rey
ERNW
Carl-Bosch-Str. 4
Heidelberg 69115
Germany

Phone: +49 6221 480390
Email: erey@ernw.de

Antonios Atlasis
NCI Agency
Oude Waalsdorperweg 61
The Hague 2597 AK
The Netherlands

Phone: +31 703743564
Email: antonios.atlasis@ncia.nato.int

OPSEC Working Group
Internet-Draft
Intended status: Standards Track
Expires: September 19, 2016

S. Winter
RESTENA
March 18, 2016

A Configuration File Format for Network Services on Leaf Devices
draft-winter-opsec-netconfig-metadata-00

Abstract

This document specifies a YANG module for transferring configuration information of deployments of network services towards leaf nodes (hosts) on the internet. Such configuration files are meant to be discovered, consumed and used by configuration agents on the host to achieve correct and secure setup of these services on the consuming device. This iteration of the I-D concentrates on Wi-Fi network setup and EAP credentials, but is extensible to cover a wide range including VPN, E-Mail and other services.

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 September 19, 2016.

Copyright Notice

Copyright (c) 2016 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.

Table of Contents

1.	Introduction	2
1.1.	Problem Statement	2
1.2.	Approach	3
1.3.	Other Approaches	4
1.4.	Requirements Language	5
1.5.	Terminology	5
2.	YANG module for Network Service Configuration	5
2.1.	Location of the YANG module and derived XML Schema	5
2.2.	Description of YANG Module Elements	5
2.2.1.	Overall structure	5
2.2.2.	The 'AuthenticationMethods' container	7
2.2.3.	The 'ProviderInfo' container	10
2.3.	Internationalisation / Multi-language support	11
3.	Derivation of formats from YANG source	12
4.	Issuer Authentication, Integrity Protection and Encryption of EAP Metadata configuration files	12
5.	XML Farget Format: File Discovery	12
5.1.	By MIME-Type: application/netconfig-metadata-xml	12
5.2.	By filename extension: .netconfig-metadata-xml	12
5.3.	By network location: SCAD	13
6.	JSON Farget Format: File Discovery	13
6.1.	By MIME-Type: application/netconfig-metadata-json	13
6.2.	By filename extension: .netconfig-metadata-json	13
6.3.	By network location: SCAD	13
7.	Design Decisions	13
7.1.	Why YANG and not directly XML, JSON or \$FOO?	13
7.2.	Shallow vs. Deep definition of EAP method properties	14
7.3.	EAP tunneling inside EAP tunnels	14
7.4.	Placement of 'OuterIdentity' inside 'AuthenticationMethod'	14
8.	Implementation Status	14
9.	Security Considerations	17
10.	IANA Considerations	17
11.	Contributors	18
12.	References	18
12.1.	Normative References	18
12.2.	Informative References	19
	Appendix A. Appendix A: MIME Type Registration Template	20

1. Introduction

1.1. Problem Statement

The IETF produces many protocols which require configuration by the respective end users. Many of those protocols can be configured securely or not, depending on whether features are turned on or off.

One random example is E-Mail: "STARTTLS when available" allows MITM attacks towards plaintext; "STARTTLS always" prevents them. Another example is the Extensible Authentication Protocol (EAP, [RFC3748]) and its numerous EAP methods (for example EAP-TTLS [RFC5281], EAP-TLS [RFC5216] and EAP-pwd [RFC5931]); the methods have many properties which need to be setup on the EAP server and matched as configuration items on the EAP peer for a secure EAP deployment.

Setting up these protocols and services is comparatively easy if the end-user devices which are to be configured are under central administrative control, e.g. in closed enterprise environments. Group policies or device provisioning by the IT department can push the settings to user devices.

In other environments, for example "BYOD" scenarios where users bring their own devices which are not under enterprise control, service configuration is significantly harder as it has to be done by potentially very non-technical end users.

In the case of Wi-Fi and EAP, correct configuration of all EAP deployment parameters is required to make the resulting authentications

- o functional (i.e. the end user can authenticate to an EAP server at all)
- o secure (i.e. the end user device can unambiguously authenticate the EAP server prior to releasing any sensitive client-side credentials)
- o privacy-preserving (i.e. the end user is able to conceal his username from the EAP authenticator)

It would be desirable to be able to convey the configuration information of a deployment in a machine parseable way to the end-user device, so that all the details need not be known/understood by the user. Instead, the configuration agent on the device could consume the configuration information and set up all details automatically.

However, there is currently no standard way of communicating configuration parameters to devices.

1.2. Approach

This specification defines such a file format for network service configuration metadata. The source definition is a YANG module which allows for automatic derivation of XML and JSON formats.

The specification contains several top-level elements which form the building blocks of service configuration:

- o a "Certificates" section which can contain trust roots, intermediate CA certificates, and client certificates.
- o a "ClientSideCredentials" section which contains a list of credentials which are valid for one or more services, possibly referencing a client certificate
- o a "EAPIdentityProviderList" section with a collection of EAP method details, possibly referencing certificates and ClientSideCredentials as defined above
- o a "IPSettingsList" with network configuration items needed to use a network after the authentication
- o one or more "WiFiNetwork" blocks which specify layer 2 details of a Wi-Fi connection, referencing IPSettings and possibly EAPIdentityProvider if the network is secured with EAP
- o exactly one "ProviderInfo" block with user-displayable information about the entity that provides this configuration file

The specification allows for unique identification of all elements by attaching a UUID to each setting. Using this unique identification, all parts of the configuration file can then refer to this particular piece of information. In particular, several different Wi-Fi networks can reference the same EAPIdentityProvider (and thus the same ClientSideCredential) to indicate that the same authentication settings are valid on all the networks. Configuration agents consuming the file can then ask for the corresponding client-side password once and apply it to all configuration blocks referencing that credential. When considering a hypothetical setup of three Wi-Fi networks, a VPN connection and an E-Mail account which all use the same username and password, all of those can be installed by asking the user for that username/password combination only once.

1.3. Other Approaches

Device manufacturers sometimes have developed their own proprietary configuration formats, examples include Apple's "mobileconfig" (MIME type application/x-apple-aspen-config), Microsoft's XML schemata for EAP methods for use with the command-line "netsh" tool, or Intel's "PRO/Set Wireless" binary configuration files. The multitude of proprietary file formats and their different levels of richness in expression of EAP details create a very heterogenous and non-interoperable landscape.

All of the solutions have their own excentricities or drawbacks. The Microsoft and Intel file formats are limited to Wi-Fi purposes. The Apple mobileconfig approach treats each service as distinct; the same hypothetical situation from the previous paragraph would trigger a username/password prompt five times consecutively during the installation which is rather annoying for end users.

New devices which would like to benefit from machine-parseable configuration information currently either have to choose to follow a competitor's approach and use that competitor's file format or have to develop their own. This situation is very unsatisfactory.

1.4. Requirements Language

In this document, several words are used to signify the requirements of the specification. 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]

1.5. Terminology

2. YANG module for Network Service Configuration

2.1. Location of the YANG module and derived XML Schema

The schema files are currently hosted on this location:

- o YANG module: <https://www.supPLICANTS.net/site/standardisation/ietf-winter-netconfig-metadata-00>
- o XML Schema: <https://www.supPLICANTS.net/site/standardisation/ietf-winter-netconfig-metadata-00.xsd>

2.2. Description of YANG Module Elements

2.2.1. Overall structure

The root of the Yang module is the container 'NetworkConfiguration', which contains any of the five elements Certificates, ClientSideCredentials, EAPIdentityProviderList, IPSettingsList, WiFiNetwork, ProviderInfo. These blocks carry the actual configuration information. Individual configuration elements are linked among each other to allow for re-use of the elements: a root CA certificate can be the trust root for multiple purposes; a client credential can be valid for multiple services.

Each linkable element is unique in the sense that it is identified by a UUID value which is required to be unique across the file. These linkable elements ("building blocks") are:

- o zero or more Certificate. A Certificate contains the actual blob of the certificate, an indicator what the type of the certificate is, and a display name to present in end user UI.
- o zero or more ClientSideCredential. Besides the core information - the ability to store username and password, or link to a client certificate, it contains some meta information such as:
 - * a 'ValidUntil' date-and-time timestamp with an indication of possible expiry of the information in the configuration file. Configuration agents importing the configuration file can use this information for example to re-assess whether the account is still valid (e.g. if the ValidUntil timestamp has passed, and authentication attempts consistently fail, the supplicant should consider the information stale and ask the user to verify his access authorisation with the identity provider). Typical use case: student account which expires at end of semester and needs to be reinstated.
 - * whether or not the system is allowed to save passwords when they are supplied by the user (allow-save)
 - * the format requirements on the username, if there are any (e.g. to enforce that users enter their username as a full NAI realm in the form user@suffix or as an AD domain identifier DOMAIN\user)

Consumers of configuration files MUST be able to fall back to user-interactive configuration for these parts if they are not specified (e.g. ask for the username and password during import of the configuration data). Configuration files which contain sensitive elements such as 'Password' MUST be handled with due care after the import on the device (e.g. ensure minimal file permissions, or delete the source file after installing).

- o zero or more EAPIdentityProvider containers with a list of EAP authentication details for services requiring EAP Authentication. The contents of this container are described in more detail in section Section 2.2.2
- o exactly one 'ProviderInfo' container can provide additional information about the supplier of the configuration file, e.g. a logo to allow visual identification of the provider to the user in a user interface, or Acceptable Use Policies pertaining to the use

of the configured services. This element is described in more detail in section Section 2.2.3

- o zero or more IPSettings. This block describes details on how to access the wider internet after authentication has completed. It includes ways to get an IP address (DHCP, SLAAC, manual) and proxy settings.
- o zero or more WiFiNetwork. This block describes Wi-Fi specific properties of a network and references IPSettings for layer 3, and possibly an EAPIdentityProvider for EAP authentication.

2.2.2. The 'AuthenticationMethods' container

'AuthenticationMethods' contains a sequence of 'AuthenticationMethod' groupings. Each such grouping specifies the properties of one supported authentication method of an EAPIdentityProvider. The content of this grouping is enumerated in section Section 2.2.2.1 The set of configuration parameters specified in the grouping depends on the particular EAP method to be configured.

For instance, EAP-PWD [RFC5931] does not require any server certificate parameters; EAP-FAST and TEAP are the only ones making use of Protected Access Credential (PAC) provisioning. On the other hand, properties such as outer ("anonymous") identity or the need for a trusted root Certification Authority are common to several EAP methods. The server- and client-side credential types of EAP methods are defined as a flat list of elements to choose from (see 'ServerSideCredential' and 'EAPClientSideCredential' below); see section Section 7.2 for a rationale.

Where the sequence of 'AuthenticationMethod' groupings contains more than one element, the order of appearance in the file indicates the server operator's preference for the supported EAP types; occurrences earlier in the file indicate a more preferred authentication method.

When a consuming device receives multiple 'AuthenticationMethod' groupings inside 'AuthenticationMethods', it should attempt to install more preferred methods first. During interactive provisioning of EAP properties, if the configuration information for a preferred method is insufficient (e.g. the 'AuthenticationMethod' is EAP-TLS, but the configuration file does not contain the client certificate/private key and the device's credential store is not pre-loaded with the client's certificate), the device should query whether this more preferred method should be used (requiring the user to supplement the missing data) or whether a less-preferred method should be configured instead. In non-interactive provisioning scenarios, all methods should be tried non-interactively in order

until one method can be installed; if no method can be installed in a fully automated way, provisioning is aborted.

2.2.2.1. Authentication Method Properties

The 'AuthenticationMethod' grouping contains

- o exactly one 'EAPMethod' leaf, which is an enumerated integer of the EAP method identifier as assigned by IANA (typedef eap-method)
- o zero or one container 'ServerSideCredential' which defines means to authenticate the EAP server to the EAP peer (for a list of the elements comprising this container, see section Section 2.2.2.2)
- o zero or one container 'EAPClientSideCredential' which defines means to authenticate the EAP peer to the EAP server (for a list of the elements comprising this container, see section Section 2.2.2.3)
- o zero or more 'InnerAuthenticationMethod' lists. Occurrence of this list indicates that a tunneled EAP method is in use, and that further server-side and/or client-side credentials are defined inside the tunnel. The presence of more than one 'InnerAuthenticationMethod' indicates that EAP Method Chaining is in use, i.e. that several inner EAP methods are to be executed in sequence inside the tunnel. The order of occurrence of the inner EAP methods defines the chaining order of the methods.

The 'InnerAuthenticationMethod' list itself contains the same 'EAPMethod', 'ServerSideCredentials' and 'EAPClientSideCredentials' elements as described in the preceding list, but differs in two points:

- o It can optionally contain the leaf 'NonEAPAuthMethod' (an enumerated integer of authentication methods not based on EAP) instead of 'EAPMethod' because some tunneled EAP types do not necessarily contain EAP inside the tunnel (e.g. TTLS-PAP, TEAP). The YANG definition ensures that EAPMethod and NonEAPAuthMethod are mutually exclusive in instantiations of the YANG module.
- o It can NOT contain a further 'InnerAuthenticationMethod' because establishing a secure tunnel inside an already established secure tunnel is considered a pathological case which needs not be considered. See section Section 7.3 for a rationale.

2.2.2.2. The 'ServerSideCredential' container

The server-side authentication of a mutually authenticating EAP method is typically based on X.509 certificates, which requires the EAP peer to be pre-provisioned with one or more trusted root Certification Authority (CA) prior to authenticating. A server is uniquely identified by presenting a certificate which is signed by these trusted CAs, and by the EAP peer verifying that the name of the server matches the expected one. Consequently, a (set of) CAs and a (set of) server names make up the ServerSideCredentials block.

Note that different EAP methods use different terminology when referring to trusted CA roots, server certificates, and server name identification. They also differ or have inherent ambiguity in their interpretation on where to extract the server name from (e.g. is the server name the CN part of the DistinguishedName, or is the server name one of the subjectAltName:DNS entries; what to do if there is a mismatch?). This specification introduces one single element for CA trust roots and naming; these notions map into the naming of the particular EAP methods very naturally. This specification can not remove the CN vs. sAN:DNS ambiguity in many EAP methods.

- o zero or more 'CA' lists: a Certification Authority which is trusted to sign the expected server certificate. The set of 'CA' occurrences SHOULD contain self-signed root certificates to establish trust, and MAY contain additional intermediate CA certificates which ultimately root in these self-signed root CAs. A configuration file can, but SHOULD NOT include only an intermediate CA certificate (i.e. without also including the corresponding self-signed root) because trusting only an intermediate CA without being able to verify to a self-signed root is an unsupported notion in many EAP peers.
- o zero or more 'ServerID' leaves: these leaves contain the expected server names in incoming X.509 EAP server certificates. For EAP methods not using X.509 certificates for their mutual authentication, these elements contain other string-based handles which identify the server (Example: EAP-pwd).

2.2.2.3. The 'EAPClientSideCredential' container

The actual ClientSideCredential is defined on the top-level and referenced here only by its UUID.

Besides the credentials themselves, there are a variety of EAP-specific properties pertaining to the credential. EAP methods make use of a subset of these properties only. One such property is the "Outer Identity" that some EAP methods support; another one the

ClientSideMTU which is relevant when the client has to send large amounts of client credential data (e.g. a large client certificate). As with server-side credentials, the terminology for the properties may differ slightly between EAP types. The naming convention in this specification maps nicely into the method-specific terminology. Not all the criteria make sense in all contexts; for EAP methods which do not support a criterion, configuration files SHOULD NOT contain the corresponding elements, and consumers of the file MUST ignore these elements.

Specifying any one of these elements except the UUID of the ClientSideCredential is optional.

Leaf 'AnonymousIdentity' is typically used on the outside of a tunneled EAP method and allows to specify which user identity should be used outside the tunnel. This string is not used for actual user authentication, but may contain routing hints to send the request to the right EAP server.

'PAC' contains the Protected Access Credential, typically used in EAP-FAST and TEAP.

'ProvisionPAC' is a boolean which indicates whether a PAC should be provisioned on the first connection. Note that this specification allows to use 'ProvisionPAC' without a CA nor ServerID in 'ServerSideCredential'. While this allows the operation mode of "Anonymous PAC Provisioning" as used in many field deployments of EAP-FAST (and is thus supported here), due to the known security vulnerabilities of anonymous PAC provisioning, this combination SHOULD NOT be used.

2.2.3. The 'ProviderInfo' container

This specification needs to consider that user interaction during the installation time may be required; the user at the very least must be empowered to decide whether the configuration file was issued by a provider he has an account with; the provider may have hints for the user (e.g. which password to use for the login), or may want to display links to helpdesk pages in case the user has problems with the setup or use of his identity.

The 'ProviderInfo' container allows to specify a range of potentially useful information for display to the user (some of which is relevant only during installation time, other pieces of information could be retained by the configuration agent and displayed e.g. in case of failed authentication):

- o 'DisplayName' specifies a user-friendly name for the configuration data provider. Consumers of this specification should be aware that this is simple text, and self-asserted by the producer of the configuration file. If more authoritative information about the issuer is available (e.g. if the file is signed with S/MIME and carries an Organisation name (O attribute) in the signing certificate) then the more authoritative information should be displayed with more prominence than the self-asserted one.
- o 'Description' specifies a generic descriptive text which should be displayed to the user prior to the installation of the configuration data.
- o 'ProviderLocation' specifies the approximate geographic location(s) of the configuration data provider and/or his Points of Presence. This can be useful if a configuration agent has stored or access to many configuration files and tries to suggest probable matching providers based on the device location.
- o 'ProviderLogo' specifies the logo of the configuration data provider. The same self-assertion considerations as for 'DisplayName' above apply.
- o 'TermsOfUse' contains terms of use to be displayed to and acknowledged by the user prior to the installation of the configuration on the user's system
- o 'Helpdesk' is a container with three possible sub-elements: 'EmailAddress', 'WebAddress' and 'Phone', all of which can be displayed to the user and possibly retained for future debugging hints.

2.3. Internationalisation / Multi-language support

Some elements in this specification contain text to be displayed in User Interfaces; depending on the user's language preferences, it would be desirable to present the information in a local language. Other elements contain contact information, and those contact points may only be able to handle requests in a number of languages; it may be desirable to present only contact points to the user which are compatible with his language capabilities.

All elements which either contain localisable text, or which point to external resources in localised languages, use the grouping 'localized-non-interactive' or 'localized-interactive'. These groupings can occur more than once in the specification, which enables an iteration of all applicable languages. If the grouping is omitted or its 'lang' leaf is set to "C", the instance of the element

is considered a default choice which is to be displayed if no other language is a better match.

If the entire file content consistently uses only one language set, e.g. all the elements are to be treated as "default" choices, the language can also be set for the entire 'EAPIdentityProvider' element in its own 'lang-tag' leaf.

3. Derivation of formats from YANG source

The utility 'pyang' is used to derive XML Schema (XSD) from the YANG source. The Schema for this Internet-Draft was generated with pyang 1.4.1.

4. Issuer Authentication, Integrity Protection and Encryption of EAP Metadata configuration files

S/MIME, XMLDSIG, JOSE or underlying transport security. Decisions TBD. Nuff said :-)

5. XML Farget Format: File Discovery

5.1. By MIME-Type: application/netconfig-metadata-xml

For transports where the categorisation of file types via MIME types is possible (e.g. HTTP, E-Mail), this document assigns the MIME type application/netconfig-metadata-xml

Edge devices can associate this MIME type to incoming files on such transports, and register the configuration agent which can consume the data in XML format as the default handler for this file type. By doing so, for example a single click or tap on a link to the file in the device's browser will invoke the configuration process.

This method of discovery is analogous to the Apple "mobileconfig" discovery on recent versions of Mac OS and iOS.

5.2. By filename extension: .netconfig-metadata-xml

In situations where file types can not be determined by MIME type meta-information (e.g. when the file gets stored on a local filesystem), this document RECOMMENDs that configuration data in XML format files be stored with the extension

.netconfig-metadata-xml

to identify the file as containing configuration information in XML format. Edge devices can register the configuration agent which can consume the data with this file extension. By doing so, for example a single click or tap on the filename in the device's User Interface will invoke the configuration process.

5.3. By network location: SCAD

6. JSON Farget Format: File Discovery

6.1. By MIME-Type: application/netconfig-metadata-json

For transports where the categorisation of file types via MIME types is possible (e.g. HTTP, E-Mail), this document assigns the MIME type

application/netconfig-metadata-json

Edge devices can associate this MIME type to incoming files on such transports, and register the configuration agent which can consume the data in JSON format as the default handler for this file type. By doing so, for example a single click or tap on a link to the file in the device's browser will invoke the configuration process.

6.2. By filename extension: .netconfig-metadata-json

In situations where file types can not be determined by MIME type meta-information (e.g. when the file gets stored on a local filesystem), this document RECOMMENDS that configuration data in JSON format files be stored with the extension

.netconfig-metadata-json

to identify the file as containing configuration information in JSON format. Edge devices can register the configuration agent which can consume the data with this file extension. By doing so, for example a single click or tap on the filename in the device's User Interface will invoke the configuration process.

6.3. By network location: SCAD

7. Design Decisions

7.1. Why YANG and not directly XML, JSON or \$FOO?

XML is a popular choice for EAP configurations: Microsoft's "netsh" files, Apple's "mobileconfig" files, the Wi-Fi Alliance's "PerProviderSubscription Managed Object", and other vendor/SDO definitions are all using XML.

JSON file formats for EAP configuration exist as well; most notable are Google's most recent efforts for their Chromebook Operating system.

YANG has a very rich feature set, and can codify restrictions on which element is allowed when in a much more fine-grained way than XML Schema could. Since YANG modules can be converted to XML Schema and be instantiated as XML or JSON, they can serve as an abstract notion of EAP configuration which can be deployed on consumer devices in either of those two more popular formats as needed by the device in question.

7.2. Shallow vs. Deep definition of EAP method properties

7.3. EAP tunneling inside EAP tunnels

7.4. Placement of 'OuterIdentity' inside 'AuthenticationMethod'

8. Implementation Status

RFC Editor Note: Please remove this section and the reference to [RFC6982] prior to publication.

This section records the status of known implementations of the protocol defined by this specification at the time of posting of this Internet-Draft, and is based on a proposal described in [RFC6982]. The description of implementations in this section is intended to assist the IETF in its decision processes in progressing drafts to RFCs. Please note that the listing of any individual implementation here does not imply endorsement by the IETF. Furthermore, no effort has been spent to verify the information presented here that was supplied by IETF contributors. This is not intended as, and must not be construed to be, a catalog of available implementations or their features. Readers are advised to note that other implementations may exist.

According to [RFC6982], "this will allow reviewers and working groups to assign due consideration to documents that have the benefit of running code, which may serve as evidence of valuable experimentation and feedback that have made the implemented protocols more mature. It is up to the individual working groups to use this information as they see fit".

All of the implementations listed below interoperate from producer-to consumer-side of the EAP metadata specification.

Producers of the configuration files

- o eduroam Configuration Assistant Tool

Organisation: Nicolaus Copernicus University, Torun, Poland

Implementation Name: eduroam Configuration Assistant Tool

This existing tool already produces EAP configuration files in various proprietary formats for hundreds of EAP Identity Providers. A module which produces configuration files in the XML variant as specified in an earlier revision of this draft (-00) is in production deployment.

Link to production version: <https://cat.eduroam.org>

Maturity: production

Coverage: entire specification; XML structure aligns with version -00 of this draft

Licensing: freely distributable with acknowledgement (BSD style)

Implementation experience: given that the specification is XML, it is easy to produce a configuration file with common XML libraries. The CAT Framework is written in PHP, which provides ample procedures to produce well-formed XML.

Contact Information: Tomasz Wolniewicz (see Section 11); the CAT software homepage at <http://forge.geant.net/CAT/>

Consumers of the configuration files

- o Android

Organisation: Swansea University, Swansea, Wales, U.K.

Implementation Name: eduroam CAT app

An Android app, compatible with API level 18 of Android (i.e. version 4.3 and above); the app consumes the -00 revision of this specification. The information in the config files is used to push settings to the SSID 'eduroam' (hard-coded) via the WifiEnterpriseConfig API. The app is in production deployment, with a 4-four digit amount of downloads one month after launch.

Link to production version: <https://play.google.com/store/apps/details?id=uk.ac.swansea.eduroamcat>

Maturity: production

Coverage: entire specification; XML structure aligns with version -00 of this draft

Licensing: Apache 2.0

Implementation experience: parsing XML is rather straightforward. The ability to verify signatures on XML files (S/MIME vs. XMLDSIG as discussed in Section 4) remains unclear at this point.

Contact Information: eduroam CAT Play Store app contact address (playstore@eduroam.org)

o Windows

Organisation: Amebis, d.o.o.i, Kamnik, Slovenia

Implementation Name: ArnesLink

A Windows supplicant/Enterprise WiFi installer/debugging assistant. The application consumes the -02 revision of this specification. The information from the XML variant of this specification is embedded in a larger XML file. The additional parts of the overall configuration file include information regarding the SSID to configure and other useful, but not EAP-specific information. The complete set of information is used to push settings into the Windows Wi-Fi configuration via the 'netsh' tool. The app is in production deployment.

Link to production version: <http://ftp.arnes.si/software/eduroam/ArnesLink/>

Maturity: production

Coverage: entire specification; XML structure aligns with version -02 of this draft

Licensing: GPL

Implementation experience: parsing XML is rather straightforward. For Wi-Fi configuration use, the lack of 802.11 specific details in the config file is an issue.

Contact Information: info@amebis.si

- o Linux: the authors of this specification are currently developing an application for UNIX-like operating systems which configure enterprise networks via the NetworkManager daemon; the application can consume the file format as defined in this draft specification (XML format) and configure the settings via Networkmanager's D-BUS interface.

9. Security Considerations

10. IANA Considerations

IANA is requested to allocate the MIME type "application/netconfig-metadata-xml" in the MIME Media Types / application registry (see section Section 5.1). The allocation should contain the following values:

- o Name: netconfig-metadata-xml
- o Template: see Appendix A (RFC editor note: remove this appendix prior to publication; replace this line with the URL to the application as posted online)
- o Reference: RFCabcd (RFC editor note: replace with the RFC number of this document)

IANA is requested to allocate the MIME type "application/netconfig-metadata-json" in the MIME Media Types / application registry (see section Section 5.1). The allocation should contain the following values:

- o Name: netconfig-metadata-json
- o Template: see Appendix A (RFC editor note: remove this appendix prior to publication; replace this line with the URL to the application as posted online)
- o Reference: RFCabcd (RFC editor note: replace with the RFC number of this document)

IANA is requested to allocate the location "TBD" in the "well-known URIs" registry. The allocation should contain the following values:

- o URI Suffix: TBD
- o Change Controller: IETF
- o Reference: RFCabcd (RFC editor note: replace with the RFC number of this document)

- o Related Information: none

IANA is requested to register the XML namespace "urn:ietf:params:xml:ns:netconfig-metadata-xml" in the "IETF XML Registry / ns". The allocation should contain the following values:

- o ID: netconfig-metadata-xml
- o URI: urn:ietf:params:xml:ns:netconfig-metadata-xml
- o Filename: <https://www.iana.org/assignments/xml-registry/ns/netconfig-metadata-xml.txt> (to be created by IANA)
- o Reference: RFCabcd (RFC editor note: replace with the RFC number of this document)

IANA is requested to register the XML schema "urn:ietf:params:xml:schema:netconfig-metadata-xml" in the "IETF XML Registry / schema". The allocation should contain the following values:

- o ID: netconfig-metadata-xml
- o URI: urn:ietf:params:xml:schema:netconfig-metadata-xml
- o Filename: <https://www.iana.org/assignments/xml-registry/schema/netconfig-metadata-xml.xsd> (to be created by IANA; current XSD file is linked to in section Section 2.1)
- o Reference: RFCabcd (RFC editor note: replace with the RFC number of this document)

11. Contributors

Tomasz Wolniewicz of Nicolaus Copernicus University in Torun, Poland, and Gareth J. Ayres of Swansea University in Swansea, United Kingdom, provided significant input into this specification.

12. References

12.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

12.2. Informative References

- [RFC3748] Aboba, B., Blunk, L., Vollbrecht, J., Carlson, J., and H. Levkowitz, "Extensible Authentication Protocol (EAP)", RFC 3748, June 2004.
- [RFC5216] Simon, D., Aboba, B., and R. Hurst, "The EAP-TLS Authentication Protocol", RFC 5216, March 2008.
- [RFC5281] Funk, P. and S. Blake-Wilson, "Extensible Authentication Protocol Tunneled Transport Layer Security Authenticated Protocol Version 0 (EAP-TTLSv0)", RFC 5281, August 2008.
- [RFC5931] Harkins, D. and G. Zorn, "Extensible Authentication Protocol (EAP) Authentication Using Only a Password", RFC 5931, August 2010.
- [RFC6982] Sheffer, Y. and A. Farrel, "Improving Awareness of Running Code: The Implementation Status Section", RFC 6982, July 2013.
- [RFC7593] Wierenga, K., Winter, S., and T. Wolniewicz, "The eduroam Architecture for Network Roaming", RFC 7593, DOI 10.17487/RFC7593, September 2015, <<http://www.rfc-editor.org/info/rfc7593>>.
- [HS20] Wi-Fi Alliance, "Hotspot 2.0 Technical Specification", 2012, <<https://www.wi-fi.org/hotspot-20-technical-specification-v100>>.

Appendix A. Appendix A: MIME Type Registration Template

The following values will be used for the online MIME type registration at <https://www.iana.org/form/media-types>

Your Name: Stefan Winter

Your Email Address: stefan.winter@restena.lu

Media Type Name: Application

Subtype name: 1) (Standards tree) netconfig-metadata-xml

Subtype name: 2) (Standards tree) netconfig-metadata-json

Required parameters: (none)

Optional parameters: (none)

Encoding Considerations: 8-Bit text

Security Considerations: This file type carries configuration information for consumer devices. It has the potential to substantially alter the consumer's device; particularly to install a new trusted Certification Authority. Applications consuming files of this type need to be cautious to explain to the end user what is being altered, so that they understand the consequences. For further explanations, see Section 9 of this draft. (Note to RFC Editor: replace with the number of this RFC once known)

Interoperability Considerations: The file content is in 1) XML version 1.0 or later; 2) JSON. The encoding SHOULD be UTF-8, but implementations consuming the file SHOULD be prepared to encounter different encodings.

Published Specification: draft-winter-opsec-netconfig-metadata
(Note to RFC Editor: replace this reference with the RFC number of this document once known)

Applications which use this media type: files of this type are intended for consumption by software on edge devices; they consume the information therein to configure authentication parameters of various network services which are then applied to network or application authentication scenarios.

Fragment Identifier Considerations: files of this type are expected to be transmitted in their entirety. If a reference to a specific part of the content is to be made, XML XPath expressions

are to be used. I.e. fragment identifier formats are not expected to be used.

Restrictions on Usage: none

Provisional registration: initial submission of this form will be executed after adoption in the IETF; it will be a provisional registration. Final registration will be done after IESG review.

Additional information:

Deprecated alias types for this name: none

Magic numbers: none

File extensions: 1) netconfig-metadata-xml

File extensions: 2) netconfig-metadata-json

Macintosh File Type Codes: TBD

Object Identifiers or OIDs: none

Intended Usage: Common (no further provisions)

Other Information/General Comment: none

Person to contact for further information:

Name: Stefan Winter

E-Mail: stefan.winter@restena.lu

Author/Change controller: IETF

DATA

Author's Address

Stefan Winter
Fondation RESTENA
6, rue Richard Coudenhove-Kalergi
Luxembourg 1359
LUXEMBOURG

Phone: +352 424409 1
Fax: +352 422473
EMail: stefan.winter@restena.lu
URI: <http://www.restena.lu>.