      Benchmarking Methodology for Network Security Device Performance
              draft-balarajah-bmwg-ngfw-performance-02

Abstract

   This document provides benchmarking terminology and methodology for
   next-generation network security devices including next-generation
   firewalls (NGFW), intrusion detection and prevention solutions (IDS/
   IPS) and unified threat management (UTM) implementations. The
   document aims to strongly improve the applicability, reproducibility
   and transparency of benchmarks and to align the test methodology with
   today's increasingly complex layer 7 application use cases. The main
   areas covered in this document are test terminology, traffic profiles
   and benchmarking methodology for NGFWs to start with.

Status of This Memo

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

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

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

   This Internet-Draft will expire on September 6, 2018.

Copyright Notice

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

   15 years have passed since IETF recommended test methodology and
   terminology for firewalls initially (RFC 2647, RFC 3511). The
   requirements for network security element performance and
   effectiveness have increased tremendously since then. Security
   function implementations have evolved to more advanced areas and have
   diversified into intrusion detection and prevention, threat
   management, analysis of encrypted traffic, etc. In an industry of
   growing importance, well-defined and reproducible key performance
   indicators (KPIs) are increasingly needed: They enable fair and
   reasonable comparison of network security functions. All these
   reasons have led to the creation of a new next-generation firewall
   benchmarking document.

2. Requirements

   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. Scope

   This document provides testing terminology and testing methodology
   next-generation firewalls and related security functions. It covers
   two main areas: Performance benchmarks and security effectiveness
   testing. The document focuses on advanced, realistic, and
   reproducible testing methods. Additionally it describes test bed
   environments, test tool requirements and test result formats.

4. Test Setup

   Test setup defined in this document will be applicable to all of the
   benchmarking test cases described in Section 7.

4.1. Testbed Configuration

   Testbed configuration MUST ensure that any performance implications
   that are discovered during the benchmark testing aren't due to the
   inherent physical network limitations such as number of physical
   links and forwarding performance capabilities (throughput and
   latency) of the network devise in the testbed. For this reason, this
   document recommends to avoid external devices such as switch and
   router in the testbed as possible.

   In the typical deployment, the security devices (DUT/SUT) will not
   have a large number of entries in MAC or ARP tables, which impact the
   actual DUT/SUT performance due to MAC and ARP table lookup processes.
   Therefore, depend on number of used IP address in client and server
   side, it is recommended to connect Layer 3 device(s) between test
   equipment and DUT/SUT as shown in Figure 1.

   If the test equipment is capable to emulate layer 3 routing
   functionality and there is no need for test equipment ports
   aggregation, it is recommended to configure the test setup as shown
   in Figure 2.

```
   +------------------+       +----------+     +--------------------+
   |Aggregation Switch/|      |          |     | Aggregation Switch/|
   | Router           +------+  DUT/SUT  +------+ Router            |
   |                  |      |          |     |                    |
   +----------+-------+       +----------+     +--------+-----------+
              |                                          |
              |                                          |
   +----------+----------+              +----------+-----------+
   |                     |              |                      |
   | +-----------------+ |              | +------------------+ |
   | | Emulated Router(s)| |            | | Emulated Router(s)| |
   | |    (Optional)   | |              | |    (Optional)    | |
   | +-----------------+ |              | +------------------+ |
   | +-----------------+ |              | +------------------+ |
   | |     Clients     | |              | |     Servers      | |
   | +-----------------+ |              | +------------------+ |
   |                     |              |                      |
   |   Test Equipment    |              |   Test Equipment     |
   +---------------------+              +----------------------+
```

                   Figure 1: Testbed Setup - Option 1

```
+----------------------+         +----------+         +----------------------+
| +------------------+ |         |          |         | +------------------+ |
| | Emulated Router(s)| |         |          |         | | Emulated Router(s)| |
| |    (Optional)    | | +------ DUT/SUT  +-----+ | |    (Optional)    | |
| +------------------+ |         |          |         | +------------------+ |
| +------------------+ |         +----------+         | +------------------+ |
| |     Clients      | |                              | |     Servers      | |
| +------------------+ |                              | +------------------+ |
|                      |                              |                      |
|   Test Equipment     |                              |   Test Equipment     |
+----------------------+                              +----------------------+
```
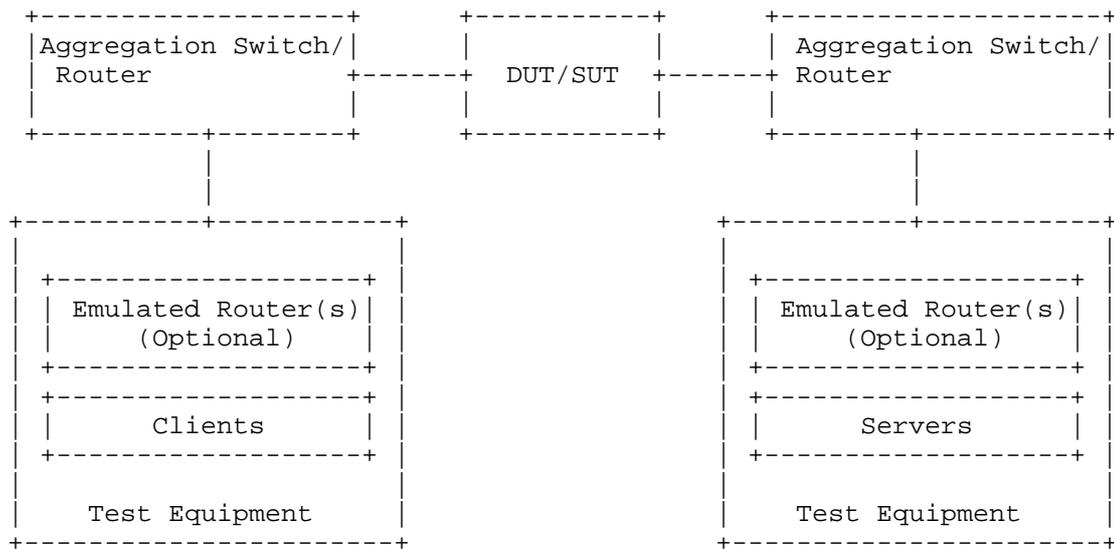
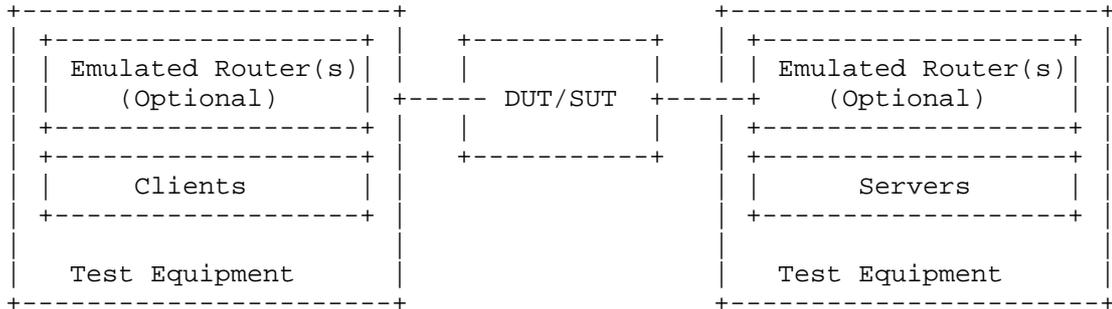   Figure 2: Testbed Setup - Option 2

4.2. DUT/SUT Configuration

   An unique DUT/SUT configuration MUST be used for all of the
   benchmarking tests described in Section 7. Since each DUT/SUT will
   have their own unique configuration, users SHOULD configure their
   device with the same parameters that would be used in the actual
   deployment of the device or a typical deployment. Also it is
   mandatory to enable all the security features on the DUT/SUT in order
   to achieve maximum security coverage for a specific deployment
   scenario.

   This document attempts to define the recommended security features
   which SHOULD be consistently enabled for all test cases. The table
   below describes the recommended sets of feature list which SHOULD be
   configured on the DUT/SUT. In order to improve repeatability, a
   summary of the DUT configuration including description of all enabled
   DUT/SUT features MUST be published with the benchmarking results.

		Included in initial Scope	Added to future Scope	NGIPS	AD	WAF	BPS	SSL Broker
				Device				
DUT Features	Feature	**NGFW**		Future test standards to be developed				
SSL Inspection	x		x					
IDS/IPS	x	x						
Web Filtering	x		x					
Antivirus	x	x						
Anti Spyware	x	x						
Anti Botnet	x	x						
DLP	x		x					
DDoS	x		x					
Certificate Validation	x		x					
Logging and Reporting	x	x						
Application Identification	x	x						

Table 1: DUT/SUT Feature List

In addition, it is also recommended to configure a realistic number of access policy rules on the DUT/SUT. This document determines the number of access policy rules for three different class of DUT/SUT. The classification of the DUT/SUT MAY be based on its maximum supported throughput performance number. This document classifies the DUT/SUT in three different categories; namely small, medium and maximum.

The recommended throughput values for the following classes are;

Small - supported throughput less than 5Gbit/s

Medium - supported throughput greater than 5Gbit/s and less than 10Gbit/s

Large - supported throughput greater than 10Gbit/s

The access rule defined in the table 2 MUST be configured from top to bottom in correct order. The configured access policy rule MUST NOT block the test traffic used for the benchmarking test scenario.

				DUT/SUT Classification # Rules		
Rules Type	Match Criteria	Description	Action	Small	Medium	Large
Application layer	Application	Any application traffic NOT included in the test traffic	block	10	20	50
Transport layer	Src IP and TCP/UDP Dst ports	Any src IP use in the test AND any dst ports NOT used in the test traffic	block	50	100	250
IP layer	Src/Dst IP	Any src/dst IP NOT used in the test	block	50	100	250
Application layer	Application	Applications included in the test traffic	allow	10	10	10
Transport layer	Src IP and TCP/UDP Dst ports	Half of the src IP used in the test AND any dst ports used in the test traffic. One rule per subnet	allow	1	1	1
IP layer	Src IP	The rest of the src IP subnet range used in the test. One rule per subnet	allow	1	1	1

Table 2: DUT/SUT Access List

4.3. Test Equipment Configuration

   In general, test equipment allows configuring parameters in different
   protocol level. These parameters thereby influencing the traffic
   flows which will be offered and impacting performance measurements.

This document attempts to explicitly specify which test equipment
parameters SHOULD be configurable, any such parameter(s) MUST be
noted in the test report.

4.3.1. Client Configuration

This section specifies which parameters SHOULD be considerable while
configuring emulated clients using test equipment. Also this section
specifies the recommended values for certain parameters.

4.3.1.1. TCP Stack Attributes

The TCP stack SHOULD use a TCP Reno variant, which include congestion
avoidance, back off and windowing, retransmission and recovery on
every TCP connection between client and server endpoints. The
default IPv4 and IPv6 MSS segments size MUST be set to 1460 bytes and
1440 bytes and a TX and RX receive windows of 32768 bytes. Delayed
ACKs are permitted, but it SHOULD be limited to either a 200 msec
delay timeout or 3000 in bytes before a forced ACK. Up to 3 retries
SHOULD be allowed before a timeout event is declared. All traffic
MUST set the TCP PSH flag to high. The source port range SHOULD be
in the range of 1024 - 65535. Internal timeout SHOULD be dynamically
scalable per RFC 793.

4.3.1.2. Client IP Address Space

The sum of the client IP space SHOULD contain the following
attributes. The traffic blocks SHOULD consist of multiple unique,
continuous static address blocks. A default gateway is permitted.
The IPv4 ToS byte should be set to '00'.

The following equation can be used to determine the required total
number of client IP address.

Desired total number of client IP = Target throughput [Mbit/s] /
Throughput per IP address [Mbit/s]

(Idea 1) 6-7 Mbps per IP= 1,400-1,700 IPs per 10Gbit/s throughput

(Idea 2) 0.1-0.2 Mbps per IP = 50,000-100,000 IPs per 10Gbit/s
          throughput

Based on deployment and usecase scenario, client IP addresses SHOULD
be distributed between IPv4 and IPv6 type. This document recommends
using the following ratio(s) between IPv4 and IPv6:

(Idea 1) 100 % IPv4, no IPv6

(Idea 2) 80 % IPv4, 20 % IPv6

(Idea 3) 50 % IPv4, 50 % IPv6

(Idea 4) 0 % IPv4, 100 % IPv6

4.3.1.3. Emulated Web Browser Attributes

   The emulated web browser contains attributes that will materially
   affect how traffic is loaded. The objective is to emulate a modern,
   typical browser attributes to improve realism of the result set.

   For HTTP traffic emulation, the emulated browser must negotiate HTTP
   1.1. HTTP persistency MAY be enabled depend on test scenario. The
   browser CAN open multiple TCP connections per Server endpoint IP at
   any time depending on how many sequential transactions are needed to
   be processed. Within the TCP connection multiple transactions can be
   processed if the emulated browser has available connections. The
   browser MUST advertise a User-Agent header. Headers will be sent
   uncompressed. The browser should enforce content length validation.

   For encrypted traffic, the following attributes shall define the
   negotiated encryption parameters. The tests must use TLSv1.2 or
   higher with a record size of 16383, commonly used cipher suite and
   key strength. Session reuse or ticket resumption may be used for
   subsequent connections to the same Server endpoint IP. The client
   endpoint must send TLS Extension SNI information when opening up a
   security tunnel. Server certificate validation should be disabled.
   Server certificate validation should be disabled. Cipher suite and
   certificate size should be defined in the parameter session of
   benchmarking tests.

4.3.2. Backend Server Configuration

   This document attempts to specify which parameters should be
   considerable while configuring emulated backend servers using test
   equipment.

4.3.2.1. TCP Stack Attributes

   The TCP stack SHOULD use a TCP Reno variant, which include congestion
   avoidance, back off and windowing, retransmission and recovery on
   every TCP connection between client and server endpoints. The
   default IPv4 MSS segment size MUST be set to 1460 bytes and a TX and
   RX receive windows of at least 32768 bytes. Delayed ACKs are
   permitted but SHOULD be limited to either a 200 msec delay timeout or
   3000 in bytes before a forced ACK. Up to 3 retries SHOULD be allowed
   before a timeout event is declared. All traffic MUST set the TCP PSH

flag to high. The source port range SHOULD be in the range of 1024 -
65535. Internal timeout should be dynamically scalable per RFC 793.

4.3.2.2. Server Endpoint IP Addressing

The server IP blocks should consist of unique, continuous static
address blocks with one IP per Server FQDN endpoint per test port.
The IPv4 ToS byte should be set to '00'. The source mac address of
the server endpoints shall be the same emulating routed behavior.
Each Server FQDN should have it's own unique IP address. The Server
IP addressing should be fixed to the same number of FQDN entries.

4.3.2.3. HTTP / HTTPS Server Pool Endpoint Attributes

The emulated server pool for HTTP should listen on TCP port 80 and
emulated HTTP version 1.1 with persistence. For HTTPS server, the
pool must have the same basic attributes of an HTTP server pool plus
attributes for SSL/TLS. The server must advertise a server type.
For HTTPS server, TLS 1.2 or higher must be used with a record size
of 16383 bytes and ticket resumption or Session ID reuse enabled.
The server must listen on port TCP 443. The server shall serve a
certificate to the client. It is required that the HTTPS server also
check Host SNI information with the Fully Qualified Domain Name
(FQDN). Client certificate validation should be disabled. Cipher
suite and certificate size should be defined in the parameter session
of benchmarking tests.

4.3.3. Traffic Flow Definition

The section describes the traffic pattern between the client and
server endpoints. At the beginning of the test, the server endpoint
initializes and will be in a ready to accept connection state
including initialization of the TCP stack as well as bound HTTP and
HTTPS servers. When a client endpoint is needed, it will initialize
and be given attributes such as the MAC and IP address. The behavior
of the client is to sweep though the given server IP space,
sequentially generating a recognizable service by the DUT. Thus, a
balanced, mesh between client endpoints and server endpoints will be
generated in a client port server port combination. Each client
endpoint performs the same actions as other endpoints, with the
difference being the source IP of the client endpoint and the target
server IP pool. The client shall use Fully Qualified Domain Names in
Host Headers and for TLS 1.2 Server Name Indication (SNI).

4.3.3.1. Description of Intra-Client Behavior

   Client endpoints are independent of other clients that are
   concurrently executing. When a client endpoint initiate traffic,
   this section will describe how the steps though different services.
   Once initialized, the user should randomly hold (perform no
   operation) for a few milliseconds to allow for better randomization
   of start of client traffic. The client will then either open up a
   new TCP connection or connect to a TCP persistence stack still open
   to that specific server. At any point that the service profile may
   require encryption, a TLS 1.2 encryption tunnel will form presenting
   the URL request to the server. The server will then perform an SNI
   name check with the proposed FQDN compared to the domain embedded in
   the certificate. Only when correct, will the server process the
   object. The initial object to the server may not have a fixed size;
   its size is based on benchmarking tests described in Section 7.
   Multiple additional sub-URLs (Objects on the service page) may be
   requested simultaneously. This may or may not be to the same server
   IP as the initial URL. Each sub-object will also use a conical FQDN
   and URL path, as observed in the traffic mix used.

4.3.4. Traffic Load Profile

   The loading of traffic will be described in this section. The
   loading of an traffic load profile has five distinct phases: Init,
   ramp up, sustain, ramp down/close, and collection.

   Within the Init phase, test bed devices including the client and
   server endpoints should negotiate layer 2-3 connectivity such as MAC
   learning and ARP. Only after successful MAC learning or ARP
   resolution shall the test iteration move to the next phase. No
   measurements are made in this phase. The minimum recommended time
   for init phase is 5 seconds. During this phase the emulated clients
   SHOULD NOT initiate any sessions with the DUT/SUT, in contrast, the
   emulated servers should be ready to accept requests from DUT/SUT or
   from emulated clients.

   In the ramp up phase, the test equipment should start to generate the
   test traffic. It should use a set approximate number of unique
   client IP addresses actively to generate traffic. The traffic should
   ramp from zero to desired target objective. The target objective
   will be defined for each benchmarking test. The duration for the
   ramp up phase must be configured long enough, so that the test
   equipment do not overwhelm DUT/SUT's supported performance metrics
   namely; connection setup rate, concurrent connection and application
   transaction. The recommended time duration for the ramp up phase is
   180- 300 seconds. No measurements are made in this phase.

In the sustain phase, the test equipment should keep to generate
traffic t constant target value for a constant number of active
client IPs. The recommended time duration for sustain phase is 600
seconds. This is the phase where measurements occur.

In the ramp down/close phase, no new connection is established and no
measurements are made. The recommend duration of this phase is
between 180 to 300 seconds.

The last phase is administrative and will be when the tester merges
and collates the report data.

5. Test Bed Considerations

This section recommends steps to control the test environment and
test equipment, specifically focusing on virtualized environments and
virtualized test equipment.

1. Ensure that any ancillary switching or routing functions between
    the system under test and the test equipment do not limit the
    performance of the traffic generator. This is specifically
    important for virtualized components (vSwitches, vRouters).

2. Verify that the performance of the test equipment matches and
    reasonably exceeds the expected maximum performance of the system
    under test.

3. Assert that the test bed characteristics are stable during the
    whole test session. A number of factors might influence
    stability specifically for virtualized test beds, for example
    additional work loads in a virtualized system, load balancing and
    movement of virtual machines during the test, or simple issues
    such as additional heat created by high workloads leading to an
    emergency CPU performance reduction.

Test bed reference pre-tests help to ensure that the desired traffic
generator aspects such as maximum throughput and the network
performance metrics such as maximum latency and maximum packet loss
are met.

Once the desired maximum performance goals for the system under test
have been identified, a safety margin of 10 % SHOULD be added for
throughput and subtracted for maximum latency and maximum packet
loss.

Test bed preparation may be performed either by configuring the DUT
in the most trivial setup (fast forwarding) or without presence of
DUT.

6. Reporting

   This section describes how the final report should be formatted and
   presented. The final test report may have two major sections;
   Introduction and result sections. The following attributes should be
   present in the introduction section of the test report.

   1. The name of the NetSecOPEN traffic mix must be prominent.

   2. The time and date of the execution of the test must be prominent.

   3. Summary of testbed software and Hardware details

       A. DUT Hardware/Virtual Configuration

           + This section should clearly identify the make and model of
              the DUT

           + iThe port interfaces, including speed and link information
              must be documented.

           + If the DUT is a virtual VNF, interface acceleration such
              as DPDK and SR-IOV must be documented as well as cores
              used, RAM used, and the pinning / resource sharing
              configuration. The Hypervisor and version must be
              documented.

           + Any additional hardware relevant to the DUT such as
              controllers must be documented

       B. DUT Software

           + The operating system name must be documented

           + The version must be documented

           + The specific configuration must be documented

       C. DUT Enabled Features

           + Specific features, such as logging, NGFW, DPI must be
              documented

           + iAttributes of those featured must be documented

           + Any additional relevant information about features must be
              documented

      D. Test equipment hardware and software

          + Test equipment vendor name

          + Hardware details including model number, interface type

          + Test equipment firmware and test application software
             version

   4. Results Summary / Executive Summary

       1. Results should resemble a pyramid in how it is reported, with
           the introduction section documenting the summary of results
           in a prominent, easy to read block.

       2. In the result section of the test report, the following
           attributes should be present for each test scenario.

           a. KPIs must be documented separately for each test
               scenario. The format of the KPI metrics should be
               presented as described in Section 6.1.

           b. The next level of detains should be graphs showing each
               of these metrics over the duration (sustain phase) of the
               test. This allows the user to see the measured
               performance stability changes over time.

6.1. Key Performance Indicators

   This section lists KPIs for overall benchmarking tests scenarios.
   All KPIs MUST be measured in whole period of sustain phase as
   described in Section 4.3.4. All KPIs MUST be measured from test
   equipment's result output.

   o TCP Concurrent Connection
      This key performance indicator will measure the average concurrent
      open TCP connections in the sustaining period.

   o TCP Connection Setup Rate
      This key performance indicator will measure the average
      established TCP connections per second in the sustaining period.
      For Session setup rate benchmarking test scenario, the KPI will
      measure average established and terminated TCP connections per
      second simultaneously.

   o Application Transaction Rate
      This key performance indicator will measure the average successful
      transactions per seconds in the sustaining period.

o TLS Handshake Rate
   This key performance indicator will measure the average TLS 1.2 or
   higher session formation rate within the sustaining period.

o Throughput
   This key performance indicator will measure the average Layer 1
   throughput within the sustaining period as well as average packets
   per seconds within the same period. The value of throughput
   should be presented in Gbps rounded to two places of precision
   with a more specific kbps in parenthesis. Optionally, goodput may
   also be logged as an average goodput rate measured over the same
   period. Goodput result shall also be presented in the same format
   as throughput.

o URL Response time / Time to Last Byte (TTLB)
   This key performance indicator will measure the minimum, average
   and maximum per URL response time in the sustaining period as well
   as the average variance in the same period.

o Application Transaction Time
   This key performance indicator will measure the minimum, average
   and maximum the amount of time to receive all objects from the
   server.

o Time to First Byte (TTFB)
   This key performance indicator will measure minimum, average and
   maximum the time to first byte. TTFB is the elapsed time between
   sending the SYN packet from the client and receiving the first
   byte of application date from the DUT/SUT. TTFB SHOULD be
   expressed in millisecond.

o TCP Connect Time
   This key performance indicator will measure minimum, average and
   maximum TCP connect time. It is elapsed between the time the
   client sends a SYN packet and the time it receives the SYN/ACK.
   TCP connect time SHOULD be expressed in millisecond.

7. Benchmarking Tests

7.1. Throughput Performance With NetSecOPEN Traffic Mix

7.1.1. Objective

   To determine the average throughput performance of the DUT/SUT when
   using application traffic mix defined in Section 7.1.3.3.

7.1.2. Test Setup

   Test bed setup MUST be configured as defined in Section 4. Any test
   scenario specific test bed configuration changes must be documented.

7.1.3. Test Parameters

   In this section, test scenario specific parameters SHOULD be defined.

7.1.3.1. DUT/SUT Configuration Parameters

   DUT/SUT parameters MUST conform to the requirements defined in
   Section 4.2. Any configuration changes for this specific test
   scenario MUST be documented.

7.1.3.2. Test Equipment Configuration Parameters

   Test equipment configuration parameters MUST conform to the
   requirements defined in Section 4.3. Following parameters MUST be
   noted for this test scenario:

      Client IP address range

      Server IP address range

      Traffic distribution ratio between IPv4 and IPv6

      Traffic load objective or specification type (e.g. Throughput,
      SimUsers and etc.)

      Target throughput: It MAY be defined based on requirements.
      Otherwise it represents aggregated line rate of interface(s) used
      in the DUT/SUT

      Initial throughput: Initial throughput MAY be up to 10% of the
      "Target throughput"

7.1.3.3. Traffic Profile

   Test scenario MUST be run with a single application traffic mix
   profile. The name of the NetSecOpen traffic mix MUST be documented.

7.1.3.4. Test Results Acceptance Criteria

   The following test Criteria is defined as test results acceptance
   criteria

   a. Number of failed Application transaction MUST be 0.01%.

   b. Number of Terminated TCP connection due to unexpected TCP RST
       sent by DUT/SUT MUST be less than 0.01%

   c. Maximum deviation (max. dev) of application transaction time /
       TTLB (Time To Last Byte) MUST be less than X (The value for "X"
       will be finalyzed and updated in future draft release)
       The following equation MUST be used to calculate the deviation of
       application transaction time or TTLB.

       max. dev = max((avg_latency - min_latency),(max_latency -
       avg_latency)) / (Initial latency)

       Where, the initial latency is calculated using the following
       equation. For this calculation, the latency values (min', avg'
       and max') MUST be measured during test procedure step 1 as
       defined in Section 7.1.4.1.
       The variable latency represents application transaction time or
       TTLB.

       Initial latency:= min((avg' latency - min' latency) | (max'
       latency - avg' latency))

   d. Maximum value of TCP connect time must be less than Xms (The
       value for "X" will be finalyzed and updated in future draft
       release). The definition for TCP connect time is found in
       Section 6.1.

   e. Maximum value of Time to First Byte must be less than 2* TCP
       connect time.

   Test Acceptance criteria for this test scenario MUST be monitored
   during the sustain phase of the traffic load profile only.

7.1.3.5. Measurement

   Following KPI metrics MUST be reported for this test scenario.

   Mandatory KPIs: average Throughput, maximum Concurrent TCP
   connection, TTLB/application transaction time (minimum, average and
   maximum) and average application transaction rate

   Optional KPIs: average TCP connection setup rate, average TLS
   handshake rate, TCP connect time and TTFB

7.1.4. Test Procedures and expected Results

   The test procedure is designed to measure the throughput performance
   of the DUT/SUT at the sustaining period of traffic load profile. The
   test procedure consists of three major steps.

7.1.4.1. Step 1: Test Initialization and Qualification

   Verify the link status of the all connected physical interfaces. All
   interfaces are expected to be "UP" status.

   Configure traffic load profile of the test equipment to generate test
   traffic at "initial throughput" rate as described in the parameters
   section. The DUT/SUT SHOULD reach the "initial throughput" during
   the sustain phase. Measure all KPI as defined in Section 7.1.3.5.
   The measured KPIs during the sustain phase MUST meet acceptance
   criteria "a" and "b" defined in Section 7.1.3.4.

   If the KPI metrics do not meet the acceptance criteria, the test
   procedure MUST NOT be continued to step 2.

7.1.4.2. Step 2: Test Run with Target Objective

   Configure test equipment to generate traffic at "Target throughput"
   rate defined in the parameter table. The test equipment SHOULD
   follow the traffic load profile definition as described in
   Section 4.3.4. The test equipment SHOULD start to measure and record
   all specified KPIs. The frequency of KPI metrics measurement MUST be
   less than 5 seconds. Continue the test until all traffic profile
   phases are completed.

   The DUT/SUT is expected to reach the desired target throughput during
   the sustain phase. In addition, the measured KPIs must meet all
   acceptance criteria. Follow the step 3, if the KPI metrics do not
   meet the acceptance criteria.

7.1.4.3. Step 3: Test Iteration with Binary Search

   Use binary search algorithm to configure the desired traffic load
   profile for each test iteration. Binary search algorithmn can be
   implemented using the parameter; Resolution =0.01* Target throughput
   and Backoff= 50%.

   Determine the maximum and average achievable throughput within the
   acceptance criteria.

7.2. Concurrent TCP Connection Capacity With HTTP Traffic

7.2.1. Objective

   Determine the maximum number of concurrent TCP connection that DUT/
   SUT sustains when using HTTP traffic.

7.2.2. Test Setup

   Test bed setup SHOULD be configured as defined in Section 4. Any
   specific test bed configuration changes such as number of interfaces
   and interface type, etc. must be documented.

7.2.3. Test Parameters

   In this section, test scenario specific parameters SHOULD be defined.

7.2.3.1. DUT/SUT Configuration Parameters

   DUT/SUT parameters MUST conform to the requirements defined in
   Section 4.2. Any configuration changes for this specific test
   scenario MUST be documented.

7.2.3.2. Test Equipment Configuration Parameters

   Test equipment configuration parameters MUST conform to the
   requirements defined in Section 4.3. Following parameters MUST be
   noted for this test scenario:

      Client IP address range

      Server IP address range

      Traffic distribution ratio between IPv4 and IPv6

      Traffic load objective or specification type (e.g Throughput,
      SimUsers and etc.)

      Target concurrent connection: It can be defined based on
      requirements

      Initial concurrent connection: 10% of "Target concurrent
      connection"

7.2.3.2.1. Client Configuration Parameters

   The client must negotiate HTTP 1.1 with persistence and each client
   can open multiple concurrent TCP connections per server endpoint IP.

   Test scenario SHOULD be run with a single traffic profile with
   following attributes:

   HTTP 1.1 with GET command requesting 10 Kbyte objects with random
   MIME type.

   The test equipment SHOULD perform HTTP transactions within each TCP
   connection subsequently. The frequency of transactions MUST be
   defined to achieve X% of total throughput that DUT can support. The
   suggested value of X is 25. It will be finalyzed and updated in the
   next draft version.

   During the sustain state of concurrent connection and traffic load ,
   a minimal % of TCP connection SHOULD be closed and re-opened.

7.2.3.3. Test Results Acceptance Criteria

   The following test Criteria is defined as test results acceptance
   criteria

   a. Number of failed Application transaction MUST be less than 0.01%
       of attempt transaction.

   b. Number of Terminated TCP connection due to unexpected TCP RST
       sent by DUT/SUT MUST be less than 0.01% of total initiated TCP
       sessions.

   c. During the sustain phase, traffic should be forwarded constantly
       at the rate defined in the parameter Section 7.2.3.

   d. Maximum deviation (max. dev) of application transaction time /
       TTLB (Time To Last Byte) MUST be less than Xms (The value for "X"
       will be finalyzed and updated in future draft release).
       The following equation MUST be used to calculate the deviation of
       application transaction time or TTLB.

       max. dev = max((avg_latency - min_latency),(max_latency -
       avg_latency)) / (Initial latency)

       Where, the initial latency is calculated using the following
       equation. For this calculation, the latency values (min', avg'
       and max') MUST be measured during test procedure step 1 as
       defined in Section 7.1.4.1.

The variable latency represents application transaction time or TTLB.

Initial latency:= min((avg' latency - min' latency) | (max' latency - avg' latency))

e. Maximum value of TCP connect time must be less than Xms (The value for "X" will be finalized and updated in future draft release). The definition for TCP connect time is found in Section 6.1.

f. Maximum value of Time to First Byte must be less than 2* TCP connect time.

Test Acceptance criteria for this test scenario MUST be monitored during the sustain phase of the traffic load profile only.

7.2.3.4. Measurement

Following KPI metrics MUST be reported for this test scenario;

average Throughput, max. Min. Avg. Concurrent TCP connection, TTLB/ application transaction time (minimum, average and maximum) and average application transaction rate.

7.2.4. Test Procedures and expected Results

The test procedure is designed to measure the concurrent TCP connection capacity of the DUT/SUT at the sustaining period of traffic load profile. The test procedure consists of three major steps. This test procedure MAY be repeated multiple times with different IPv4 and IPv6 traffic distribution.

7.2.4.1. Step 1: Test Initialization and Qualification

Verify the link status of the all connected physical interfaces. All interfaces are expected to be "UP" status.

Configure traffic load profile of the test equipment to establish "initial concurrent connection" as defined in the parameters section. The traffic load profile should be defined as described in Section 4.3.4.

The DUT/SUT SHOULD reach the "initial concurrent connection" during the sustain phase. The measured KPIs during the sustain phase MUST meet the acceptance criteria "a" and "b" defined in Section 7.2.3.3

If the KPI metrics do not meet the acceptance criteria, the test
procedure MUST NOT be continued to "Step 2".

7.2.4.2. Step 2: Test Run with Target Objective

Configure test equipment to establish "Target concurrent connection"
defined in the parameters table. The test equipment SHOULD follow
the traffic load profile definition as described in Section 4.3.4.

During the ramp up and sustain phase, the other KPIs such as
throughput, TCP connection rate and application transaction MUST NOT
reach to the maximum value that the DUT/SUT can support. Throughput,
TCP connection rate and application transaction should not be reached
more than X% of maximum value that DUT can support. The suggested
value of X is 25. It will be finalized and updated in the next draft
version.

The test equipment SHOULD start to measure and record all specified
KPIs. The frequency of measurement MUST be less than 5 seconds.
Continue the test until all traffic profile phases are completed.

The DUT/SUT is expected to reach the desired target concurrent
connection at the sustain phase. In addition, the measured KPIs must
meet all acceptance criteria.

Follow the step 3, if the KPI metrics do not meet the acceptance
criteria.

7.2.4.3. Step 3: Test Iteration with Binary Search

Use binary search algorithm to configure the desired traffic load
profile for each test iteration. Binary search algorithmn can be
implemented using the parameter; Resolution =0.01* "Target concurrent
connection" and Backoff= 50%.

Determine the maximum and average achievable throughput within the
acceptance criteria.

7.3. TCP/HTTP Connections Per Second

7.3.1. Objective

Using HTTP traffic, determine the maximum and average value of TCP
session establishment rate supported by the DUT/SUT.

Test parameters and test test procedures will be added in the future
release.

7.4. HTTP Transactions Per Second

7.4.1. Objective

   Determine maximum and average HTTP transacton rate supported by the
   DUT/SUT.

   Test parameters and test test procedures will be added in the future
   release.

7.5. HTTP Throughput

7.5.1. Objective

   Determine the average throughput performance of the DUT/SUT when
   using HTTP traffic.

   Test parameters and test test procedures will be added in the future
   release.

7.6. HTTP Transaction Latency

7.6.1. Objective

   Determine the minimum, average and maximum values of HTTP transaction
   latency at 80% throughput rate measured in "HTTP Throughput" test
   scenario.

   Test parameters and test test procedures will be added in the future
   release.

7.7. Concurrent SSL/TLS Connection Capacity

7.7.1. Objective

   Usin encrypted traffic (HTTPS), determine the maximum number of
   concurrent TCP connection that DUT/SUT sustains.

   Test parameters and test test procedures will be added in the future
   release.

7.8. SSL/TLS Handshake Rate

7.8.1. Objective

   Determine the maximum and average SSL/TLS handshake rate supported by
   the DUT/SUT.

Test parameters and test test procedures will be added in the future release.

7.9. HTTPS Transaction Per Second

7.9.1. Objective

Determine maximum and average HTTPS transacton rate supported by the DUT/SUT.

Test parameters and test test procedures will be added in the future release.

7.10. HTTPS Throughput

7.10.1. Objective

Determine the average throughput performance of the DUT/SUT when using HTTPS traffic.

Test parameters and test test procedures will be added in the future release.

7.11. HTTPS Transaction Latency

7.11.1. Objective

Determine the minimum, average and maximum values of HTTPS transaction latency at 80% throughput rate measured in "HTTPS Throughput" test scenario.

Test parameters and test test procedures will be added in the future release.

8. Formal Syntax

9. IANA Considerations

This document makes no request of IANA.

Note to RFC Editor: this section may be removed on publication as an RFC.

10. Security Considerations

Security consideration will be added in the future release.

11. Acknowledgements

   Acknowledgements will be added in the future release.

12. 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>.

Appendix A. An Appendix

   Details about NetSecOPEN traffic mix will be added in next draft
   release.

Authors' Addresses

   Balamuhunthan Balarajah
   EANTC AG
   Salzufer 14
   Berlin 10587
   Germany

   Email: balarajah@eantc.de

   Carsten Rossenhoevel
   EANTC AG
   Salzufer 14
   Berlin 10587
   Germany

   Email: cross@eantc.de

          Benchmarking Methodology for Network Security Device Performance
                  draft-balarajah-bmwg-ngfw-performance-05

Abstract

   This document provides benchmarking terminology and methodology for
   next-generation network security devices including next-generation
   firewalls (NGFW), intrusion detection and prevention solutions (IDS/
   IPS) and unified threat management (UTM) implementations. The
   document aims to strongly improve the applicability, reproducibility,
   and transparency of benchmarks and to align the test methodology with
   today's increasingly complex layer 7 application use cases. The main
   areas covered in this document are test terminology, traffic profiles
   and benchmarking methodology for NGFWs to start with.

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 April 17, 2019.

Table of Contents

1. Introduction

   15 years have passed since IETF recommended test methodology and
   terminology for firewalls initially ([RFC2647], [RFC3511]). The
   requirements for network security element performance and
   effectiveness have increased tremendously since then. Security
   function implementations have evolved to more advanced areas and have
   diversified into intrusion detection and prevention, threat
   management, analysis of encrypted traffic, etc. In an industry of

growing importance, well-defined and reproducible key performance
indicators (KPIs) are increasingly needed: They enable fair and
reasonable comparison of network security functions. All these
reasons have led to the creation of a new next-generation firewall
benchmarking document.

2. Requirements

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. Scope

This document provides testing terminology and testing methodology
for next-generation firewalls and related security functions. It
covers two main areas: Performance benchmarks and security
effectiveness testing. The document focuses on advanced, realistic,
and reproducible testing methods. Additionally, it describes test
bed environments, test tool requirements and test result formats.

4. Test Setup

Test setup defined in this document is applicable to all benchmarking
test scenarios described in Section 7.

4.1. Testbed Configuration

Testbed configuration MUST ensure that any performance implications
that are discovered during the benchmark testing aren't due to the
inherent physical network limitations such as number of physical
links and forwarding performance capabilities (throughput and
latency) of the network devise in the testbed. For this reason, this
document recommends avoiding external devices such as switch and
router in the testbed as possible.

However, in the typical deployment, the security devices (DUT/SUT)
are connected to routers and switches which will reduce the number of
entries in MAC or ARP tables of the DUT/SUT. If MAC or ARP tables
have many entries, this may impact the actual DUT/SUT performance due
to MAC and ARP/ND table lookup processes. Therefore, it is
RECOMMENDED to connect Layer 3 device(s) between test equipment and
DUT/SUT as shown in Figure 1.

If the test equipment is capable to emulate layer 3 routing
functionality and there is no need for test equipment ports
aggregation, it is RECOMMENDED to configure the test setup as shown
in Figure 2.

```
+------------------+      +----------+      +------------------+
|Aggregation Switch/|      |          |      | Aggregation Switch/|
| Router           +------+ DUT/SUT  +------+ Router           |
|                  |      |          |      |                  |
+----------+-------+      +----------+      +--------+---------+
           |                                         |
           |                                         |
+----------+----------+              +----------+----------+
|          |          |              |          |          |
| +------------------+ |              | +------------------+ |
| | Emulated Router(s)| |              | | Emulated Router(s)| |
| |    (Optional)    | |              | |    (Optional)    | |
| +------------------+ |              | +------------------+ |
| +------------------+ |              | +------------------+ |
| |     Clients      | |              | |     Servers      | |
| +------------------+ |              | +------------------+ |
|                     |              |                     |
|   Test Equipment    |              |   Test Equipment    |
+---------------------+              +---------------------+
```

Figure 1: Testbed Setup - Option 1

```
+---------------------+      +----------+      +---------------------+
| +------------------+ |      |          |      | +------------------+ |
| | Emulated Router(s)| |      |          |      | | Emulated Router(s)| |
| |    (Optional)    | +------ DUT/SUT  +------+ |    (Optional)    | |
| +------------------+ |      |          |      | +------------------+ |
| +------------------+ |      +----------+      | +------------------+ |
| |     Clients      | |                        | |     Servers      | |
| +------------------+ |                        | +------------------+ |
|                     |                        |                     |
|   Test Equipment    |                        |   Test Equipment    |
+---------------------+                        +---------------------+
```
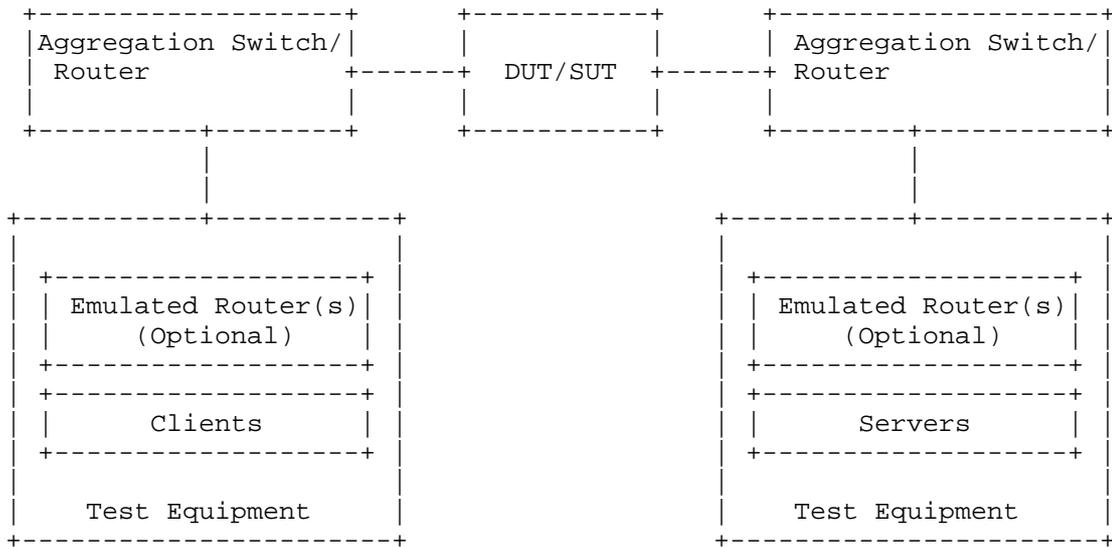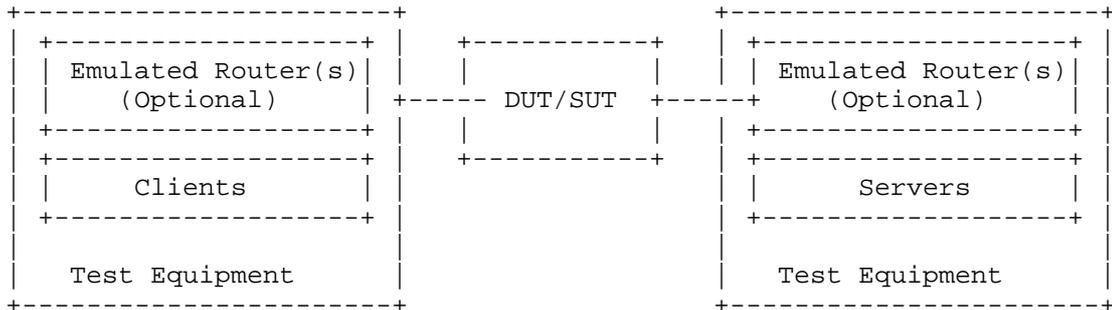
Figure 2: Testbed Setup - Option 2

4.2. DUT/SUT Configuration

   A unique DUT/SUT configuration MUST be used for all benchmarking
   tests described in Section 7. Since each DUT/SUT will have their own
   unique configuration, testers SHOULD configure their device with the
   same parameters that would be used in the actual deployment of the
   device or a typical deployment. Users MUST enable security features
   on the DUT/SUT to achieve maximum security coverage for a specific
   deployment scenario.

   This document attempts to define the recommended security features
   which SHOULD be consistently enabled for all the benchmarking tests

described in Section 7. The table 1 below describes the RECOMMENDED
sets of feature list which SHOULD be configured on the DUT/SUT.

Based on customer use case, user can take a decision to enable or
disable SSL inspection feature for "Throughput Performance with
NetSecOPEN Traffic Mix" test scenario described in Section 7.1

To improve repeatability, a summary of the DUT configuration
including description of all enabled DUT/SUT features MUST be
published with the benchmarking results.

	NGFW
DUT Features	Feature	Included in initial Scope	Added to future Scope
SSL Inspection	x	x	
IDS/IPS	x	x	
Web Filtering	x		x
Antivirus	x	x	
Anti Spyware	x	x	
Anti Botnet	x	x	
DLP	x		x
DDoS	x		x
Certificate Validation	x		x
Logging and Reporting	x	x	
Application Identification	x	x	

Table 1: DUT/SUT Feature List

In summary, DUT/SUT SHOULD be configured as follows:

o All security inspection enabled

o Disposition of all traffic is logged - Logging to an external
   device is permissible

o CVEs matching the following characteristics when serving the NVD

   * CVSS Version: 2

   * CVSS V2 Metrics: AV:N/Au:N/I:C/A:C

   * AV=Attack Vector, Au=Authentication, I=Integrity and
      A=Availability

   * CVSS V2 Severity: High (7-10)

   * If doing a group test the published start date and published
      end date should be the same

o Geographical location filtering and Application Identification and
   Control configured to be triggered based on a site or application
   from the defined traffic mix

In addition, it is also RECOMMENDED to configure a realistic number
of access policy rules on the DUT/SUT. This document determines the
number of access policy rules for three different class of DUT/SUT.
The classification of the DUT/SUT MAY be based on its maximum
supported firewall throughput performance number defined in the
vendor data sheet. This document classifies the DUT/SUT in three
different categories; namely small, medium, and maximum.

The RECOMMENDED throughput values for the following classes are:

Extra Small (XS) - supported throughput less than 1Gbit/s

Small (S) - supported throughput less than 5Gbit/s

Medium (M) - supported throughput greater than 5Gbit/s and less than
10Gbit/s

Large (L) - supported throughput greater than 10Gbit/s

The access rule defined in the table 2 MUST be configured from top to
bottom in correct order shown in the table. The configured access
policy rule MUST NOT block the test traffic used for the benchmarking
test scenarios.

| | | | | UD/SUT lCssification #ules | | | |
Rules Type	Match Criteria	Description	Action	XS	S	M	L
Application layer	Application	Any application traffic NOT included in the test traffic	block	5	10	20	50
Transport layer	Src IP and TCP/UDP Dst ports	Any src IP use in the test AND any dst ports NOT used in the test traffic	block	25	50	100	250
IP layer	Src/Dst IP	Any src/dst IP NOT used in the test	block	25	50	100	250
Application layer	Application	Applications included in the test traffic	allow	10	10	10	10
Transport layer	Src IP and TCP/UDP Dst ports	Half of the src IP used in the test AND any dst ports used in the test traffic. One rule per subnet	allow	1	1	1	1
IP layer	Src IP	The rest of the src IP subnet range used in the test. One rule per subnet	allow	1	1	1	1

Table 2: DUT/SUT Access List

4.3. Test Equipment Configuration

   In general, test equipment allows configuring parameters in different
   protocol level. These parameters thereby influencing the traffic
   flows which will be offered and impacting performance measurements.

This document specifies common test equipment configuration
parameters applicable for all test scenarios defined in Section 7.
Any test scenario specific parameters are described under test setup
section of each test scenario individually.

4.3.1. Client Configuration

This section specifies which parameters SHOULD be considered while
configuring clients using test equipment. Also, this section
specifies the recommended values for certain parameters.

4.3.1.1. TCP Stack Attributes

The TCP stack SHOULD use a TCP Reno variant, which include congestion
avoidance, back off and windowing, retransmission, and recovery on
every TCP connection between client and server endpoints. The
default IPv4 and IPv6 MSS segments size MUST be set to 1460 bytes and
1440 bytes respectively and a TX and RX receive windows of 32768
bytes. Client initial congestion window MUST NOT exceed 10 times the
MSS. Delayed ACKs are permitted and the maximum client delayed Ack
MUST NOT exceed 10 times the MSS before a forced ACK. Up to 3
retries SHOULD be allowed before a timeout event is declared. All
traffic MUST set the TCP PSH flag to high. The source port range
SHOULD be in the range of 1024 - 65535. Internal timeout SHOULD be
dynamically scalable per RFC 793. Client SHOULD initiate and close
TCP connections. TCP connections MUST be closed via FIN.

4.3.1.2. Client IP Address Space

The sum of the client IP space SHOULD contain the following
attributes. The traffic blocks SHOULD consist of multiple unique,
discontinuous static address blocks. A default gateway is permitted.
The IPv4 ToS byte or IPv6 traffic class should be set to '00' or
'000000' respectively.

The following equation can be used to determine the required total
number of client IP address.

Desired total number of client IP = Target throughput [Mbit/s] /
Throughput per IP address [Mbit/s]

(Idea 1) 6-7 Mbps per IP (e.g. 1,400-1,700 IPs per 10Gbit/s
          throughput)

(Idea 2) 0.1-0.2 Mbps per IP (e.g. 50,000-100,000 IPs per 10Gbit/s
          throughput)

Based on deployment and use case scenario, client IP addresses SHOULD be distributed between IPv4 and IPv6 type. This document recommends using the following ratio(s) between IPv4 and IPv6:

(Idea 1)   100 % IPv4, no IPv6

(Idea 2)   80 % IPv4, 20 % IPv6

(Idea 3)   50 % IPv4, 50 % IPv6

(Idea 4)   0 % IPv4, 100 % IPv6

4.3.1.3. Emulated Web Browser Attributes

The emulated web browser contains attributes that will materially affect how traffic is loaded. The objective is to emulate a modern, typical browser attributes to improve realism of the result set.

For HTTP traffic emulation, the emulated browser MUST negotiate HTTP 1.1. HTTP persistency MAY be enabled depending on test scenario. The browser MAY open multiple TCP connections per Server endpoint IP at any time depending on how many sequential transactions are needed to be processed. Within the TCP connection multiple transactions MAY be processed if the emulated browser has available connections. The browser SHOULD advertise a User-Agent header. Headers MUST be sent uncompressed. The browser SHOULD enforce content length validation.

For encrypted traffic, the following attributes shall define the negotiated encryption parameters. The tests MUST use TLSv1.2 or higher with a record size of 16383, commonly used cipher suite and key strength. Depending on test scenario, Session reuse or ticket resumption MAY be used for subsequent connections to the same Server endpoint IP. The client endpoint MUST send TLS Extension Server Name Indication (SNI) information when opening a security tunnel. Cipher suite and certificate size should be defined in the parameter session of each test scenario.

4.3.2. Backend Server Configuration

This document specifies which parameters should be considerable while configuring emulated backend servers using test equipment.

4.3.2.1. TCP Stack Attributes

The TCP stack SHOULD use a TCP Reno variant, which include congestion avoidance, back off and windowing, retransmission, and recovery on every TCP connection between client and server endpoints. The default IPv4 and IPv6 MSS segment size MUST be set to 1460 bytes and

1440 bytes respectively and a TX and RX receive windows of at least
32768 bytes. Server initial congestion window MUST NOT exceed 10
times the MSS. Delayed ACKs are permitted and the maximum server
delayed Ack MUST NOT exceed 10 times the MSS before a forced ACK. Up
to 3 retries SHOULD be allowed before a timeout event is declared.
All traffic MUST set the TCP PSH flag to high. The source port range
SHOULD be in the range of 1024 - 65535. Internal timeout should be
dynamically scalable per RFC 793.

4.3.2.2. Server Endpoint IP Addressing

The server IP blocks SHOULD consist of unique, discontinuous static
address blocks with one IP per Server Fully Qualified Domain Name
(FQDN) endpoint per test port. The IPv4 ToS byte and IPv6 traffic
class bytes should be set to '00' and '000000' respectively.

4.3.2.3. HTTP / HTTPS Server Pool Endpoint Attributes

The server pool for HTTP SHOULD listen on TCP port 80 and emulate
HTTP version 1.1 with persistence. The server MUST advertise a
server type. For HTTPS server, TLS 1.2 or higher MUST be used with a
record size of 16383 bytes and ticket resumption or Session ID reuse
SHOULD be enabled based on test scenario. The server MUST listen on
port TCP 443. The server shall serve a certificate to the client.
It is REQUIRED that the HTTPS server also check Host SNI information
with the FQDN. Cipher suite and certificate size should be defined
in the parameter section of each test scenario.

4.3.3. Traffic Flow Definition

The section describes the traffic pattern between the client and
server endpoints. At the beginning of the test, the server endpoint
initializes and will be in a ready to accept connection state
including initialization of the TCP stack as well as bound HTTP and
HTTPS servers. When a client endpoint is needed, it will initialize
and be given attributes such as the MAC and IP address. The behavior
of the client is to sweep though the given server IP space,
sequentially generating a recognizable service by the DUT. Thus, a
balanced, mesh between client endpoints and server endpoints will be
generated in a client port server port combination. Each client
endpoint performs the same actions as other endpoints, with the
difference being the source IP of the client endpoint and the target
server IP pool. The client shall use Fully Qualified Domain Names
(FQDN) in Host Headers and for TLS Server Name Indication (SNI).

4.3.3.1. Description of Intra-Client Behavior

   Client endpoints are independent of other clients that are
   concurrently executing. When a client endpoint initiates traffic,
   this section describes how the client steps though different
   services. Once initialized, the client should randomly hold (perform
   no operation) for a few milliseconds to allow for better
   randomization of start of client traffic. The client will then
   either open a new TCP connection or connect to a TCP persistence
   stack still open to that specific server. At any point that the
   service profile may require encryption, a TLS encryption tunnel will
   form presenting the URL request to the server. The server will then
   perform an SNI name check with the proposed FQDN compared to the
   domain embedded in the certificate. Only when correct, will the
   server process the HTTPS response object. The initial response
   object to the server MUST NOT have a fixed size; its size is based on
   benchmarking tests described in Section 7. Multiple additional sub-
   URLs (response objects on the service page) MAY be requested
   simultaneously. This may or may not be to the same server IP as the
   initial URL. Each sub-object will also use a conical FQDN and URL
   path, as observed in the traffic mix used.

4.3.4. Traffic Load Profile

   The loading of traffic is described in this section. The loading of
   a traffic load profile has five distinct phases: Init, ramp up,
   sustain, ramp down, and collection.

   During the Init phase, test bed devices including the client and
   server endpoints should negotiate layer 2-3 connectivity such as MAC
   learning and ARP. Only after successful MAC learning or ARP/ND
   resolution shall the test iteration move to the next phase. No
   measurements are made in this phase. The minimum RECOMMEND time for
   Init phase is 5 seconds. During this phase, the emulated clients
   SHOULD NOT initiate any sessions with the DUT/SUT, in contrast, the
   emulated servers should be ready to accept requests from DUT/SUT or
   from emulated clients.

   In the ramp up phase, the test equipment SHOULD start to generate the
   test traffic. It SHOULD use a set approximate number of unique
   client IP addresses actively to generate traffic. The traffic should
   ramp from zero to desired target objective. The target objective
   will be defined for each benchmarking test. The duration for the
   ramp up phase MUST be configured long enough, so that the test
   equipment does not overwhelm DUT/SUT's supported performance metrics
   namely; connections per second, concurrent TCP connections, and
   application transactions per second. The RECOMMENDED time duration

for the ramp up phase is 180-300 seconds. No measurements are made
in this phase.

In the sustain phase, the test equipment SHOULD continue generating
traffic to constant target value for a constant number of active
client IPs. The RECOMMENDED time duration for sustain phase is 600
seconds. This is the phase where measurements occur.

In the ramp down/close phase, no new connections are established, and
no measurements are made. The time duration for ramp up and ramp
down phase SHOULD be same. The RECOMMENDED duration of this phase is
between 180 to 300 seconds.

The last phase is administrative and will be when the tester merges
and collates the report data.

5. Test Bed Considerations

   This section recommends steps to control the test environment and
   test equipment, specifically focusing on virtualized environments and
   virtualized test equipment.

   1. Ensure that any ancillary switching or routing functions between
       the system under test and the test equipment do not limit the
       performance of the traffic generator. This is specifically
       important for virtualized components (vSwitches, vRouters).

   2. Verify that the performance of the test equipment matches and
       reasonably exceeds the expected maximum performance of the system
       under test.

   3. Assert that the test bed characteristics are stable during the
       entire test session. Several factors might influence stability
       specifically for virtualized test beds, for example additional
       workloads in a virtualized system, load balancing and movement of
       virtual machines during the test, or simple issues such as
       additional heat created by high workloads leading to an emergency
       CPU performance reduction.

   Test bed reference pre-tests help to ensure that the desired traffic
   generator aspects such as maximum throughput and the network
   performance metrics such as maximum latency and maximum packet loss
   are met.

   Once the desired maximum performance goals for the system under test
   have been identified, a safety margin of 10% SHOULD be added for
   throughput and subtracted for maximum latency and maximum packet
   loss.

Test bed preparation may be performed either by configuring the DUT in the most trivial setup (fast forwarding) or without presence of DUT.

6. Reporting

This section describes how the final report should be formatted and presented. The final test report MAY have two major sections; Introduction and result sections. The following attributes SHOULD be present in the introduction section of the test report.

1. The name of the NetSecOPEN traffic mix (see Appendix A) MUST be prominent.

2. The time and date of the execution of the test MUST be prominent.

3. Summary of testbed software and Hardware details

    A. DUT Hardware/Virtual Configuration

        + This section SHOULD clearly identify the make and model of the DUT

        + The port interfaces, including speed and link information MUST be documented.

        + If the DUT is a virtual VNF, interface acceleration such as DPDK and SR-IOV MUST be documented as well as cores used, RAM used, and the pinning / resource sharing configuration. The Hypervisor and version MUST be documented.

        + Any additional hardware relevant to the DUT such as controllers MUST be documented

    B. DUT Software

        + The operating system name MUST be documented

        + The version MUST be documented

        + The specific configuration MUST be documented

    C. DUT Enabled Features

        + Specific features, such as logging, NGFW, DPI MUST be documented

          + Attributes of those featured MUST be documented

          + Any additional relevant information about features MUST be documented

      D. Test equipment hardware and software

          + Test equipment vendor name

          + Hardware details including model number, interface type

          + Test equipment firmware and test application software version

4. Results Summary / Executive Summary

   1. Results should resemble a pyramid in how it is reported, with the introduction section documenting the summary of results in a prominent, easy to read block.

   2. In the result section of the test report, the following attributes should be present for each test scenario.

      a. KPIs MUST be documented separately for each test scenario. The format of the KPI metrics should be presented as described in Section 6.1.

      b. The next level of details SHOULD be graphs showing each of these metrics over the duration (sustain phase) of the test. This allows the user to see the measured performance stability changes over time.

6.1. Key Performance Indicators

   This section lists KPIs for overall benchmarking tests scenarios. All KPIs MUST be measured during the of sustain phase of the traffic load profile described in Section 4.3.4. All KPIs MUST be measured from the result output of test equipment.

   o Concurrent TCP Connections
      This key performance indicator measures the average concurrent open TCP connections in the sustaining period.

   o TCP Connections Per Second
      This key performance indicator measures the average established TCP connections per second in the sustaining period. For "TCP/ HTTP(S) Connection Per Second" benchmarking test scenario, the KPI

is measured average established and terminated TCP connections per
second simultaneously.

o Application Transactions Per Second
   This key performance indicator measures the average successfully
   completed application transactions per second in the sustaining
   period.

o TLS Handshake Rate
   This key performance indicator measures the average TLS 1.2 or
   higher session formation rate within the sustaining period.

o Throughput
   This key performance indicator measures the average Layer 2
   throughput within the sustaining period as well as average packets
   per seconds within the same period. The value of throughput
   SHOULD be presented in Gbit/s rounded to two places of precision
   with a more specific kbps in parenthesis. Optionally, goodput MAY
   also be logged as an average goodput rate measured over the same
   period. Goodput result SHALL also be presented in the same format
   as throughput.

o URL Response time / Time to Last Byte (TTLB)
   This key performance indicator measures the minimum, average and
   maximum per URL response time in the sustaining period. The
   latency is measured at Client and in this case would be the time
   duration between sending a GET request from Client and the
   receival of the complete response from the server.

o Application Transaction Latency
   This key performance indicator measures the minimum, average and
   maximum the amount of time to receive all objects from the server.
   The value of application transaction latency SHOULD be presented
   in millisecond rounded to zero decimal.

o Time to First Byte (TTFB)
   This key performance indicator will measure minimum, average and
   maximum the time to first byte. TTFB is the elapsed time between
   sending the SYN packet from the client and receiving the first
   byte of application date from the DUT/SUT. TTFB SHOULD be
   expressed in millisecond.

7. Benchmarking Tests

7.1. Throughput Performance With NetSecOPEN Traffic Mix

7.1.1. Objective

   Using NetSecOPEN traffic mix, determine the maximum sustainable
   throughput performance supported by the DUT/SUT. (see Appendix A for
   details about traffic mix)

7.1.2. Test Setup

   Test bed setup MUST be configured as defined in Section 4. Any test
   scenario specific test bed configuration changes MUST be documented.

7.1.3. Test Parameters

   In this section, test scenario specific parameters SHOULD be defined.

7.1.3.1. DUT/SUT Configuration Parameters

   DUT/SUT parameters MUST conform to the requirements defined in
   Section 4.2. Any configuration changes for this specific test
   scenario MUST be documented.

   This test scenario is RECOMMENDED to perform twice; one with SSL
   inspection feature enabled and the second scenario with SSL
   inspection feature disabled on the DUT/SUT.

7.1.3.2. Test Equipment Configuration Parameters

   Test equipment configuration parameters MUST conform to the
   requirements defined in Section 4.3. Following parameters MUST be
   noted for this test scenario:

      Client IP address range defined in Section 4.3.1.2

      Server IP address range defined in Section 4.3.2.2

      Traffic distribution ratio between IPv4 and IPv6 defined in
      Section 4.3.1.2

      Traffic load objective or specification type (e.g. Throughput,
      SimUsers and etc.)

      Target throughput: It can be defined based on requirements.
      Otherwise it represents aggregated line rate of interface(s) used
      in the DUT/SUT

      Initial throughput: 10% of the "Target throughput"

7.1.3.3. Traffic Profile

   Traffic profile: Test scenario MUST be run with a single application
   traffic mix profile (see Appendix A for details about traffic mix).
   The name of the NetSecOPEN traffic mix MUST be documented.

7.1.3.4. Test Results Acceptance Criteria

   The following test Criteria is defined as test results acceptance
   criteria. Test results acceptance criteria MUST be monitored during
   the whole sustain phase of the traffic load profile.

   a. Number of failed Application transaction MUST be less than 0.01%
       of total attempt transactions

   b. Number of Terminated TCP connections due to unexpected TCP RST
       sent by DUT/SUT MUST be less than 0.01% of total initiated TCP
       connections

   c. Maximum deviation (max. dev) of application transaction time or
       TTLB (Time To Last Byte) MUST be less than X (The value for "X"
       will be finalized and updated after completion of PoC test)
       The following equation MUST be used to calculate the deviation of
       application transaction latency or TTLB
       max. dev = max((avg_latency - min_latency),(max_latency -
       avg_latency)) / (Initial latency)
       Where, the initial latency is calculated using the following
       equation. For this calculation, the latency values (min', avg'
       and max') MUST be measured during test procedure step 1 as
       defined in Section 7.1.4.1.
       The variable latency represents application transaction latency
       or TTLB.
       Initial latency:= min((avg' latency - min' latency) | (max'
       latency - avg' latency))

   d. Maximum value of Time to First Byte MUST be less than X

7.1.3.5. Measurement

   Following KPI metrics MUST be reported for this test scenario.

   Mandatory KPIs: average Throughput, average Concurrent TCP
   connections, TTLB/application transaction latency (minimum, average
   and maximum) and average application transactions per second

   Optional KPIs: average TCP connections per second, average TLS
   handshake rate and TTFB

7.1.4. Test Procedures and expected Results

   The test procedures are designed to measure the throughput
   performance of the DUT/SUT at the sustaining period of traffic load
   profile. The test procedure consists of three major steps.

7.1.4.1. Step 1: Test Initialization and Qualification

   Verify the link status of the all connected physical interfaces. All
   interfaces are expected to be "UP" status.

   Configure traffic load profile of the test equipment to generate test
   traffic at "initial throughput" rate as described in the parameters
   section. The test equipment SHOULD follow the traffic load profile
   definition as described in Section 4.3.4. The DUT/SUT SHOULD reach
   the "initial throughput" during the sustain phase. Measure all KPI
   as defined in Section 7.1.3.5. The measured KPIs during the sustain
   phase MUST meet acceptance criteria "a" and "b" defined in
   Section 7.1.3.4.

   If the KPI metrics do not meet the acceptance criteria, the test
   procedure MUST NOT be continued to step 2.

7.1.4.2. Step 2: Test Run with Target Objective

   Configure test equipment to generate traffic at "Target throughput"
   rate defined in the parameter table. The test equipment SHOULD
   follow the traffic load profile definition as described in
   Section 4.3.4. The test equipment SHOULD start to measure and record
   all specified KPIs. The frequency of KPI metric measurements MUST be
   less than 5 seconds. Continue the test until all traffic profile
   phases are completed.

   The DUT/SUT is expected to reach the desired target throughput during
   the sustain phase. In addition, the measured KPIs MUST meet all
   acceptance criteria. Follow the step 3, if the KPI metrics do not
   meet the acceptance criteria.

7.1.4.3. Step 3: Test Iteration

   Determine the maximum and average achievable throughput within the
   acceptance criteria. Final test iteration MUST be performed for the
   test duration defined in Section 4.3.4.

7.2. TCP/HTTP Connections Per Second

7.2.1. Objective

   Using HTTP traffic, determine the maximum sustainable TCP connection
   establishment rate supported by the DUT/SUT under different
   throughput load conditions.

   To measure connections per second, test iterations MUST use different
   fixed HTTP response object sizes defined in the test equipment
   configuration parameters section 7.2.3.2.

7.2.2. Test Setup

   Test bed setup SHOULD be configured as defined in section 4. Any
   specific test bed configuration changes such as number of interfaces
   and interface type, etc. MUST be documented.

7.2.3. Test Parameters

   In this section, test scenario specific parameters SHOULD be defined.

7.2.3.1. DUT/SUT Configuration Parameters

   DUT/SUT parameters MUST conform to the requirements defined in the
   section 4.2. Any configuration changes for this specific test
   scenario MUST be documented.

7.2.3.2. Test Equipment Configuration Parameters

   Test equipment configuration parameters MUST conform to the
   requirements defined in the section 4.3. Following parameters MUST
   be documented for this test scenario:

   Client IP address range defined in Section 4.3.1.2

   Server IP address range defined in Section 4.3.2.2

   Traffic distribution ratio between IPv4 and IPv6 defined in
   Section 4.3.1.2

   Target connections per second: Initial value from product data sheet
   (if known)

   Initial connections per second: 10% of "Target connections per
   second"

The client SHOULD negotiate HTTP 1.1 and close the connection with FIN immediately after completion of one transaction. In each test iteration, client MUST send GET command requesting a fixed HTTP response object size.

The RECOMMENDED response object sizes are 1, 2, 4, 16, 64 KByte

7.2.3.3. Test Results Acceptance Criteria

The following test Criteria is defined as test results acceptance criteria. Test results acceptance criteria MUST be monitored during the whole sustain phase of the traffic load profile.

a. Number of failed Application transaction MUST be less than 0.01% of total attempt transactions

b. Number of Terminated TCP connections due to unexpected TCP RST sent by DUT/SUT MUST be less than 0.01% of total initiated TCP connections

c. During the sustain phase, traffic should be forwarded at a constant rate

d. Concurrent TCP connections SHOULD be constant during steady state. The deviation of concurrent TCP connections MUST be less than 10%. This confirms that DUT open and close the TCP connections almost at the same rate

7.2.3.4. Measurement

Following KPI metrics MUST be reported for each test iteration.

Mandatory KPIs: average TCP connections per second, average Throughput and Average Time to First Byte (TTFB).

7.2.4. Test Procedures and Expected Results

The test procedure is designed to measure the TCP connections per second rate of the DUT/SUT at the sustaining period of traffic load profile. The test procedure consists of three major steps. This test procedure MAY be repeated multiple times with different IP types; IPv4 only, IPv6 only and IPv4 and IPv6 mixed traffic distribution.

7.2.4.1. Step 1: Test Initialization and Qualification

   Verify the link status of the all connected physical interfaces. All
   interfaces are expected to be "UP" status.

   Configure traffic load profile of the test equipment to establish
   "initial connections per second" as defined in the parameters
   section. The traffic load profile SHOULD be defined as described in
   the section 4.3.4.

   The DUT/SUT SHOULD reach the "initial connections per second" before
   the sustain phase. The measured KPIs during the sustain phase MUST
   meet the acceptance criteria a, b, c, and d defined in section
   7.3.3.3.

   If the KPI metrics do not meet the acceptance criteria, the test
   procedure MUST NOT be continued to "Step 2".

7.2.4.2. Step 2: Test Run with Target Objective

   Configure test equipment to establish "Target connections per second"
   defined in the parameters table. The test equipment SHOULD follow
   the traffic load profile definition as described in the section
   4.3.4.

   During the ramp up and sustain phase of each test iteration, other
   KPIs such as throughput, concurrent TCP connections and application
   transactions per second MUST NOT reach to the maximum value the DUT/
   SUT can support. The test results for specific test iterations
   SHOULD NOT be reported, if the above mentioned KPI (especially
   throughput) reaches to the maximum value. (Example: If the test
   iteration with 64Kbyte of HTTP response object size reached the
   maximum throughput limitation of the DUT, the test iteration MAY be
   interrupted and the result for 64kbyte SHOULD NOT be reported).

   The test equipment SHOULD start to measure and record all specified
   KPIs. The frequency of measurement MUST be less than 5 seconds.
   Continue the test until all traffic profile phases are completed.

   The DUT/SUT is expected to reach the desired target connections per
   second rate at the sustain phase. In addition, the measured KPIs
   MUST meet all acceptance criteria.

   Follow the step 3, if the KPI metrics do not meet the acceptance
   criteria.

7.2.4.3. Step 3: Test Iteration

   Determine the maximum and average achievable connections per second
   within the acceptance criteria.

7.3. HTTP Transaction per Second

7.3.1. Objective

   Using HTTP 1.1 traffic, determine the maximum sustainable HTTP
   transactions per second supported by the DUT/SUT under different
   throughput load conditions.

   To measure transactions per second performance under a variety of DUT
   Security inspection load conditions, each test iteration MUST use
   different fixed HTTP response object sizes defined in the test
   equipment configuration parameters section 7.3.3.2.

7.3.2. Test Setup

   Test bed setup SHOULD be configured as defined in section 4. Any
   specific test bed configuration changes such as number of interfaces
   and interface type, etc. MUST be documented.

7.3.3. Test Parameters

   In this section, test scenario specific parameters SHOULD be defined.

7.3.3.1. DUT/SUT Configuration Parameters

   DUT/SUT parameters MUST conform to the requirements defined in
   section 4.2. Any configuration changes for this specific test
   scenario MUST be documented.

7.3.3.2. Test Equipment Configuration Parameters

   Test equipment configuration parameters MUST conform to the
   requirements defined in the section 4.3. Following parameters MUST
   be documented for this test scenario:

   Client IP address range defined in Section 4.3.1.2

   Server IP address range defined in Section 4.3.2.2

   Traffic distribution ratio between IPv4 and IPv6 defined in
   Section 4.3.1.2

Target Transactions per second: Initial value from product data sheet
(if known)

Initial Transactions per second: 10% of "Target Transactions per
second"

Test scenario SHOULD be run with a single traffic profile with
following attributes:

The client MUST negotiate HTTP 1.1 and close the connections with FIN
immediately after completion of 10 transactions. In each test
iteration, client MUST send GET command requesting a fixed HTTP
response object size. The RECOMMENDED object sizes are 1, 16 and 64
KByte

7.3.3.3. Test Results Acceptance Criteria

The following test Criteria is defined as test results acceptance
criteria. Test results acceptance criteria MUST be monitored during
the whole sustain phase of the traffic load profile.

a. Number of failed Application transactions MUST be zero

b. Number of Terminated HTTP connections due to unexpected TCP RST
    sent by DUT/SUT MUST be less than 0.01% of total initiated HTTP
    sessions

c. Traffic should be forwarded at a constant rate

d. Average Time to TCP First Byte MUST be constant and not increase
    more than 10%

e. The deviation of concurrent TCP connection Must be less than 10%

7.3.3.4. Measurement

Following KPI metrics MUST be reported for this test scenario.

average TCP Connections per second, average Throughput, Average Time
to TCP First Byte and average application transaction latency.

7.3.4. Test Procedures and Expected Results

The test procedure is designed to measure the HTTP transactions per
second of the DUT/SUT at the sustaining period of traffic load
profile. The test procedure consists of three major steps. This
test procedure MAY be repeated multiple times with different IP

types; IPv4 only, IPv6 only and IPv4 and IPv6 mixed traffic
distribution.

7.3.4.1. Step 1: Test Initialization and Qualification

Verify the link status of the all connected physical interfaces. All
interfaces are expected to be "UP" status.

Configure traffic load profile of the test equipment to establish
"initial HTTP transactions per second" as defined in the parameters
section. The traffic load profile CAN be defined as described in the
section 4.3.4.

The DUT/SUT SHOULD reach the "initial HTTP transactions per second"
before the sustain phase. The measured KPIs during the sustain phase
MUST meet the acceptance criteria a, b, c, and d defined in section
7.3.3.3.

If the KPI metrics do not meet the acceptance criteria, the test
procedure MUST NOT be continued to "Step 2".

7.3.4.2. Step 2: Test Run with Target Objective

Configure test equipment to establish "Target HTTP transactions per
second" defined in the parameters table. The test equipment SHOULD
follow the traffic load profile definition as described in the
section 4.3.4.

During the ramp up and sustain phase of each test iteration, other
KPIs such as throughput, concurrent TCP connections and connection
per second MUST NOT reach to the maximum value the DUT/SUT can
support. The test results for specific test iterations SHOULD NOT be
reported, if the above mentioned KPI (especially throughput) reaches
to the maximum value. (Example: If the test iteration with 64Kbyte
of HTTP response object size reached the maximum throughput
limitation of the DUT, the test iteration MAY be interrupted and the
result for 64kbyte SHOULD NOT be reported).

The test equipment SHOULD start to measure and record all specified
KPIs. The frequency of measurement MUST be less than 5 seconds.
Continue the test until all traffic profile phases are completed.

The DUT/SUT is expected to reach the desired target HTTP transactions
per second at the sustain phase. In addition, the measured KPIs MUST
meet all acceptance criteria.

Follow the step 3, if the KPI metrics do not meet the acceptance
criteria.

7.3.4.3. Step 3: Test Iteration

   Determine the maximum and average achievable HTTP transactions per
   second within the acceptance criteria. Final test iteration MUST be
   performed for the test duration defined in Section 4.3.4.

7.4. TCP/HTTP Transaction Latency

7.4.1. Objective

   Using HTTP traffic, determine the average HTTP transaction latency
   when DUT is running with sustainable HTTP transactions per second
   supported by the DUT/SUT under different HTTP response object sizes.

   Test iterations MUST be performed with different HTTP response object
   sizes twice, one with a single transaction and the other with
   multiple transactions within a single TCP connection. For
   consistency both single and multiple transaction test needs to be
   configured with HTTP 1.1.

7.4.2. Test Setup

   Test bed setup SHOULD be configured as defined in section 4. Any
   specific test bed configuration changes such as number of interfaces
   and interface type, etc. MUST be documented.

7.4.3. Test Parameters

   In this section, test scenario specific parameters SHOULD be defined.

7.4.3.1. DUT/SUT Configuration Parameters

   DUT/SUT parameters MUST conform to the requirements defined in the
   section 4.2. Any configuration changes for this specific test
   scenario MUST be documented.

7.4.3.2. Test Equipment Configuration Parameters

   Test equipment configuration parameters MUST conform to the
   requirements defined in the section 4.3. Following parameters MUST
   be documented for this test scenario:

   Client IP address range defined in Section 4.3.1.2

   Server IP address range defined in Section 4.3.2.2

   Traffic distribution ratio between IPv4 and IPv6 defined in
   Section 4.3.1.2

Target connections per second:50% of the value measured in test
scenario TCP/HTTP Connections Per Second (Section 7.2)

Initial connections per second: 10% of "Target connections per
second"

HTTP transaction per TCP connection: one test scenario with single
transaction and another scenario with 10 transactions

Test scenario SHOULD be run with a single traffic profile with
following attributes:

To measure application transaction latency with a single connection
per transaction and a single connection with multiple transactions
the tests should run twice:

1st test run: The client MUST negotiate HTTP 1.1 and close the
connection with FIN immediately after completion of the transaction.

2nd test run: The client MUST negotiate HTTP 1.1 and close the
connection after 10 transactions (GET and RESPONSE) within a single
TCP connection.

HTTP 1.1 with GET command requesting a single 1, 16 or 64 Kbyte
objects. For each test iteration, client MUST request a single HTTP
response object size.

7.4.3.3. Test Results Acceptance Criteria

The following test Criteria is defined as test results acceptance
criteria. Test results acceptance criteria MUST be monitored during
the whole sustain phase of the traffic load profile. Ramp up and
ramp down phase SHOULD NOT be considered.

Generica criteria:

a. Number of failed Application transaction MUST be zero.

b. Number of Terminated TCP connection due to unexpected TCP RST
    sent by DUT/SUT MUST be zero.

c. During the sustain phase, traffic should be forwarded at a
    constant rate.

d. During the sustain phase, Average connect time and average
    transaction time MUST be constant and latency deviation SHOULD
    not increase more than 10%.

    e. Concurrent TCP connections should be constant during steady
       state. This confirms the DUT opens and closes TCP connections at
       the same rate.

    f. After ramp up the DUT MUST achieve the target connections per
       second objective defined in the parameter section 7.4.3.2 and it
       remains in that state for the entire test duration (sustain
       phase).

7.4.3.4. Measurement

Following KPI metrics MUST be reported for each test scenario and
HTTP response object sizes separately:

average TCP connections per second and average application
transaction latency needs to be recorded.

All KPI's are measured once the target connections per second
achieves the steady state.

7.4.4. Test Procedures and Expected Results

The test procedure is designed to measure the average application
transaction latencies or TTLB when the DUT is operating close to 50%
of its maximum achievable connections per second. , This test
procedure CAN be repeated multiple times with different IP types
(IPv4 only, IPv6 only and IPv4 and IPv6 mixed traffic distribution),
HTTP response object sizes and single and multiple transactions per
connection scenarios.

7.4.4.1. Step 1: Test Initialization and Qualification

Verify the link status of the all connected physical interfaces. All
interfaces are expected to be "UP" status.

Configure traffic load profile of the test equipment to establish
"initial connections per second" as defined in the parameters
section. The traffic load profile CAN be defined as described in the
section 4.3.4.

The DUT/SUT SHOULD reach the "initial connections per second" before
the sustain phase. The measured KPIs during the sustain phase MUST
meet the acceptance criteria a, b, c, d ,e and f defined in section
7.4.3.3.

If the KPI metrics do not meet the acceptance criteria, the test
procedure MUST NOT be continued to "Step 2".

7.4.4.2. Step 2: Test Run with Target Objective

   Configure test equipment to establish "Target connections per second"
   defined in the parameters table. The test equipment SHOULD follow
   the traffic load profile definition as described in the section
   4.3.4.

   During the ramp up and sustain phase, other KPIs such as throughput,
   concurrent TCP connections and application transactions per second
   MUST NOT reach to the maximum value that DUT/SUT can support. The
   test results for specific test iterations SHOULD NOT be reported, if
   the above mentioned KPI (especially throughput) reaches to the
   maximum value. (Example: If the test iteration with 64Kbyte of HTTP
   response object size reached the maximum throughput limitation of the
   DUT, the test iteration MAY be interrupted and the result for 64kbyte
   SHOULD NOT be reported).

   The test equipment SHOULD start to measure and record all specified
   KPIs. The frequency of measurement MUST be less than 5 seconds.
   Continue the test until all traffic profile phases are completed.
   DUT/SUT is expected to reach the desired target connections per
   second rate at the sustain phase. In addition, the measured KPIs
   must meet all acceptance criteria.

   Follow the step 3, if the KPI metrics do not meet the acceptance
   criteria.

7.4.4.3. Step 3: Test Iteration

   Determine the maximum achievable connections per second within the
   acceptance criteria and measure the latency values.

7.5. HTTP Throughput

7.5.1. Objective

   Determine the throughput for HTTP transactions varying the HTTP
   response object size.

7.5.2. Test Setup

   Test bed setup SHOULD be configured as defined in section 4. Any
   specific test bed configuration changes such as number of interfaces
   and interface type, etc. must be documented.

7.5.3. Test Parameters

   In this section, test scenario specific parameters SHOULD be defined.

7.5.3.1. DUT/SUT Configuration Parameters

   DUT/SUT parameters MUST conform to the requirements defined in the
   section 4.2. Any configuration changes for this specific test
   scenario MUST be documented.

7.5.3.2. Test Equipment Configuration Parameters

   Test equipment configuration parameters MUST conform to the
   requirements defined in the section 4.3. Following parameters MUST
   be documented for this test scenario:

   Client IP address range defined in Section 4.3.1.2

   Server IP address range defined in Section 4.3.2.2

   Traffic distribution ratio between IPv4 and IPv6 defined in
   Section 4.3.1.2

   Target Throughput: Initial value from product data sheet (if known)

   Number of HTTP response object requests (transactions) per
   connection: 10

   HTTP response object size: 16KB, 64KB, 256KB and mixed objects
   defined in the table

Object size (KByte)	Number of requests/ Weight
0.2	1
6	1
8	1
9	1
10	1
25	1
26	1
35	1
59	1
347	1

Table 3: Mixed Objects

7.5.3.3. Test Results Acceptance Criteria

   The following test Criteria is defined as test results acceptance
   criteria. Test results acceptance criteria MUST be monitored during
   the whole sustain phase of the traffic load profile

   a. Number of failed Application transaction MUST be less than 0.01%
       of attempt transaction.

   b. Traffic should be forwarded constantly.

   c. The deviation of concurrent TCP connection Must be less than 10%

   d. The deviation of average HTTP transaction latency MUST be less
       than 10%

7.5.3.4. Measurement

   The KPI metrics MUST be reported for this test scenario:

Average Throughput, concurrent connections, and average TCP
connections per second.

7.5.4. Test Procedures and Expected Results

The test procedure is designed to measure HTTP throughput of the DUT/
SUT. The test procedure consists of three major steps. This test
procedure MAY be repeated multiple times with different IPv4 and IPv6
traffic distribution and HTTP response object sizes.

7.5.4.1. Step 1: Test Initialization and Qualification

Verify the link status of the all connected physical interfaces. All
interfaces are expected to be "UP" status.

Configure traffic load profile of the test equipment to establish
"initial throughput" as defined in the parameters section.

The traffic load profile SHOULD be defined as described in
Section 4.3.4. The DUT/SUT SHOULD reach the "initial throughput"
during the sustain phase. Measure all KPI as defined in
Section 7.5.3.4.

The measured KPIs during the sustain phase MUST meet the acceptance
criteria "a" defined in Section 7.5.3.3.

If the KPI metrics do not meet the acceptance criteria, the test
procedure MUST NOT be continued to "Step 2".

7.5.4.2. Step 2: Test Run with Target Objective

The test equipment SHOULD start to measure and record all specified
KPIs. The frequency of measurement MUST be less than 5 seconds.
Continue the test until all traffic profile phases are completed.

The DUT/SUT is expected to reach the desired target throughput at the
sustain phase. In addition, the measured KPIs must meet all
acceptance criteria.

Perform the test separately for each HTTP response object size (16k,
64k, 256k and mixed HTTP response objects).

Follow the step 3, if the KPI metrics do not meet the acceptance
criteria.

7.5.4.3. Step 3: Test Iteration

   Determine the maximum and average achievable throughput within the
   acceptance criteria. Final test iteration MUST be performed for the
   test duration defined in Section 4.3.4.

7.6. Concurrent TCP/HTTP Connection Capacity

7.6.1. Objective

   Determine the maximum number of concurrent TCP connections that DUT/
   SUT sustains when using HTTP traffic.

7.6.2. Test Setup

   Test bed setup SHOULD be configured as defined in Section 4. Any
   specific test bed configuration changes such as number of interfaces
   and interface type, etc. must be documented.

7.6.3. Test Parameters

   In this section, test scenario specific parameters SHOULD be defined.

7.6.3.1. DUT/SUT Configuration Parameters

   DUT/SUT parameters MUST conform to the requirements defined in
   Section 4.2. Any configuration changes for this specific test
   scenario MUST be documented.

7.6.3.2. Test Equipment Configuration Parameters

   Test equipment configuration parameters MUST conform to the
   requirements defined in Section 4.3. Following parameters MUST be
   noted for this test scenario:

      Client IP address range defined in Section 4.3.1.2

      Server IP address range defined in Section 4.3.2.2

      Traffic distribution ratio between IPv4 and IPv6 defined in
      Section 4.3.1.2

      Target concurrent connection: Initial value from product data
      sheet (if known)

      Initial concurrent connection: 10% of "Target concurrent
      connection"

The client must negotiate HTTP 1.1 with persistence and each client MAY open multiple concurrent TCP connections per server endpoint IP.

Each client sends 10 GET commands requesting 1Kbyte HTTP response object in the same TCP connection (10 transactions/TCP connection) and the delay (think time) between the transaction MUST be X seconds. The value for think time (X) MUST be defined to achieve 15% of maximum throughput measured in test scenario 7.5.

The established connections SHOULD remain open until the ramp down phase of the test. During the ramp down phase, all connections should be successfully closed with FIN.

7.6.3.3. Test Results Acceptance Criteria

The following test Criteria is defined as test results acceptance criteria. Test results acceptance criteria MUST be monitored during the whole sustain phase of the traffic load profile.

a. Number of failed Application transaction MUST be zero

b. Number of Terminated TCP connections due to unexpected TCP RST sent by DUT/SUT MUST be less than 0.01% of total initiated TCP connections

c. During the sustain phase, traffic should be forwarded constantly at the rate defined in the parameter section 7.6.3.2

d. During the sustain phase, the maximum deviation (max. dev) of application transaction latency or TTLB (Time To Last Byte) MUST be less than 10%

7.6.3.4. Measurement

Following KPI metrics MUST be reported for this test scenario:

average Throughput, max. Min. Avg. Concurrent TCP connections, TTLB/ application transaction latency (minimum, average and maximum) and average application transactions per second.

7.6.4. Test Procedures and expected Results

The test procedure is designed to measure the concurrent TCP connection capacity of the DUT/SUT at the sustaining period of traffic load profile. The test procedure consists of three major steps. This test procedure MAY be repeated multiple times with different IPv4 and IPv6 traffic distribution.

7.6.4.1. Step 1: Test Initialization and Qualification

   Verify the link status of the all connected physical interfaces. All
   interfaces are expected to be "UP" status.

   Configure test equipment to generate background traffic ad defined in
   section 7.6.3.2. Measure throughput, concurrent TCP connections, and
   TCP connections per second.

   While generating the background traffic, configure another traffic
   profile on the test equipment to establish "initial concurrent TCP
   connections" defined in the section 7.6.3.2. The traffic load
   profile CAN be defined as described in the section Error: Reference
   source not found.

   During the sustain phase, the DUT/SUT SHOULD reach the "initial
   concurrent TCP connections" plus concurrent TCP connections measured
   in background traffic. The measured KPIs during the sustain phase
   MUST meet the acceptance criteria "a" and "b" defined in the section
   Error: Reference source not found

   If the KPI metrics do not meet the acceptance criteria, the test
   procedure MUST NOT be continued to "Step 2".

7.6.4.2. Step 2: Test Run with Target Objective

   Configure test equipment to establish "Target concurrent TCP
   connections" defined in the parameters table. The test equipment
   SHOULD follow the traffic load profile definition as described in
   Section 4.3.4.

   Configure test equipment to establish "Target concurrent TCP
   connections" minus concurrent TCP connections measured in background
   traffic. The test equipment SHOULD follow the traffic load profile
   definition as described in the section Error: Reference source not
   found.

   During the ramp up and sustain phase, the other KPIs such as
   throughput, TCP connections per second and application transactions
   per second MUST NOT reach to the maximum value that the DUT/SUT can
   support.

   The test equipment SHOULD start to measure and record KPIs defined in
   section 7.6.3.4. The frequency of measurement MUST be less than 5
   seconds. Continue the test until all traffic profile phases are
   completed.

The DUT/SUT is expected to reach the desired target concurrent connection at the sustain phase. In addition, the measured KPIs must meet all acceptance criteria.

Follow the step 3, if the KPI metrics do not meet the acceptance criteria.

7.6.4.3. Step 3: Test Iteration

Determine the maximum and average achievable concurrent TCP connections capacity within the acceptance criteria.

7.7. TCP/HTTPS Connections per second

7.7.1. Objective

Using HTTPS traffic, determine the maximum sustainable SSL/TLS session establishment rate supported by the DUT/SUT under different throughput load conditions.

Test iterations MUST include common cipher suites and key strengths as well as forward looking stronger keys. Specific test iterations MUST include ciphers and keys defined in the parameter section 7.7.3.2

For each cipher suite and key strengths, test iterations MUST use a single HTTPS response object size defined in the test equipment configuration parameters section 7.7.3.2 to measure connections per second performance under a variety of DUT Security inspection load conditions.

7.7.2. Test Setup

Test bed setup SHOULD be configured as defined in section 4. Any specific test bed configuration changes such as number of interfaces and interface type, etc. must be documented.

7.7.3. Test Parameters

In this section, test scenario specific parameters SHOULD be defined.

7.7.3.1. DUT/SUT Configuration Parameters

DUT/SUT parameters MUST conform to the requirements defined in the section 4.2. Any configuration changes for this specific test scenario MUST be documented.

7.7.3.2. Test Equipment Configuration Parameters

   Test equipment configuration parameters MUST conform to the
   requirements defined in the section 4.3. Following parameters MUST
   be documented for this test scenario:

   Client IP address range defined in Section 4.3.1.2

   Server IP address range defined in Section 4.3.2.2

   Traffic distribution ratio between IPv4 and IPv6 defined in
   Section 4.3.1.2

   Target connections per second: Initial value from product data sheet
   (if known)

   Initial connections per second: 10% of "Target connections per
   second"

   Ciphers and keys:

   1. ECHDE-ECDSA-AES128-GCM-SHA256 with Prime256v1 (Signature Hash
       Algorithmn: ecdsa_secp256r1_sha256 and Supported group:
       sepc256r1)

   2. ECDHE-RSA-AES128-GCM-SHA256 with RSA 2048 (Signature Hash
       Algorithmn: rsa_pkscs1_sha256 and Supported group: sepc256)

   3. ECDHE-ECDSA-AES256-GCM-SHA384 with Secp521 (Signature Hash
       Algorithmn: ecdsa_secp256r1_sha384 and Supported group:
       sepc521r1)

   4. ECDHE-RSA-AES256-GCM-SHA384 with RSA 4096 (Signature Hash
       Algorithmn: rsa_pkcs1_sha384 and Supported group: secp256)

   The client MUST negotiate HTTPS 1.1 and close the connection with FIN
   immediately after completion of one transaction. In each test
   iteration, client MUST send GET command requesting a fixed HTTPS
   response object size. The RECOMMENDED object sizes are 1, 2, 4, 16,
   64 Kbyte.

   Each client connection MUST perform a full handshake with server
   certificate (no Certificate on client side) and MUST NOT use session
   reuse or resumption. TLS record size MAY be optimized for the HTTPS
   response object size up to a record size of 16K.

7.7.3.3. Test Results Acceptance Criteria

The following test Criteria is defined as test results acceptance
criteria:

a. Number of failed Application transaction MUST be less than 0.01%
    of attempt transactions

b. Number of Terminated TCP connections due to unexpected TCP RST
    sent by DUT/SUT MUST be less than 0.01% of total initiated TCP
    connections

c. During the sustain phase, traffic should be forwarded at a
    constant rate

d. Concurrent TCP connections SHOULD be constant during steady
    state. This confirms that DUT open and close the TCP connections
    at the same rate

7.7.3.4. Measurement

Following KPI metrics MUST be reported for this test scenario:

Mandatory KPIs: average TCP connections per second, average
Throughput and Average Time to TCP First Byte.

7.7.4. Test Procedures and expected Results

The test procedure is designed to measure the TCP connections per
second rate of the DUT/SUT at the sustaining period of traffic load
profile. The test procedure consists of three major steps. This
test procedure MAY be repeated multiple times with different IPv4 and
IPv6 traffic distribution.

7.7.4.1. Step 1: Test Initialization and Qualification

Verify the link status of the all connected physical interfaces. All
interfaces are expected to be "UP" status.

Configure traffic load profile of the test equipment to establish
"initial connections per second" as defined in the parameters
section. The traffic load profile CAN be defined as described in the
section 4.3.4.

The DUT/SUT SHOULD reach the "initial connections per second" before
the sustain phase. The measured KPIs during the sustain phase MUST
meet the acceptance criteria a, b, c, and d defined in section
7.7.3.3.

If the KPI metrics do not meet the acceptance criteria, the test
procedure MUST NOT be continued to "Step 2".

7.7.4.2. Step 2: Test Run with Target Objective

Configure test equipment to establish "Target connections per second"
defined in the parameters table. The test equipment SHOULD follow
the traffic load profile definition as described in the section
4.3.4.

During the ramp up and sustain phase, other KPIs such as throughput,
concurrent TCP connections and application transactions per second
MUST NOT reach to the maximum value the DUT/SUT can support. The
test results for specific test iteration SHOULD NOT be reported, if
the above mentioned KPI (especially throughput) reaches to the
maximum value. (Example: If the test iteration with 64Kbyte of HTTPS
response object size reached the maximum throughput limitation of the
DUT, the test iteration can be interrupted and the result for 64kbyte
SHOULD NOT be reported).

The test equipment SHOULD start to measure and record all specified
KPIs. The frequency of measurement MUST be less than 5 seconds.
Continue the test until all traffic profile phases are completed.

The DUT/SUT is expected to reach the desired target connections per
second rate at the sustain phase. In addition, the measured KPIs
must meet all acceptance criteria.

Follow the step 3, if the KPI metrics do not meet the acceptance
criteria.

7.7.4.3. Step 3: Test Iteration

Determine the maximum and average achievable connections per second
within the acceptance criteria.

7.8. HTTPS Transaction per Second

7.8.1. Objective

Using HTTPS traffic, determine the maximum sustainable HTTPS
transactions per second supported by the DUT/SUT under different
throughput load conditions.

To measure transactions per second performance under a variety of DUT
Security inspection load conditions, each test iteration MUST use
different fixed HTTPS transaction object sizes defined in the test
equipment configuration parameters section 7.8.3.2.

Test iterations MUST include common cipher suites and key strengths
as well as forward looking stronger keys. Specific test iterations
MUST include the ciphers and keys defined in the parameter section
7.8.3.2.

7.8.2. Test Setup

Test bed setup SHOULD be configured as defined in section 4. Any
specific test bed configuration changes such as number of interfaces
and interface type, etc. must be documented.

7.8.3. Test Parameters

In this section, test scenario specific parameters SHOULD be defined.

7.8.3.1. DUT/SUT Configuration Parameters

DUT/SUT parameters MUST conform to the requirements defined in the
section 4.2. Any configuration changes for this specific test
scenario MUST be documented.

7.8.3.2. Test Equipment Configuration Parameters

Test equipment configuration parameters MUST conform to the
requirements defined in the section 4.3. Following parameters MUST
be documented for this test scenario:

Client IP address range defined in Section 4.3.1.2

Server IP address range defined in Section 4.3.2.2

Traffic distribution ratio between IPv4 and IPv6 defined in
Section 4.3.1.2

Target Transactions per second: Initial value from product data sheet
(if known)

Initial Transactions per second: 10% of "Target Transactions per
second"

Ciphers and keys:

1. ECHDE-ECDSA-AES128-GCM-SHA256 with Prime256v1 (Signature Hash
    Algorithmn: ecdsa_secp256r1_sha256 and Supported group:
    sepc256r1)

2. ECDHE-RSA-AES128-GCM-SHA256 with RSA 2048 (Signature Hash
    Algorithmn: rsa_pkscs1_sha256 and Supported group: sepc256)

3. ECDHE-ECDSA-AES256-GCM-SHA384 with Secp521 (Signature Hash
    Algorithmn: ecdsa_secp256r1_sha384 and Supported group:
    sepc521r1)

4. ECDHE-RSA-AES256-GCM-SHA384 with RSA 4096 (Signature Hash
    Algorithmn: rsa_pkcs1_sha384 and Supported group: secp256)

The client MUST negotiate HTTPS 1.1 and close the connection with FIN
immediately after completion of 10 transactions.

HTTPS 1.1 with GET command requesting a single 1, 16 and 64 KByte
objects.

Each client connection MUST perform a full handshake with server
certificate and SHOULD NOT use session reuse or resumption.

TLS record size MAY be optimized for the object size up to a record
size of 16K.

7.8.3.3. Test Results Acceptance Criteria

The following test Criteria is defined as test results acceptance
criteria. Test results acceptance criteria MUST be monitored during
the whole sustain phase of the traffic load profile. Ramp up and
ramp down phase SHOULD NOT be considered.

a. Number of failed Application transactions MUST be zero

b. Number of Terminated HTTP connections due to unexpected TCP RST
    sent by DUT/SUT MUST be less than 0.01% of total initiated HTTP
    sessions

c. Average Time to TCP First Byte MUST be constant and not increase
    more than 10%

d. The deviation of concurrent TCP connection Must be less than 10%

7.8.3.4. Measurement

Following KPI metrics MUST be reported for this test scenario.

average TCP connections per second, average Throughput, Average Time
to TCP First Byte and average application transaction latency.

7.8.4. Test Procedures and Expected Results

   The test procedure is designed to measure the HTTPS transactions per
   second rate of the DUT/SUT at the sustaining period of traffic load
   profile. The test procedure consists of three major steps. This
   test procedure MAY be repeated multiple times with different IPv4 and
   IPv6 traffic distribution, HTTPS response object sizes and ciphers
   and keys.

7.8.4.1. Step 1: Test Initialization and Qualification

   Verify the link status of the all connected physical interfaces. All
   interfaces are expected to be "UP" status.

   Configure traffic load profile of the test equipment to establish
   "initial HTTPS transactions per second" as defined in the parameters
   section. The traffic load profile CAN be defined as described in the
   section 4.3.4.

   The DUT/SUT SHOULD reach the "initial HTTPS transactions per second"
   before the sustain phase. The measured KPIs during the sustain phase
   MUST meet the acceptance criteria a, b, c, and d defined in section
   7.8.3.3.

   If the KPI metrics do not meet the acceptance criteria, the test
   procedure MUST NOT be continued to "Step 2".

7.8.4.2. Step 2: Test Run with Target Objective

   Configure test equipment to establish "Target HTTPS transactions per
   second" defined in the parameters table. The test equipment SHOULD
   follow the traffic load profile definition as described in the
   section 4.3.4.

   During the ramp up and sustain phase of each test iteration, other
   KPIs such as throughput, concurrent TCP connections and connections
   per second MUST NOT reach to the maximum value the DUT/SUT can
   support. The test results for specific test iterations SHOULD NOT be
   reported, if the above mentioned KPI (especially throughput) reaches
   to the maximum value. (Example: If the test iteration with 64Kbyte
   of HTTP response object size reached the maximum throughput
   limitation of the DUT, the test iteration MAY be interrupted and the
   result for 64kbyte SHOULD NOT be reported).

   The test equipment SHOULD start to measure and record all specified
   KPIs. The frequency of measurement MUST be less than 5 seconds.
   Continue the test until all traffic profile phases are completed.

The DUT/SUT is expected to reach the desired target HTTPS transactions per second rate at the sustain phase. In addition, the measured KPIs must meet all acceptance criteria.

Follow the step 3, if the KPI metrics do not meet the acceptance criteria.

7.8.4.3. Step 3: Test Iteration

Determine the maximum and average achievable HTTPS transactions per second within the acceptance criteria. Final test iteration MUST be performed for the test duration defined in Section 4.3.4.

7.9. HTTPS Transaction Latency

7.9.1. Objective

Using HTTPS traffic, determine the average HTTPS transaction latency when DUT is running with sustainable HTTPS transactions per second supported by the DUT/SUT under different HTTPS response object size.

Test iterations MUST be performed with different HTTPS response object sizes twice, one with a single transaction and the other with multiple transactions within a single TCP connection.

7.9.2. Test Setup

Test bed setup SHOULD be configured as defined in section 4. Any specific test bed configuration changes such as number of interfaces and interface type, etc. must be documented.

7.9.3. Test Parameters

In this section, test scenario specific parameters SHOULD be defined.

7.9.3.1. DUT/SUT Configuration Parameters

DUT/SUT parameters MUST conform to the requirements defined in the section 4.2. Any configuration changes for this specific test scenario MUST be documented.

7.9.3.2. Test Equipment Configuration Parameters

Test equipment configuration parameters MUST conform to the requirements defined in the section 4.3. Following parameters MUST be documented for this test scenario:

Client IP address range defined in Section 4.3.1.2

Server IP address range defined in Section 4.3.2.2

Traffic distribution ratio between IPv4 and IPv6 defined in Section 4.3.1.2

Cipher suites and key size: ECDHE-ECDSA-AES256-GCM-SHA384 with Secp521 bits key size (Signature Hash Algorithmn: ecdsa_secp256r1_sha384 and Supported group: sepc521r1)

Target connections per second:50% of the value measured in test scenario TCP/HTTPS Connections per second (Section 7.7)

Initial Transactions per second: 10% of "Target Transactions per second"

HTTPS transaction per connection: one test scenario with a single transaction and another scenario with 10 transactions

Test scenario SHOULD be run with a single traffic profile with following attributes:

To measure application transaction latency with a single connection per transaction and single connection with multiple transactions the tests should run twice:

1st test run: The client MUST negotiate HTTPS 1.1 and close the connection with FIN immediately after completion of the transaction.

2nd test run: The client MUST negotiate HTTPS 1.1 and close the connection after 10 transactions (GET and RESPONSE) within a single TCP connection.

HTTPS 1.1 with GET command requesting a single 1, 16 or 64 Kbyte objects. For each test iteration, client MUST request a single HTTPS response object size.

7.9.3.3. Test Results Acceptance Criteria

The following test Criteria is defined as test results acceptance criteria. Test results acceptance criteria MUST be monitored during the whole sustain phase of the traffic load profile. Ramp up and ramp down phase SHOULD NOT be considered.

Generic creteria:

a. Number of failed Application transactions MUST be zero

   b. Number of Terminated TCP connections due to unexpected TCP RST
       sent by DUT/SUT MUST be zero.

   c. During the sustain phase, traffic should be forwarded at a
       constant rate.

   d. During the sustain phase and average application transaction
       latency MUST be constant and latency deviation SHOULD NOT
       increase more than 10%.

   e. Concurrent TCP connections SHOULD be constant during steady
       state. This confirms the DUT opens and closes the TCP
       connections at the same rate.

   f. After ramp up the DUT MUST achieve the target connections per
       second objective defined in the parameter section and remain in
       that state for the entire duration of the sustain phase.

7.9.3.4. Measurement

   Following KPI metrics MUST be reported for each test scenario and
   HTTPS response object sizes separately:

   average TCP connections per second and average application
   transaction latency or TTLB needs to be recorded.

   All KPI's are measured once the target connections per second
   achieves the steady state.

7.9.4. Test Procedures and Expected Results

   The test procedure is designed to measure average application
   transaction latency or TTLB when the DUT is operating close to 50% of
   its maximum achievable connections per second. , This test procedure
   CAN be repeated multiple times with different IP types (IPv4 only,
   IPv6 only and IPv4 and IPv6 mixed traffic distribution), HTTPS
   response object sizes and single and multiple transactions per
   connection scenarios.

7.9.4.1. Step 1: Test Initialization and Qualification

   Verify the link status of the all connected physical interfaces. All
   interfaces are expected to be "UP" status.

   Configure traffic load profile of the test equipment to establish
   "initial connections per second" as defined in the parameters
   section. The traffic load profile CAN be defined as described in the
   section 4.3.4.

The DUT/SUT SHOULD reach the "initial connections per second" before
the sustain phase. The measured KPIs during the sustain phase MUST
meet the acceptance criteria a, b, c, d ,e and f defined in section
7.4.3.3.

If the KPI metrics do not meet the acceptance criteria, the test
procedure MUST NOT be continued to "Step 2".

7.9.4.2. Step 2: Test Run with Target Objective

Configure test equipment to establish "Target connections per second"
defined in the parameters table. The test equipment SHOULD follow
the traffic load profile definition as described in the section
4.3.4.

During the ramp up and sustain phase, other KPIs such as throughput,
concurrent TCP connections and application transactions per second
MUST NOT reach to the maximum value the DUT/SUT can support.

The test equipment SHOULD start to measure and record all specified
KPIs. The frequency of measurement MUST be less than 5 seconds.
Continue the test until all traffic profile phases are completed.
DUT/SUT is expected to reach the desired target connections per
second rate at the sustain phase. In addition, the measured KPIs
must meet all acceptance criteria.

The DUT/SUT is expected to reach the desired target HTTPS
transactions per second rate at the sustain phase. In addition, the
measured KPIs must meet all acceptance criteria.

Follow the step 3, if the KPI metrics do not meet the acceptance
criteria.

7.9.4.3. Step 3: Test Iteration

Determine the maximum achievable connections per second within the
acceptance criteria and measure the latency values.

7.10. HTTPS Throughput

7.10.1. Objective

Determine the throughput for HTTPS transactions varying the HTTPS
response object size.

Test iterations MUST include common cipher suites and key strengths
as well as forward looking stronger keys. Specific test iterations

MUST include the ciphers and keys defined in the parameter section 7.10.3.2.

7.10.2. Test Setup

Test bed setup SHOULD be configured as defined in section 4. Any specific test bed configuration changes such as number of interfaces and interface type, etc. must be documented.

7.10.3. Test Parameters

In this section, test scenario specific parameters SHOULD be defined.

7.10.3.1. DUT/SUT Configuration Parameters

DUT/SUT parameters MUST conform to the requirements defined in the section 4.2. Any configuration changes for this specific test scenario MUST be documented.

7.10.3.2. Test Equipment Configuration Parameters

Test equipment configuration parameters MUST conform to the requirements defined in the section 4.3. Following parameters MUST be documented for this test scenario:

Client IP address range defined in Section 4.3.1.2

Server IP address range defined in Section 4.3.2.2

Traffic distribution ratio between IPv4 and IPv6 defined in Section 4.3.1.2

Target Throughput: Initial value from product data sheet (if known)

Number of HTPPS response object requests (transactions) per connection: 10

Ciphers and keys:

1. ECHDE-ECDSA-AES128-GCM-SHA256 with Prime256v1 (Signature Hash Algorithmn: ecdsa_secp256r1_sha256 and Supported group: sepc256r1)

2. ECDHE-RSA-AES128-GCM-SHA256 with RSA 2048 (Signature Hash Algorithmn: rsa_pkscs1_sha256 and Supported group: sepc256)

3. ECDHE-ECDSA-AES256-GCM-SHA384 with Secp521 (Signature Hash
    Algorithmn: ecdsa_secp256r1_sha384 and Supported group:
    sepc521r1)

4. ECDHE-RSA-AES256-GCM-SHA384 with RSA 4096 (Signature Hash
    Algorithmn: rsa_pkcs1_sha384 and Supported group: secp256)

HTTPS response object size: 16KB, 64KB, 256KB and mixed object
defined in the table below.

Object size (KByte)	Number of requests/ Weight
0.2	1
6	1
8	1
9	1
10	1
25	1
26	1
35	1
59	1
347	1

Table 4: Mixed Objects

Each client connection MUST perform a full handshake with server
certificate (no Certificate on client side) and 50% of connection
SHOULD use session reuse or resumption.

TLS record size MAY be optimized for the HTTPS response object size
up to a record size of 16K.

7.10.3.3. Test Results Acceptance Criteria

   The following test Criteria is defined as test results acceptance
   criteria. Test results acceptance criteria MUST be monitored during
   the whole sustain phase of the traffic load profile.

   a. Number of failed Application transaction MUST be less than 0.01%
       of attempt transaction.

   b. Traffic should be forwarded constantly.

   c. The deviation of concurrent TCP connection Must be less than 10%

   d. The deviation of average application transaction latency MUST be
       less than 10%

7.10.3.4. Measurement

   The KPI metrics MUST be reported for this test scenario:

   Average Throughput, concurrent connections, and average TCP
   connections per second.

7.10.4. Test Procedures and Expected Results

   The test procedure consists of three major steps. This test
   procedure MAY be repeated multiple times with different IPv4 and IPv6
   traffic distribution and HTTPS response object sizes.

7.10.4.1. Step 1: Test Initialization and Qualification

   Verify the link status of the all connected physical interfaces. All
   interfaces are expected to be "UP" status.

   Configure traffic load profile of the test equipment to establish
   "initial throughput" as defined in the parameters section.

   The traffic load profile should be defined as described in
   Section 4.3.4. The DUT/SUT SHOULD reach the "initial throughput"
   during the sustain phase. Measure all KPI as defined in
   Section 7.10.3.4.

   The measured KPIs during the sustain phase MUST meet the acceptance
   criteria "a" defined in Section 7.10.3.3.

   If the KPI metrics do not meet the acceptance criteria, the test
   procedure MUST NOT be continued to "Step 2".

7.10.4.2. Step 2: Test Run with Target Objective

   The test equipment SHOULD start to measure and record all specified
   KPIs. The frequency of measurement MUST be less than 5 seconds.
   Continue the test until all traffic profile phases are completed.

   The DUT/SUT is expected to reach the desired target throughput at the
   sustain phase. In addition, the measured KPIs must meet all
   acceptance criteria.

   Perform the test separately for each HTTPS response object size (16k,
   64k, 256k and mixed HTTPS response objects).

   Follow the step 3, if the KPI metrics do not meet the acceptance
   criteria.

7.10.4.3. Step 3: Test Iteration

   Determine the maximum and average achievable throughput within the
   acceptance criteria. Final test iteration MUST be performed for the
   test duration defined in Section 4.3.4.

7.11. Concurrent TCP/HTTPS Connection Capacity

7.11.1. Objective

   Determine the maximum number of concurrent TCP connections that DUT/
   SUT sustains when using HTTPS traffic.

7.11.2. Test Setup

   Test bed setup SHOULD be configured as defined in section 4. Any
   specific test bed configuration changes such as number of interfaces
   and interface type, etc. must be documented.

7.11.3. Test Parameters

   In this section, test scenario specific parameters SHOULD be defined.

7.11.3.1. DUT/SUT Configuration Parameters

   DUT/SUT parameters MUST conform to the requirements defined in the
   section 4.2. Any configuration changes for this specific test
   scenario MUST be documented.

7.11.3.2. Test Equipment Configuration Parameters

   Test equipment configuration parameters MUST conform to the
   requirements defined in the section Error: Reference source not
   found. Following parameters MUST be documented for this test
   scenario:

      Client IP address range defined in Section 4.3.1.2

      Server IP address range defined in Section 4.3.2.2

      Traffic distribution ratio between IPv4 and IPv6 defined in
      Section 4.3.1.2

      Cipher suites and key size: ECDHE-ECDSA-AES256-GCM-SHA384 with
      Secp521 bits key size (Signature Hash Algorithmn:
      ecdsa_secp256r1_sha384 and Supported group: sepc521r1)

      Target concurrent connection: Initial value from product data
      sheet (if known)

      Initial concurrent connection: 10% of "Target concurrent
      connection"

      Maximum connections per second during ramp up phase: 50% of
      maximum connections per second measured in test scenario TCP/HTTPS
      Connections per second (Section 7.7)

      Throughput for background traffic: 10% of maximum throughput
      measured in test scenario HTTPS Throughput (Section 7.10)7.10
      using an HTTPS response object size of 16Kbyte with a matching
      cipher and key size to what is being tested in this test

   The client must perform HTTPS transaction with persistence and each
   client can open multiple concurrent TCP connections per server
   endpoint IP.

   Each client sends 10 times of GET commands requesting 1Kbyte HTTPS
   response object in the same TCP connections (10 transactions/TCP
   connection) and the delay (think time) between the transaction MUST
   be X seconds. The value for think time (X) MUST be defined to
   achieve 15% of maximum throughput measured in test scenario 7.10.

   The established connections (except background traffic connection)
   SHOULD remain open until the end phase of the test. During the ramp
   down phase, all connections should be successfully closed with FIN.

7.11.3.3. Test Results Acceptance Criteria

The following test Criteria is defined as test results acceptance
criteria. Test results acceptance criteria MUST be monitored during
the whole sustain phase of the traffic load profile.

a. Number of failed Application transactions MUST be zero.

b. Number of Terminated TCP connections due to unexpected TCP RST
    sent by DUT/SUT MUST be less than 0.01% of total initiated TCP
    connections

c. During the sustain phase, traffic should be forwarded constantly
    at the rate defined in the parameter section 7.11.3.2

d. During the sustain phase, then maximum deviation (max. dev) of
    application transaction latency or TTLB (Time To Last Byte) MUST
    be less than 10%

7.11.3.4. Measurement

Following KPI metrics MUST be reported for this test scenario:

Average Throughput, max. Min. Avg. Concurrent TCP connections, TTLB/
application transaction latency and average application transactions
per second

7.11.4. Test Procedures and expected Results

The test procedure is designed to measure the concurrent TCP
connection capacity of the DUT/SUT at the sustaining period of
traffic load profile. The test procedure consists of three major
steps. This test procedure MAY be repeated multiple times with
different IPv4 and IPv6 traffic distribution.

7.11.4.1. Step 1: Test Initialization and Qualification

Verify the link status of the all connected physical interfaces. All
interfaces are expected to be "UP" status.

Configure test equipment to generate background traffic ad defined in
section 7.3.11.2. Measure throughput, concurrent TCP connections,
and connections per second.

While generating the background traffic, configure another traffic
profile on the test equipment to establish "initial concurrent TCP
connections" defined in the section 7.11.3.2. The traffic load

profile CAN be defined as described in the section Error: Reference source not found

During the sustain phase, the DUT/SUT SHOULD reach the "initial concurrent TCP connections" plus concurrent TCP connections measured in background traffic. The measured KPIs during the sustain phase MUST meet the acceptance criteria "a" and "b" defined in the section Error: Reference source not found

If the KPI metrics do not meet the acceptance criteria, the test procedure MUST NOT be continued to "Step 2".

7.11.4.2. Step 2: Test Run with Target Objective

Configure test equipment to establish "Target concurrent TCP connections" minus concurrent TCP connections measured in background traffic. The test equipment SHOULD follow the traffic load profile definition as described in the section 4.3.4

During the ramp up and sustain phase, the other KPIs such as throughput, TCP connections per second and application transactions per second MUST NOT reach to the maximum value that the DUT/SUT can support.

The test equipment SHOULD start to measure and record KPIs defined in section 7.11.3.4. The frequency of measurement MUST be less than 5 seconds. Continue the test until all traffic profile phases are completed.

The DUT/SUT is expected to reach the desired target concurrent TCP connections at the sustain phase. In addition, the measured KPIs must meet all acceptance criteria.

Follow the step 3, if the KPI metrics do not meet the acceptance criteria.

7.11.4.3. Step 3: Test Iteration

Determine the maximum and average achievable concurrent TCP connections within the acceptance criteria.

8. Formal Syntax

9. IANA Considerations

This document makes no request of IANA.

Note to RFC Editor: this section may be removed on publication as an
RFC.

10. Acknowledgements

Acknowledgements will be added in the future release.

11. Contributors

The authors would like to thank the many people that contributed
their time and knowledge to this effort.

Specifically to the co-chairs of the NetSecOPEN Test Methodology
working group and the NetSecOPEN Security Effectiveness working group
- Alex Samonte, Aria Eslambolchizadeh, Carsten Rossenhoevel and David
DeSanto.

Additionally the following people provided input, comments and spent
time reviewiing the myriad of drafts. If we have missed anyone the
fault is entirely our own. Thanks to - Amritam Putatunda,
Balamuhunthan Balarajah, Brian Monkman, Chris Chapman, Chris Pearson,
Chuck McAuley, David White, Jurrie Van Den Breekel, Michelle Rhines,
Rob Andrews, Samaresh Nair, and Tim Winters.

12. References

12.1. 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>.

12.2. Informative References

[RFC2647] Newman, D., "Benchmarking Terminology for Firewall
           Performance", RFC 2647, DOI 10.17487/RFC2647, August 1999,
           <https://www.rfc-editor.org/info/rfc2647>.

[RFC3511] Hickman, B., Newman, D., Tadjudin, S., and T. Martin,
           "Benchmarking Methodology for Firewall Performance",
           RFC 3511, DOI 10.17487/RFC3511, April 2003,
           <https://www.rfc-editor.org/info/rfc3511>.

Appendix A.   NetSecOPEN Basic Traffic Mix

   A traffic mix for testing performance of next generation firewalls
   MUST scale to stress the DUT based on real-world conditions. In
   order to achieve this the following MUST be included:

   o Clients connecting to multiple different server FQDNs per
      application

   o Clients loading apps and pages with connections and objects in
      specific orders

   o Multiple unique certificates for HTTPS/TLS

   o A wide variety of different object sizes

   o Different URL paths

   o Mix of HTTP and HTTPS

   A traffic mix for testing performance of next generation firewalls
   MUST also facility application identification using different
   detection methods with and without decryption of the traffic. Such
   as:

   o HTTP HOST based application detection

   o HTTPS/TLS Server Name Indication (SNI)

   o Certificate Subject Common Name (CN)

   The mix MUST be of sufficient complexity and volume to render
   differences in individual apps as statistically insignificant. For
   example, changes in like to like apps - such as one type of video
   service vs. another both consist of larger objects whereas one news
   site vs. another both typically have more connections then other apps
   because of trackers and embedded advertising content. To achieve
   sufficient complexity, a mix MUST have:

   o Thousands of URLs each client walks thru

   o Hundreds of FQDNs each client connects to

   o Hundreds of unique certificates for HTTPS/TLS

   o Thousands of different object sizes per client in orders matching
      applications

The following is a description of what a popular application in an
enterprise traffic mix contains.

Table 5 lists the FQDNs, number of transactions and bytes transferred
as an example client interacts with Office 365 Outlook, Word, Excel,
Powerpoint, Sharepoint and Skype.

Office365 FQDN	Bytes	Transaction
===	===	===
r1.res.office365.com	14,056,960	192
s1-word-edit-15.cdn.office.net	6,731,019	22
company1-my.sharepoint.com	6,269,492	42
swx.cdn.skype.com	6,100,027	12
static.sharepointonline.com	6,036,947	41
spoprod-a.akamaihd.net	3,904,250	25
s1-excel-15.cdn.office.net	2,767,941	16
outlook.office365.com	2,047,301	86
shellprod.msocdn.com	1,008,370	11
word-edit.officeapps.live.com	932,080	25
res.delve.office.com	760,146	2
s1-powerpoint-15.cdn.office.net	557,604	3
appsforoffice.microsoft.com	511,171	5
powerpoint.officeapps.live.com	471,625	14
excel.officeapps.live.com	342,040	14
s1-officeapps-15.cdn.office.net	331,343	5
webdir0a.online.lync.com	66,930	15
portal.office.com	13,956	1
config.edge.skype.com	6,911	2

clientlog.portal.office.com	6,608	8	
webdir.online.lync.com	4,343	5	
graph.microsoft.com	2,289	2	
nam.loki.delve.office.com	1,812	5	
login.microsoftonline.com	464	2	
login.windows.net	232	1	

Table 5: Office365

Clients MUST connect to multiple server FQDNs in the same order as
real applications. Connections MUST be made when the client is
interacting with the application and NOT first setup up all
connections. Connections SHOULD stay open per client for subsequent
transactions to the same FQDN similar to how a web browser behaves.
Clients MUST use different URL Paths and Object sizes in orders as
they are observed in real Applications. Clients MAY also setup
multiple connections per FQDN to process multiple transactions in a
sequence at the same time. Table 6 has a partial example sequence of
the Office 365 Word application transactions.

FQDN	URL Path	Object size
company1-my.sharepoint.com	/personal...	23,132
word-edit.officeapps.live.com	/we/WsaUpload.ashx	2
static.sharepointonline.com	/bld/.../blank.js	454
static.sharepointonline.com	/bld/.../ initstrings.js	23,254
static.sharepointonline.com	/bld/.../init.js	292,740
company1-my.sharepoint.com	/ScriptResource...	102,774
company1-my.sharepoint.com	/ScriptResource...	40,329
company1-my.sharepoint.com	/WebResource...	23,063
word-edit.officeapps.live.com	/we/wordeditorframe.	60,657

	aspx...	
static.sharepointonline.com	/bld/_layouts/.../ blank.js	454
s1-word-edit-15.cdn.office.net	/we/s/.../ EditSurface.css	19,201
s1-word-edit-15.cdn.office.net	/we/s/.../ WordEditor.css	221,397
s1-officeapps-15.cdn.office.net	/we/s/.../ Microsoft Ajax.js	107,571
s1-word-edit-15.cdn.office.net	/we/s/.../ wacbootwe.js	39,981
s1-officeapps-15.cdn.office.net	/we/s/.../ CommonIntl.js	51,749
s1-word-edit-15.cdn.office.net	/we/s/.../ Compat.js	6,050
s1-word-edit-15.cdn.office.net	/we/s/.../ Box4Intl.js	54,158
s1-word-edit-15.cdn.office.net	/we/s/.../ WoncaIntl.js	24,946
s1-word-edit-15.cdn.office.net	/we/s/.../ WordEditorIntl.js	53,515
s1-word-edit-15.cdn.office.net	/we/s/.../ WordEditorExp.js	1,978,712
s1-word-edit-15.cdn.office.net	/we/s/.../jSanity.js	10,912
word-edit.officeapps.live.com	/we/OneNote.ashx	145,708

Table 6: Office365 Word Transactions

For application identification the HTTPS/TLS traffic MUST include
realistic Certificate Subject Common Name (CN) data as well as Server
Name Indications. For example, a DUT may detect Facebook Chat
traffic by inspecting the certificate and detecting *.facebook.com in
the certificate subject CN and subsequently detect the word chat in

the FQDN 5-edge-chat.facebook.com and identify traffic on the
connection to be Facebook Chat.

Table 7 includes further examples in SNI and CN pairs for several
FQDNs of Office 365.

Server Name Indication (SNI)	Certificate Subject Common Name (CN)
r1.res.office365.com	*.res.outlook.com
login.windows.net	graph.windows.net
webdir0a.online.lync.com	*.online.lync.com
login.microsoftonline.com	stamp2.login.microsoftonline.com
webdir.online.lync.com	*.online.lync.com
graph.microsoft.com	graph.microsoft.com
outlook.office365.com	outlook.com
appsforoffice.microsoft.com	appsforoffice.microsoft.com

Table 7: Office365 SNI and CN Pairs Examples

NetSecOPEN has provided a reference enterprise perimeter traffic mix
with dozens of applications, hundreds of connections, and thousands
of transactions.

The enterprise perimeter traffic mix consists of 70% HTTPS and 30%
HTTP by Bytes, 58% HTTPS and 42% HTTP by Transactions. By
connections with a single connection per FQDN the mix consists of 43%
HTTPS and 57% HTTP. With multiple connections per FQDN the HTTPS
percentage is higher.

Table 8 is a summary of the NetSecOPEN enterprise perimeter traffic
mix sorted by bytes with unique FQDNs and transactions per
applications.

Application	FQDNs	Transactions	Bytes
Office365	26	558	52,931,947

Box	4	90	23,276,089
Salesforce	6	365	23,137,548
Gmail	13	139	16,399,289
Linkedin	10	206	15,040,918
DailyMotion	8	77	14,751,514
GoogleDocs	2	71	14,205,476
Wikia	15	159	13,909,777
Foxnews	82	499	13,758,899
Yahoo Finance	33	254	13,134,011
Youtube	8	97	13,056,216
Facebook	4	207	12,726,231
CNBC	77	275	11,939,566
Lightreading	27	304	11,200,864
BusinessInsider	16	142	11,001,575
Alexa	5	153	10,475,151
CNN	41	206	10,423,740
Twitter Video	2	72	10,112,820
Cisco Webex	1	213	9,988,417
Slack	3	40	9,938,686
Google Maps	5	191	8,771,873
SpectrumIEEE	7	145	8,682,629
Yelp	9	146	8,607,645
Vimeo	12	74	8,555,960
Wikihow	11	140	8,042,314

Netflix	3	31	7,839,256
Instagram	3	114	7,230,883
Morningstar	30	150	7,220,121
Docusign	5	68	6,972,738
Twitter	1	100	6,939,150
Tumblr	11	70	6,877,200
Whatsapp	3	46	6,829,848
Imdb	16	251	6,505,227
NOAAgov	1	44	6,316,283
IndustryWeek	23	192	6,242,403
Spotify	18	119	6,231,013
AutoNews	16	165	6,115,354
Evernote	3	47	6,063,168
NatGeo	34	104	6,026,344
BBC News	18	156	5,898,572
Investopedia	38	241	5,792,038
Pinterest	8	102	5,658,994
Succesfactors	2	112	5,049,001
AbaJournal	6	93	4,985,626
Pbworks	4	78	4,670,980
NetworkWorld	42	153	4,651,354
WebMD	24	280	4,416,736
OilGasJournal	14	105	4,095,255
Trello	5	39	4,080,182

BusinessWire	5	109	4,055,331
Dropbox	5	17	4,023,469
Nejm	20	190	4,003,657
OilGasDaily	7	199	3,970,498
Chase	6	52	3,719,232
MedicalNews	6	117	3,634,187
Marketwatch	25	142	3,291,226
Imgur	5	48	3,189,919
NPR	9	83	3,184,303
Onelogin	2	31	3,132,707
Concur	2	50	3,066,326
Service-now	1	37	2,985,329
Apple itunes	14	80	2,843,744
BerkeleyEdu	3	69	2,622,009
MSN	39	203	2,532,972
Indeed	3	47	2,325,197
MayoClinic	6	56	2,269,085
Ebay	9	164	2,219,223
UCLAedu	3	42	1,991,311
ConstructionDive	5	125	1,828,428
EducationNews	4	78	1,605,427
BofA	12	68	1,584,851
ScienceDirect	7	26	1,463,951
Reddit	8	55	1,441,909

FoodBusinessNews	5	49	1,378,298
Amex	8	42	1,270,696
Weather	4	50	1,243,826
Wikipedia	3	27	958,935
Bing	1	52	697,514
ADP	1	30	508,654
Grand Total	983	10021	569,819,095

Table 8: Summary of NetSecOPEN Enterprise Perimeter Traffic Mix

Authors' Addresses

Balamuhunthan Balarajah
EANTC AG
Salzufer 14
Berlin 10587
Germany

Email: balarajah@eantc.de

Carsten Rossenhoevel
EANTC AG
Salzufer 14
Berlin 10587
Germany

Email: cross@eantc.de

Network Working Group   F. Andreasen
Internet-Draft   N. Cam-Winget
Intended status: Informational   E. Wang
Expires: January 9, 2020   Cisco Systems
                                                          July 8, 2019

              TLS 1.3 Impact on Network-Based Security
                  draft-camwinget-tls-use-cases-05

Abstract

   Network-based security solutions are used by enterprises, public
   sector, and cloud service providers today in order to both complement
   and enhance host-based security solutions. TLS 1.3 introduces
   several changes to TLS 1.2 with a goal to improve the overall
   security and privacy provided by TLS. However some of these changes
   have a negative impact on network-based security solutions and
   deployments that adopt a multi-layered approach to security. While
   this may be viewed as a feature, there are several real-life use case
   scenarios where the same functionality and security can not be
   offered without such network-based security solutions. In this
   document, we identify the TLS 1.3 changes that may impact such use
   cases.

Status of This Memo

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

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

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

   This Internet-Draft will expire on January 9, 2020.

Copyright Notice

1. Introduction

   Enterprises, public sector, and cloud service providers need to
   defend their information systems from attacks originating from both
   inside and outside their networks. Protection and detection are
   typically done both on end hosts and in the network. Host agents
   have deep visibility on the devices where they are installed, whereas
   the network has broader visibility. With such network and security
   devices in the network, it can provide, among other functions,
   homogenous security controls across heterogenous endpoints, covering
   devices for which no host monitoring is available (which is common
   today and is increasingly so in the Internet of Things). This helps
   protect against unauthorized devices installed by insiders, and
   provides a fallback in case the infection of a host disables its
   security agent. Because of these advantages, network-based security
   mechanisms are widely used. In fact, regulatory standards such as
   NERC CIP [NERCCIP] place strong requirements about network perimeter
   security and its ability to have visibility to provide security
   information to the security management and control systems. At the
   same time, the privacy of employees, customers, and other users must
   be respected by minimizing the collection of personal data and
   controlling access to what data is collected. These imperatives hold
   for both end host and network based security monitoring.

   Network-based security solutions such as Firewalls (FW) and Intrusion
   Prevention Systems (IPS) rely on some level of network traffic
   inspection to implement perimeter-based security policies. In many
   use cases, only the metadata or visible aspects of the network
   traffic is inspected. Depending on the security functions required,
   these middleboxes can either be deployed as traffic monitoring
   devices or active in-line devices. A traffic monitoring middlebox
   may for example perform vulnerability detection, intrusion detection,
   crypto audit, compliance monitoring, etc. An active in-line
   middlebox may for example prevent malware download, block known
   malicious URLs, enforce use of strong ciphers, stop data
   exfiltration, etc. A portion of such security policies require
   clear-text traffic inspection above Layer 4, which becomes
   problematic when traffic is encrypted with Transport Layer Security

(TLS) [RFC5246]. Today, network-based security solutions typically address this problem by becoming a man-in-the-middle (MITM) for the TLS session according to one of the following two scenarios:

1. Outbound Session, where the TLS session originates from a client inside the perimeter towards an entity on the outside

2. Inbound Session, where the TLS session originates from a client outside the perimeter towards an entity on the inside

For the outbound session scenario, MITM is enabled by generating a local root certificate and an accompanying (local) public/private key pair. The local root certificate is installed on the inside entities for which TLS traffic is to be inspected, and the network security device(s) store a copy of the private key. During the TLS handshake, the network security device (hereafter referred to as a middlebox) makes a policy decision on the current TLS session. The policy decision could be pass-through, decrypt, deny, etc. On a "decrypt" policy action, the middlebox becomes a TLS proxy for this TLS session. It modifies the certificate provided by the (outside) server and (re)signs it with the private key from the local root certificate. From here on, the middlebox has visibility into further exchanges between the client and server which enables it to decrypt and inspect subsequent network traffic. Alternatively, based on policy, the middlebox may allow the current session to pass through if the session starts in monitoring mode, and then decrypt the next session from the same client.

For the inbound session scenario, the TLS proxy on the middlebox is configured with a copy of the local servers' certificate(s) and corresponding private key(s). Based on the server certificate presented, the TLS proxy determines the corresponding private key, which again enables the middlebox to gain visibility into further exchanges between the client and server and hence decrypt subsequent network traffic.

To date, there are a number of use case scenarios that rely on the above capabilities when used with TLS 1.2 [RFC5246] or earlier. TLS 1.3 [RFC8446] introduces several changes which prevent a number of these use case scenarios from being satisfied with the types of TLS proxy based capabilities that exist today.

It has been noted, that currently deployed TLS proxies on middleboxes may reduce the security of the TLS connection itself due to a combination of poor implementation and configuration, and they may compromise privacy when decrypting a TLS session. As such, it has been argued that preventing TLS proxies from working should be viewed as a feature of TLS 1.3 and that the proper way of solving these

issues is to solely rely on endpoint (client and server) based
solutions instead. We believe this is an overly constrained view of
the problem that ignores a number of important real-life use case
scenarios that improve the overall security posture. For instance,
it goes against a layered defense approach. We also note that
current endpoint-based TLS proxies suffer from many of the same
security issues as the network-based TLS proxies do [HTTPSintercept].

The purpose of this document is to provide a representative set of
network based security use case scenarios that are impacted by TLS
1.3. For each use case scenario, we highlight the specific aspect(s)
of TLS 1.3 that make the use case problematic with a TLS proxy based
solution.

It should be noted that this document addresses only _security_ use
cases with a focus on identifying the problematic ones. The document
does not offer specific solutions to these as the goal is to describe
how current network security solutions rely on network traffic
inspection to address customer requirements and use cases.

1.1. Requirements notation

In this document, the key words "MUST", "MUST NOT", "REQUIRED",
"SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY",
and "OPTIONAL" are to be interpreted as described in BCP 14, RFC 2119
[RFC2119].

2. TLS 1.3 Change Impact Overview

Aiming to improve its overall security and privacy, TLS 1.3
introduces several changes to TLS 1.2, but some of the changes
present a negative impact on network based security. In this
section, we describe those TLS 1.3 changes and briefly outline some
scenario impacts. We divide the changes into two groups; those that
impact inbound sessions and those that impact outbound sessions.

2.1. Inbound Session Change Impacts

2.1.1. Removal of Static RSA and Diffie-Hellman Cipher Suites

TLS 1.2 supports static RSA and Diffie-Hellman(DH) cipher suites,
which enables the server's private key to be shared with server-side
middleboxes. TLS 1.3 has removed support for these cipher suites in
favor of supporting only ephemeral mode Diffie-Hellman in order to
provide perfect forward secrecy (PFS). As a result of this, it is no
longer possible for a server to share a key with the middlebox a
priori, which in turn implies that the middlebox cannot gain access
to the TLS session data.

   Example scenarios that are impacted by this include network
   monitoring, troubleshooting, compliance, etc.

   For further details (and a suggested solution), please refer to
   [I-D.green-tls-static-dh-in-tls13].

2.2. Outbound Session Change Impacts

2.2.1. Encrypted Server Certificate

   In TLS, the ClientHello message is sent to the server's transport
   address (IP and port). The ClientHello message may include the
   Server Name Indication (SNI) to specify the hostname the client
   wishes to contact. This is useful when multiple "virtual servers"
   are hosted on a given transport address (IP and port). It also
   provides passive observers and security devices information about the
   domain the client is attempting to reach. Note that while SNI is
   optional in TLS 1.2, it is mandatory in TLS 1.3.

   The server replies with a ServerHello message, which contains the
   selected connection parameters, followed by a Certificate message,
   which contains the server's certificate and hence its identity.

   Note that even _if_ the SNI is provided by the client, there is no
   guarantee that the actual server responding is the one indicated in
   the SNI from the client. SNI alone, without comparison of the server
   certificate, does not provide reliable information about the server
   that the client attempts to reach. Where a client has been
   compromised by malware and connects to a command and control server,
   but presents an innocuous SNI to bypass protective filters, it is
   undetectable under TLS 1.3.

   In TLS 1.2, the ClientHello, ServerHello and Certificate messages are
   all sent in clear-text, however in TLS 1.3, the Certificate message
   is encrypted thereby hiding the server identity from any
   intermediary.

   Example scenarios that are impacted by this involve selective network
   security policies on the server, such as whitelists or blacklists
   based on security intelligence, regulatory requirements, categories
   (e.g. financial services), etc. Under TLS 1.3, these scenarios now
   require the middlebox to perform decryption and inspection of every
   connection to have the same information to make policy decisions.
   Further, the middlebox is not able to make the policy decisions
   without actively engaging in the TLS 1.3 session from the beginning
   of the handshake, and it cannot step out of the connection once it
   has been determined to be benign, without dropping the whole
   connection. In TLS 1.2, middleboxes could be more selective in

choosing what connections to engage with, and make decisions based on
the certificate without actively decrypting the connection to access
the certificate(s).

While conformant clients can generate the SNI and check that the
server certificate contains a name matching the SNI, there are non-
conformant clients that do not and some enterprises also require a
level of validation. Thus, from a network infrastructure
perspective, policies to validate SNI against the Server Certificate
can not be validated in TLS 1.3 as the Server certificate is now
obscured to the middlebox. This is an example where the network
infrastructure is using one measure to protect the enterprise from
non-conformant (e.g. evasive) clients and a conformant server. As a
general practice, security functions conduct cross checks and
consistency checks wherever possible to mitigate imperfect or
malicious implementations; even if they are deemed redundant with
fully conformant implementations.

2.2.2. Resumption and Pre-Shared Key

In TLS 1.2 and below, session resumption is provided by "session IDs"
and "session tickets" [RFC5077]. If the server does not want to
honor a ticket, then it can simply initiate a full TLS handshake with
the client as usual.

In TLS 1.3, the above mechanism is replaced by Pre-Shared Keys (PSK),
which can be negotiated as part of an initial handshake and then used
in a subsequent handshake to perform resumption using the PSK. TLS
1.3 states that the client SHOULD include a "key_share" extension to
enable the server to decline resumption and fall back to a full
handshake, however it is not an absolute requirement.

Example scenarios that are impacted by this are middleboxes that were
not part of the initial handshake, and hence do not know the PSK. If
the client does not include the "key_share" extension, the middlebox
cannot force a fallback to the full handshake. If the middlebox
policy requires it to inspect the session, it will have to fail the
connection instead.

Note that in practice though, it is unlikely that clients using
session resumption will not allow for fallback to a full handshake
since the server may treat a ticket as valid for a shorter period of
time that what is stated in the ticket_lifetime [RFC8446]. As long
as the client advertises a supported DH group, the server (or
middlebox) can always send a HelloRetryRequest to force the client to
send a key_share and hence a full handshake.

Clients that truly only support PSK mode of operation (provisioned out of band) will of course not negotiate a new key, however that is not a change in TLS 1.3.

2.2.3. Version Negotiation and Downgrade Protection

In TLS, the ClientHello message includes a list of supported protocol versions. The server will select the highest supported version and indicate its choice in the ServerHello message.

TLS 1.3 changes the way in which version negotiation is performed. The ClientHello message will indicate TLS version 1.3 in the new "supported_versions" extension, however for backwards compatibility with TLS 1.2, the ClientHello message will indicate TLS version 1.2 in the "legacy_version" field. A TLS 1.3 server will recognize that TLS 1.3 is being negotiated, whereas a TLS 1.2 server will simply see a TLS 1.2 ClientHello and proceed with TLS 1.2 negotiation.

In TLS 1.3, the random value in the ServerHello message includes a special value in the last eight bytes when the server negotiates either TLS 1.2 or TLS 1.1 and below. The special value(s) enable a TLS 1.3 client to detect an active attacker launching a downgrade attack when the client did indeed reach a TLS 1.3 server, provided ephemeral ciphers are being used.

From a network security point of view, the primary impact is that TLS 1.3 requires the TLS proxy to be an active man-in-the-middle from the start of the handshake. It is also required that a TLS 1.2 and below middlebox implementation must handle unsupported extensions gracefully, which is a requirement for a conformant middlebox.

2.2.4. SNI Encryption in TLS Through Tunneling

As noted above, with server certificates encrypted, the Server Name Indication (SNI) in the ClientHello message is the only information available in cleartext to indicate the client's targeted server, and the actual server reached may differ.

[I-D.ietf-tls-sni-encryption] proposes to hide the SNI in the ClientHello from middleboxes.

Example scenarios that are impacted by this involve selective network security, such as whitelists or blacklists based on security intelligence, regulatory requirements, categories (e.g. financial services), etc. An added challenge is that some of these scenarios require the middlebox to perform inspection, whereas other scenarios require the middlebox to not perform inspection. Without the SNI,

however, the middlebox may not have the information required to determine the actual scenario before it is too late.

3. Inbound Session Use Cases

In this section we explain how a set of real-life inbound use case scenarios are affected by some of the TLS 1.3 changes.

3.1. Use Case I1 - Data Center Protection

Services deployed in the data center may be offered for access by external and untrusted hosts. Network security functions such as IPS and Web Application Firewall (WAF) are deployed to monitor and control the transactions to these services. While an Application level load balancer is not a security function strictly speaking, it is also an important function that resides in front of these services

These network security functions are usually deployed in two modes: monitoring and inline. In either case, they need to access the L7 and application data such as HTTP transactions which could be protected by TLS encryption. They may monitor the TLS handshakes for additional visibility and control.

The TLS handshake monitoring function will be impacted by TLS 1.3.

For additional considerations on this scenario, see also [I-D.green-tls-static-dh-in-tls13].

3.2. Use Case I2 - Application Operation over NAT

The Network Address Translation (NAT) function translates L3 and L4 addresses and ports as the packet traverses the network device. Sophisticated NAT devices may also implement application inspection engines to correct L3/L4 data embedded in the control messages (e.g., FTP control message, SIP signaling messages) so that they are consistent with the outer L3/L4 headers.

Without the correction, the secondary data (FTP) or media (SIP) connections will likely reach a wrong destination.

The embedded address and port correction operation requires access to the L7 payload which could be protected by encryption.

3.3. Use Case I3 - Compliance

Many regulations exist today that include cyber security requirements requiring close inspection of the information traversing through the network. For example, organizations that require PCI-DSS [PCI-DSS]

compliance must provide the ability to regularly monitor the network
to prevent, detect and minimize impact of a data compromise.
[PCI-DSS] Requirement #2 (and Appendix A2 as it concerns TLS)
describes the need to be able to detect protocol and protocol usage
correctness. Further, [PCI-DSS] Requirement #10 detailing monitoring
capabilities also describe the need to provide network-based audit to
ensure that the protocols and configurations are properly used.

Deployments today still use factory or default credentials and
settings that must be observed, and to meet regulatory compliance,
must be audited, logged and reported. As the server (certificate)
credential is now encrypted in TLS 1.3, the ability to verify the
appropriate (or compliant) use of these credentials are lost, unless
the middlebox always becomes an active MITM.

3.4. Use Case I4 - Crypto Security Audit

Organizations may have policies around acceptable ciphers and
certificates on their servers. Examples include no use of self-
signed certificates, black or white-list Certificate Authority, valid
certificate experitation time, etc. In TLS 1.2, the Certificate
message was sent in clear-text, however in TLS 1.3 the message is
encrypted thereby preventing both a network-based audit and policy
enforcement around acceptable server certificates.

While the audits and policy enforcements could in theory be done on
the servers themselves, the premise of the use case is that not all
servers are configured correctly and hence such an approach is
unlikely to work in practice. A common example where this occurs
includes lab servers.

4. Outbound Session Use Cases

In this section we explain a set of real-life outbound session use
case scenarios. These scenarios remain functional with TLS 1.3
though the TLS proxy's performance is impacted by participating in
the DHE key exchange from the beginning of the handshake. Similarly,
while with TLS 1.2 the handshake packets could be passively
inspected, with TLS 1.3 the TLS proxy may have to perform full
decryption to inspect the certificates or to affect other policies
impacting its performance.

4.1. Use Case O1 - Acceptable Use Policy (AUP)

Enterprises deploy security devices to enforce Acceptable Use Policy
(AUP) according to the IT and workplace policies. The security
devices, such as firewall/next-gen firewall, web proxy and an

application on the endpoints, act as middleboxes to scan traffic in the enterprise network for policy enforcement.

Sample AUP policies are:

o "Employees are not allowed to access 'gaming' websites from enterprise network"

o "Temporary workers are not allowed to use enterprise network to upload video clips to Internet, but are allowed to watch video clips"

Such enforcements are accomplished by controlling the DNS transactions and HTTP transactions. A coarse control can currently be achieved by controlling the DNS response (though this may become infeasible if it is also protected by TLS), however, in many cases, granular control is required at HTTP URL or Method levels, to distinguish a specific web page on a hosting site, or to differentiate between uploading and downloading operations.

The security device requires access to plain text HTTP header for granular AUP control.

4.2. Use Case O2 - Malware and Threat Protection

Enterprises adopt a multi-technology approach when it comes to malware and threat protection for the network assets. This includes solutions deployed on the endpoint, network and cloud.

While endpoint application based solution may be effective, to an extent, at detecting and preventing some types of attack, defense in depth is widely considered to be best security practice because it provides additional protection against compromise of endpoints. For example, network-based solutions can detect malware and threats based on network visibility and provide discovery to a compromised endpoint, even though the logs of such a compromised endpoint appear normal. That is, network based solutions provide such additional detection, prevention and mitigation of attacks with the benefit of rapid and centralized updates.

The network based solutions utilise network traffic for a range of purposes, including but not limited to: preventing malware landing on the endpoint through signatures, detecting abnormal data exfiltration, allowing 0-day analysis and mitigation of successful attacks.".

The security functions require access to clear text HTTP or other application level streams on a needed basis.

4.3. Use Case O3 - IoT Endpoints

   As the Internet of Everything continues to evolve, more and more
   endpoints become connected to the Internet. From a security point of
   view, some of the challenges presented by these are:

   o Constrained devices with limited resources (CPU, memory, battery
      life, etc.)

   o Lack of ability to install and update endpoint protection
      software.

   o Lack of software updates as new vulnerabilities are discovered.

   In short, the security posture of such devices is expected to be
   weak, especially as they get older, and the only way to improve this
   posture is to supplement them with a network-based solution. IoT
   deployments are further challenged in that they host a variety of
   these devices, each with different update cycles and often, are very
   slow to update their software or firmware to ensure availability and
   safe of the environments they operate. This in turn requires network
   based solutions to afford a consistant security baseline. This
   solution can range from selective passive monitoring to a full and
   active MiTM.

4.4. Use Case O4 - Unpatched Endpoints

   New vulnerabilities appear constantly and in spite of many advances
   in recent years in terms of automated software updates, especially in
   reaction to security vulnerabilities, the reality is that a very
   large number of endpoints continue to run versions of software with
   known vulnerabilities.

   In theory, these endpoints should of course be patched, but in
   practice, it is often not done which leaves the endpoint open to the
   vulnerability in question. A network-based security solution can
   look for attempted exploits of such vulnerabilities and stop them
   before they reach the unpatched endpoint.

4.5. Use Case O5 - Rapid Containment of New Vulnerability and Campaigns

   When a new vulnerability is discovered or an attack campaign is
   launched, it is important to patch the vulnerability or contain the
   campaign as quickly as possible. Patches however are not usually
   available immediately for every device on the network, and even when
   they are, most endpoints are in practice not patched immediately,
   which leaves them open to the attack.

A network-based security solution can look for attempted exploits of such new vulnerabilities or recognize an attack being launched based on security intelligence related to the campaign and stop them before they reach the vulnerable endpoint.

4.6. Use Case O6 - End-of-Life Endpoint

Older endpoints (and in some cases even new ones) will not receive any software updates. As new vulnerabilities inevitably are discovered, these endpoints will be permanently vulnerable to exploits without security solutions that are not endpoint-based.

A network-based security solution can help prevent such exploits with the MITM functions.

4.7. Use Case O7 - Compliance

This use case is similar to the inbound compliance use case described in Section 3.3, except its from the client point of view.

4.8. Use Case O8 - Crypto Security Audit

This is a variation of the use case in Section 3.4.

Organizations may have policies around acceptable ciphers and certificates for client sessions, possibly based on the destination. Examples include no use of self-signed certificates, black or white-list Certificate Authority, etc. In TLS 1.2, the Certificate message was sent in clear-text, however in TLS 1.3 the message is encrypted thereby preventing either a network-based audit or policy enforcement around acceptable server certificates.

It is not possible to implement a full security solution by relying on the client alone in this case. For example, in the many cases where the device is not under configuration control of the organisation (i.e. "Bring Your Own Device" devices, which are present in many modern organisations), as audits and policy enforcements can't be done on such clients or on clients that are not properly configured.

5. IANA considerations

This document does not include IANA considerations.

6. Security Considerations

   This document describes existing functionality and use case scenarios
   and as such does not introduce any new security considerations.

7. Acknowledgements

   The authors thank Eric Rescorla, the National Cyber Security Center
   and Dan Wing who provided several comments on technical accuracy and
   middlebox security implications.

8. Change Log

8.1. Version -01

   Updates based on comments from Eric Rescorla.

8.2. Version -03

   Updates based on EKR's comments

9. Version -04

   Updates based on Kirsty's comments

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>.

   [RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security
              (TLS) Protocol Version 1.2", RFC 5246,
              DOI 10.17487/RFC5246, August 2008,
              <https://www.rfc-editor.org/info/rfc5246>.

   [RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol
              Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018,
              <https://www.rfc-editor.org/info/rfc8446>.

10.2. Informative References

   [HTTPSintercept]
              "The Security Impact of HTTPS Interception", n.d.,
              <https://jhalderm.com/pub/papers/interception-ndss17.pdf>.

   [I-D.green-tls-static-dh-in-tls13]
             Green, M., Droms, R., Housley, R., Turner, P., and S.
             Fenter, "Data Center use of Static Diffie-Hellman in TLS
             1.3", draft-green-tls-static-dh-in-tls13-01 (work in
             progress), July 2017.

   [I-D.ietf-tls-sni-encryption]
             Huitema, C. and E. Rescorla, "Issues and Requirements for
             SNI Encryption in TLS", draft-ietf-tls-sni-encryption-04
             (work in progress), November 2018.

   [NERCCIP] "North American Electric Reliability Corporation, (CIP)
             Critical Infrastructure Protection", n.d.,
             <http://www.nerc.com/pa/stand/Pages/ReliabilityStandardsUn
             itedStates.aspx?jurisdiction=United%20States>.

   [PCI-DSS] "Payment Card Industry (PCI): Data Security Standard",
             n.d., <https://www.pcisecuritystandards.org/documents/
             PCI_DSS_v3-2.pdf>.

   [RFC5077] Salowey, J., Zhou, H., Eronen, P., and H. Tschofenig,
             "Transport Layer Security (TLS) Session Resumption without
             Server-Side State", RFC 5077, DOI 10.17487/RFC5077,
             January 2008, <https://www.rfc-editor.org/info/rfc5077>.

Authors' Addresses

   Flemming Andreasen
   Cisco Systems
   111 Wood Avenue South
   Iselin, NJ 08830
   USA

   Email: fandreas@cisco.com

   Nancy Cam-Winget
   Cisco Systems
   3550 Cisco Way
   San Jose, CA 95134
   USA

   Email: ncamwing@cisco.com

Eric Wang
Cisco Systems
3550 Cisco Way
San Jose, CA   95134
USA

Email: ejwang@cisco.com

        Recommendations on the Filtering of IPv6 Packets Containing IPv6
                            Extension Headers
                   draft-ietf-opsec-ipv6-eh-filtering-05

Abstract

   It is common operator practice to mitigate security risks by
   enforcing appropriate packet filtering. This document analyzes both
   the general security implications of IPv6 Extension Headers and the
   specific security implications of each Extension Header and Option
   type. Additionally, it discusses the operational and
   interoperability implications of discarding packets based on the IPv6
   Extension Headers and IPv6 options they contain. Finally, it
   provides advice on the filtering of such IPv6 packets at transit
   routers for traffic *not* directed to them, for those cases in which
   such filtering is deemed as necessary.

Status of this Memo

Table of Contents

1. Introduction

   Recent studies (see e.g. [RFC7872]) suggest that there is widespread
   dropping of IPv6 packets that contain IPv6 Extension Headers (EHs).
   In some cases, such packet drops occur at transit routers. While
   some operators "officially" drop packets that contain IPv6 EHs, it is
   possible that some of the measured packet drops be the result of
   improper configuration defaults, or inappropriate advice in this
   area.

   This document analyzes both the general security implications of IPv6
   EHs and the specific security implications of each EH and Option
   type, and provides advice on the filtering of IPv6 packets based on
   the IPv6 EHs and the IPv6 options they contain. Since various
   protocols may use IPv6 EHs (possibly with IPv6 options), discarding
   packets based on the IPv6 EHs or IPv6 options they contain may have
   implications on the proper functioning of such protocols. Thus, this
   document also attempts to discuss the operational and
   interoperability implications of such filtering policies.

   The filtering policy typically depends on where in the network such
   policy is enforced: when the policy is enforced in a transit network,
   the policy typically follows a "black-list" approach, where only
   packets with clear negative implications are dropped. On the other
   hand, when the policy is enforced closer to the destination systems,
   the policy typically follows a "white-list" approach, where only
   traffic that is expected to be received is allowed. The advice in
   this document is aimed only at transit routers that may need to
   enforce a filtering policy based on the EHs and IPv6 options a packet
   may contain, following a "black-list" approach, and hence is likely
   to be much more permissive that a filtering policy to be employed
   e.g. at the edge of an enterprise network. The advice in this
   document is meant to improve the current situation of the dropping of
   packets with IPv6 EHs in the Internet [RFC7872].

   This document is similar in nature to [RFC7126], which addresses the
   same problem for the IPv4 case. However, in IPv6, the problem space
   is compounded by the fact that IPv6 specifies a number of IPv6 EHs,
   and a number of IPv6 options which may be valid only when included in
   specific EH types.

   This document completes and complements the considerations for
   protecting the control plane from packets containing IP options that
   can be found in [RFC6192].

   Section 2 of this document specifies the terminology and conventions
   employed throughout this document. Section 3 of this document
   discusses IPv6 EHs and provides advice in the area of filtering IPv6

packets that contain such IPv6 EHs. Section 4 of this document discusses IPv6 options and provides advice in the area of filtering IPv6 packets that contain such options.

2. Terminology and Conventions Used in This Document

2.1. Terminology

The terms "fast path", "slow path", and associated relative terms ("faster path" and "slower path") are loosely defined as in Section 2 of [RFC6398].

The terms "permit" (allow the traffic), "drop" (drop with no notification to sender), and "reject" (drop with appropriate notification to sender) are employed as defined in [RFC3871]. Throughout this document we also employ the term "discard" as a generic term to indicate the act of discarding a packet, irrespective of whether the sender is notified of such drops, and irrespective of whether the specific filtering action is logged.

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

2.2. Applicability Statement

This document provides advice on the filtering of IPv6 packets with EHs at transit routers for traffic *not* explicitly destined to such transit routers, for those cases in which such filtering is deemed as necessary.

2.3. Conventions

This document assumes that nodes comply with the requirements in [RFC7045]. Namely (from [RFC7045]),

o If a forwarding node discards a packet containing a standard IPv6 EH, it MUST be the result of a configurable policy and not just the result of a failure to recognise such a header.

o The discard policy for each standard type of EH MUST be individually configurable.

o The default configuration SHOULD allow all standard IPv6 EHs.

The advice provided in this document is only meant to guide an operator in configuring forwarding devices, and is *not* to be

interpreted as advice regarding default configuration settings for
network devices. That is, this document provides advice with respect
to operational configurations, but does not change the implementation
defaults required by [RFC7045].

We recommend that configuration options are made available to govern
the processing of each IPv6 EH type and each IPv6 option type. Such
configuration options may include the following possible settings:

o Permit this IPv6 EH or IPv6 Option type

o Discard (and log) packets containing this IPv6 EH or option type

o Reject (and log) packets containing this IPv6 EH or option type
   (where the packet drop is signaled with an ICMPv6 error message)

o Rate-limit traffic containing this IPv6 EH or option type

o Ignore this IPv6 EH or option type (as if it was not present) and
   forward the packet. We note that if a packet carries forwarding
   information (e.g., in an IPv6 Routing Header) this might be an
   inappropriate or undesirable action.

We note that special care needs to be taken when devices log packet
drops/rejects. Devices should count the number of packets dropped/
rejected, but the logging of drop/reject events should be limited so
as to not overburden device resources.

Finally, we note that when discarding packets, it is generally
desirable that the sender be signaled of the packet drop, since this
is of use for trouble-shooting purposes. However, throughout this
document (when recommending that packets be discarded) we generically
refer to the action as "discard" without specifying whether the
sender is signaled of the packet drop.

3. IPv6 Extension Headers

3.1. General Discussion

   IPv6 [RFC8200] EHs allow for the extension of the IPv6 protocol.
   Since both IPv6 EHs and upper-layer protocols share the same
   namespace ("Next Header" registry/namespace), [RFC7045] identifies
   which of the currently assigned Internet Protocol numbers identify
   IPv6 EHs vs. upper-layer protocols. This document discusses the
   filtering of packets based on the IPv6 EHs (as specified by
   [RFC7045]) they contain.

NOTE: [RFC7112] specifies that non-fragmented IPv6 datagrams and IPv6 First-Fragments MUST contain the entire IPv6 header chain [RFC7112]. Therefore, intermediate systems can enforce the filtering policies discussed in this document, or resort to simply discarding the offending packets when they fail to comply with the requirements in [RFC7112]. We note that, in order to implement filtering rules on the fast path, it may be necessary for the filtering device to limit the depth into the packet that can be inspected before giving up. In circumstances where there is such a limitation, it is recommended that implementations discard packets if, when trying to determine whether to discard or permit a packet, the aforementioned limit is encountered.

3.2. General Security Implications

In some specific device architectures, IPv6 packets that contain IPv6 EHs may cause the corresponding packets to be processed on the slow path, and hence may be leveraged for the purpose of Denial of Service (DoS) attacks [I-D.gont-v6ops-ipv6-ehs-packet-drops] [Cisco-EH] [FW-Benchmark].

Operators are urged to consider IPv6 EH filtering and IPv6 options handling capabilities of different devices as they make deployment decisions in future.

3.3. Summary of Advice on the Handling of IPv6 Packets with Specific IPv6 Extension Headers

This section summarizes the advice provided in Section 3.4, providing references to the specific sections in which a detailed analysis can be found.

EH type	Filtering policy	Reference
IPv6 Hop-by-Hop Options (Proto=0)	Drop or Ignore	Section 3.4.1
Routing Header for IPv6 (Proto=43)	Drop only RTH0, Permit other RH Types	Section 3.4.2
Fragment Header for IPv6 (Proto=44)	Permit	Section 3.4.3
Encapsulating Security Payload (Proto=50)	Permit	Section 3.4.4

Authentication Header (Proto=51)	Permit	Section 3.4.5
Destination Options for IPv6 (Proto=60)	Permit	Section 3.4.6
Mobility Header (Proto=135)	Permit	Section 3.4.7
Host Identity Protocol (Proto=139)	Permit	Section 3.4.8
Shim6 Protocol (Proto=140)	Permit	Section 3.4.9
Use for experimentation and testing (Proto=253 and 254)	Drop	Section 3.4.10

        Table 1: Summary of Advice on the Handling of IPv6 Packets with
                    Specific IPv6 Extension Headers

3.4. Advice on the Handling of IPv6 Packets with Specific IPv6
      Extension Headers

3.4.1. IPv6 Hop-by-Hop Options (Protocol Number=0)

3.4.1.1. Uses

   The Hop-by-Hop Options header is used to carry optional information
   that may be examined by every node along a packet's delivery path.
   It is expected that nodes will examine the Hop-by-Hop Options header
   if explicitly configured to do so.

   NOTE: [RFC2460] required that all nodes examined and processed the
   Hop-by-Hop Options header. However, even before the publication of
   [RFC8200] a number of implementations already provided the option of
   ignoring this header unless explicitly configured to examine it.

3.4.1.2. Specification

   This EH is specified in [RFC8200]. At the time of this writing, the
   following options have been specified for the Hop-by-Hop Options EH:

   o Type 0x00: Pad1 [RFC8200]

   o Type 0x01: PadN [RFC8200]

   o Type 0x05: Router Alert [RFC2711]

   o Type 0x07: CALIPSO [RFC5570]

   o Type 0x08: SMF_DPD [RFC6621]

   o Type 0x23: RPL Option [I-D.ietf-roll-useofrplinfo]

   o Type 0x26: Quick-Start [RFC4782]

   o Type 0x4D: (Deprecated)

   o Type 0x63: RPL Option [RFC6553]

   o Type 0x6D: MPL Option [RFC7731]

   o Type 0x8A: Endpoint Identification (Deprecated)
      [draft-ietf-nimrod-eid]

   o Type 0xC2: Jumbo Payload [RFC2675]

   o Type 0xEE: IPv6 DFF Header [RFC6971]

   o Type 0x1E: RFC3692-style Experiment [RFC4727]

   o Type 0x3E: RFC3692-style Experiment [RFC4727]

   o Type 0x5E: RFC3692-style Experiment [RFC4727]

   o Type 0x7E: RFC3692-style Experiment [RFC4727]

   o Type 0x9E: RFC3692-style Experiment [RFC4727]

   o Type 0xBE: RFC3692-style Experiment [RFC4727]

   o Type 0xDE: RFC3692-style Experiment [RFC4727]

   o Type 0xFE: RFC3692-style Experiment [RFC4727]

3.4.1.3. Specific Security Implications

   Legacy nodes that may process this extencion header could be subject
   to Denial of Service attacks.

   NOTE: While [RFC8200] has removed this requirement, the deployed base
   may still reflect the traditional behavior for a while, and hence the

potential security problems of this EH are still of concern.

3.4.1.4. Operational and Interoperability Impact if Blocked

Discarding packets containing a Hop-by-Hop Options EH would break any
of the protocols that rely on it for proper functioning. For
example, it would break RSVP [RFC2205] and multicast deployments, and
would cause IPv6 jumbograms to be discarded.

3.4.1.5. Advice

Nodes implementing [RFC8200] would already ignore this extension
header unless explicitly required to process it. For legacy
([RFC2460] nodes, the recommended configuration for the processing of
these packets depends on the features and capabilities of the
underlying platform. On platforms that allow forwarding of packets
with HBH Options on the fast path, we recommend that packets with a
HBH Options EH be forwarded as normal. Otherwise, on platforms in
which processing of packets with a IPv6 HBH Options EH is carried out
in the slow path, and an option is provided to rate-limit these
packets, we recommend that this option be selected. Finally, when
packets containing a HBH Options EH are processed in the slow-path,
and the underlying platform does not have any mitigation options
available for attacks based on these packets, we recommend that such
platforms discard packets containing IPv6 HBH Options EHs.

Finally, we note that, for obvious reasons, RPL (Routing Protocol for
Low-Power and Lossy Networks) [RFC6550] routers must not discard
packets based on the presence of an IPv6 Hop-by-Hop Options EH.

3.4.2. Routing Header for IPv6 (Protocol Number=43)

3.4.2.1. Uses

The Routing header is used by an IPv6 source to list one or more
intermediate nodes to be "visited" on the way to a packet's
destination.

3.4.2.2. Specification

This EH is specified in [RFC8200]. [RFC2460] had originally
specified the Routing Header Type 0, which was later obsoleted by
[RFC5095], and thus removed from [RFC8200].

At the time of this writing, the following Routing Types have been
specified:

o Type 0: Source Route (DEPRECATED) [RFC2460] [RFC5095]

o Type 1: Nimrod (DEPRECATED)

o Type 2: Type 2 Routing Header [RFC6275]

o Type 3: RPL Source Route Header [RFC6554]

o Types 4-252: Unassigned

o Type 253: RFC3692-style Experiment 1 [RFC4727]

o Type 254: RFC3692-style Experiment 2 [RFC4727]

o Type 255: Reserved

3.4.2.3. Specific Security Implications

The security implications of RHT0 have been discussed in detail in
[Biondi2007] and [RFC5095].

3.4.2.4. Operational and Interoperability Impact if Blocked

Blocking packets containing a RHT0 or RTH1 has no operational
implications. However, blocking packets employing other routing
header types will break the protocols that rely on them.

3.4.2.5. Advice

Intermediate systems should discard packets containing a RHT0 or
RHT1. RHT2 and RHT3 should be permitted, as required by [RFC7045].
Other routing header types should be discarded.

3.4.3. Fragment Header for IPv6 (Protocol Number=44)

3.4.3.1. Uses

This EH provides the fragmentation functionality for IPv6.

3.4.3.2. Specification

This EH is specified in [RFC8200].

3.4.3.3. Specific Security Implications

The security implications of the Fragment Header range from Denial of
Service attacks (e.g. based on flooding a target with IPv6 fragments)
to information leakage attacks [RFC7739].

3.4.3.4. Operational and Interoperability Impact if Blocked

   Blocking packets that contain a Fragment Header will break any
   protocol that may rely on fragmentation (e.g., the DNS [RFC1034]).

3.4.3.5. Advice

   Intermediate systems should permit packets that contain a Fragment
   Header.

3.4.4. Encapsulating Security Payload (Protocol Number=50)

3.4.4.1. Uses

   This EH is employed for the IPsec suite [RFC4303].

3.4.4.2. Specification

   This EH is specified in [RFC4303].

3.4.4.3. Specific Security Implications

   Besides the general implications of IPv6 EHs, this EH could be
   employed to potentially perform a DoS attack at the destination
   system by wasting CPU resources in validating the contents of the
   packet.

3.4.4.4. Operational and Interoperability Impact if Blocked

   Discarding packets that employ this EH would break IPsec deployments.

3.4.4.5. Advice

   Intermediate systems should permit packets containing the
   Encapsulating Security Payload EH.

3.4.5. Authentication Header (Protocol Number=51)

3.4.5.1. Uses

   The Authentication Header can be employed for provide authentication
   services in IPv4 and IPv6.

3.4.5.2. Specification

   This EH is specified in [RFC4302].

3.4.5.3. Specific Security Implications

   Besides the general implications of IPv6 EHs, this EH could be
   employed to potentially perform a DoS attack at the destination
   system by wasting CPU resources in validating the contents of the
   packet.

3.4.5.4. Operational and Interoperability Impact if Blocked

   Discarding packets that employ this EH would break IPsec deployments.

3.4.5.5. Advice

   Intermediate systems should permit packets containing an
   Authentication Header.

3.4.6. Destination Options for IPv6 (Protocol Number=60)

3.4.6.1. Uses

   The Destination Options header is used to carry optional information
   that needs be examined only by a packet's destination node(s).

3.4.6.2. Specification

   This EH is specified in [RFC8200]. At the time of this writing, the
   following options have been specified for this EH:

   o Type 0x00: Pad1 [RFC8200]

   o Type 0x01: PadN [RFC8200]

   o Type 0x04: Tunnel Encapsulation Limit [RFC2473]

   o Type 0x4D: (Deprecated)

   o Type 0xC9: Home Address [RFC6275]

   o Type 0x8A: Endpoint Identification (Deprecated)
      [draft-ietf-nimrod-eid]

   o Type 0x8B: ILNP Nonce [RFC6744]

   o Type 0x8C: Line-Identification Option [RFC6788]

   o Type 0x1E: RFC3692-style Experiment [RFC4727]

o Type 0x3E: RFC3692-style Experiment [RFC4727]

o Type 0x5E: RFC3692-style Experiment [RFC4727]

o Type 0x7E: RFC3692-style Experiment [RFC4727]

o Type 0x9E: RFC3692-style Experiment [RFC4727]

o Type 0xBE: RFC3692-style Experiment [RFC4727]

o Type 0xDE: RFC3692-style Experiment [RFC4727]

o Type 0xFE: RFC3692-style Experiment [RFC4727]

3.4.6.3. Specific Security Implications

   No security implications are known, other than the general
   implications of IPv6 EHs. For a discussion of possible security
   implications of specific options specified for the DO header, please
   see the Section 4.3.

3.4.6.4. Operational and Interoperability Impact if Blocked

   Discarding packets that contain a Destination Options header would
   break protocols that rely on this EH type for conveying information,
   including protocols such as ILNP [RFC6740] and Mobile IPv6 [RFC6275],
   and IPv6 tunnels that employ the Tunnel Encapsulation Limit option.

3.4.6.5. Advice

   Intermediate systems should permit packets that contain a Destination
   Options Header.

3.4.7. Mobility Header (Protocol Number=135)

3.4.7.1. Uses

   The Mobility Header is an EH used by mobile nodes, correspondent
   nodes, and home agents in all messaging related to the creation and
   management of bindings in Mobile IPv6.

3.4.7.2. Specification

   This EH is specified in [RFC6275].

3.4.7.3. Specific Security Implications

   A thorough security assessment of the security implications of the
   Mobility Header and related mechanisms can be found in Section 15 of
   [RFC6275].

3.4.7.4. Operational and Interoperability Impact if Blocked

   Discarding packets containing this EH would break Mobile IPv6.

3.4.7.5. Advice

   Intermediate systems should permit packets containing this EH.

3.4.8. Host Identity Protocol (Protocol Number=139)

3.4.8.1. Uses

   This EH is employed with the Host Identity Protocol (HIP), an
   experimental protocol that allows consenting hosts to securely
   establish and maintain shared IP-layer state, allowing separation of
   the identifier and locator roles of IP addresses, thereby enabling
   continuity of communications across IP address changes.

3.4.8.2. Specification

   This EH is specified in [RFC5201].

3.4.8.3. Specific Security Implications

   The security implications of the HIP header are discussed in detail
   in Section 8 of [RFC6275].

3.4.8.4. Operational and Interoperability Impact if Blocked

   Discarding packets that contain the Host Identity Protocol would
   break HIP deployments.

3.4.8.5. Advice

   Intermediate systems should permit packets that contain a Host
   Identity Protocol EH.

3.4.9. Shim6 Protocol (Protocol Number=140)

3.4.9.1. Uses

   This EH is employed by the Shim6 [RFC5533] Protocol.

3.4.9.2. Specification

   This EH is specified in [RFC5533].

3.4.9.3. Specific Security Implications

   The specific security implications are discussed in detail in Section
   16 of [RFC5533].

3.4.9.4. Operational and Interoperability Impact if Blocked

   Discarding packets that contain this EH will break Shim6.

3.4.9.5. Advice

   Intermediate systems should permit packets containing this EH.

3.4.10. Use for experimentation and testing (Protocol Numbers=253 and
        254)

3.4.10.1. Uses

   These IPv6 EHs are employed for performing RFC3692-Style experiments
   (see [RFC3692] for details).

3.4.10.2. Specification

   These EHs are specified in [RFC3692] and [RFC4727].

3.4.10.3. Specific Security Implications

   The security implications of these EHs will depend on their specific
   use.

3.4.10.4. Operational and Interoperability Impact if Blocked

   For obvious reasons, discarding packets that contain these EHs limits
   the ability to perform legitimate experiments across IPv6 routers.

3.4.10.5. Advice

   Intermediate systems should discard packets containing these EHs.
   Only in specific scenarios in which RFC3692-Style experiments are to
   be performed should these EHs be permitted.

3.5. Advice on the Handling of Packets with Unknown IPv6 Extension
      Headers

   We refer to IPv6 EHs that have not been assigned an Internet Protocol
   Number by IANA (and marked as such) in [IANA-PROTOCOLS] as "unknown
   IPv6 extension headers" ("unknown IPv6 EHs").

3.5.1. Uses

   New IPv6 EHs may be specified as part of future extensions to the
   IPv6 protocol.

   Since IPv6 EHs and Upper-layer protocols employ the same namespace,
   it is impossible to tell whether an unknown "Internet Protocol
   Number" is being employed for an IPv6 EH or an Upper-Layer protocol.

3.5.2. Specification

   The processing of unknown IPv6 EHs is specified in [RFC8200] and
   [RFC7045].

3.5.3. Specific Security Implications

   For obvious reasons, it is impossible to determine specific security
   implications of unknown IPv6 EHs. However, from security standpoint,
   a device should discard IPv6 extension headers for which the security
   implications cannot be determined. We note that this policy is
   allowed by [RFC7045].

3.5.4. Operational and Interoperability Impact if Blocked

   As noted in [RFC7045], discarding unknown IPv6 EHs may slow down the
   deployment of new IPv6 EHs and transport protocols. The
   corresponding IANA registry ([IANA-PROTOCOLS]) should be monitored
   such that filtering rules are updated as new IPv6 EHs are
   standardized.

   We note that since IPv6 EHs and upper-layer protocols share the same
   numbering space, discarding unknown IPv6 EHs may result in packets
   encapsulating unknown upper-layer protocols being discarded.

3.5.5. Advice

   Intermediate systems should discard packets containing unknown IPv6
   EHs.

4. IPv6 Options

4.1. General Discussion

   The following subsections describe specific security implications of
   different IPv6 options, and provide advice regarding filtering
   packets that contain such options.

4.2. General Security Implications of IPv6 Options

   The general security implications of IPv6 options are closely related
   to those discussed in Section 3.2 for IPv6 EHs. Essentially, packets
   that contain IPv6 options might need to be processed by an IPv6
   router's general-purpose CPU,and hence could present a DDoS risk to
   that router's general-purpose CPU (and thus to the router itself).
   For some architectures, a possible mitigation would be to rate-limit
   the packets that are to be processed by the general-purpose CPU (see
   e.g. [Cisco-EH]).

4.3. Advice on the Handling of Packets with Specific IPv6 Options

   The following subsections contain a description of each of the IPv6
   options that have so far been specified, a summary of the security
   implications of each of such options, a discussion of possible
   interoperability implications if packets containing such options are
   discarded, and specific advice regarding whether packets containing
   these options should be permitted.

4.3.1. Pad1 (Type=0x00)

4.3.1.1. Uses

   This option is used when necessary to align subsequent options and to
   pad out the containing header to a multiple of 8 octets in length.

4.3.1.2. Specification

   This option is specified in [RFC8200].

4.3.1.3. Specific Security Implications

   None.

4.3.1.4. Operational and Interoperability Impact if Blocked

   Discarding packets that contain this option would potentially break
   any protocol that relies on IPv6 EHs.

4.3.1.5. Advice

   Intermediate systems should not discard packets based on the presence
   of this option.

4.3.2. PadN (Type=0x01)

4.3.2.1. Uses

   This option is used when necessary to align subsequent options and to
   pad out the containing header to a multiple of 8 octets in length.

4.3.2.2. Specification

   This option is specified in [RFC8200].

4.3.2.3. Specific Security Implications

   Because of the possible size of this option, it could be leveraged as
   a large-bandwidth covert channel.

4.3.2.4. Operational and Interoperability Impact if Blocked

   Discarding packets that contain this option would potentially break
   any protocol that relies on IPv6 EHs.

4.3.2.5. Advice

   Intermediate systems should not discard IPv6 packets based on the
   presence of this option.

4.3.3. Jumbo Payload (Type=0XC2)

4.3.3.1. Uses

   The Jumbo payload option provides the means of specifying payloads
   larger than 65535 bytes.

4.3.3.2. Specification

   This option is specified in [RFC2675].

4.3.3.3. Specific Security Implications

   There are no specific issues arising from this option, except for
   improper validity checks of the option and associated packet lengths.

4.3.3.4. Operational and Interoperability Impact if Blocked

   Discarding packets based on the presence of this option will cause
   IPv6 jumbograms to be discarded.

4.3.3.5. Advice

   Intermediate systems should discard packets that contain this option.
   An operator should permit this option only in specific scenarios in
   which support for IPv6 jumbograms is desired.

4.3.4. RPL Option (Type=0x63)

4.3.4.1. Uses

   The RPL Option provides a mechanism to include routing information
   with each datagram that an RPL router forwards.

4.3.4.2. Specification

   This option was originally specified in [RFC6553]. It has been
   deprecated by [I-D.ietf-roll-useofrplinfo].

4.3.4.3. Specific Security Implications

   Those described in [RFC6553].

4.3.4.4. Operational and Interoperability Impact if Blocked

   This option is meant to be employed within an RPL instance. As a
   result, discarding packets based on the presence of this option (e.g.
   at an ISP) will not result in interoperability implications.

4.3.4.5. Advice

   Non-RPL routers should discard packets that contain an RPL option.

4.3.5. RPL Option (Type=0x23)

4.3.5.1. Uses

   The RPL Option provides a mechanism to include routing information
   with each datagram that an RPL router forwards.

4.3.5.2. Specification

   This option is specified in [I-D.ietf-roll-useofrplinfo].

4.3.5.3. Specific Security Implications

   Those described in [I-D.ietf-roll-useofrplinfo].

4.3.5.4. Operational and Interoperability Impact if Blocked

   This option is meant to survive outside of an RPL instance. As a
   result, discarding packets based on the presence of this option would
   break some use cases for RPL (see [I-D.ietf-roll-useofrplinfo]).

4.3.5.5. Advice

   Intermediate systems should not discard IPv6 packets based on the
   presence of this option.

4.3.6. Tunnel Encapsulation Limit (Type=0x04)

4.3.6.1. Uses

   The Tunnel Encapsulation Limit option can be employed to specify how
   many further levels of nesting the packet is permitted to undergo.

4.3.6.2. Specification

   This option is specified in [RFC2473].

4.3.6.3. Specific Security Implications

   Those described in [RFC2473].

4.3.6.4. Operational and Interoperability Impact if Blocked

   Discarding packets based on the presence of this option could result
   in tunnel traffic being discarded.

4.3.6.5. Advice

   Intermediate systems should not discard packets based on the presence
   of this option.

4.3.7. Router Alert (Type=0x05)

4.3.7.1. Uses

   The Router Alert option [RFC2711] is typically employed for the RSVP
   protocol [RFC2205] and the MLD protocol [RFC2710].

4.3.7.2. Specification

   This option is specified in [RFC2711].

4.3.7.3. Specific Security Implications

   Since this option causes the contents of the packet to be inspected
   by the handling device, this option could be leveraged for performing
   DoS attacks.

4.3.7.4. Operational and Interoperability Impact if Blocked

   Discarding packets that contain this option would break RSVP and
   multicast deployments.

4.3.7.5. Advice

   Intermediate systems should discard packets that contain this option.
   Only in specific environments where support for RSVP, multicast
   routing, or similar protocols is desired, should this option be
   permitted.

4.3.8. Quick-Start (Type=0x26)

4.3.8.1. Uses

   This IP Option is used in the specification of Quick-Start for TCP
   and IP, which is an experimental mechanism that allows transport
   protocols, in cooperation with routers, to determine an allowed
   sending rate at the start and, at times, in the middle of a data
   transfer (e.g., after an idle period) [RFC4782].

4.3.8.2. Specification

   This option is specified in [RFC4782], on the "Experimental" track.

4.3.8.3. Specific Security Implications

   Section 9.6 of [RFC4782] notes that Quick-Start is vulnerable to two
   kinds of attacks:

   o attacks to increase the routers' processing and state load, and,

   o attacks with bogus Quick-Start Requests to temporarily tie up
      available Quick-Start bandwidth, preventing routers from approving
      Quick-Start Requests from other connections.

   We note that if routers in a given environment do not implement and

enable the Quick-Start mechanism, only the general security
implications of IP options (discussed in Section 4.2) would apply.

4.3.8.4. Operational and Interoperability Impact if Blocked

The Quick-Start functionality would be disabled, and additional
delays in TCP's connection establishment (for example) could be
introduced. (Please see Section 4.7.2 of [RFC4782].) We note,
however, that Quick-Start has been proposed as a mechanism that could
be of use in controlled environments, and not as a mechanism that
would be intended or appropriate for ubiquitous deployment in the
global Internet [RFC4782].

4.3.8.5. Advice

Intermediate systems should not discard IPv6 packets based on the
presence of this option.

4.3.9. CALIPSO (Type=0x07)

4.3.9.1. Uses

This option is used for encoding explicit packet Sensitivity Labels
on IPv6 packets. It is intended for use only within Multi-Level
Secure (MLS) networking environments that are both trusted and
trustworthy.

4.3.9.2. Specification

This option is specified in [RFC5570].

4.3.9.3. Specific Security Implications

Presence of this option in a packet does not by itself create any
specific new threat. Packets with this option ought not normally be
seen on the global public Internet.

4.3.9.4. Operational and Interoperability Impact if Blocked

If packets with this option are discarded or if the option is
stripped from the packet during transmission from source to
destination, then the packet itself is likely to be discarded by the
receiver because it is not properly labeled. In some cases, the
receiver might receive the packet but associate an incorrect
sensitivity label with the received data from the packet whose
CALIPSO was stripped by an intermediate router or firewall.
Associating an incorrect sensitivity label can cause the received
information either to be handled as more sensitive than it really is

("upgrading") or as less sensitive than it really is ("downgrading"), either of which is problematic.

4.3.9.5. Advice

Intermediate systems that do not operate in Multi-Level Secure (MLS) networking environments should discard packets that contain this option.

4.3.10. SMF_DPD (Type=0x08)

4.3.10.1. Uses

This option is employed in the (experimental) Simplified Multicast Forwarding (SMF) for unique packet identification for IPv6 I-DPD, and as a mechanism to guarantee non-collision of hash values for different packets when H-DPD is used.

4.3.10.2. Specification

This option is specified in [RFC6621].

4.3.10.3. Specific Security Implications

None. The use of identifiers is subject to the security and privacy considerations discussed in [I-D.gont-predictable-numeric-ids].

4.3.10.4. Operational and Interoperability Impact if Blocked

Dropping packets containing this option within a MANET domain would break SMF. However, dropping such packets at the border of such domain would have no negative impact.

4.3.10.5. Advice

Intermediate system should discard packets that contain this option.

4.3.11. Home Address (Type=0xC9)

4.3.11.1. Uses

The Home Address option is used by a Mobile IPv6 node while away from home, to inform the recipient of the mobile node's home address.

4.3.11.2. Specification

This option is specified in [RFC6275].

4.3.11.3. Specific Security Implications

   No (known) additional security implications than those described in
   [RFC6275].

4.3.11.4. Operational and Interoperability Impact if Blocked

   Discarding IPv6 packets based on the presence of this option will
   break Mobile IPv6.

4.3.11.5. Advice

   Intermediate systems should not discard IPv6 packets based on the
   presence of this option.

4.3.12. Endpoint Identification (Type=0x8A)

4.3.12.1. Uses

   The Endpoint Identification option was meant to be used with the
   Nimrod routing architecture [NIMROD-DOC], but has never seen
   widespread deployment.

4.3.12.2. Specification

   This option is specified in [NIMROD-DOC].

4.3.12.3. Specific Security Implications

   Undetermined.

4.3.12.4. Operational and Interoperability Impact if Blocked

   None.

4.3.12.5. Advice

   Intermediate systems should discard packets that contain this option.

4.3.13. ILNP Nonce (Type=0x8B)

4.3.13.1. Uses

   This option is employed by Identifier-Locator Network Protocol for
   IPv6 (ILNPv6) for providing protection against off-path attacks for
   packets when ILNPv6 is in use, and as a signal during initial
   network-layer session creation that ILNPv6 is proposed for use with
   this network-layer session, rather than classic IPv6.

4.3.13.2. Specification

   This option is specified in [RFC6744].

4.3.13.3. Specific Security Implications

   Those described in [RFC6744].

4.3.13.4. Operational and Interoperability Impact if Blocked

   Discarding packets that contain this option will break INLPv6
   deployments.

4.3.13.5. Advice

   Intermediate systems should not discard packets based on the presence
   of this option.

4.3.14. Line-Identification Option (Type=0x8C)

4.3.14.1. Uses

   This option is used by an Edge Router to identify the subscriber
   premises in scenarios where several subscriber premises may be
   logically connected to the same interface of an Edge Router.

4.3.14.2. Specification

   This option is specified in [RFC6788].

4.3.14.3. Specific Security Implications

   Those described in [RFC6788].

4.3.14.4. Operational and Interoperability Impact if Blocked

   Since this option is meant to be employed in Router Solicitation
   messages, discarding packets based on the presence of this option at
   intermediate systems will result in no interoperability implications.

4.3.14.5. Advice

   Intermediate devices should discard packets that contain this option.

4.3.15. Deprecated (Type=0x4D)

4.3.15.1. Uses

   No information has been found about this option type.

4.3.15.2. Specification

   No information has been found about this option type.

4.3.15.3. Specific Security Implications

   No information has been found about this option type, and hence it
   has been impossible to perform the corresponding security assessment.

4.3.15.4. Operational and Interoperability Impact if Blocked

   Unknown.

4.3.15.5. Advice

   Intermediate systems should discard packets that contain this option.

4.3.16. MPL Option (Type=0x6D)

4.3.16.1. Uses

   This option is used with the Multicast Protocol for Low power and
   Lossy Networks (MPL), that provides IPv6 multicast forwarding in
   constrained networks.

4.3.16.2. Specification

   This option is specified in [RFC7731], and is meant to be included
   only in Hop-by-Hop Option headers.

4.3.16.3. Specific Security Implications

   Those described in [RFC7731].

4.3.16.4. Operational and Interoperability Impact if Blocked

   Dropping packets that contain an MPL option within an MPL network
   would break the Multicast Protocol for Low power and Lossy Networks
   (MPL). However, dropping such packets at the border of such networks
   will have no negative impact.

4.3.16.5. Advice

   Intermediate systems should not discard packets based on the presence
   of this option. However, since this option has been specified for
   the Hop-by-Hop Options, such systems should consider the discussion
   in Section 3.4.1.

4.3.17. IP_DFF (Type=0xEE)

4.3.17.1. Uses

   This option is employed with the (Experimental) Depth-First
   Forwarding (DFF) in Unreliable Networks.

4.3.17.2. Specification

   This option is specified in [RFC6971].

4.3.17.3. Specific Security Implications

   Those specified in [RFC6971].

4.3.17.4. Operational and Interoperability Impact if Blocked

   Dropping packets containing this option within a routing domain that
   is running DFF would break DFF. However, droping such packets at the
   border of such domains will have no security implications.

4.3.17.5. Advice

   Intermediate systems that do not operate within a routing domain that
   is running DFF should discard packets containing this option.

4.3.18. RFC3692-style Experiment (Types = 0x1E, 0x3E, 0x5E, 0x7E, 0x9E,
         0xBE, 0xDE, 0xFE)

4.3.18.1. Uses

   These options can be employed for performing RFC3692-style
   experiments. It is only appropriate to use these values in
   explicitly configured experiments; they must not be shipped as
   defaults in implementations.

4.3.18.2. Specification

   Specified in RFC 4727 [RFC4727] in the context of RFC3692-style
   experiments.

4.3.18.3. Specific Security Implications

   The specific security implications will depend on the specific use of
   these options.

4.3.18.4. Operational and Interoperability Impact if Blocked

   For obvious reasons, discarding packets that contain these options
   limits the ability to perform legitimate experiments across IPv6
   routers.

4.3.18.5. Advice

   Intermediate systems should discard packets that contain these
   options. Only in specific environments where RFC3692-style
   experiments are meant to be performed should these options be
   permitted.

4.4. Advice on the handling of Packets with Unknown IPv6 Options

   We refer to IPv6 options that have not been assigned an IPv6 option
   type in the corresponding registry ([IANA-IPV6-PARAM]) as "unknown
   IPv6 options".

4.4.1. Uses

   New IPv6 options may be specified as part of future protocol work.

4.4.2. Specification

   The processing of unknown IPv6 options is specified in [RFC8200].

4.4.3. Specific Security Implications

   For obvious reasons, it is impossible to determine specific security
   implications of unknown IPv6 options.

4.4.4. Operational and Interoperability Impact if Blocked

   Discarding unknown IPv6 options may slow down the deployment of new
   IPv6 options. As noted in [draft-gont-6man-ipv6-opt-transmit], the
   corresponding IANA registry ([IANA-IPV6-PARAM] should be monitored
   such that IPv6 option filtering rules are updated as new IPv6 options
   are standardized.

4.4.5. Advice

   Enterprise intermediate systems that process the contents of IPv6 EHs
   should discard packets that contain unknown options. Other
   intermediate systems that process the contents of IPv6 EHs should
   permit packets that contain unknown options.

5. IANA Considerations

   This document has no actions for IANA.

6. Security Considerations

   This document provides advice on the filtering of IPv6 packets that
   contain IPv6 EHs (and possibly IPv6 options) at IPv6 transit routers.
   It is meant to improve the current situation of widespread dropping
   of such IPv6 packets in those cases where the drops result from
   improper configuration defaults, or inappropriate advice in this
   area.

7. Acknowledgements

   The authors would like to thank Ron Bonica for his work on earlier
   versions of this document.

   The authors of this document would like to thank (in alphabetical
   order) Mikael Abrahamsson, Brian Carpenter, Mike Heard, Bob Hinden,
   Jen Linkova, Carlos Pignataro, Maria Ines Robles, Donald Smith,
   Pascal Thubert, Ole Troan, Gunter Van De Velde, and Eric Vyncke, for
   providing valuable comments on earlier versions of this document.

   This document borrows some text and analysis from [RFC7126], authored
   by Fernando Gont, Randall Atkinson, and Carlos Pignataro.

8. References

8.1. Normative References

   [I-D.ietf-roll-useofrplinfo]
              Robles, I., Richardson, M., and P. Thubert, "When to use
              RFC 6553, 6554 and IPv6-in-IPv6",
              draft-ietf-roll-useofrplinfo-22 (work in progress),
              March 2018.

   [RFC1034] Mockapetris, P., "Domain names - concepts and facilities",
              STD 13, RFC 1034, DOI 10.17487/RFC1034, November 1987,
              <https://www.rfc-editor.org/info/rfc1034>.

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

   [RFC2205] Braden, R., Ed., Zhang, L., Berson, S., Herzog, S., and S.
              Jamin, "Resource ReSerVation Protocol (RSVP) -- Version 1
              Functional Specification", RFC 2205, DOI 10.17487/RFC2205,
              September 1997, <https://www.rfc-editor.org/info/rfc2205>.

   [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>.

   [RFC2473] Conta, A. and S. Deering, "Generic Packet Tunneling in
              IPv6 Specification", RFC 2473, DOI 10.17487/RFC2473,
              December 1998, <https://www.rfc-editor.org/info/rfc2473>.

   [RFC2675] Borman, D., Deering, S., and R. Hinden, "IPv6 Jumbograms",
              RFC 2675, DOI 10.17487/RFC2675, August 1999,
              <https://www.rfc-editor.org/info/rfc2675>.

   [RFC2710] Deering, S., Fenner, W., and B. Haberman, "Multicast
              Listener Discovery (MLD) for IPv6", RFC 2710,
              DOI 10.17487/RFC2710, October 1999,
              <https://www.rfc-editor.org/info/rfc2710>.

   [RFC2711] Partridge, C. and A. Jackson, "IPv6 Router Alert Option",
              RFC 2711, DOI 10.17487/RFC2711, October 1999,
              <https://www.rfc-editor.org/info/rfc2711>.

   [RFC3692] Narten, T., "Assigning Experimental and Testing Numbers
              Considered Useful", BCP 82, RFC 3692, DOI 10.17487/
              RFC3692, January 2004,
              <https://www.rfc-editor.org/info/rfc3692>.

   [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>.

   [RFC4304] Kent, S., "Extended Sequence Number (ESN) Addendum to
              IPsec Domain of Interpretation (DOI) for Internet Security
              Association and Key Management Protocol (ISAKMP)",
              RFC 4304, DOI 10.17487/RFC4304, December 2005,
              <https://www.rfc-editor.org/info/rfc4304>.

   [RFC4727] Fenner, B., "Experimental Values In IPv4, IPv6, ICMPv4,
              ICMPv6, UDP, and TCP Headers", RFC 4727, DOI 10.17487/
              RFC4727, November 2006,
              <https://www.rfc-editor.org/info/rfc4727>.

   [RFC4782] Floyd, S., Allman, M., Jain, A., and P. Sarolahti, "Quick-
              Start for TCP and IP", RFC 4782, DOI 10.17487/RFC4782,
              January 2007, <https://www.rfc-editor.org/info/rfc4782>.

   [RFC5095] Abley, J., Savola, P., and G. Neville-Neil, "Deprecation
              of Type 0 Routing Headers in IPv6", RFC 5095,
              DOI 10.17487/RFC5095, December 2007,
              <https://www.rfc-editor.org/info/rfc5095>.

   [RFC5201] Moskowitz, R., Nikander, P., Jokela, P., Ed., and T.
              Henderson, "Host Identity Protocol", RFC 5201,
              DOI 10.17487/RFC5201, April 2008,
              <https://www.rfc-editor.org/info/rfc5201>.

   [RFC5533] Nordmark, E. and M. Bagnulo, "Shim6: Level 3 Multihoming
              Shim Protocol for IPv6", RFC 5533, DOI 10.17487/RFC5533,
              June 2009, <https://www.rfc-editor.org/info/rfc5533>.

   [RFC5570] StJohns, M., Atkinson, R., and G. Thomas, "Common
              Architecture Label IPv6 Security Option (CALIPSO)",
              RFC 5570, DOI 10.17487/RFC5570, July 2009,
              <https://www.rfc-editor.org/info/rfc5570>.

   [RFC6275] Perkins, C., Ed., Johnson, D., and J. Arkko, "Mobility
              Support in IPv6", RFC 6275, DOI 10.17487/RFC6275,
              July 2011, <https://www.rfc-editor.org/info/rfc6275>.

   [RFC6398] Le Faucheur, F., Ed., "IP Router Alert Considerations and
              Usage", BCP 168, RFC 6398, DOI 10.17487/RFC6398,
              October 2011, <https://www.rfc-editor.org/info/rfc6398>.

   [RFC6550] Winter, T., Ed., Thubert, P., Ed., Brandt, A., Hui, J.,
              Kelsey, R., Levis, P., Pister, K., Struik, R., Vasseur,
              JP., and R. Alexander, "RPL: IPv6 Routing Protocol for
              Low-Power and Lossy Networks", RFC 6550, DOI 10.17487/
              RFC6550, March 2012,
              <https://www.rfc-editor.org/info/rfc6550>.

   [RFC6553] Hui, J. and JP. Vasseur, "The Routing Protocol for Low-
              Power and Lossy Networks (RPL) Option for Carrying RPL
              Information in Data-Plane Datagrams", RFC 6553,
              DOI 10.17487/RFC6553, March 2012,
              <https://www.rfc-editor.org/info/rfc6553>.

   [RFC6554] Hui, J., Vasseur, JP., Culler, D., and V. Manral, "An IPv6
              Routing Header for Source Routes with the Routing Protocol
              for Low-Power and Lossy Networks (RPL)", RFC 6554,
              DOI 10.17487/RFC6554, March 2012,
              <https://www.rfc-editor.org/info/rfc6554>.

   [RFC6621] Macker, J., Ed., "Simplified Multicast Forwarding",
              RFC 6621, DOI 10.17487/RFC6621, May 2012,
              <https://www.rfc-editor.org/info/rfc6621>.

   [RFC6740] Atkinson, RJ. and SN. Bhatti, "Identifier-Locator Network
              Protocol (ILNP) Architectural Description", RFC 6740,
              DOI 10.17487/RFC6740, November 2012,
              <https://www.rfc-editor.org/info/rfc6740>.

   [RFC6744] Atkinson, RJ. and SN. Bhatti, "IPv6 Nonce Destination
              Option for the Identifier-Locator Network Protocol for
              IPv6 (ILNPv6)", RFC 6744, DOI 10.17487/RFC6744,
              November 2012, <https://www.rfc-editor.org/info/rfc6744>.

   [RFC6788] Krishnan, S., Kavanagh, A., Varga, B., Ooghe, S., and E.
              Nordmark, "The Line-Identification Option", RFC 6788,
              DOI 10.17487/RFC6788, November 2012,
              <https://www.rfc-editor.org/info/rfc6788>.

   [RFC6971] Herberg, U., Ed., Cardenas, A., Iwao, T., Dow, M., and S.
              Cespedes, "Depth-First Forwarding (DFF) in Unreliable
              Networks", RFC 6971, DOI 10.17487/RFC6971, June 2013,
              <https://www.rfc-editor.org/info/rfc6971>.

   [RFC7045] Carpenter, B. and S. Jiang, "Transmission and Processing
              of IPv6 Extension Headers", RFC 7045, DOI 10.17487/
              RFC7045, December 2013,
              <https://www.rfc-editor.org/info/rfc7045>.

   [RFC7112] Gont, F., Manral, V., and R. Bonica, "Implications of
              Oversized IPv6 Header Chains", RFC 7112, DOI 10.17487/
              RFC7112, January 2014,
              <https://www.rfc-editor.org/info/rfc7112>.

   [RFC7731] Hui, J. and R. Kelsey, "Multicast Protocol for Low-Power
              and Lossy Networks (MPL)", RFC 7731, DOI 10.17487/RFC7731,

                February 2016, <https://www.rfc-editor.org/info/rfc7731>.

   [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>.

   [draft-gont-6man-ipv6-opt-transmit]
              Gont, F., Liu, W., and R. Bonica, "Transmission and
              Processing of IPv6 Options", IETF Internet Draft, work in
              progress, August 2014.

8.2. Informative References

   [Biondi2007]
              Biondi, P. and A. Ebalard, "IPv6 Routing Header Security",
              CanSecWest 2007 Security Conference, 2007,
              <http://www.secdev.org/conf/IPv6_RH_security-csw07.pdf>.

   [Cisco-EH]
              Cisco Systems, "IPv6 Extension Headers Review and
              Considerations", Whitepaper. October 2006, <http://
              www.cisco.com/en/US/technologies/tk648/tk872/
              technologies_white_paper0900aecd8054d37d.pdf>.

   [FW-Benchmark]
              Zack, E., "Firewall Security Assessment and Benchmarking
              IPv6 Firewall Load Tests", IPv6 Hackers Meeting #1,
              Berlin, Germany. June 30, 2013, <http://
              www.ipv6hackers.org/meetings/ipv6-hackers-1/
              zack-ipv6hackers1-firewall-security-assessment-and-
              benchmarking.pdf>.

   [I-D.gont-predictable-numeric-ids]
              Gont, F. and I. Arce, "Security and Privacy Implications
              of Numeric Identifiers Employed in Network Protocols",
              draft-gont-predictable-numeric-ids-02 (work in progress),
              February 2018.

   [I-D.gont-v6ops-ipv6-ehs-packet-drops]
              Gont, F., Hilliard, N., Doering, G., (Will), S., and W.
              Kumari, "Operational Implications of IPv6 Packets with
              Extension Headers",
              draft-gont-v6ops-ipv6-ehs-packet-drops-03 (work in
              progress), March 2016.

   [I-D.ietf-6man-hbh-header-handling]
              Baker, F. and R. Bonica, "IPv6 Hop-by-Hop Options

Extension Header", draft-ietf-6man-hbh-header-handling-03
                (work in progress), March 2016.

   [IANA-IPV6-PARAM]
                Internet Assigned Numbers Authority, "Internet Protocol
                Version 6 (IPv6) Parameters", December 2013, <http://
                www.iana.org/assignments/ipv6-parameters/
                ipv6-parameters.xhtml>.

   [IANA-PROTOCOLS]
                Internet Assigned Numbers Authority, "Protocol Numbers",
                2014, <http://www.iana.org/assignments/protocol-numbers/
                protocol-numbers.xhtml>.

   [NIMROD-DOC]
                Nimrod Documentation Page,
                "http://ana-3.lcs.mit.edu/~jnc/nimrod/".

   [RFC3871]   Jones, G., Ed., "Operational Security Requirements for
                Large Internet Service Provider (ISP) IP Network
                Infrastructure", RFC 3871, DOI 10.17487/RFC3871,
                September 2004, <https://www.rfc-editor.org/info/rfc3871>.

   [RFC6192]   Dugal, D., Pignataro, C., and R. Dunn, "Protecting the
                Router Control Plane", RFC 6192, DOI 10.17487/RFC6192,
                March 2011, <https://www.rfc-editor.org/info/rfc6192>.

   [RFC7126]   Gont, F., Atkinson, R., and C. Pignataro, "Recommendations
                on Filtering of IPv4 Packets Containing IPv4 Options",
                BCP 186, RFC 7126, DOI 10.17487/RFC7126, February 2014,
                <https://www.rfc-editor.org/info/rfc7126>.

   [RFC7739]   Gont, F., "Security Implications of Predictable Fragment
                Identification Values", RFC 7739, DOI 10.17487/RFC7739,
                February 2016, <https://www.rfc-editor.org/info/rfc7739>.

   [RFC7872]   Gont, F., Linkova, J., Chown, T., and W. Liu,
                "Observations on the Dropping of Packets with IPv6
                Extension Headers in the Real World", RFC 7872,
                DOI 10.17487/RFC7872, June 2016,
                <https://www.rfc-editor.org/info/rfc7872>.

   [draft-ietf-nimrod-eid]
                Lynn, C., "Endpoint Identifier Destination Option", IETF
                Internet Draft, draft-ietf-nimrod-eid-00.txt,
                November 1995.

Authors' Addresses

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

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

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

Email: liushucheng@huawei.com

         Recommendations on the Filtering of IPv6 Packets Containing IPv6
                            Extension Headers
                  draft-ietf-opsec-ipv6-eh-filtering-06

Abstract

   It is common operator practice to mitigate security risks by
   enforcing appropriate packet filtering. This document analyzes both
   the general security implications of IPv6 Extension Headers and the
   specific security implications of each Extension Header and Option
   type. Additionally, it discusses the operational and
   interoperability implications of discarding packets based on the IPv6
   Extension Headers and IPv6 options they contain. Finally, it
   provides advice on the filtering of such IPv6 packets at transit
   routers for traffic *not* directed to them, for those cases in which
   such filtering is deemed as necessary.

Status of This Memo

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

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

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

   This Internet-Draft will expire on January 3, 2019.

Table of Contents

1. Introduction

   Recent studies (see e.g. [RFC7872]) suggest that there is widespread
   dropping of IPv6 packets that contain IPv6 Extension Headers (EHs).
   In some cases, such packet drops occur at transit routers. While
   some operators "officially" drop packets that contain IPv6 EHs, it is
   possible that some of the measured packet drops be the result of
   improper configuration defaults, or inappropriate advice in this
   area.

This document analyzes both the general security implications of IPv6
EHs and the specific security implications of each EH and Option
type, and provides advice on the filtering of IPv6 packets based on
the IPv6 EHs and the IPv6 options they contain. Since various
protocols may use IPv6 EHs (possibly with IPv6 options), discarding
packets based on the IPv6 EHs or IPv6 options they contain may have
implications on the proper functioning of such protocols. Thus, this
document also attempts to discuss the operational and
interoperability implications of such filtering policies.

The filtering policy typically depends on where in the network such
policy is enforced: when the policy is enforced in a transit network,
the policy typically follows a "black-list" approach, where only
packets with clear negative implications are dropped. On the other
hand, when the policy is enforced closer to the destination systems,
the policy typically follows a "white-list" approach, where only
traffic that is expected to be received is allowed. The advice in
this document is aimed only at transit routers that may need to
enforce a filtering policy based on the EHs and IPv6 options a packet
may contain, following a "black-list" approach, and hence is likely
to be much more permissive that a filtering policy to be employed
e.g. at the edge of an enterprise network. The advice in this
document is meant to improve the current situation of the dropping of
packets with IPv6 EHs in the Internet [RFC7872].

This document is similar in nature to [RFC7126], which addresses the
same problem for the IPv4 case. However, in IPv6, the problem space
is compounded by the fact that IPv6 specifies a number of IPv6 EHs,
and a number of IPv6 options which may be valid only when included in
specific EH types.

This document completes and complements the considerations for
protecting the control plane from packets containing IP options that
can be found in [RFC6192].

Section 2 of this document specifies the terminology and conventions
employed throughout this document. Section 3 of this document
discusses IPv6 EHs and provides advice in the area of filtering IPv6
packets that contain such IPv6 EHs. Section 4 of this document
discusses IPv6 options and provides advice in the area of filtering
IPv6 packets that contain such options.

2. Terminology and Conventions Used in This Document

2.1. Terminology

   The terms "fast path", "slow path", and associated relative terms
   ("faster path" and "slower path") are loosely defined as in Section 2
   of [RFC6398].

   The terms "permit" (allow the traffic), "drop" (drop with no
   notification to sender), and "reject" (drop with appropriate
   notification to sender) are employed as defined in [RFC3871].
   Throughout this document we also employ the term "discard" as a
   generic term to indicate the act of discarding a packet, irrespective
   of whether the sender is notified of such drops, and irrespective of
   whether the specific filtering action is logged.

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

2.2. Applicability Statement

   This document provides advice on the filtering of IPv6 packets with
   EHs at transit routers for traffic *not* explicitly destined to such
   transit routers, for those cases in which such filtering is deemed as
   necessary.

2.3. Conventions

   This document assumes that nodes comply with the requirements in
   [RFC7045]. Namely (from [RFC7045]),

   o If a forwarding node discards a packet containing a standard IPv6
      EH, it MUST be the result of a configurable policy and not just
      the result of a failure to recognise such a header.

   o The discard policy for each standard type of EH MUST be
      individually configurable.

   o The default configuration SHOULD allow all standard IPv6 EHs.

   The advice provided in this document is only meant to guide an
   operator in configuring forwarding devices, and is *not* to be
   interpreted as advice regarding default configuration settings for
   network devices. That is, this document provides advice with respect
   to operational configurations, but does not change the implementation
   defaults required by [RFC7045].

We recommend that configuration options are made available to govern the processing of each IPv6 EH type and each IPv6 option type. Such configuration options may include the following possible settings:

o Permit this IPv6 EH or IPv6 Option type

o Discard (and log) packets containing this IPv6 EH or option type

o Reject (and log) packets containing this IPv6 EH or option type (where the packet drop is signaled with an ICMPv6 error message)

o Rate-limit traffic containing this IPv6 EH or option type

o Ignore this IPv6 EH or option type (as if it was not present) and forward the packet. We note that if a packet carries forwarding information (e.g., in an IPv6 Routing Header) this might be an inappropriate or undesirable action.

We note that special care needs to be taken when devices log packet drops/rejects. Devices should count the number of packets dropped/ rejected, but the logging of drop/reject events should be limited so as to not overburden device resources.

Finally, we note that when discarding packets, it is generally desirable that the sender be signaled of the packet drop, since this is of use for trouble-shooting purposes. However, throughout this document (when recommending that packets be discarded) we generically refer to the action as "discard" without specifying whether the sender is signaled of the packet drop.

3. IPv6 Extension Headers

3.1. General Discussion

IPv6 [RFC8200] EHs allow for the extension of the IPv6 protocol. Since both IPv6 EHs and upper-layer protocols share the same namespace ("Next Header" registry/namespace), [RFC7045] identifies which of the currently assigned Internet Protocol numbers identify IPv6 EHs vs. upper-layer protocols. This document discusses the filtering of packets based on the IPv6 EHs (as specified by [RFC7045]) they contain.

   NOTE: [RFC7112] specifies that non-fragmented IPv6 datagrams and IPv6 First-Fragments MUST contain the entire IPv6 header chain [RFC7112]. Therefore, intermediate systems can enforce the filtering policies discussed in this document, or resort to simply discarding the offending packets when they fail to comply with the requirements in [RFC7112]. We note that, in order to implement

filtering rules on the fast path, it may be necessary for the
filtering device to limit the depth into the packet that can be
inspected before giving up. In circumstances where there is such
a limitation, it is recommended that implementations discard
packets if, when trying to determine whether to discard or permit
a packet, the aforementioned limit is encountered.

3.2. General Security Implications

In some specific device architectures, IPv6 packets that contain IPv6
EHs may cause the corresponding packets to be processed on the slow
path, and hence may be leveraged for the purpose of Denial of Service
(DoS) attacks [I-D.gont-v6ops-ipv6-ehs-packet-drops] [Cisco-EH]
[FW-Benchmark].

Operators are urged to consider IPv6 EH filtering and IPv6 options
handling capabilities of different devices as they make deployment
decisions in future.

3.3. Summary of Advice on the Handling of IPv6 Packets with Specific IPv6 Extension Headers

This section summarizes the advice provided in Section 3.4, providing
references to the specific sections in which a detailed analysis can
be found.

EH type	Filtering policy	Reference
IPv6 Hop-by-Hop Options (Proto=0)	Drop or Ignore	Section 3.4.1
Routing Header for IPv6 (Proto=43)	Drop only RTH0 and RTH1. Permit other RH Types	Section 3.4.2
Fragment Header for IPv6 (Proto=44)	Permit	Section 3.4.3
Encapsulating Security Payload (Proto=50)	Permit	Section 3.4.4
Authentication Header (Proto=51)	Permit	Section 3.4.5
Destination Options for IPv6 (Proto=60)	Permit	Section 3.4.6
Mobility Header (Proto=135)	Permit	Section 3.4.7
Host Identity Protocol (Proto=139)	Permit	Section 3.4.8
Shim6 Protocol (Proto=140)	Permit	Section 3.4.9
Use for experimentation and testing (Proto=253 and 254)	Drop	Section 3.4.10

Table 1: Summary of Advice on the Handling of IPv6 Packets with
Specific IPv6 Extension Headers

3.4. Advice on the Handling of IPv6 Packets with Specific IPv6
      Extension Headers

3.4.1. IPv6 Hop-by-Hop Options (Protocol Number=0)

3.4.1.1. Uses

   The Hop-by-Hop Options header is used to carry optional information
   that may be examined by every node along a packet's delivery path.
   It is expected that nodes will examine the Hop-by-Hop Options header
   if explicitly configured to do so.

   NOTE: [RFC2460] required that all nodes examined and processed the
   Hop-by-Hop Options header. However, even before the publication of
   [RFC8200] a number of implementations already provided the option of
   ignoring this header unless explicitly configured to examine it.

3.4.1.2. Specification

   This EH is specified in [RFC8200]. At the time of this writing, the
   following options have been specified for the Hop-by-Hop Options EH:

   o Type 0x00: Pad1 [RFC8200]

   o Type 0x01: PadN [RFC8200]

   o Type 0x05: Router Alert [RFC2711]

   o Type 0x07: CALIPSO [RFC5570]

   o Type 0x08: SMF_DPD [RFC6621]

   o Type 0x23: RPL Option [I-D.ietf-roll-useofrplinfo]

   o Type 0x26: Quick-Start [RFC4782]

   o Type 0x4D: (Deprecated)

   o Type 0x63: RPL Option [RFC6553]

   o Type 0x6D: MPL Option [RFC7731]

   o Type 0x8A: Endpoint Identification (Deprecated)
      [draft-ietf-nimrod-eid]

   o Type 0xC2: Jumbo Payload [RFC2675]

   o Type 0xEE: IPv6 DFF Header [RFC6971]

   o Type 0x1E: RFC3692-style Experiment [RFC4727]

   o Type 0x3E: RFC3692-style Experiment [RFC4727]

o Type 0x5E: RFC3692-style Experiment [RFC4727]

o Type 0x7E: RFC3692-style Experiment [RFC4727]

o Type 0x9E: RFC3692-style Experiment [RFC4727]

o Type 0xBE: RFC3692-style Experiment [RFC4727]

o Type 0xDE: RFC3692-style Experiment [RFC4727]

o Type 0xFE: RFC3692-style Experiment [RFC4727]

3.4.1.3. Specific Security Implications

   Legacy nodes that may process this extencion header could be subject
   to Denial of Service attacks.

   NOTE: While [RFC8200] has removed this requirement, the deployed base
   may still reflect the traditional behavior for a while, and hence the
   potential security problems of this EH are still of concern.

3.4.1.4. Operational and Interoperability Impact if Blocked

   Discarding packets containing a Hop-by-Hop Options EH would break any
   of the protocols that rely on it for proper functioning. For
   example, it would break RSVP [RFC2205] and multicast deployments, and
   would cause IPv6 jumbograms to be discarded.

3.4.1.5. Advice

   Nodes implementing [RFC8200] would already ignore this extension
   header unless explicitly required to process it. For legacy
   ([RFC2460] nodes, the recommended configuration for the processing of
   these packets depends on the features and capabilities of the
   underlying platform. On platforms that allow forwarding of packets
   with HBH Options on the fast path, we recommend that packets with a
   HBH Options EH be forwarded as normal. Otherwise, on platforms in
   which processing of packets with a IPv6 HBH Options EH is carried out
   in the slow path, and an option is provided to rate-limit these
   packets, we recommend that this option be selected. Finally, when
   packets containing a HBH Options EH are processed in the slow-path,
   and the underlying platform does not have any mitigation options
   available for attacks based on these packets, we recommend that such
   platforms discard packets containing IPv6 HBH Options EHs.

   Finally, we note that, for obvious reasons, RPL (Routing Protocol for
   Low-Power and Lossy Networks) [RFC6550] routers must not discard
   packets based on the presence of an IPv6 Hop-by-Hop Options EH.

3.4.2. Routing Header for IPv6 (Protocol Number=43)

3.4.2.1. Uses

   The Routing header is used by an IPv6 source to list one or more
   intermediate nodes to be "visited" on the way to a packet's
   destination.

3.4.2.2. Specification

   This EH is specified in [RFC8200]. [RFC2460] had originally
   specified the Routing Header Type 0, which was later obsoleted by
   [RFC5095], and thus removed from [RFC8200].

   At the time of this writing, the following Routing Types have been
   specified:

   o Type 0: Source Route (DEPRECATED) [RFC2460] [RFC5095]

   o Type 1: Nimrod (DEPRECATED)

   o Type 2: Type 2 Routing Header [RFC6275]

   o Type 3: RPL Source Route Header [RFC6554]

   o Types 4-252: Unassigned

   o Type 253: RFC3692-style Experiment 1 [RFC4727]

   o Type 254: RFC3692-style Experiment 2 [RFC4727]

   o Type 255: Reserved

3.4.2.3. Specific Security Implications

   The security implications of RHT0 have been discussed in detail in
   [Biondi2007] and [RFC5095].

3.4.2.4. Operational and Interoperability Impact if Blocked

   Blocking packets containing a RHT0 or RTH1 has no operational
   implications, since both have been deprecated. However, blocking
   packets employing other routing header types will break the protocols
   that rely on them.

3.4.2.5. Advice

   Intermediate systems should discard packets containing a RHT0 or
   RHT1. Other routing header types should be permitted, as required by
   [RFC7045].

3.4.3. Fragment Header for IPv6 (Protocol Number=44)

3.4.3.1. Uses

   This EH provides the fragmentation functionality for IPv6.

3.4.3.2. Specification

   This EH is specified in [RFC8200].

3.4.3.3. Specific Security Implications

   The security implications of the Fragment Header range from Denial of
   Service attacks (e.g. based on flooding a target with IPv6 fragments)
   to information leakage attacks [RFC7739].

3.4.3.4. Operational and Interoperability Impact if Blocked

   Blocking packets that contain a Fragment Header will break any
   protocol that may rely on fragmentation (e.g., the DNS [RFC1034]).

3.4.3.5. Advice

   Intermediate systems should permit packets that contain a Fragment
   Header.

3.4.4. Encapsulating Security Payload (Protocol Number=50)

3.4.4.1. Uses

   This EH is employed for the IPsec suite [RFC4303].

3.4.4.2. Specification

   This EH is specified in [RFC4303].

3.4.4.3. Specific Security Implications

   Besides the general implications of IPv6 EHs, this EH could be
   employed to potentially perform a DoS attack at the destination
   system by wasting CPU resources in validating the contents of the
   packet.

3.4.4.4. Operational and Interoperability Impact if Blocked

   Discarding packets that employ this EH would break IPsec deployments.

3.4.4.5. Advice

   Intermediate systems should permit packets containing the
   Encapsulating Security Payload EH.

3.4.5. Authentication Header (Protocol Number=51)

3.4.5.1. Uses

   The Authentication Header can be employed for provide authentication
   services in IPv4 and IPv6.

3.4.5.2. Specification

   This EH is specified in [RFC4302].

3.4.5.3. Specific Security Implications

   Besides the general implications of IPv6 EHs, this EH could be
   employed to potentially perform a DoS attack at the destination
   system by wasting CPU resources in validating the contents of the
   packet.

3.4.5.4. Operational and Interoperability Impact if Blocked

   Discarding packets that employ this EH would break IPsec deployments.

3.4.5.5. Advice

   Intermediate systems should permit packets containing an
   Authentication Header.

3.4.6. Destination Options for IPv6 (Protocol Number=60)

3.4.6.1. Uses

   The Destination Options header is used to carry optional information
   that needs be examined only by a packet's destination node(s).

3.4.6.2. Specification

   This EH is specified in [RFC8200]. At the time of this writing, the
   following options have been specified for this EH:

   o Type 0x00: Pad1 [RFC8200]

   o Type 0x01: PadN [RFC8200]

   o Type 0x04: Tunnel Encapsulation Limit [RFC2473]

   o Type 0x4D: (Deprecated)

   o Type 0xC9: Home Address [RFC6275]

   o Type 0x8A: Endpoint Identification (Deprecated)
      [draft-ietf-nimrod-eid]

   o Type 0x8B: ILNP Nonce [RFC6744]

   o Type 0x8C: Line-Identification Option [RFC6788]

   o Type 0x1E: RFC3692-style Experiment [RFC4727]

   o Type 0x3E: RFC3692-style Experiment [RFC4727]

   o Type 0x5E: RFC3692-style Experiment [RFC4727]

   o Type 0x7E: RFC3692-style Experiment [RFC4727]

   o Type 0x9E: RFC3692-style Experiment [RFC4727]

   o Type 0xBE: RFC3692-style Experiment [RFC4727]

   o Type 0xDE: RFC3692-style Experiment [RFC4727]

   o Type 0xFE: RFC3692-style Experiment [RFC4727]

3.4.6.3. Specific Security Implications

   No security implications are known, other than the general
   implications of IPv6 EHs. For a discussion of possible security
   implications of specific options specified for the DO header, please
   see the Section 4.3.

3.4.6.4. Operational and Interoperability Impact if Blocked

   Discarding packets that contain a Destination Options header would
   break protocols that rely on this EH type for conveying information,
   including protocols such as ILNP [RFC6740] and Mobile IPv6 [RFC6275],
   and IPv6 tunnels that employ the Tunnel Encapsulation Limit option.

3.4.6.5. Advice

   Intermediate systems should permit packets that contain a Destination
   Options Header.

3.4.7. Mobility Header (Protocol Number=135)

3.4.7.1. Uses

   The Mobility Header is an EH used by mobile nodes, correspondent
   nodes, and home agents in all messaging related to the creation and
   management of bindings in Mobile IPv6.

3.4.7.2. Specification

   This EH is specified in [RFC6275].

3.4.7.3. Specific Security Implications

   A thorough security assessment of the security implications of the
   Mobility Header and related mechanisms can be found in Section 15 of
   [RFC6275].

3.4.7.4. Operational and Interoperability Impact if Blocked

   Discarding packets containing this EH would break Mobile IPv6.

3.4.7.5. Advice

   Intermediate systems should permit packets containing this EH.

3.4.8. Host Identity Protocol (Protocol Number=139)

3.4.8.1. Uses

   This EH is employed with the Host Identity Protocol (HIP), an
   experimental protocol that allows consenting hosts to securely
   establish and maintain shared IP-layer state, allowing separation of
   the identifier and locator roles of IP addresses, thereby enabling
   continuity of communications across IP address changes.

3.4.8.2. Specification

   This EH is specified in [RFC5201].

3.4.8.3. Specific Security Implications

   The security implications of the HIP header are discussed in detail
   in Section 8 of [RFC6275].

3.4.8.4. Operational and Interoperability Impact if Blocked

   Discarding packets that contain the Host Identity Protocol would
   break HIP deployments.

3.4.8.5. Advice

   Intermediate systems should permit packets that contain a Host
   Identity Protocol EH.

3.4.9. Shim6 Protocol (Protocol Number=140)

3.4.9.1. Uses

   This EH is employed by the Shim6 [RFC5533] Protocol.

3.4.9.2. Specification

   This EH is specified in [RFC5533].

3.4.9.3. Specific Security Implications

   The specific security implications are discussed in detail in
   Section 16 of [RFC5533].

3.4.9.4. Operational and Interoperability Impact if Blocked

   Discarding packets that contain this EH will break Shim6.

3.4.9.5. Advice

   Intermediate systems should permit packets containing this EH.

3.4.10. Use for experimentation and testing (Protocol Numbers=253 and
         254)

3.4.10.1. Uses

   These IPv6 EHs are employed for performing RFC3692-Style experiments
   (see [RFC3692] for details).

3.4.10.2. Specification

   These EHs are specified in [RFC3692] and [RFC4727].

3.4.10.3. Specific Security Implications

   The security implications of these EHs will depend on their specific
   use.

3.4.10.4. Operational and Interoperability Impact if Blocked

   For obvious reasons, discarding packets that contain these EHs limits
   the ability to perform legitimate experiments across IPv6 routers.

3.4.10.5. Advice

   Intermediate systems should discard packets containing these EHs.
   Only in specific scenarios in which RFC3692-Style experiments are to
   be performed should these EHs be permitted.

3.5. Advice on the Handling of Packets with Unknown IPv6 Extension
      Headers

   We refer to IPv6 EHs that have not been assigned an Internet Protocol
   Number by IANA (and marked as such) in [IANA-PROTOCOLS] as "unknown
   IPv6 extension headers" ("unknown IPv6 EHs").

3.5.1. Uses

   New IPv6 EHs may be specified as part of future extensions to the
   IPv6 protocol.

   Since IPv6 EHs and Upper-layer protocols employ the same namespace,
   it is impossible to tell whether an unknown "Internet Protocol
   Number" is being employed for an IPv6 EH or an Upper-Layer protocol.

3.5.2. Specification

   The processing of unknown IPv6 EHs is specified in [RFC8200] and
   [RFC7045].

3.5.3. Specific Security Implications

   For obvious reasons, it is impossible to determine specific security
   implications of unknown IPv6 EHs. However, from security standpoint,
   a device should discard IPv6 extension headers for which the security
   implications cannot be determined. We note that this policy is
   allowed by [RFC7045].

3.5.4. Operational and Interoperability Impact if Blocked

   As noted in [RFC7045], discarding unknown IPv6 EHs may slow down the
   deployment of new IPv6 EHs and transport protocols. The
   corresponding IANA registry ([IANA-PROTOCOLS]) should be monitored
   such that filtering rules are updated as new IPv6 EHs are
   standardized.

   We note that since IPv6 EHs and upper-layer protocols share the same
   numbering space, discarding unknown IPv6 EHs may result in packets
   encapsulating unknown upper-layer protocols being discarded.

3.5.5. Advice

   Intermediate systems should discard packets containing unknown IPv6
   EHs.

4. IPv6 Options

4.1. General Discussion

   The following subsections describe specific security implications of
   different IPv6 options, and provide advice regarding filtering
   packets that contain such options.

4.2. General Security Implications of IPv6 Options

   The general security implications of IPv6 options are closely related
   to those discussed in Section 3.2 for IPv6 EHs. Essentially, packets
   that contain IPv6 options might need to be processed by an IPv6
   router's general-purpose CPU,and hence could present a DDoS risk to
   that router's general-purpose CPU (and thus to the router itself).
   For some architectures, a possible mitigation would be to rate-limit
   the packets that are to be processed by the general-purpose CPU (see
   e.g. [Cisco-EH]).

4.3. Advice on the Handling of Packets with Specific IPv6 Options

   The following subsections contain a description of each of the IPv6
   options that have so far been specified, a summary of the security
   implications of each of such options, a discussion of possible
   interoperability implications if packets containing such options are
   discarded, and specific advice regarding whether packets containing
   these options should be permitted.

4.3.1. Pad1 (Type=0x00)

4.3.1.1. Uses

   This option is used when necessary to align subsequent options and to
   pad out the containing header to a multiple of 8 octets in length.

4.3.1.2. Specification

   This option is specified in [RFC8200].

4.3.1.3. Specific Security Implications

   None.

4.3.1.4. Operational and Interoperability Impact if Blocked

   Discarding packets that contain this option would potentially break
   any protocol that relies on IPv6 EHs.

4.3.1.5. Advice

   Intermediate systems should not discard packets based on the presence
   of this option.

4.3.2. PadN (Type=0x01)

4.3.2.1. Uses

   This option is used when necessary to align subsequent options and to
   pad out the containing header to a multiple of 8 octets in length.

4.3.2.2. Specification

   This option is specified in [RFC8200].

4.3.2.3. Specific Security Implications

   Because of the possible size of this option, it could be leveraged as
   a large-bandwidth covert channel.

4.3.2.4. Operational and Interoperability Impact if Blocked

   Discarding packets that contain this option would potentially break
   any protocol that relies on IPv6 EHs.

4.3.2.5. Advice

   Intermediate systems should not discard IPv6 packets based on the
   presence of this option.

4.3.3. Jumbo Payload (Type=0XC2)

4.3.3.1. Uses

   The Jumbo payload option provides the means of specifying payloads
   larger than 65535 bytes.

4.3.3.2. Specification

   This option is specified in [RFC2675].

4.3.3.3. Specific Security Implications

   There are no specific issues arising from this option, except for
   improper validity checks of the option and associated packet lengths.

4.3.3.4. Operational and Interoperability Impact if Blocked

   Discarding packets based on the presence of this option will cause
   IPv6 jumbograms to be discarded.

4.3.3.5. Advice

   Intermediate systems should discard packets that contain this option.
   An operator should permit this option only in specific scenarios in
   which support for IPv6 jumbograms is desired.

4.3.4. RPL Option (Type=0x63)

4.3.4.1. Uses

   The RPL Option provides a mechanism to include routing information
   with each datagram that an RPL router forwards.

4.3.4.2. Specification

   This option was originally specified in [RFC6553]. It has been
   deprecated by [I-D.ietf-roll-useofrplinfo].

4.3.4.3. Specific Security Implications

   Those described in [RFC6553].

4.3.4.4. Operational and Interoperability Impact if Blocked

   This option is meant to be employed within an RPL instance. As a
   result, discarding packets based on the presence of this option (e.g.
   at an ISP) will not result in interoperability implications.

4.3.4.5. Advice

   Non-RPL routers should discard packets that contain an RPL option.

4.3.5. RPL Option (Type=0x23)

4.3.5.1. Uses

   The RPL Option provides a mechanism to include routing information
   with each datagram that an RPL router forwards.

4.3.5.2. Specification

   This option is specified in [I-D.ietf-roll-useofrplinfo].

4.3.5.3. Specific Security Implications

   Those described in [I-D.ietf-roll-useofrplinfo].

4.3.5.4. Operational and Interoperability Impact if Blocked

   This option is meant to survive outside of an RPL instance. As a
   result, discarding packets based on the presence of this option would
   break some use cases for RPL (see [I-D.ietf-roll-useofrplinfo]).

4.3.5.5. Advice

   Intermediate systems should not discard IPv6 packets based on the
   presence of this option.

4.3.6. Tunnel Encapsulation Limit (Type=0x04)

4.3.6.1. Uses

   The Tunnel Encapsulation Limit option can be employed to specify how
   many further levels of nesting the packet is permitted to undergo.

4.3.6.2. Specification

   This option is specified in [RFC2473].

4.3.6.3. Specific Security Implications

   Those described in [RFC2473].

4.3.6.4. Operational and Interoperability Impact if Blocked

   Discarding packets based on the presence of this option could result
   in tunnel traffic being discarded.

4.3.6.5. Advice

   Intermediate systems should not discard packets based on the presence
   of this option.

4.3.7. Router Alert (Type=0x05)

4.3.7.1. Uses

   The Router Alert option [RFC2711] is typically employed for the RSVP
   protocol [RFC2205] and the MLD protocol [RFC2710].

4.3.7.2. Specification

   This option is specified in [RFC2711].

4.3.7.3. Specific Security Implications

   Since this option causes the contents of the packet to be inspected
   by the handling device, this option could be leveraged for performing
   DoS attacks.

4.3.7.4. Operational and Interoperability Impact if Blocked

   Discarding packets that contain this option would break RSVP and
   multicast deployments.

4.3.7.5. Advice

   Intermediate systems should discard packets that contain this option.
   Only in specific environments where support for RSVP, multicast
   routing, or similar protocols is desired, should this option be
   permitted.

4.3.8. Quick-Start (Type=0x26)

4.3.8.1. Uses

   This IP Option is used in the specification of Quick-Start for TCP
   and IP, which is an experimental mechanism that allows transport
   protocols, in cooperation with routers, to determine an allowed
   sending rate at the start and, at times, in the middle of a data
   transfer (e.g., after an idle period) [RFC4782].

4.3.8.2. Specification

   This option is specified in [RFC4782], on the "Experimental" track.

4.3.8.3. Specific Security Implications

   Section 9.6 of [RFC4782] notes that Quick-Start is vulnerable to two
   kinds of attacks:

   o attacks to increase the routers' processing and state load, and,

   o attacks with bogus Quick-Start Requests to temporarily tie up
      available Quick-Start bandwidth, preventing routers from approving
      Quick-Start Requests from other connections.

   We note that if routers in a given environment do not implement and
   enable the Quick-Start mechanism, only the general security
   implications of IP options (discussed in Section 4.2) would apply.

4.3.8.4. Operational and Interoperability Impact if Blocked

   The Quick-Start functionality would be disabled, and additional
   delays in TCP's connection establishment (for example) could be
   introduced. (Please see Section 4.7.2 of [RFC4782].) We note,
   however, that Quick-Start has been proposed as a mechanism that could
   be of use in controlled environments, and not as a mechanism that
   would be intended or appropriate for ubiquitous deployment in the
   global Internet [RFC4782].

4.3.8.5. Advice

   Intermediate systems should not discard IPv6 packets based on the
   presence of this option.

4.3.9. CALIPSO (Type=0x07)

4.3.9.1. Uses

   This option is used for encoding explicit packet Sensitivity Labels
   on IPv6 packets. It is intended for use only within Multi-Level
   Secure (MLS) networking environments that are both trusted and
   trustworthy.

4.3.9.2. Specification

   This option is specified in [RFC5570].

4.3.9.3. Specific Security Implications

   Presence of this option in a packet does not by itself create any
   specific new threat. Packets with this option ought not normally be
   seen on the global public Internet.

4.3.9.4. Operational and Interoperability Impact if Blocked

   If packets with this option are discarded or if the option is
   stripped from the packet during transmission from source to
   destination, then the packet itself is likely to be discarded by the
   receiver because it is not properly labeled. In some cases, the
   receiver might receive the packet but associate an incorrect
   sensitivity label with the received data from the packet whose
   CALIPSO was stripped by an intermediate router or firewall.
   Associating an incorrect sensitivity label can cause the received
   information either to be handled as more sensitive than it really is
   ("upgrading") or as less sensitive than it really is ("downgrading"),
   either of which is problematic.

4.3.9.5. Advice

   Intermediate systems that do not operate in Multi-Level Secure (MLS)
   networking environments should discard packets that contain this
   option.

4.3.10. SMF_DPD (Type=0x08)

4.3.10.1. Uses

   This option is employed in the (experimental) Simplified Multicast
   Forwarding (SMF) for unique packet identification for IPv6 I-DPD, and
   as a mechanism to guarantee non-collision of hash values for
   different packets when H-DPD is used.

4.3.10.2. Specification

   This option is specified in [RFC6621].

4.3.10.3. Specific Security Implications

   None. The use of identifiers is subject to the security and privacy
   considerations discussed in [I-D.gont-predictable-numeric-ids].

4.3.10.4. Operational and Interoperability Impact if Blocked

   Dropping packets containing this option within a MANET domain would
   break SMF. However, dropping such packets at the border of such
   domain would have no negative impact.

4.3.10.5. Advice

   Intermediate system should discard packets that contain this option.

4.3.11. Home Address (Type=0xC9)

4.3.11.1. Uses

   The Home Address option is used by a Mobile IPv6 node while away from
   home, to inform the recipient of the mobile node's home address.

4.3.11.2. Specification

   This option is specified in [RFC6275].

4.3.11.3. Specific Security Implications

   No (known) additional security implications than those described in
   [RFC6275].

4.3.11.4. Operational and Interoperability Impact if Blocked

   Discarding IPv6 packets based on the presence of this option will
   break Mobile IPv6.

4.3.11.5. Advice

   Intermediate systems should not discard IPv6 packets based on the
   presence of this option.

4.3.12. Endpoint Identification (Type=0x8A)

4.3.12.1. Uses

   The Endpoint Identification option was meant to be used with the
   Nimrod routing architecture [NIMROD-DOC], but has never seen
   widespread deployment.

4.3.12.2. Specification

   This option is specified in [NIMROD-DOC].

4.3.12.3. Specific Security Implications

   Undetermined.

4.3.12.4. Operational and Interoperability Impact if Blocked

   None.

4.3.12.5. Advice

   Intermediate systems should discard packets that contain this option.

4.3.13. ILNP Nonce (Type=0x8B)

4.3.13.1. Uses

   This option is employed by Identifier-Locator Network Protocol for
   IPv6 (ILNPv6) for providing protection against off-path attacks for
   packets when ILNPv6 is in use, and as a signal during initial
   network-layer session creation that ILNPv6 is proposed for use with
   this network-layer session, rather than classic IPv6.

4.3.13.2. Specification

   This option is specified in [RFC6744].

4.3.13.3. Specific Security Implications

   Those described in [RFC6744].

4.3.13.4. Operational and Interoperability Impact if Blocked

   Discarding packets that contain this option will break INLPv6
   deployments.

4.3.13.5. Advice

   Intermediate systems should not discard packets based on the presence
   of this option.

4.3.14. Line-Identification Option (Type=0x8C)

4.3.14.1. Uses

   This option is used by an Edge Router to identify the subscriber
   premises in scenarios where several subscriber premises may be
   logically connected to the same interface of an Edge Router.

4.3.14.2. Specification

   This option is specified in [RFC6788].

4.3.14.3. Specific Security Implications

   Those described in [RFC6788].

4.3.14.4. Operational and Interoperability Impact if Blocked

   Since this option is meant to be employed in Router Solicitation
   messages, discarding packets based on the presence of this option at
   intermediate systems will result in no interoperability implications.

4.3.14.5. Advice

   Intermediate devices should discard packets that contain this option.

4.3.15. Deprecated (Type=0x4D)

4.3.15.1. Uses

   No information has been found about this option type.

4.3.15.2. Specification

   No information has been found about this option type.

4.3.15.3. Specific Security Implications

   No information has been found about this option type, and hence it
   has been impossible to perform the corresponding security assessment.

4.3.15.4. Operational and Interoperability Impact if Blocked

   Unknown.

4.3.15.5. Advice

   Intermediate systems should discard packets that contain this option.

4.3.16. MPL Option (Type=0x6D)

4.3.16.1. Uses

   This option is used with the Multicast Protocol for Low power and
   Lossy Networks (MPL), that provides IPv6 multicast forwarding in
   constrained networks.

4.3.16.2. Specification

   This option is specified in [RFC7731], and is meant to be included
   only in Hop-by-Hop Option headers.

4.3.16.3. Specific Security Implications

   Those described in [RFC7731].

4.3.16.4. Operational and Interoperability Impact if Blocked

   Dropping packets that contain an MPL option within an MPL network
   would break the Multicast Protocol for Low power and Lossy Networks
   (MPL). However, dropping such packets at the border of such networks
   will have no negative impact.

4.3.16.5. Advice

   Intermediate systems should not discard packets based on the presence
   of this option. However, since this option has been specified for
   the Hop-by-Hop Options, such systems should consider the discussion
   in Section 3.4.1.

4.3.17. IP_DFF (Type=0xEE)

4.3.17.1. Uses

   This option is employed with the (Experimental) Depth-First
   Forwarding (DFF) in Unreliable Networks.

4.3.17.2. Specification

   This option is specified in [RFC6971].

4.3.17.3. Specific Security Implications

   Those specified in [RFC6971].

4.3.17.4. Operational and Interoperability Impact if Blocked

   Dropping packets containing this option within a routing domain that
   is running DFF would break DFF. However, droping such packets at the
   border of such domains will have no security implications.

4.3.17.5. Advice

   Intermediate systems that do not operate within a routing domain that
   is running DFF should discard packets containing this option.

4.3.18. RFC3692-style Experiment (Types = 0x1E, 0x3E, 0x5E, 0x7E, 0x9E,
         0xBE, 0xDE, 0xFE)

4.3.18.1. Uses

   These options can be employed for performing RFC3692-style
   experiments. It is only appropriate to use these values in
   explicitly configured experiments; they must not be shipped as
   defaults in implementations.

4.3.18.2. Specification

   Specified in RFC 4727 [RFC4727] in the context of RFC3692-style
   experiments.

4.3.18.3. Specific Security Implications

   The specific security implications will depend on the specific use of
   these options.

4.3.18.4. Operational and Interoperability Impact if Blocked

   For obvious reasons, discarding packets that contain these options
   limits the ability to perform legitimate experiments across IPv6
   routers.

4.3.18.5. Advice

   Intermediate systems should discard packets that contain these
   options. Only in specific environments where RFC3692-style
   experiments are meant to be performed should these options be
   permitted.

4.4. Advice on the handling of Packets with Unknown IPv6 Options

   We refer to IPv6 options that have not been assigned an IPv6 option
   type in the corresponding registry ([IANA-IPV6-PARAM]) as "unknown
   IPv6 options".

4.4.1. Uses

   New IPv6 options may be specified as part of future protocol work.

4.4.2. Specification

   The processing of unknown IPv6 options is specified in [RFC8200].

4.4.3. Specific Security Implications

   For obvious reasons, it is impossible to determine specific security
   implications of unknown IPv6 options.

4.4.4. Operational and Interoperability Impact if Blocked

   Discarding unknown IPv6 options may slow down the deployment of new
   IPv6 options. As noted in [draft-gont-6man-ipv6-opt-transmit], the
   corresponding IANA registry ([IANA-IPV6-PARAM] should be monitored
   such that IPv6 option filtering rules are updated as new IPv6 options
   are standardized.

4.4.5. Advice

   Enterprise intermediate systems that process the contents of IPv6 EHs
   should discard packets that contain unknown options. Other
   intermediate systems that process the contents of IPv6 EHs should
   permit packets that contain unknown options.

5. IANA Considerations

   This document has no actions for IANA.

6. Security Considerations

   This document provides advice on the filtering of IPv6 packets that
   contain IPv6 EHs (and possibly IPv6 options) at IPv6 transit routers.
   It is meant to improve the current situation of widespread dropping
   of such IPv6 packets in those cases where the drops result from
   improper configuration defaults, or inappropriate advice in this
   area.

7. Acknowledgements

   The authors would like to thank Ron Bonica for his work on earlier
   versions of this document.

   The authors of this document would like to thank (in alphabetical
   order) Mikael Abrahamsson, Brian Carpenter, Darren Dukes, Mike Heard,
   Bob Hinden, Jen Linkova, Carlos Pignataro, Maria Ines Robles, Donald
   Smith, Pascal Thubert, Ole Troan, Gunter Van De Velde, and Eric
   Vyncke, for providing valuable comments on earlier versions of this
   document.

   This document borrows some text and analysis from [RFC7126], authored
   by Fernando Gont, Randall Atkinson, and Carlos Pignataro.

   Fernando Gont would like to thank Eric Vyncke for his guidance.

8. References

8.1. Normative References

   [draft-gont-6man-ipv6-opt-transmit]
              Gont, F., Liu, W., and R. Bonica, "Transmission and
              Processing of IPv6 Options", IETF Internet Draft, work in
              progress, August 2014.

   [I-D.ietf-roll-useofrplinfo]
              Robles, I., Richardson, M., and P. Thubert, "When to use
              RFC 6553, 6554 and IPv6-in-IPv6", draft-ietf-roll-
              useofrplinfo-23 (work in progress), May 2018.

   [RFC1034] Mockapetris, P., "Domain names - concepts and facilities",
              STD 13, RFC 1034, DOI 10.17487/RFC1034, November 1987,
              <https://www.rfc-editor.org/info/rfc1034>.

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

   [RFC2205] Braden, R., Ed., Zhang, L., Berson, S., Herzog, S., and S.
              Jamin, "Resource ReSerVation Protocol (RSVP) -- Version 1
              Functional Specification", RFC 2205, DOI 10.17487/RFC2205,
              September 1997, <https://www.rfc-editor.org/info/rfc2205>.

   [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>.

   [RFC2473] Conta, A. and S. Deering, "Generic Packet Tunneling in
              IPv6 Specification", RFC 2473, DOI 10.17487/RFC2473,
              December 1998, <https://www.rfc-editor.org/info/rfc2473>.

   [RFC2675] Borman, D., Deering, S., and R. Hinden, "IPv6 Jumbograms",
              RFC 2675, DOI 10.17487/RFC2675, August 1999,
              <https://www.rfc-editor.org/info/rfc2675>.

   [RFC2710] Deering, S., Fenner, W., and B. Haberman, "Multicast
              Listener Discovery (MLD) for IPv6", RFC 2710,
              DOI 10.17487/RFC2710, October 1999,
              <https://www.rfc-editor.org/info/rfc2710>.

   [RFC2711] Partridge, C. and A. Jackson, "IPv6 Router Alert Option",
              RFC 2711, DOI 10.17487/RFC2711, October 1999,
              <https://www.rfc-editor.org/info/rfc2711>.

   [RFC3692] Narten, T., "Assigning Experimental and Testing Numbers
              Considered Useful", BCP 82, RFC 3692,
              DOI 10.17487/RFC3692, January 2004,
              <https://www.rfc-editor.org/info/rfc3692>.

   [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>.

   [RFC4304] Kent, S., "Extended Sequence Number (ESN) Addendum to
              IPsec Domain of Interpretation (DOI) for Internet Security
              Association and Key Management Protocol (ISAKMP)",
              RFC 4304, DOI 10.17487/RFC4304, December 2005,
              <https://www.rfc-editor.org/info/rfc4304>.

   [RFC4727] Fenner, B., "Experimental Values In IPv4, IPv6, ICMPv4,
              ICMPv6, UDP, and TCP Headers", RFC 4727,
              DOI 10.17487/RFC4727, November 2006,
              <https://www.rfc-editor.org/info/rfc4727>.

   [RFC4782] Floyd, S., Allman, M., Jain, A., and P. Sarolahti, "Quick-
              Start for TCP and IP", RFC 4782, DOI 10.17487/RFC4782,
              January 2007, <https://www.rfc-editor.org/info/rfc4782>.

   [RFC5095] Abley, J., Savola, P., and G. Neville-Neil, "Deprecation
              of Type 0 Routing Headers in IPv6", RFC 5095,
              DOI 10.17487/RFC5095, December 2007,
              <https://www.rfc-editor.org/info/rfc5095>.

   [RFC5201] Moskowitz, R., Nikander, P., Jokela, P., Ed., and T.
              Henderson, "Host Identity Protocol", RFC 5201,
              DOI 10.17487/RFC5201, April 2008,
              <https://www.rfc-editor.org/info/rfc5201>.

   [RFC5533] Nordmark, E. and M. Bagnulo, "Shim6: Level 3 Multihoming
              Shim Protocol for IPv6", RFC 5533, DOI 10.17487/RFC5533,
              June 2009, <https://www.rfc-editor.org/info/rfc5533>.

   [RFC5570] StJohns, M., Atkinson, R., and G. Thomas, "Common
              Architecture Label IPv6 Security Option (CALIPSO)",
              RFC 5570, DOI 10.17487/RFC5570, July 2009,
              <https://www.rfc-editor.org/info/rfc5570>.

   [RFC6275] Perkins, C., Ed., Johnson, D., and J. Arkko, "Mobility
              Support in IPv6", RFC 6275, DOI 10.17487/RFC6275, July
              2011, <https://www.rfc-editor.org/info/rfc6275>.

   [RFC6398] Le Faucheur, F., Ed., "IP Router Alert Considerations and
              Usage", BCP 168, RFC 6398, DOI 10.17487/RFC6398, October
              2011, <https://www.rfc-editor.org/info/rfc6398>.

   [RFC6550] Winter, T., Ed., Thubert, P., Ed., Brandt, A., Hui, J.,
              Kelsey, R., Levis, P., Pister, K., Struik, R., Vasseur,
              JP., and R. Alexander, "RPL: IPv6 Routing Protocol for
              Low-Power and Lossy Networks", RFC 6550,
              DOI 10.17487/RFC6550, March 2012,
              <https://www.rfc-editor.org/info/rfc6550>.

   [RFC6553] Hui, J. and JP. Vasseur, "The Routing Protocol for Low-
              Power and Lossy Networks (RPL) Option for Carrying RPL
              Information in Data-Plane Datagrams", RFC 6553,
              DOI 10.17487/RFC6553, March 2012,
              <https://www.rfc-editor.org/info/rfc6553>.

   [RFC6554] Hui, J., Vasseur, JP., Culler, D., and V. Manral, "An IPv6
              Routing Header for Source Routes with the Routing Protocol
              for Low-Power and Lossy Networks (RPL)", RFC 6554,
              DOI 10.17487/RFC6554, March 2012,
              <https://www.rfc-editor.org/info/rfc6554>.

   [RFC6621] Macker, J., Ed., "Simplified Multicast Forwarding",
              RFC 6621, DOI 10.17487/RFC6621, May 2012,
              <https://www.rfc-editor.org/info/rfc6621>.

   [RFC6740] Atkinson, RJ. and SN. Bhatti, "Identifier-Locator Network
              Protocol (ILNP) Architectural Description", RFC 6740,
              DOI 10.17487/RFC6740, November 2012,
              <https://www.rfc-editor.org/info/rfc6740>.

   [RFC6744] Atkinson, RJ. and SN. Bhatti, "IPv6 Nonce Destination
              Option for the Identifier-Locator Network Protocol for
              IPv6 (ILNPv6)", RFC 6744, DOI 10.17487/RFC6744, November
              2012, <https://www.rfc-editor.org/info/rfc6744>.

   [RFC6788] Krishnan, S., Kavanagh, A., Varga, B., Ooghe, S., and E.
              Nordmark, "The Line-Identification Option", RFC 6788,
              DOI 10.17487/RFC6788, November 2012,
              <https://www.rfc-editor.org/info/rfc6788>.

   [RFC6971] Herberg, U., Ed., Cardenas, A., Iwao, T., Dow, M., and S.
              Cespedes, "Depth-First Forwarding (DFF) in Unreliable
              Networks", RFC 6971, DOI 10.17487/RFC6971, June 2013,
              <https://www.rfc-editor.org/info/rfc6971>.

   [RFC7045] Carpenter, B. and S. Jiang, "Transmission and Processing
              of IPv6 Extension Headers", RFC 7045,
              DOI 10.17487/RFC7045, December 2013,
              <https://www.rfc-editor.org/info/rfc7045>.

   [RFC7112] Gont, F., Manral, V., and R. Bonica, "Implications of
              Oversized IPv6 Header Chains", RFC 7112,
              DOI 10.17487/RFC7112, January 2014,
              <https://www.rfc-editor.org/info/rfc7112>.

   [RFC7731] Hui, J. and R. Kelsey, "Multicast Protocol for Low-Power
              and Lossy Networks (MPL)", RFC 7731, DOI 10.17487/RFC7731,
              February 2016, <https://www.rfc-editor.org/info/rfc7731>.

   [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>.

8.2. Informative References

   [Biondi2007]
             Biondi, P. and A. Ebalard, "IPv6 Routing Header Security",
             CanSecWest 2007 Security Conference, 2007,
             <http://www.secdev.org/conf/IPv6_RH_security-csw07.pdf>.

   [Cisco-EH]
             Cisco Systems, "IPv6 Extension Headers Review and
             Considerations", Whitepaper. October 2006,
             <http://www.cisco.com/en/US/technologies/tk648/tk872/
             technologies_white_paper0900aecd8054d37d.pdf>.

   [draft-ietf-nimrod-eid]
             Lynn, C., "Endpoint Identifier Destination Option", IETF
             Internet Draft, draft-ietf-nimrod-eid-00.txt, November
             1995.

   [FW-Benchmark]
             Zack, E., "Firewall Security Assessment and Benchmarking
             IPv6 Firewall Load Tests", IPv6 Hackers Meeting #1,
             Berlin, Germany. June 30, 2013,
             <http://www.ipv6hackers.org/meetings/ipv6-hackers-1/zack-
             ipv6hackers1-firewall-security-assessment-and-
             benchmarking.pdf>.

   [I-D.gont-predictable-numeric-ids]
             Gont, F. and I. Arce, "Security and Privacy Implications
             of Numeric Identifiers Employed in Network Protocols",
             draft-gont-predictable-numeric-ids-02 (work in progress),
             February 2018.

   [I-D.gont-v6ops-ipv6-ehs-packet-drops]
             Gont, F., Hilliard, N., Doering, G., (Will), S., and W.
             Kumari, "Operational Implications of IPv6 Packets with
             Extension Headers", draft-gont-v6ops-ipv6-ehs-packet-
             drops-03 (work in progress), March 2016.

   [I-D.ietf-6man-hbh-header-handling]
             Baker, F. and R. Bonica, "IPv6 Hop-by-Hop Options
             Extension Header", draft-ietf-6man-hbh-header-handling-03
             (work in progress), March 2016.

   [IANA-IPV6-PARAM]
             Internet Assigned Numbers Authority, "Internet Protocol
             Version 6 (IPv6) Parameters", December 2013,
             <http://www.iana.org/assignments/ipv6-parameters/
             ipv6-parameters.xhtml>.

   [IANA-PROTOCOLS]
              Internet Assigned Numbers Authority, "Protocol Numbers",
              2014, <http://www.iana.org/assignments/protocol-numbers/
              protocol-numbers.xhtml>.

   [NIMROD-DOC]
              Nimrod Documentation Page,
              "http://ana-3.lcs.mit.edu/~jnc/nimrod/".

   [RFC3871] Jones, G., Ed., "Operational Security Requirements for
              Large Internet Service Provider (ISP) IP Network
              Infrastructure", RFC 3871, DOI 10.17487/RFC3871, September
              2004, <https://www.rfc-editor.org/info/rfc3871>.

   [RFC6192] Dugal, D., Pignataro, C., and R. Dunn, "Protecting the
              Router Control Plane", RFC 6192, DOI 10.17487/RFC6192,
              March 2011, <https://www.rfc-editor.org/info/rfc6192>.

   [RFC7126] Gont, F., Atkinson, R., and C. Pignataro, "Recommendations
              on Filtering of IPv4 Packets Containing IPv4 Options",
              BCP 186, RFC 7126, DOI 10.17487/RFC7126, February 2014,
              <https://www.rfc-editor.org/info/rfc7126>.

   [RFC7739] Gont, F., "Security Implications of Predictable Fragment
              Identification Values", RFC 7739, DOI 10.17487/RFC7739,
              February 2016, <https://www.rfc-editor.org/info/rfc7739>.

   [RFC7872] Gont, F., Linkova, J., Chown, T., and W. Liu,
              "Observations on the Dropping of Packets with IPv6
              Extension Headers in the Real World", RFC 7872,
              DOI 10.17487/RFC7872, June 2016,
              <https://www.rfc-editor.org/info/rfc7872>.

Authors' Addresses

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

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

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

Email: liushucheng@huawei.com

             Operational Security Considerations for IPv6 Networks
                         draft-ietf-opsec-v6-13

   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.

   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 1, 2018.

Copyright Notice

Table of Contents

1. Introduction

   Running an IPv6 network is new for most operators not only because
   they are not yet used to large scale IPv6 networks but also because
   there are subtle differences between IPv4 and IPv6 especially with
   respect to security. For example, all layer-2 interactions are now
   done using Neighbor Discovery Protocol [RFC4861] rather than using
   Address Resolution Protocol [RFC0826]. Also, there are subtle
   differences between NAT44 [RFC2993] and NPTv6 [RFC6296] which are
   explicitly pointed out in the latter's security considerations
   section.

   IPv6 networks are deployed using a variety of techniques, each of
   which have their own specific security concerns.

   This document complements [RFC4942] by listing all security issues
   when operating a network utilizing varying transition technologies
   and updating with ones that have been standardized since 2007. It
   also provides more recent operational deployment experiences where
   warranted.

1.1. Requirements Language

   The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT",
   "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this
   document are to be interpreted as described in [RFC2119] when they
   appear in ALL CAPS. These words may also appear in this document in
   lower case as plain English words, absent their normative meanings.

2. Generic Security Considerations

2.1. Addressing Architecture

   IPv6 address allocations and overall architecture are an important
   part of securing IPv6. Initial designs, even if intended to be
   temporary, tend to last much longer than expected. Although
   initially IPv6 was thought to make renumbering easy, in practice, it
   may be extremely difficult to renumber without a good IP Addresses
   Management (IPAM) system.

   Once an address allocation has been assigned, there should be some
   thought given to an overall address allocation plan. With the
   abundance of address space available, an address allocation may be
   structured around services along with geographic locations, which
   then can be a basis for more structured security policies to permit
   or deny services between geographic regions.

   A common question is whether companies should use PI vs PA space
   [RFC7381], but from a security perspective there is little
   difference. However, one aspect to keep in mind is who has
   administrative ownership of the address space and who is technically
   responsible if/when there is a need to enforce restrictions on
   routability of the space due to malicious criminal activity. 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 network, 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).

2.1.1. Statically Configured Addresses

   When considering how to assign statically configured addresses it is
   necessary to take into consideration the effectiveness of perimeter
   security in a given environment. There is a trade-off between ease
   of 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 common multicast groups which are defined
for important networked devices and the use of commonly repeated
addresses could make it easy to figure out which devices are name
servers, routers or other critical devices; 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 unmanaged environments obfuscating addresses could be
considered a benefit; it is a better practice to ensure that
perimeter rules are actively checked and enforced and that statically
configured addresses follow some logical allocation scheme for ease
of operation (as simplicity always helps security).

2.1.2. Use of ULAs

Unique Local Addresses (ULAs) RFC4193 [RFC4193] are intended for
scenarios where IP addresses are not globally reachable, despite
formally having global scope. They must not appear in the routing
system outside the administrative domain where they are considered
valid. Therefore, packets with ULA source and/or destination
addresses MUST be filtered at the domain boundary.

ULAs are assigned within pseudo-random /48 prefixes created as
specified in RFC4193 [RFC4193]. They could be useful for
infrastructure hiding as described in RFC4864 [RFC4864].

ULAs may be used for internal communication, in conjunction with
globally reachable unicast addresses (GUAs) for hosts that also
require external connectivity through a firewall. For this reason,
no form of address translation is required in conjunction with ULAs.

Using ULAs as described here might simplify the filtering rules
needed at the domain boundary, by allowing a regime in which only
hosts that require external connectivity possess a globally reachable
address. However, this does not remove the need for careful design
of the filtering rules. Routers with ULA on their interfaces may
also leak their address to the Internet when generating ICMP messages
or ICMP error messages can also include ULA address as they contain a
copy of the offending packet.

2.1.3. Point-to-Point Links

   RFC6164 [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 [RFC4443]. The previous recommendation of RFC3627 [RFC3627]
   has been obsoleted and marked Historic by RFC6547 [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 [RFC7404].

2.1.4. Temporary Addresses - Privacy Extensions for SLAAC

   Normal stateless address autoconfiguration (SLAAC) relies on the
   automatically generated EUI-64 address, which together with the /64
   prefix makes up the global unique IPv6 address. The EUI-64 address
   is generated from the MAC address. Randomly generating an interface
   ID, as described in [RFC4941], is part of SLAAC with so-called
   privacy extension addresses and used to address some privacy
   concerns. Privacy extension addresses a.k.a. temporary addresses may
   help to mitigate the correlation of activities of a node within the
   same network, and may also reduce the attack exposure window.

   As privacy extension addresses could also be used to obfuscate some
   malevolent activities (whether on purpose or not), it is advised in
   scenarios where user attribution is important to rely on a layer-2
   authentication mechanism such as IEEE 802.1X [IEEE-802.1X] with the
   appropriate RADIUS accounting (Section 2.6.1.6) or to disable SLAAC
   and rely only on DHCPv6. However, in scenarios where anonymity is a
   strong desire (protecting user privacy is more important than user
   attribution), privacy extension addresses should be used. When
   [RFC8064] is available, the stable temporary address are probably a
   good balance between privacy (among multiple networks) and security/
   user attribution (within a network).

   Using privacy extension addresses prevents the operator from building
   a priori host specific access control lists (ACLs). It must be noted
   that recent versions of Windows do not use the MAC address anymore to
   build the stable address but use a mechanism similar to the one
   described in [RFC7217], this also means that such an ACL cannot be
   configured based solely on the MAC address of the nodes, diminishing
   the value of such ACL. On the other hand, different VLANs are often
   used to segregate users, in this case ACL can rely on a /64 prefix
   per VLAN rather than a per host ACL entry.

The decision to utilize privacy extension addresses can come down to
whether the network is managed versus unmanaged. In some
environments full visibility into the network is required at all
times which requires that all traffic be attributable to where it is
sourced or where it is destined to within a specific network. This
situation is dependent on what level of logging is performed. If
logging considerations include utilizing accurate timestamps and
logging a node's source ports [RFC6302] then there should always
exist appropriate user attribution needed to get to the source of any
malware originator or source of criminal activity.

Disabling SLAAC and privacy extensions addresses can be done 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.

2.1.5. Privacy consideration of Addresses

The reader can learn more about privacy considerations for IPv6
addresses in RFC7721 [RFC7721].

2.1.6. DHCP/DNS Considerations

Many environments use DHCPv6 to allocate addresses to ensure audit-
ability 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 2.3.1).

While there are no fundamental differences with IPv4 and IPv6
security concerns about DNS, there are specific consideration in
DNS64 RFC6147 [RFC6147] environments that need to be understood.
Specifically the interactions and potential to interference with
DNSSEC implementation need to be understood - these are pointed out
in detail in Section 2.7.3.2.

2.1.7. Using a /64 per host

An interesting approach is using a /64 per host as proposed in
RFC8273 [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 [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
   [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 [RFC7045]. The uniform TLV
   format to be used for defining future extension headers is described
   in RFC6564 [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 [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 formated
   packets.

   A firewall or any edge device able to enforce the recommended order
   and number of occurences of extension headers is recommended.

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 [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 [RFC8200], is more sensible to this respect as the
   processing of hop-by-hop options header is optional.

2.2.3. Fragment Header

   The fragment header is used by the source when it has to fragment
   packets. RFC7112 [RFC7112] and section 4.5 of RFC8200 [RFC8200]
   explain why it is important to:

      firewall and security devices should drop first fragment not
      containing an upper-layer header;

      destination nodes should discard first fragments not containing an
      upper-layer header.

   Else, stateless filtering could be bypassed by an hostile party.
   RFC6980 [RFC6980] applies the same rule to NDP and the RA-guard
   function described in RFC6105 [RFC6105].

2.2.4. IP Security Extension Header

   The IPsec [RFC4301] [RFC4301] extension headers (AH [RFC4302] and ESP
   [RFC4303]) are required if IPsec is to be utilized for network level
   security functionality.

2.3. Link-Layer Security

   IPv6 relies heavily on the Neighbor Discovery protocol (NDP) RFC4861
   [RFC4861] to perform a variety of link operations such as discovering
   other nodes on the link, resolving their link-layer addresses, and
   finding routers on the link. If not secured, NDP is vulnerable to
   various attacks such as router/neighbor message spoofing, redirect
   attacks, Duplicate Address Detection (DAD) DoS attacks, etc. many of
   these security threats to NDP have been documented in IPv6 ND Trust
   Models and Threats RFC3756 [RFC3756] and in RFC6583 [RFC6583].

2.3.1. Securing DHCP

   Dynamic Host Configuration Protocol for IPv6 (DHCPv6), as detailed in
   RFC3315 [RFC3315], enables DHCP servers to pass configuration
   parameters such as IPv6 network addresses and other configuration
   information to IPv6 nodes. DHCP plays an important role in any large
   network by providing robust stateful configuration and
   autoregistration of DNS Host Names.

   The two most common threats to DHCP clients come from malicious
   (a.k.a. rogue) or unintentionally misconfigured DHCP servers. A
   malicious DHCP server is established with the intent of providing
   incorrect configuration information to the client to cause a denial
   of service attack or mount a man in the middle attack. While
   unintentionall, a misconfigured DHCP server can have the same impact.
   Additional threats against DHCP are discussed in the security
   considerations section of RFC3315 [RFC3315]DHCP-shield.

   RFC7610 [RFC7610], 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 [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.

2.3.2. ND/RA Rate Limiting

   Neighbor Discovery (ND) can be vulnerable to denial of service (DoS)
   attacks in which a router is forced to perform address resolution for
   a large number of unassigned addresses. Possible side effects of
   this attack preclude new devices from joining the network or even
   worse rendering the last hop router ineffective due to high CPU
   usage. Easy mitigative steps include rate limiting Neighbor
   Solicitations, restricting the amount of state reserved for
   unresolved solicitations, and clever cache/timer management.

   RFC6583 [RFC6583] discusses the potential for DoS in detail and
   suggests implementation improvements and operational mitigation
   techniques that may be used to mitigate or alleviate the impact of
   such attacks. Here are some feasible mitigation options that can be
   employed by network operators today:

o Ingress filtering of unused addresses by ACL, route filtering,
   longer than /64 prefix; These require static configuration of the
   addresses.

o Tuning of NDP process (where supported).

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

2.3.3. ND/RA Filtering

Router Advertisement spoofing is a well-known attack vector and has
been extensively documented. The presence of rogue RAs, either
intentional or malicious, can cause partial or complete failure of
operation of hosts on an IPv6 link. For example, a host can select
an incorrect router address which can be used as a man-in-the-middle
(MITM) attack or can assume wrong prefixes to be used for stateless
address configuration (SLAAC). RFC6104 [RFC6104] summarizes the
scenarios in which rogue RAs may be observed and presents a list of
possible solutions to the problem. RFC6105 [RFC6105] (RA-Guard)
describes a solution framework for the rogue RA problem where network
segments are designed around switching devices that are capable of
identifying invalid RAs and blocking them before the attack packets
actually reach the target nodes.

However, several evasion techniques that circumvent the protection
provided by RA-Guard have surfaced. A key challenge to this
mitigation technique is introduced by IPv6 fragmentation. An
attacker can conceal the attack by fragmenting his packets into
multiple fragments such that the switching device that is responsible
for blocking invalid RAs cannot find all the necessary information to
perform packet filtering in the same packet. RFC7113 [RFC7113]
describes such evasion techniques, and provides advice to RA-Guard
implementers such that the aforementioned evasion vectors can be
eliminated.

Given that the IPv6 Fragmentation Header can be leveraged to
circumvent current implementations of RA-Guard, RFC6980 [RFC6980]
updates RFC4861 [RFC4861] such that use of the IPv6 Fragmentation
Header is forbidden in all Neighbor Discovery messages except
"Certification Path Advertisement", thus allowing for simple and
effective measures to counter Neighbor Discovery attacks.

The Source Address Validation Improvements (SAVI) working group has
worked on other ways to mitigate the effects of such attacks.
RFC7513 [RFC7513] would help in creating bindings between a DHCPv4
RFC2131 [RFC2131] /DHCPv6 RFC3315 [RFC3315] assigned source IP
address and a binding anchor RFC7039 [RFC7039] on a SAVI device.
Also, RFC6620 [RFC6620] describes how to glean similar bindings when
DHCP is not used. The bindings can be used to filter packets
generated on the local link with forged source IP address.

It is still recommended that RA-Guard be be employed as a first line
of defense against common attack vectors including misconfigured
hosts. The generated log should also be analyzed to act on
violations.

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 [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 [RFC3971],
   is a mechanism that was designed to secure ND messages. This
   approach involves the use of new NDP options to carry public key
   based signatures. Cryptographically Generated Addresses (CGA), as
   described in RFC3972 [RFC3972], are used to ensure that the sender of
   a Neighbor Discovery message is the actual "owner" of the claimed
   IPv6 address. A new NDP option, the CGA option, was introduced and
   is used to carry the public key and associated parameters. Another
   NDP option, the RSA Signature option, is used to protect all messages
   relating to neighbor and Router discovery.

   SeND protects against:

   o Neighbor Solicitation/Advertisement Spoofing

   o Neighbor Unreachability Detection Failure

   o Duplicate Address Detection DoS Attack

   o Router Solicitation and Advertisement Attacks

   o Replay Attacks

   o Neighbor Discovery DoS Attacks

   SeND does NOT:

   o Protect statically configured addresses

   o Protect addresses configured using fixed identifiers (i.e. EUI-
      64)

   o Provide confidentiality for NDP communications

   o Compensate for an unsecured link - SEND does not require that the
      addresses on the link and Neighbor Advertisements correspond

   However, at this time and after many years after their
   specifications, CGA and SeND do not have wide support from generic
   operating systems; hence, their usefulness is limited.

2.4. Control Plane Security

   RFC6192 [RFC6192] defines the router control plane. This definition
   is repeated here for the reader's convenience.

Modern router architecture design maintains a strict separation of
forwarding and router control plane hardware and software. The
router control plane supports routing and management functions. It
is generally described as the router architecture hardware and
software components for handling packets destined to the device
itself as well as building and sending packets originated locally on
the device. The forwarding plane is typically described as the
router architecture hardware and software components responsible for
receiving a packet on an incoming interface, performing a lookup to
identify the packet's IP next hop and determine the best outgoing
interface towards the destination, and forwarding the packet out
through the appropriate outgoing interface.

While the forwarding plane is usually implemented in high-speed
hardware, the control plane is implemented by a generic processor
(named router processor RP) and cannot process packets at a high
rate. Hence, this processor can be attacked by flooding its input
queue with more packets than it can process. The control plane
processor is then unable to process valid control packets and the
router can lose OSPF or BGP adjacencies which can cause a severe
network disruption.

The mitigation technique is:

o To drop non-legit control packet before they are queued to the RP
   (this can be done by a forwarding plane ACL) and

o To rate limit the remaining packets to a rate that the RP can
   sustain. Protocol specific protection should also be done (for
   example, a spoofed OSPFv3 packet could trigger the execution of
   the Dijkstra algorithm, therefore the number of Dijsktra 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 [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 [RFC6980] and more generally RFC7112 [RFC7112]
highlights the security implications of oversized extension header
chains on routers and updates RFC2460 [RFC2460] such that the first
fragment of a packet is required to contain the entire IPv6 header
chain.

An ingress ACL cannot help to mitigate a control plane attack using
those packet exceptions. The only protection for the RP is to limit
the rate of those packet exceptions forwarded to the RP, this means
that some data plane packets will be dropped without any ICMP
messages back to the source which 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 [RFC5340]
   or RFC2740 [RFC2740] that was obsoleted by the former. However, the
   document which specifically describes how IPsec should be implemented
   for OSPFv3 RFC4552 [RFC4552] specifically states that ESP-Null MUST
   and AH MAY be implemented since it follows the overall IPsec
   standards wordings. OSPFv3 can also use normal ESP to encrypt the
   OSPFv3 payload to hide the routing information.

   RFC7166 [RFC7166] (which obsoletes RFC6506 [RFC6506] changes OSPFv3's
   reliance on IPsec by appending an authentication trailer to the end
   of the OSPFv3 packets; 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
   RFC6434 [RFC6434] is now 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
   [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
   [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 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:

```perl
#!/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 [RFC6241] to collect the state of the
   neighbor cache;

o also by connecting over a secure management channel (such as SSH)
   and explicitely 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
[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.

2.6.2.1. Forensic

   The forensic use case is when the network operator must locate an
   IPv6 address that was present in the network at a certain time or is
   still currently in the network.

   The source of information can be, in decreasing order, neighbor
   cache, DHCP lease file. Then, the procedure is:

   1. based on the IPv6 prefix of the IPv6 address find the router(s)
       which are used to reach this prefix (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,

   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 stil
   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 RFC6761 [RFC6762] and
RFC6763 [RFC6763].

Enumerating DNS zones, especially looking at reverse DNS records and
CNAMES, is another common method employed by various tools. As
already metioned in RFC7707 [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.

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 network 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 most existing
network operators without an MPLS core where 6PE RFC4798 [RFC4798] is
quite common. 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 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 is 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 being described in RFC 6169 [RFC6169]);

o tunnel injection: a malevolent person knowing a few pieces of
   information (for example the tunnel endpoints and the used
   protocol) can forge a packet which looks like a legit and valid
   encapsulated packet that will gladly be accepted by the
   destination tunnel endpoint, this is a specific case of spoofing;

o traffic interception: no confidentiality is provided by the tunnel
   protocols (without the use of IPsec), therefore anybody on the
   tunnel path can intercept the traffic and have access to the
   clear-text IPv6 packet; 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 not detect an
   malevolent IPv6 traffic contained in the tunnel.

To mitigate the bypassing of security policies, it is recomended 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 2.7.2.6) 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 and the tunnel endpoint are
   usually managed by a single entity; therefore, audit trail and strict
   anti-spoofing are usually possible and this raises the overall
   security.

   Special care must be taken to avoid looping attack by implementing
   the measures of RFC 6324 [RFC6324] and of RFC6964 [RFC6964].

   IPsec [RFC4301] in transport or tunnel mode can be used to secure the
   IPv4 ISATAP traffic to provide IPv6 traffic confidentiality and
   prevent service theft.

2.7.2.3. 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 [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 [RFC4659] to enable IPv6 access over MPLS. As 6PE and
   6VPE are really similar to BGP/MPLS IP VPN described in RFC4364
   [RFC4364], the security of these networks is also similar to the one
   described in RFC4381 [RFC4381]. It relies on:

   o Address space, routing and traffic seperation 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. 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.

2.7.2.7. 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 [RFC3068] which has been
   deprecated by RFC7526 [RFC7526], and is no more used by recent
   Operating Systems. Some security considerations are explained in
   RFC3694 [RFC3964].

   RFC6343 [RFC6343] points out that if an operator provides well-
   managed servers and relays for 6to4, non-encapsulated IPv6 packets
   will pass through well- defined points (the native IPv6 interfaces of
   those servers and relays) at which security mechanisms may be
   applied. Client usage of 6to4 by default is now discouraged, and
   significant precautions are needed to avoid operational problems.

2.7.2.8. 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 manager.

2.7.3. Translation Mechanisms

   Translation mechanisms between IPv4 and IPv6 networks are alternative
   coexistence strategies while networks transition to IPv6. While a
   framework is described in [RFC6144] the specific security
   considerations are documented in each individual mechanism. For the
   most part they specifically mention interference with IPsec or DNSSEC
   deployments, how to mitigate spoofed traffic and what some effective
   filtering strategies may be.

2.7.3.1. Carrier-Grade Nat (CGN)

   Carrier-Grade NAT (CGN), also called NAT444 CGN or Large Scale NAT
   (LSN) or SP NAT is described in [RFC6264] and is utilized as an
   interim measure to prolong the use of IPv4 in a large service
   provider network until the provider can deploy and 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]).

RFC7422 [RFC7422] suggests the use of deterministic address mapping in order to reduce logging requirements for CGN. The idea is to have an algorithm mapping back and forth the internal subscriber to public ports.

2.7.3.2. NAT64/DNS64

Stateful NAT64 translation [RFC6146] allows IPv6-only clients to contact IPv4 servers using unicast UDP, TCP, or ICMP. It can be used in conjunction with DNS64 [RFC6147], a mechanism which synthesizes AAAA records from existing A records. There is also a stateless NAT64 [RFC6145] which is similar for the security aspects with the added benefit of being stateless, so, less prone to a state exhaustion attack.

The Security Consideration sections of [RFC6146] and [RFC6147] list the comprehensive issues. A specific issue with the use of NAT64 is that it will interfere with most IPsec deployments unless UDP encapsulation is used. DNS64 has an incidence on DNSSEC see section 3.1 of [RFC7050].

2.7.3.3. DS-Lite

Dual-Stack Lite (DS-Lite) [RFC6333] is a transition technique that enables a service provider to share IPv4 addresses among customers by combining two well-known technologies: IP in IP (IPv4-in-IPv6) and Network Address and Port Translation (NAPT).

Security considerations with respect to DS-Lite mainly revolve around logging data, preventing DoS attacks from rogue devices (as the AFTR function is stateful) and restricting service offered by the AFTR only to registered customers.

Section 11 of [RFC6333] describes important security issues associated with this technology.

2.8. General Device Hardening

There are many environments which rely too much on the network infrastructure to disallow malicious traffic to get access to critical hosts. In new IPv6 deployments it has been common to see IPv6 traffic enabled but none of the typical access control mechanisms enabled for IPv6 device access. With the possibility of network device configuration mistakes and the growth of IPv6 in the overall Internet it is important to ensure that all individual devices are hardened agains 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::/64 as discard prefix RFC6666 [RFC6666].

4.2. Transition Mechanism

   SP will typically use transition mechanisms such as 6rd, 6PE, MAP,
   DS-Lite which have been analyzed in the transition Section 2.7.2
   section.

4.3. Lawful Intercept

   The Lawful Intercept requirements are similar for IPv6 and IPv4
   architectures and will be subject to the laws enforced in varying
   geographic regions. The local issues with each jurisdiction can make
   this challenging and both corporate legal and privacy personnel
   should be involved in discussions pertaining to what information gets
   logged and what the logging retention policies will be.

   The target of interception will usually be a residential subscriber
   (e.g. his/her PPP session or physical line or CPE MAC address). With
   the absence of NAT on the CPE, IPv6 has the provision to allow for
   intercepting the traffic from a single host (a /128 target) rather
   than the whole set of hosts of a subscriber (which could be a /48, a
   /60 or /64).

   In contrast, in mobile environments, since the 3GPP specifications
   allocate a /64 per device, it may be sufficient to intercept traffic
   from the /64 rather than specific /128's (since each time the device
   powers up it gets a new IID).

A sample architecture which was written for informational purposes is
found in [RFC3924].

5. Residential Users Security Considerations

The IETF Homenet working group is working on how IPv6 residential
network should be done; this obviously includes operational security
considerations; but, this is still work in progress.

Residential users have usually less experience and knowledge about
security or networking. As most of the recent hosts, smartphones,
tablets have all IPv6 enabled by default, IPv6 security is important
for those users. Even with an IPv4-only ISP, those users can get
IPv6 Internet access with the help of Teredo tunnels. Several peer-
to-peer programs (notably Bittorrent) support IPv6 and those programs
can initiate a Teredo tunnel through the IPv4 residential gateway,
with the consequence of making the internal host reachable from any
IPv6 host on the Internet. It is therefore recommended that all host
security products (personal firewall, ...) are configured with a
dual-stack security policy.

If the Residential Gateway has IPv6 connectivity, [RFC7084] (which
obsoletes [RFC6204]) defines the requirements of an IPv6 CPE and does
not take position on the debate of default IPv6 security policy as
defined in [RFC6092]:

o outbound only: allowing all internally initiated connections and
   block all externally initiated ones, which is a common default
   security policy enforced by IPv4 Residential Gateway doing NAT-PT
   but it also breaks the end-to-end reachability promise of IPv6.
   [RFC6092] lists several recommendations to design such a CPE;

o open/transparent: allowing all internally and externally initiated
   connections, therefore restoring the end-to-end nature of the
   Internet for the IPv6 traffic but having a different security
   policy for IPv6 than for IPv4.

[RFC6092] REC-49 states that a choice must be given to the user to
select one of those two policies.

There is also an alternate solution which has been deployed notably
by Swisscom: open to all outbound and inbound connections at the
exception of an handful of TCP and UDP ports known as vulnerable.

6. Further Reading

   There are several documents that describe in more details the
   security of an IPv6 network; these documents are not written by the
   IETF but are listed here for your convenience:

   1. Guidelines for the Secure Deployment of IPv6 [NIST]

   2. North American IPv6 Task Force Technology Report - IPv6 Security
       Technology Paper [NAv6TF_Security]

   3. IPv6 Security [IPv6_Security_Book]

7. Acknowledgements

   The authors would like to thank the following people for their useful
   comments: Mikael Abrahamsson, Fred Baker, Brian Carpenter, Tim Chown,
   Markus deBruen, Tobias Fiebig, Fernando Gont, Jeffry Handal, Panos
   Kampanakis, Erik Kline, Jouni Korhonen, Mark Lentczner, Bob
   Sleigh,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>.

   [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>.

   [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>.

   [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]
              "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., LIU, W., and R. Bonica, "Recommendations on the
              Filtering of IPv6 Packets Containing IPv6 Extension
              Headers", draft-ietf-opsec-ipv6-eh-filtering-04 (work in
              progress), October 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 and Vyncke, "IPv6 Security", ISBN 1-58705-594-5,
              Publisher CiscoPress, December 2008.

   [NAv6TF_Security]
              Kaeo, Green, Bound, and Pouffary, "North American IPv6
              Task Force Technology Report - IPv6 Security Technology
              Paper", 2006, <http://www.ipv6forum.com/dl/white/
              NAv6TF_Security_Report.pdf>.

   [NIST]   Frankel, Graveman, Pearce, and Rooks, "Guidelines for the
              Secure Deployment of IPv6", 2010,
              <http://csrc.nist.gov/publications/nistpubs/800-119/
              sp800-119.pdf>.

   [RFC0826] Plummer, D., "Ethernet Address Resolution Protocol: Or
              Converting Network Protocol Addresses to 48.bit Ethernet
              Address for Transmission on Ethernet Hardware", STD 37,
              RFC 826, DOI 10.17487/RFC0826, November 1982,
              <https://www.rfc-editor.org/info/rfc826>.

   [RFC2131] Droms, R., "Dynamic Host Configuration Protocol",
              RFC 2131, DOI 10.17487/RFC2131, March 1997,
              <https://www.rfc-editor.org/info/rfc2131>.

   [RFC2529] Carpenter, B. and C. Jung, "Transmission of IPv6 over IPv4
              Domains without Explicit Tunnels", RFC 2529,
              DOI 10.17487/RFC2529, March 1999,
              <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>.

   [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., Soininen, J., Patil, B., Savolainen,
              T., Bajko, G., and K. Iisakkila, "IPv6 in 3rd Generation
              Partnership Project (3GPP) Evolved Packet System (EPS)",
              RFC 6459, DOI 10.17487/RFC6459, January 2012,
              <https://www.rfc-editor.org/info/rfc6459>.

   [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>.

   [RFC7217] Gont, F., "A Method for Generating Semantically Opaque
              Interface Identifiers with IPv6 Stateless Address
              Autoconfiguration (SLAAC)", RFC 7217,
              DOI 10.17487/RFC7217, April 2014,
              <https://www.rfc-editor.org/info/rfc7217>.

   [RFC7381] Chittimaneni, K., Chown, T., Howard, L., Kuarsingh, V.,
              Pouffary, Y., and E. Vyncke, "Enterprise IPv6 Deployment
              Guidelines", RFC 7381, DOI 10.17487/RFC7381, October 2014,
              <https://www.rfc-editor.org/info/rfc7381>.

   [RFC7404] Behringer, M. and E. Vyncke, "Using Only Link-Local
              Addressing inside an IPv6 Network", RFC 7404,
              DOI 10.17487/RFC7404, November 2014,
              <https://www.rfc-editor.org/info/rfc7404>.

   [RFC7422] Donley, C., Grundemann, C., Sarawat, V., Sundaresan, K.,
              and O. Vautrin, "Deterministic Address Mapping to Reduce
              Logging in Carrier-Grade NAT Deployments", RFC 7422,
              DOI 10.17487/RFC7422, December 2014,
              <https://www.rfc-editor.org/info/rfc7422>.

   [RFC7454] Durand, J., Pepelnjak, I., and G. Doering, "BGP Operations
              and Security", BCP 194, RFC 7454, DOI 10.17487/RFC7454,
              February 2015, <https://www.rfc-editor.org/info/rfc7454>.

   [RFC7513] Bi, J., Wu, J., Yao, G., and F. Baker, "Source Address
              Validation Improvement (SAVI) Solution for DHCP",
              RFC 7513, DOI 10.17487/RFC7513, May 2015,
              <https://www.rfc-editor.org/info/rfc7513>.

   [RFC7526] Troan, O. and B. Carpenter, Ed., "Deprecating the Anycast
              Prefix for 6to4 Relay Routers", BCP 196, RFC 7526,
              DOI 10.17487/RFC7526, May 2015,
              <https://www.rfc-editor.org/info/rfc7526>.

   [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>.

   [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>.

   [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>.

   [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
Dropbox Inc.
185 Berry Street, Suite 400
San Francisco, CA 94107
USA

Email: kk@dropbox.com

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

Phone: +12066696394
Email: merike@doubleshotsecurity.com

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

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

             Operational Security Considerations for IPv6 Networks
                          draft-ietf-opsec-v6-21

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 places of a network such as the Internet of Things
   are not discussed in this document.

Copyright Notice

Table of Contents

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].

   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 it with 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 proper IP Addresses
   Management (IPAM) system.

   A key task for a successful IPv6 deployment is to prepare an
   addressing plan. Because an abundance of address space available,
   structuring an address plan around both services and geographic
   locations allow address space to become 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 e.g. due to
   malicious criminal activity originating from it.

   In [RFC7934], it is recommended that IPv6 network deployments provide
   multiple IPv6 addresses from each prefix to general-purpose hosts and
   it specifically does not recommend limiting a host to only one IPv6
   address per prefix. It also recommends that the network give the
   host the ability to use new addresses without requiring explicit
   requests (for example by using SLAAC). Having multiple IPv6
   addresses per interface is a major change compared to the unique IPv4
   address per interface; especially for audits (see section
   Section 2.6.2.3).

2.1.1. Use of ULAs

   Unique Local Addresses (ULAs) [RFC4193] are intended for scenarios
   where interfaces are not globally reachable, despite being routed
   within a domain. They formally have global scope, but [RFC4193]
   specifies that they must be filtered out at domain boundaries. ULAs
   are different from [RFC1918] addresses and have different use cases.
   One use of ULA is described in [RFC4864].

2.1.2. Point-to-Point Links

   [RFC6164] in section 5.1 specifies the rationale of using /127 for
   inter-router point-to-point links; 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].

2.1.3. Loopback Addresses

   Many operators reserve a /64 block for all loopback addresses in
   their infrastructure and allocate a /128 out of this reserved /64
   prefix for each loopback interface. This practice allows for an easy
   to write Access Control List (ACL) to enforce a security policy about
   those loopback addresses.

2.1.4. 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 feasible than expected; see
   also [RFC7707]. The use of well-known IPv6 addresses (such as
   ff02::1 for all link-local nodes) or the use of commonly repeated
   addresses could make it easy to figure out which devices are name
   servers, routers or other critical devices; even a simple traceroute
   will expose most of the routers on a path. There are many scanning
   techniques possible and operators should not rely on the 'impossible
   to find because my address is random' paradigm, even if it is common

practice to have the statically configured addresses randomly
distributed across /64 subnets and to always use DNS.

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.

2.1.5. 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 is used
to address some privacy concerns. Privacy extension addresses a.k.a.
temporary addresses may help to mitigate the correlation of
activities of a node within the same network, and may also reduce the
attack exposure window.

Using [RFC4941] privacy extension addresses might prevent the
operator from building host specific access control lists (ACLs).
The [RFC4941] privacy extension addresses could also be used to
obfuscate some malevolent activities and specific user attribution/
accountability procedures should be put in place as described in
Section 2.6.

[RFC8064] specifies another way to generate an address while still
keeping the same IID for each network prefix; this allows SLAAC nodes
to always have the same stable IPv6 address on a specific network
while having different IPv6 address on different networks.

In some specific use cases where user accountability is more
important than user privacy, network operators may consider disabling
SLAAC and relying only on DHCPv6; but, not all operating systems
support DHCPv6 so some hosts will not get any IPv6 connectivity.
Disabling SLAAC and privacy extensions addresses can be done for most
operating systems by sending RA messages with a hint to get addresses
via DHCPv6 by setting the M-bit but also disabling SLAAC by resetting
all A-bits in all prefix information options. However, attackers

could still find ways to bypass this mechanism if not enforced at the
switch/ router level.

However, in scenarios where anonymity is a strong desire (protecting
user privacy is more important than user attribution), privacy
extension addresses should be used. When [RFC8064] is available, the
stable privacy address is probably a good balance between privacy
(among different networks) and security/user attribution (within a
network).

2.1.6. DHCP/DNS Considerations

Many environments use DHCPv6 to provision addresses and other
parameters in order to ensure audit-ability and traceability (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). It must
be noted that as opposed to DHCPv4, DHCPv6 can lease several IPv6
addresses per client and the lease is not bound to the link-layer
address of the client but to the DHCP Unique ID (DUID) of the client
that is not always bound to the client link-layer address.

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 the potential of interference with
DNSSEC implementation need to be understood - these are pointed out
in more 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 for easier user attribution (typically based
on the host MAC address) as its /64 prefix is stable even if
applications within the host can change their IPv6 address within
this /64.

2.1.8. Privacy consideration of Addresses

Beside the security aspects of IPv6 addresses, there are also privacy
considerations: mainly because they are of global scope and visible
globally. [RFC7721] goes in more details about the privacy
considerations of IPv6 addresses by comparing the manually
configured, DHCPv6 or SLAAC.

2.2. Extension Headers

   The extension headers are an important difference between IPv4 and
   IPv6. In IPv4-based packets, it's trivial to find the upper layer
   protocol type and protocol header, while in IPv6 it is more complex
   since the extension header chain must be parsed completely. 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 packets with
   existing or future extension headers is found in [RFC7045]. The
   uniform TLV format to be used for defining future extension headers
   is described in [RFC6564].

   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 led to
   nodes crashing when receiving or forwarding wrongly formatted
   packets.

   A firewall or edge device should be used to enforce the recommended
   order and 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 enables denial of service attacks 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], has
   taken this attack vector into account and made the processing of hop-
   by-hop options header by intermediate routers 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 that:

      firewall and security devices should drop first fragments that do
      not contain the entire ipv6 header chain (including the transport-
      layer header);

      destination nodes should discard first fragments that do not
      contain the entire ipv6 header chain (including the transport-
      layer header).

   If those requirements are not met, stateless filtering could be
   bypassed by a hostile party. [RFC6980] applies a stricter rule to
   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; for example, when a router is forced to perform address
   resolution for a large number of unassigned addresses. This can keep
   new devices from joining the network or render the last hop router
   ineffective due to high CPU usage. Easy mitigative steps include
   rate limiting Neighbor Solicitations, restricting the amount of state
   reserved for unresolved solicitations, and clever cache/timer
   management.

   [RFC6583] discusses the potential for DoS in detail and suggests
   implementation improvements and operational mitigation techniques
   that may be used to mitigate or alleviate the impact of such attacks.
   Here are some feasible mitigation options that can be employed by
   network operators today:

   o Ingress filtering of unused addresses by ACL. These require
      static configuration of the addresses; for example, allocating the
      addresses out of a /120 and using a specific ACL to only allow
      traffic to this /120 (of course, the actual hosts are configured
      with a /64 prefix for the link).

   o Tuning of NDP process (where supported).

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

   o Using link-local addresses only on links where there are only
      routers see [RFC7404]

   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 wireless
   networks, such as 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 wireless 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] helps in creating bindings between a DHCPv4 [RFC2131]
/DHCPv6 [RFC8415] assigned source IP address and a binding anchor
[RFC7039] on a SAVI device. Also, [RFC6620] describes how to glean
similar bindings when DHCP is not used. The bindings can be used to
filter packets generated on the local link with forged source IP
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 most effective when incomplete
fragments are dropped by routers and switches as described in
Section 2.2.3. The generated log should also be analyzed to act on
violations.

A drastic technique to prevent all NDP attacks is based on isolation
of all hosts with specific configurations. Hosts (i.e. all nodes
that are not routers) are unable to send data-link layer frames to
other hosts, therefore, no host to host attacks can happen. This

specific set-up can be established on some switches or wireless
access points. Of course, this is not always easily feasible when
hosts need to communicate with other hosts.

2.3.3. Securing DHCP

Dynamic Host Configuration Protocol for IPv6 (DHCPv6), as detailed in
[RFC8415], enables DHCP servers to pass configuration parameters such
as IPv6 network addresses and other configuration information to IPv6
nodes. DHCP plays an important role in most large networks by
providing robust stateful configuration and in the context of
automated system provisioning.

The two most common threats to DHCP clients come from malicious
(a.k.a. rogue) or unintentionally misconfigured DHCP servers. 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 to 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. Furthermore, extension headers could be leveraged to bypass
DHCPv6-Shield unless [RFC7112] is enforced.

It is recommended to use DHCPv6-Shield and to analyze the
corresponding log messages.

2.3.4. 3GPP Link-Layer Security

The 3GPP link is a point-to-point like link that has no link-layer
address. This implies there can only be 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 details. In some mobile network, DHCPv6 and DHCP-PD are also used.

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 over a decade since their original
   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. Please note that the
   definition is completely protocol-version agnostic (most of this
   section applies to IPv6 in the same way as to IPv4).

   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 techniques are:

   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 Dijsktra
      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 will depend on the available resources of the router
(CPU, TCAM, ...).

2.4.2. Management Protocols

This class includes: SSH, SNMP, syslog, NTP, etc.

An ingress ACL to be applied on all the router interfaces (or at
ingress interfaces of the security perimeter or by using specific
features of the platform) should be configured 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 will depend on the available resources of the router.

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 mitigate a control plane attack using these
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 and also to prevent an amplification

attack using the router as a reflector. It is worth noting that some
platforms implement this rate-limiting in hardware. Of course, a
consequence of not generating an ICMP message will break some IPv6
mechanisms such as Path MTU discovery or a simple traceroute.

2.5. Routing Security

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.

[RFC5082] is also applicable to IPv6 and can ensure that routing
protocol packets are coming from the local network; it must also be
noted that in IPv6 all interior gateway protocols use link-local
addresses.

2.5.1. Authenticating Neighbors

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].
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] changes OSPFv3 reliance on IPsec by appending an
authentication trailer to the end of the OSPFv3 packets; it does not
specifically authenticate the specific originator of an OSPFv3
packet; rather, it allows a router to confirm that the packet has
been issued by a router that had access to the shared authentication
key.

With all authentication mechanisms, operators should confirm that
implementations can support re-keying mechanisms that do not cause
outages. There have been instances where any re-keying cause outages
and therefore, the tradeoff between utilizing this functionality
needs to be weighed against the protection it provides.

As for IPv4, it is recommended to enable a routing protocol only on
interface where it is required.

2.5.2. Securing Routing Updates

IPv6 initially mandated the provisioning of IPsec capability in all
nodes. However, in the updated IPv6 Nodes Requirement standard
[RFC8504] is a 'SHOULD' and not a 'MUST' implement. Theoretically it
is possible that communication between two IPv6 nodes, especially
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, it
has also an operational cost 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 routes for bogon and reserved space (see [CYMRU] and
   [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 [RFC8210]

Some good recommendations for filtering can be found from Team CYMRU
at [CYMRU]. [RFC7454] is another valuable resource of guidance in
this space.

A valid routing table can also be used apply network ingress
filtering (see [RFC2827]).

2.6. Logging/Monitoring

In order to perform forensic research in case of a security incident
or to detect abnormal behaviors, network operators should log
multiple pieces of information.

This logging should include:

o logs of all applications using the network (including user space
   and kernel space) when available (for example web servers);

o data from IP Flow Information Export [RFC7011] also known as
   IPfix;

o data from various SNMP MIB [RFC4293];

o historical data of Neighbour Cache entries;

o stateful DHCPv6 [RFC8415] lease cache, especially when a relay
   agent [RFC6221] is used;

o Source Address Validation Improvement (SAVI) [RFC7039] events,
   especially the binding of an IPv6 address to a MAC address and a
   specific switch or router interface;

o RADIUS [RFC2866] accounting records.

Please note that there are privacy issues or regulations related to
how those logs are collected, kept and safely discarded. Operators
are urged to check their country legislation (e.g. GDPR in the
European Union).

All those pieces of information can be used for:

o forensic (Section 2.6.2.1) investigations such as who did what and
   when?

o correlation (Section 2.6.2.3): which IP addresses were used by a
   specific node (assuming the use of privacy extensions addresses
   [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 an 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, for example 2001:db8::1 can be
   written in multiple ways such as:

   o 2001:DB8::1 (in uppercase)

   o 2001:0db8::0001 (with leading 0)

   o and many other ways including the reverse DNS mapping into a FQDN
      (which should not be trusted).

   [RFC5952] explains this problem in detail and recommends the use of a
   single canonical format. This document recommends the use of
   canonical format [RFC5952] for IPv6 addresses in all possible cases.
   If the existing application cannot log under the canonical format,
   then it is recommended to use an external program in order to
   canonicalize all IPv6 addresses.

   For example, this perl script can be used:

```
<CODE BEGINS>

#!/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" ;
}

<CODE ENDS>
```

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. There are multiple ways to
   collect the current entries in the Neighbor Cache, notably but not
   limited to:

   o the SNMP MIB (Section 2.6.1.3) as explained above;

   o using streaming telemetry or NETCONF [RFC6241] to collect the
      operational state of the neighbor cache;

   o also by connecting over a secure management channel (such as SSH)
      and explicitly requesting a neighbor cache dump via the Command
      Line Interface or another 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 at an
   interval which does not exhaust the router resources and still
   provides valuable information (suggested value is 30 seconds but to
   be checked in the actual set-up) 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.

   When using one /64 per host (Section 2.1.7) or DHCP-PD, it is
   sufficient to keep the history of the allocated prefixes when

combined with strict source address prefix enforcement on the routers
and layer-2 switches to prevent IPv6 spoofing.

2.6.1.5. Stateful DHCPv6 Lease

In some networks, IPv6 addresses/prefixes are managed by a stateful
DHCPv6 server [RFC8415] that leases IPv6 addresses/prefixes to
clients. It is indeed quite similar to DHCP for IPv4 so it can be
tempting to use this DHCP lease file to discover the mapping between
IPv6 addresses/prefixes and data-link layer addresses as 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 saved 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. If possible, it is recommended to use DHCPv6 servers that keep
the relayed data-link layer address in addition to the DUID in the
lease file as those servers have the equivalent information to IPv4
DHCP servers.

The mapping between data-link layer address and the IPv6 address can
be secured by 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 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.

   To locate an IPv6 address in an enterprise network where the operator
   has control over all resources, 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
       the IPv6 address, retrieve the data-link address from live
       neighbor cache, from the historical data of the neighbor cache or
       from SAVI events, or retrieve the data-link address from the DHCP
       lease file (Section 2.6.1.5);

3. based on the data-link layer address, look-up on which switch
    interface was this data-link layer address. In the case of
    wireless LAN with RADIUS accounting (see Section 2.6.1.6), the
    RADIUS log has the mapping between the user identification and
    the MAC address. If a Configuration Management Data Base (CMDB)
    is used, then it can be used to map the data-link layer address
    to a switch port.

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

To identify the subscriber of an IPv6 address in a residential
Internet Service Provider, the starting point is the DHCP-PD leased
prefix covering the IPv6 address; this prefix can often be linked to
a subscriber via the RADIUS log. Alternatively, the Forwarding
Information Base of the CMTS or BNG indicates the CPE of the
subscriber and the RADIUS log can be used to retrieve the actual
subscriber.

More generally, a mix of the above techniques can be used in most if
not all networks.

2.6.2.2. Inventory

RFC 7707 [RFC7707] 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 in a format with only 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 using multiple IPv6 addresses per
prefix, so, the correlation must also be done among those multiple
IPv6 addresses, for example by discovering in the NDP cache
(Section 2.6.1.4) all IPv6 addresses associated with the same MAC
address and interface.

2.6.2.4. Abnormal Behavior Detection

Abnormal 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
   (network path, quality of service, ...). Dual stack enables a
   gradual turn off of the IPv4 operations when the IPv6 network is
   ready for prime time. On the other hand, the operators have to
   manage two network stacks with the added complexities.

   From an operational security perspective, this now means that 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 or security perimeter:

   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. The enforced IPv6
security must be congruent with the IPv4 security policy, else the
attacker will use the protocol version having the more relax security
policy. Maintaining the congruence between security policies can be
challenging (especially over time); it is recommended to use a
firewall or an ACL manager that is dual-stack, i.e., a system that
can apply a single ACL entry to a mixed group of IPv4 and IPv6
addresses.

Also, given the end-to-end connectivity that IPv6 provides, it is
recommended that hosts be fortified against threats. General device
hardening guidelines are provided in Section 2.8.

For many years, all host operating systems have IPv6 enabled by
default, so, it is possible even in an 'IPv4-only' network to attack
layer-2 adjacent victims over their IPv6 link-local address or over a
global IPv6 address when the attacker provides rogue RAs or a rogue
DHCPv6 service.

2.7.2. Encapsulation 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 as 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 or alternative encryption
   methods), therefore anybody on the tunnel path can intercept the
   traffic and have access to the clear-text IPv6 packet; combined
   with the absence of authentication, 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 may 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 IPv4 packets matching:

o IP protocol 41: this will block ISATAP (Section 2.7.2.2), 6to4 (Section 2.7.2.7), 6rd (Section 2.7.2.3) as well as 6in4 (Section 2.7.2.1) tunnels;

o IP protocol 47: this will block GRE (Section 2.7.2.1) tunnels;

o UDP protocol 3544: this will block the default encapsulation of Teredo (Section 2.7.2.8) tunnels. Teredo is now mostly never used and it is no more automated in most environment, so, it is less of a threat, however, special consideration must be taken in case of devices with older or non-updated operating systems may be present, which by default were running Teredo.

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 often implies that those systems are usually managed
   by a single entity; therefore, audit trail and strict anti-spoofing
   are usually possible and this raises the overall security.

   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, 6VPE, and LDPv6

   Organizations using MPLS in their core can also use 6PE [RFC4798] and
   6VPE [RFC4659] to enable IPv6 access over MPLS. As 6PE and 6VPE are
   really similar to BGP/MPLS IP 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
      VRFs (only applicable to 6VPE);

   o Hiding the IPv4 core, hence removing all attacks against
      P-routers;

   o Securing the routing protocol between CE and PE; in the case of
      6PE and 6VPE, link-local addresses (see [RFC7404]) can be used and
      as these addresses cannot be reached from outside of the link, the
      security of 6PE and 6VPE is even higher than the IPv4 BGP/MPLS IP
      VPN.

   LDPv6 itself does not induce new risks, see also [RFC7552].

2.7.2.5. DS-Lite

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

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 should implement all the security considerations
   of [RFC7597]; notably, ensuring that the mapping of the IPv4 address
   and port are consistent with the configuration. As MAP has a
   predictable IPv4 address and port mapping, the audit logs are easier
   to 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 was historically
   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 Teredo easily traverses an IPv4 NAPT device thanks to its UDP
   encapsulation. Teredo tunnels connect a single host to the IPv6

Internet. Teredo shares the same issues as other tunnels: no
authentication, no confidentiality, possible spoofing and reflection
attacks.

IPsec [RFC4301] for the transported IPv6 traffic is recommended.

The biggest threat to Teredo is probably for 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
enough 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 no more enabled by default in
most environment, so, it is less of a threat, however, special
consideration must be taken in case of devices with older or non-
updated operating systems may be present, which by default were
running Teredo.

2.7.3. Translation Mechanisms

Translation mechanisms between IPv4 and IPv6 networks are alternate
coexistence strategies while networks transition to IPv6. While a
framework is described in [RFC6144] the specific security
considerations are documented 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.

While not really a transition mechanism to IPv6, this section also
includes the discussion about the use of heavy IPv4 to IPv4 network
address and port translation to prolong the life of IPv4-only
network.

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 and 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 and 464XLAT

Stateful NAT64 translation [RFC6146] allows IPv6-only clients to
contact IPv4 servers using unicast UDP, TCP, or ICMP. It can be used
in conjunction with DNS64 [RFC6147], a mechanism which synthesizes
AAAA records from existing A records. There is also a stateless
NAT64 [RFC7915] which 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. DNSSEC has an impact on DNS64 see section 3.1
of [RFC7050].

Another translation mechanism relying on a combination of stateful
and stateless translation, 464XLAT [RFC6877], can be used to do host
local translation from IPv4 to IPv6 and a network provider
translation from IPv6 to IPv4, i.e., giving IPv4-only application
access to IPv4-only server over an IPv6-only network. 464XLAT shares
the same security considerations as NAT64 and DNS64, however it can
be used without DNS64, avoiding the DNSSEC implications.

2.7.3.3. DS-Lite

Dual-Stack Lite (DS-Lite) [RFC6333] is a transition technique that
enables a service provider to share IPv4 addresses among customers by

combining two well-known technologies: IP in IP (IPv4-in-IPv6) and
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
Address Family Translation Router, AFTR [RFC6333] 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 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. This section also applies to the enterprise part
   of all ISP, i.e., the part of the network where the ISP employees are
   connected.

   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] and [RFC8190]

   o Accept certain ICMPv6 messages to allow proper operation of ND and
      PMTUD, see also [RFC4890] or [REY_PF] for hosts

   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.ietf-opsec-ipv6-eh-filtering]

   o Filter packets having an illegal IPv6 headers chain at the
      perimeter (and if possible, inside the network as well), see
      Section 2.2

   o Filter unneeded services at the perimeter

   o Implement ingress and egress anti-spoofing in the forwarding and
      control planes

       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.

   When site-to-site VPNs are used it should be kept in mind that, given
   the global scope of IPv6 global addresses as opposed to the common
   use of IPv4 private address space [RFC1918], sites might be able to
   communicate with each other over the Internet even when the VPN
   mechanism is not available and hence no traffic encryption is
   performed and traffic could be injected from the Internet into the
   site, see [WEBER_VPN]. It is recommended to filter at the Internet
   connection(s) packets having a source or destination address
   belonging to the site internal prefix(es); this should be done for
   ingress and egress traffic.

   Hosts need to be hardened directly through security policy to protect
   against security threats. The host firewall default capabilities
   have to be clearly understood. In some cases, 3rd party firewalls
   have no IPv6 support whereas the native firewall installed by default
   has IPv6 support. General device hardening guidelines are provided
   in xxref target="device"/>

   It should also be noted that many hosts still use IPv4 for
   transporting logs for RADIUS, TACACS+, SYSLOG, etc. Operators cannot
   rely on an IPv6-only security policy to secure such protocols that
   are still using IP4.

4. Service Providers Security Considerations

4.1. BGP

   The threats and mitigation techniques are identical between IPv4 and
   IPv6. Broadly speaking they are:

   o Authenticating the TCP session;

   o TTL security (which becomes hop-limit security in IPv6) as
      [RFC5082];

   o bogon AS filtering;

   o Prefix filtering.

   These are explained in more detail in Section 2.5. Also, the
   recommendations of [RFC7454] should be considered.

4.1.1. Remote Triggered Black Hole Filtering

   RTBH [RFC5635] works identically in IPv4 and IPv6. IANA has
   allocated the 100::/64 prefix to be used as the discard prefix
   [RFC6666].

4.2. Transition/Coexistence Mechanism

   SP will typically use transition mechanisms such as 6rd, 6PE, MAP,
   NAT64 which have been analyzed in the transition and coexistence i
   (Section 2.7).

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 IPv6 NAT on the CPE, IPv6 has the possibility 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.

Some residential users have 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 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 (including personal firewalls) are configured with a dual-
stack security policy.

If the residential CPE has IPv6 connectivity, [RFC7084] defines the
requirements of an IPv6 CPE and does not take 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 a 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 and some of them are dated 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 de Bruen, Tobias
   Fiebig, Fernando Gont, Jeffry Handal, Lee Howard, Panos Kampanakis,
   Erik Kline, Jouni Korhonen, Warren Kumari, Mark Lentczner, Jen
   Linkova (and her detailed review), Gyan S. Mishra, Jordi Palet, Bob
   Sleigh, Donal Smith, 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>.

   [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC
              2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174,
              May 2017, <https://www.rfc-editor.org/info/rfc8174>.

   [RFC8200] Deering, S. and R. Hinden, "Internet Protocol, Version 6
              (IPv6) Specification", STD 86, RFC 8200,
              DOI 10.17487/RFC8200, July 2017,
              <https://www.rfc-editor.org/info/rfc8200>.

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.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.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>.

   [REY_PF]   Rey, E., "Local Packet Filtering with IPv6", July 2017,
               <https://labs.ripe.net/Members/enno_rey/local-packet-
               filtering-with-ipv6>.

   [RFC0826]   Plummer, D., "An Ethernet Address Resolution Protocol: Or
               Converting Network Protocol Addresses to 48.bit Ethernet
               Address for Transmission on Ethernet Hardware", STD 37,
               RFC 826, DOI 10.17487/RFC0826, November 1982,
               <https://www.rfc-editor.org/info/rfc826>.

   [RFC1918]   Rekhter, Y., Moskowitz, B., Karrenberg, D., de Groot, G.,
               and E. Lear, "Address Allocation for Private Internets",
               BCP 5, RFC 1918, DOI 10.17487/RFC1918, February 1996,
               <https://www.rfc-editor.org/info/rfc1918>.

   [RFC2131]   Droms, R., "Dynamic Host Configuration Protocol",
               RFC 2131, DOI 10.17487/RFC2131, March 1997,
               <https://www.rfc-editor.org/info/rfc2131>.

   [RFC2460]   Deering, S. and R. Hinden, "Internet Protocol, Version 6
               (IPv6) Specification", RFC 2460, DOI 10.17487/RFC2460,
               December 1998, <https://www.rfc-editor.org/info/rfc2460>.

   [RFC2529]   Carpenter, B. and C. Jung, "Transmission of IPv6 over IPv4
               Domains without Explicit Tunnels", RFC 2529,
               DOI 10.17487/RFC2529, March 1999,
               <https://www.rfc-editor.org/info/rfc2529>.

   [RFC2784]   Farinacci, D., Li, T., Hanks, S., Meyer, D., and P.
               Traina, "Generic Routing Encapsulation (GRE)", RFC 2784,
               DOI 10.17487/RFC2784, March 2000,
               <https://www.rfc-editor.org/info/rfc2784>.

   [RFC2827]   Ferguson, P. and D. Senie, "Network Ingress Filtering:
               Defeating Denial of Service Attacks which employ IP Source
               Address Spoofing", BCP 38, RFC 2827, DOI 10.17487/RFC2827,
               May 2000, <https://www.rfc-editor.org/info/rfc2827>.

   [RFC2866]   Rigney, C., "RADIUS Accounting", RFC 2866,
               DOI 10.17487/RFC2866, June 2000,
               <https://www.rfc-editor.org/info/rfc2866>.

   [RFC3056] Carpenter, B. and K. Moore, "Connection of IPv6 Domains
              via IPv4 Clouds", RFC 3056, DOI 10.17487/RFC3056, February
              2001, <https://www.rfc-editor.org/info/rfc3056>.

   [RFC3068] Huitema, C., "An Anycast Prefix for 6to4 Relay Routers",
              RFC 3068, DOI 10.17487/RFC3068, June 2001,
              <https://www.rfc-editor.org/info/rfc3068>.

   [RFC3627] Savola, P., "Use of /127 Prefix Length Between Routers
              Considered Harmful", RFC 3627, DOI 10.17487/RFC3627,
              September 2003, <https://www.rfc-editor.org/info/rfc3627>.

   [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>.

   [RFC5082] Gill, V., Heasley, J., Meyer, D., Savola, P., Ed., and C.
              Pignataro, "The Generalized TTL Security Mechanism
              (GTSM)", RFC 5082, DOI 10.17487/RFC5082, October 2007,
              <https://www.rfc-editor.org/info/rfc5082>.

   [RFC5214] Templin, F., Gleeson, T., and D. Thaler, "Intra-Site
              Automatic Tunnel Addressing Protocol (ISATAP)", RFC 5214,
              DOI 10.17487/RFC5214, March 2008,
              <https://www.rfc-editor.org/info/rfc5214>.

   [RFC5340] Coltun, R., Ferguson, D., Moy, J., and A. Lindem, "OSPF
              for IPv6", RFC 5340, DOI 10.17487/RFC5340, July 2008,
              <https://www.rfc-editor.org/info/rfc5340>.

   [RFC5635] Kumari, W. and D. McPherson, "Remote Triggered Black Hole
              Filtering with Unicast Reverse Path Forwarding (uRPF)",
              RFC 5635, DOI 10.17487/RFC5635, August 2009,
              <https://www.rfc-editor.org/info/rfc5635>.

   [RFC5952] Kawamura, S. and M. Kawashima, "A Recommendation for IPv6
              Address Text Representation", RFC 5952,
              DOI 10.17487/RFC5952, August 2010,
              <https://www.rfc-editor.org/info/rfc5952>.

   [RFC5969] Townsley, W. and O. Troan, "IPv6 Rapid Deployment on IPv4
              Infrastructures (6rd) -- Protocol Specification",
              RFC 5969, DOI 10.17487/RFC5969, August 2010,
              <https://www.rfc-editor.org/info/rfc5969>.

   [RFC6092] Woodyatt, J., Ed., "Recommended Simple Security
              Capabilities in Customer Premises Equipment (CPE) for
              Providing Residential IPv6 Internet Service", RFC 6092,
              DOI 10.17487/RFC6092, January 2011,
              <https://www.rfc-editor.org/info/rfc6092>.

   [RFC6104] Chown, T. and S. Venaas, "Rogue IPv6 Router Advertisement
              Problem Statement", RFC 6104, DOI 10.17487/RFC6104,
              February 2011, <https://www.rfc-editor.org/info/rfc6104>.

   [RFC6105] Levy-Abegnoli, E., Van de Velde, G., Popoviciu, C., and J.
              Mohacsi, "IPv6 Router Advertisement Guard", RFC 6105,
              DOI 10.17487/RFC6105, February 2011,
              <https://www.rfc-editor.org/info/rfc6105>.

   [RFC6144] Baker, F., Li, X., Bao, C., and K. Yin, "Framework for
              IPv4/IPv6 Translation", RFC 6144, DOI 10.17487/RFC6144,
              April 2011, <https://www.rfc-editor.org/info/rfc6144>.

   [RFC6146] Bagnulo, M., Matthews, P., and I. van Beijnum, "Stateful
              NAT64: Network Address and Protocol Translation from IPv6
              Clients to IPv4 Servers", RFC 6146, DOI 10.17487/RFC6146,
              April 2011, <https://www.rfc-editor.org/info/rfc6146>.

   [RFC6147] Bagnulo, M., Sullivan, A., Matthews, P., and I. van
              Beijnum, "DNS64: DNS Extensions for Network Address
              Translation from IPv6 Clients to IPv4 Servers", RFC 6147,
              DOI 10.17487/RFC6147, April 2011,
              <https://www.rfc-editor.org/info/rfc6147>.

   [RFC6164] Kohno, M., Nitzan, B., Bush, R., Matsuzaki, Y., Colitti,
              L., and T. Narten, "Using 127-Bit IPv6 Prefixes on Inter-
              Router Links", RFC 6164, DOI 10.17487/RFC6164, April 2011,
              <https://www.rfc-editor.org/info/rfc6164>.

   [RFC6169] Krishnan, S., Thaler, D., and J. Hoagland, "Security
              Concerns with IP Tunneling", RFC 6169,
              DOI 10.17487/RFC6169, April 2011,
              <https://www.rfc-editor.org/info/rfc6169>.

   [RFC6192] Dugal, D., Pignataro, C., and R. Dunn, "Protecting the
              Router Control Plane", RFC 6192, DOI 10.17487/RFC6192,
              March 2011, <https://www.rfc-editor.org/info/rfc6192>.

   [RFC6221] Miles, D., Ed., Ooghe, S., Dec, W., Krishnan, S., and A.
              Kavanagh, "Lightweight DHCPv6 Relay Agent", RFC 6221,
              DOI 10.17487/RFC6221, May 2011,
              <https://www.rfc-editor.org/info/rfc6221>.

   [RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed.,
              and A. Bierman, Ed., "Network Configuration Protocol
              (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011,
              <https://www.rfc-editor.org/info/rfc6241>.

   [RFC6264] Jiang, S., Guo, D., and B. Carpenter, "An Incremental
              Carrier-Grade NAT (CGN) for IPv6 Transition", RFC 6264,
              DOI 10.17487/RFC6264, June 2011,
              <https://www.rfc-editor.org/info/rfc6264>.

   [RFC6269] Ford, M., Ed., Boucadair, M., Durand, A., Levis, P., and
              P. Roberts, "Issues with IP Address Sharing", RFC 6269,
              DOI 10.17487/RFC6269, June 2011,
              <https://www.rfc-editor.org/info/rfc6269>.

   [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>.

   [RFC6459] Korhonen, J., Ed., Soininen, J., Patil, B., Savolainen,
              T., Bajko, G., and K. Iisakkila, "IPv6 in 3rd Generation
              Partnership Project (3GPP) Evolved Packet System (EPS)",
              RFC 6459, DOI 10.17487/RFC6459, January 2012,
              <https://www.rfc-editor.org/info/rfc6459>.

   [RFC6547] George, W., "RFC 3627 to Historic Status", RFC 6547,
              DOI 10.17487/RFC6547, February 2012,
              <https://www.rfc-editor.org/info/rfc6547>.

   [RFC6564] Krishnan, S., Woodyatt, J., Kline, E., Hoagland, J., and
              M. Bhatia, "A Uniform Format for IPv6 Extension Headers",
              RFC 6564, DOI 10.17487/RFC6564, April 2012,
              <https://www.rfc-editor.org/info/rfc6564>.

   [RFC6583] Gashinsky, I., Jaeggli, J., and W. Kumari, "Operational
              Neighbor Discovery Problems", RFC 6583,
              DOI 10.17487/RFC6583, March 2012,
              <https://www.rfc-editor.org/info/rfc6583>.

   [RFC6598] Weil, J., Kuarsingh, V., Donley, C., Liljenstolpe, C., and
              M. Azinger, "IANA-Reserved IPv4 Prefix for Shared Address
              Space", BCP 153, RFC 6598, DOI 10.17487/RFC6598, April
              2012, <https://www.rfc-editor.org/info/rfc6598>.

   [RFC6620] Nordmark, E., Bagnulo, M., and E. Levy-Abegnoli, "FCFS
              SAVI: First-Come, First-Served Source Address Validation
              Improvement for Locally Assigned IPv6 Addresses",
              RFC 6620, DOI 10.17487/RFC6620, May 2012,
              <https://www.rfc-editor.org/info/rfc6620>.

   [RFC6666] Hilliard, N. and D. Freedman, "A Discard Prefix for IPv6",
              RFC 6666, DOI 10.17487/RFC6666, August 2012,
              <https://www.rfc-editor.org/info/rfc6666>.

   [RFC6762] Cheshire, S. and M. Krochmal, "Multicast DNS", RFC 6762,
              DOI 10.17487/RFC6762, February 2013,
              <https://www.rfc-editor.org/info/rfc6762>.

   [RFC6763] Cheshire, S. and M. Krochmal, "DNS-Based Service
              Discovery", RFC 6763, DOI 10.17487/RFC6763, February 2013,
              <https://www.rfc-editor.org/info/rfc6763>.

   [RFC6877] Mawatari, M., Kawashima, M., and C. Byrne, "464XLAT:
              Combination of Stateful and Stateless Translation",
              RFC 6877, DOI 10.17487/RFC6877, April 2013,
              <https://www.rfc-editor.org/info/rfc6877>.

   [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] Chittimaneni, K., Chown, T., Howard, L., Kuarsingh, V.,
              Pouffary, Y., and E. Vyncke, "Enterprise IPv6 Deployment
              Guidelines", RFC 7381, DOI 10.17487/RFC7381, October 2014,
              <https://www.rfc-editor.org/info/rfc7381>.

   [RFC7404] Behringer, M. and E. Vyncke, "Using Only Link-Local
              Addressing inside an IPv6 Network", RFC 7404,
              DOI 10.17487/RFC7404, November 2014,
              <https://www.rfc-editor.org/info/rfc7404>.

   [RFC7422] Donley, C., Grundemann, C., Sarawat, V., Sundaresan, K.,
              and O. Vautrin, "Deterministic Address Mapping to Reduce
              Logging in Carrier-Grade NAT Deployments", RFC 7422,
              DOI 10.17487/RFC7422, December 2014,
              <https://www.rfc-editor.org/info/rfc7422>.

   [RFC7454] Durand, J., Pepelnjak, I., and G. Doering, "BGP Operations
              and Security", BCP 194, RFC 7454, DOI 10.17487/RFC7454,
              February 2015, <https://www.rfc-editor.org/info/rfc7454>.

   [RFC7513] Bi, J., Wu, J., Yao, G., and F. Baker, "Source Address
              Validation Improvement (SAVI) Solution for DHCP",
              RFC 7513, DOI 10.17487/RFC7513, May 2015,
              <https://www.rfc-editor.org/info/rfc7513>.

   [RFC7526] Troan, O. and B. Carpenter, Ed., "Deprecating the Anycast
              Prefix for 6to4 Relay Routers", BCP 196, RFC 7526,
              DOI 10.17487/RFC7526, May 2015,
              <https://www.rfc-editor.org/info/rfc7526>.

   [RFC7552] Asati, R., Pignataro, C., Raza, K., Manral, V., and R.
              Papneja, "Updates to LDP for IPv6", RFC 7552,
              DOI 10.17487/RFC7552, June 2015,
              <https://www.rfc-editor.org/info/rfc7552>.

   [RFC7597] Troan, O., Ed., Dec, W., Li, X., Bao, C., Matsushima, S.,
              Murakami, T., and T. Taylor, Ed., "Mapping of Address and
              Port with Encapsulation (MAP-E)", RFC 7597,
              DOI 10.17487/RFC7597, July 2015,
              <https://www.rfc-editor.org/info/rfc7597>.

   [RFC7599] Li, X., Bao, C., Dec, W., Ed., Troan, O., Matsushima, S.,
              and T. Murakami, "Mapping of Address and Port using
              Translation (MAP-T)", RFC 7599, DOI 10.17487/RFC7599, July
              2015, <https://www.rfc-editor.org/info/rfc7599>.

   [RFC7610] Gont, F., Liu, W., and G. Van de Velde, "DHCPv6-Shield:
              Protecting against Rogue DHCPv6 Servers", BCP 199,
              RFC 7610, DOI 10.17487/RFC7610, August 2015,
              <https://www.rfc-editor.org/info/rfc7610>.

   [RFC7707] Gont, F. and T. Chown, "Network Reconnaissance in IPv6
              Networks", RFC 7707, DOI 10.17487/RFC7707, March 2016,
              <https://www.rfc-editor.org/info/rfc7707>.

   [RFC7721] Cooper, A., Gont, F., and D. Thaler, "Security and Privacy
              Considerations for IPv6 Address Generation Mechanisms",
              RFC 7721, DOI 10.17487/RFC7721, March 2016,
              <https://www.rfc-editor.org/info/rfc7721>.

   [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>.

   [RFC8190] Bonica, R., Cotton, M., Haberman, B., and L. Vegoda,
              "Updates to the Special-Purpose IP Address Registries",
              BCP 153, RFC 8190, DOI 10.17487/RFC8190, June 2017,
              <https://www.rfc-editor.org/info/rfc8190>.

   [RFC8210] Bush, R. and R. Austein, "The Resource Public Key
              Infrastructure (RPKI) to Router Protocol, Version 1",
              RFC 8210, DOI 10.17487/RFC8210, September 2017,
              <https://www.rfc-editor.org/info/rfc8210>.

   [RFC8273] Brzozowski, J. and G. Van de Velde, "Unique IPv6 Prefix
              per Host", RFC 8273, DOI 10.17487/RFC8273, December 2017,
              <https://www.rfc-editor.org/info/rfc8273>.

   [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]
              Barnes, R., Altmann, R., and D. Kerr, "Mapping the Great
              Void - Smarter scanning for IPv6", February 2012,
              <http://www.caida.org/workshops/isma/1202/slides/
              aims1202_rbarnes.pdf>.

   [WEBER_VPN]
              Weber, J., "Dynamic IPv6 Prefix - Problems and VPNs",
              March 2018, <https://blog.webernetz.net/wp-
              content/uploads/2018/03/TR18-Johannes-Weber-Dynamic-IPv6-
              Prefix-Problems-and-VPNs.pdf>.

Authors' Addresses

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

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

    Kiran Kumar Chittimaneni
    WeWork
    415 Mission St.
    San Francisco 94105
    USA

    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-Wuertemberg 69115
    Germany

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

Network Working Group   J. Paillisse
Internet-Draft   UPC-BarcelonaTech
Intended status: Informational   A. Rodriguez-Natal
Expires: May 2, 2018   V. Ermagan
                                                                F. Maino
                                                           Cisco Systems
                                                             A. Cabellos
                                                        UPC-BarcelonaTech
                                                        October 29, 2017

        An analysis of the applicability of blockchain to secure IP addresses
                     allocation, delegation and bindings.
                    draft-paillisse-sidrops-blockchain-01.txt

Abstract

   This document analyzes how blockchain technology can be used to
   secure the allocation, delegation and binding to topological
   information of the IP address space. The main outcomes of the
   analysis are that blockchain is suitable in environments with
   multiple distrusting parties and that Proof of Stake is a potential
   candidate for a consensus algorithm.

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 May 2, 2018.

Copyright Notice

Table of Contents

1. Introduction

   Blockchain [Bitcoin] is attracting a lot of attention among the
   security community since it provides means for exchanging information
   among a set of distrusting entities without the use of digital
   certificates and centralized control. Blockchain provides means for
   the distrusting parties to reach consensus in a distributed way.
   Formally, it is regarded as a new solution to the Byzantine Generals
   problem, well-known in fault-tolerant distributed systems.

   Although at the time of this writing the main application of
   blockchain are financial systems, their use in the field of
   networking is being explored (e.g., [Hari2016]). Some successful
   systems exist such as [Blockstack] and [Namecoin], which aim at
   building a secure DNS.

   The main goal of this document is to represent a first step towards
   the understanding of the properties of blockchains and their
   applicability in the Internet infrastructure, specifically securing
   the allocation, delegation and bindings of IP addresses. First, it
   introduces blockchain, then it analyzes how blockchain could be used
   to secure the delegation of IP addresses. Finally, it presents an
   initial design for such an infrastructure. This document also
   includes a preliminary security analysis of such system. It is
   important to note that the goal of this document is not to provide a
   complete architecture that secures IP address allocation, delegation
   and bindings.

2. Definition of Terms

   TBC

3. Blockchain in a nutshell

3.1. Overview

   Conceptually, a blockchain is a distributed, secure and trustless
   database. It can also be regarded as a state machine with rules that
   clearly state which transitions can be performed. Participants in
   the blockchain communicate through a P2P network. The smallest data
   unit of a blockchain is a transaction. Users attach data to a
   transaction along with its signature and their associated public key.
   Usually, the attached data is an asset or a token, something that is
   unique and should not be replicated (e.g., coins in Bitcoin). Then
   they broadcast this transaction to the other participants. The rest
   of the nodes in the network store temporarily this transaction. At
   some fixed intervals in time, one of the nodes takes a set of these
   transactions and groups them in a block. It then broadcasts this

block back to the network. When the other nodes receive this block
they verify it, remove the transactions contained in the block from
the temporary storage and add it after the previous block, thus
creating a chain of blocks. It should be noted that all nodes store
the entire blockchain locally. In addition, most blockchains give
some sort of reward to nodes that add new blocks, although this is
not strictly necessary. Figure 1 presents an overview of the most
common elements in a block.

```
+--------+--------------+----------------+--------------+
| Block  | Hash(Previous | Hash(All Block | Block Creator |
| Number |     Block)    | Transactions)  |   Signature   |
+--------+--------------+----------------+--------------+
|                     Transaction 1                     |
+------------------------------------------------------+
|                     Transaction 2                     |
+------------------------------------------------------+
+                        ...                            +
+                        ...                            +
+------------------------------------------------------+
|                     Transaction N                     |
+------------------------------------------------------+
```

Figure 1.- Common structure of a block

Two basic mechanisms are used to protect the chained data: a chain of
signatures and a consensus algorithm.

3.1.1. Chain of signatures

The chain of signatures operates at transaction level. Consider the
sender and receiver of a token, each with its public-private keypair.
To change the owner of a token, the sender signs the data and the
receiver's public key. It then puts together its public key, the
signature, the data and the hash of the receiver's public key
(Figure 2) to form a transaction.

```
+-------------+------------------+--------+--------------+
|    Sender   | Signature Sender |  Data  | Hash(Receiver|
|  Public Key |    Private Key   |        |   Public Key)|
+-------------+------------------+--------+--------------+
```

Figure 2.- Common transaction structure in a blockchain

In conclusion, the rules of the blockchain enforce that:

o The owner of the receiver private key has total control over the
   contents of the transaction. In Bitcoin this translates in a
   central property: only this owner can spend a coin.

o When an owner sends a token to the new owner, it irreversibly
   transfers the control of the contents to the new owner.

3.1.2. Consensus algorithm

The consensus algorithm is the central part of blockchain and it
controls the chaining of data blocks. The main role of the algorithm
is to provide a set of well-defined rules so that participants agree
on a consistent view of the database. For this it has the following
main functions. First, forks (multiple chains) can exist, this may
happen for instance due to varying network latency among
participants. In this case the participants must agree on which is
the valid chain. And second, another important function of the
consensus algorithm is to determine which participants are allowed to
add a new data blocks. Section 3.3 contains more information
regarding available consensus algorithms.

It is important to note that regardless of the consensus algorithm,
in blockchain data blocks are always added, never deleted nor
modified. This creates a tamper-proof, shared ledger among all
participants. Transactions can be tracked back by inspecting past
blocks, thus enabling the verification of claims by certain parties.

3.2. Features

The following list tries to briefly summarize the main
characteristics of the blockchain technology:

Decentralized: No central entity controls the blockchain, it is
   shared among all participants.

No CAs: No digital certificates, Certification Authorities or CRLs
   are needed.

Limited prior trust: It is not required to trust other nodes. It is
   worth noting that some consensus algorithms rely on some limited
   levels of trust.

Tamper-proof: Since data can be only added but never modified,
   attempts to alter previous records are detected.

Non-repudiation: All nodes share a common, immutable view on the
   status of the blockchain, and blockchain provides non-repudiation
   mechanisms.

Censorship-resistant: Gaining control over a transaction involves
   having access to the associated private key.

Append-only: Data is always added, but never modified nor deleted.

Privacy: Entities participating in the blockchain can achieve
   privacy using anonymous keys, i.e. randomly-generated keys not
   related to their identity. In addition, a new keypair should be
   generated for each new transaction in order to prevent tracking
   [Bitcoin], section 10.

Slow updates: New transactions have to be verified, added to a block
   and received by all nodes. This results in a delay since the
   transaction is created until it is finally available to all the
   nodes. This delay will depend on the consensus algorithm and the
   block creation rate.

Large storage: The size of the blockchain keeps growing forever,
   because data blocks are always added. This may result in
   scalability issues.

3.3. Description of consensus algorithms

   The two more popular consensus algorithms are: Proof of Work and
   Proof of Stake.

3.3.1. Proof of Work (PoW)

   In Proof of Work nodes have to solve a complex mathematical problem
   to add a block, thus requiring some computational effort, this is
   commonly know as mining. For example in Bitcoin the problem is to
   find a hash starting with a fixed amount of zeroes, the only known
   way to solve this problem is by brute force. The valid chain is the
   one with most accumulated computing power, this chain is also the
   more expensive in terms of computing power to modify. This is
   because modifying a block going N blocks back from the tip of the
   chain would require redoing the computations for all these N blocks.

As a result, an attacker should have more computational power than
the power required to create the N blocks to be able to modify the
chain. Overall, it is commonly assumed that if more than half of the
nodes are honest the blockchain is considered as secure.

PoW offers relevant features, adding new blocks requires an external
resource (CPU power) that has an economical cost. However this also
results in some relevant drawbacks:

Risk of overtaking: The security of PoW is entirely based on
   computation power. This means that if an entity has access to
   more than half of the total blockchain's computing power it can
   control the chain. As a result and in order to keep blockchain
   secure, the incentive of taking control of the chain must be lower
   than the cost of acquiring and operating the hardware that
   provides the equivalent to half of the participants computing
   power. This is hard to guarantee since the economy of the
   blockchain and the economy of the required hardware are
   independent. As an example an attacker can acquire the required
   hardware and operate it, take control of the blockchain to obtain
   an economical benefit and finally sell the hardware to reduce the
   final cost of the attack.

Hardware dependency: Bitcoin automatically increases -over time- the
   complexity of the mathematical problem that needs to be solved in
   order to add a block. This is done to account for Moore's law.
   As a result the community has designed mining specific hardware
   (ASICs) that provides a competitive advantage. In this context
   blockchain becomes less democratic, since the cost of
   participating in it increases.

Energy inefficiency: PoW requires large amounts of energy to perform
   the computations (e.g., [miningfarm]).

3.3.2. Proof of Stake (PoS)

   The main idea behind Proof of Stake is that participants with more
   assets (or stake) in the blockchain are more likely to add blocks.
   With this, the control of the chain is given to entities who own more
   stake. For each new block, a signer is selected randomly from the
   list of participants typically weighted according to their stake. A
   fundamental assumption behind PoS is that such entities have more
   incentives for honest behaviour since they have more assets in the
   chain.

   Proof of Stake is seen as an alternative to PoW. At the time of this
   writing major players in the blockchain environment such as
   [Ethereum] are preparing a shift towards PoS, moreover several

blockchains based on PoS already exist (eg. [Peercoin]). The main
reason behind this paradigm shift is that PoS addresses some of PoW's
main drawbacks:

o It does not require special hardware nor computationally or
   energy-expensive calculations.

o An attacker must get hold of a significant part of the assets in
   order to gain control of the blockchain. As opposed to PoS the
   investment required to gain control of the chain lies within the
   chain, and does not involve using external resources.

On the other side, Proof of Stake introduces new sources of attacks:

o In Proof of Stake the signer is selected randomly among the
   stakers. In this context attackers can manipulate the source of
   randomness to sign more blocks and ultimately gain control over
   the chain.

o As opposed to PoW, creating forks is very inexpensive, since no
   computational power is required. The PoS must provide means to
   select the valid chain, which is typically the longer one.

o Collusions of high-stakers can create alternate chains which can
   appear to be valid.

4. Blockchain for IP addresses

4.1. Problem statement

   The objective of this section is to analyze if an infrastructure
   using blockchain can provide a similar degree of security to
   traditional PKI-based architectures. Specifically we aim to secure:

   o Binding of IP address blocks to the holder (private key holder).

   o The allocations and delegations of IP address blocks among their
      holders.

   o Binding of IP address blocks to their topological locators (eg.
      AS numbers allocations).

   This information is public and shared among a set of distrusting
   entities over the Internet. The architecture must be able to:

   o Allow anyone to verify the legitimate holder of a block of
      addresses

o Let participating entities allocate address blocks without
   requiring a trusted third party.

o Restrict the allocation of a block of addresses to only its
   legitimate holder.

o Prevent data modification without the consent of its holder.

4.2. Analysis

The main rationale behind using blockchain to secure IP address
allocations is that IPs can be understood as coins, both concepts
share some fundamental characteristics:

o They are unambiguously allocated to entities.

o Can be transferred between them.

o Cannot be assigned to two entities at the same time.

o Can be divided up to a certain limit.

Such similar properties make it possible to envisage a blockchain
that allows its participants delegate IP address blocks, similarly to
how Bitcoin transfers coins. For example, IANA could write a
transaction allocating addresses to the RIRs, which in turn could
allocate them to the LIRs, etc. Complex management logic can be
defined as needed for example, rejecting a transaction that allocates
of a block of addresses originated by an entity that does not hold
that block. In addition, transactions accept multiple inputs and
outputs, i.e. an arbitrary amount of public keys either as senders or
receivers. This means that it is possible to break and merge blocks
of addresses as required. Section 5 provides more detailed
information about this architecture.

4.3. A consensus algorithm for IP addresses

As stated before, the consensus algorithm is a central part of a
blockchain. The first consensus algorithm designed for blockchain
was PoW, and it is a common choice for new blockchain
implementations. However it presents several drawbacks
(Section 3.3.1) for the IP address scenario.

Using computing power as a means to secure blockchains has been
proved to work in financial environments. However, the capability to
add new blocks and the security of the chain itself depends on the
computing power of the participants, which is not always aligned with
their interest in the well-being of the blockchain. Depending on the

objectives of the attacker, certain attacks can become profitable.
Namely, buying a large quantity of hardware to be able to rewrite the
blockchain with false data (e.g., incorrect delegations of IP
addresses). This is because the incentives of the participants of
the IP addresses blockchain are not linked with their computing
power.

In contrast, with Proof of Stake the capability to alter the
blockchain remains within it. This aspect is of particular
importance in the context of securing IP addresses: it would mean
that AS domains holding large blocks of IP addresses are more likely
to add blocks. These parties have a reduced incentive in tampering
the blockchain because they would suffer the consequences: an
insecure Internet. Typically ASes that hold large blocks of IP
address space have their business within the Internet and as such,
have clear incentives in the correct operation and security of the
Internet.

Furthermore, in such blockchain the risk of takeover is reduced
compared to PoW, the reason is that accumulating a large amount of IP
addresses is typically more complex than accumulating computing
power. The risk of takeover is also mitigated compared to other PoS-
based blockchains. In this blockchain an attacker would buy tokens
from the other parties, who receive a monetary compensation to
participate in the attack. However, in a blockchain for IP addresses
this would mean buying IP addresses from other parties, who do not
have a clear incentive to sell their blocks of addresses to the
attacker. Because of this, PoS appears to be a firm candidate for a
consensus algorithm in a blockchain for securing IP addresses
allocations and delegations.

5. Overview of the architecture

   This architecture mimics the hierarchy of IP address allocation
   present in today's Internet, with IANA on top of it. All nodes trust
   IANA's public key, which writes a genesis transaction assigning all
   of the address space to itself (figure 3).

```
+--------------+----------------+----------+----------------+
|    IANA      | Signature IANA | Allocate |   Hash(IANA    |
| Public Key 1 |  Private Key 1 | 0/0      |  Public Key 2) |
+--------------+----------------+----------+----------------+
```

Figure 3.- Genesis transaction

It then begins allocating each block of addresses to the IP address
holders. Each transaction allocates part of the address space to the
legitimate holder, and the rest of space is given back to IANA using
a new keypair (figure 4).

```
+--------------+----------------+----------+----------------+
|              |                | Rest of  | Hash(IANA      |
|    IANA      | Signature IANA | space    | Public Key 3)  |
| Public Key 2 | Private Key 2  +----------+----------------+
|              |                | Allocate | Hash(APNIC     |
|              |                | 1/8      | Public Key 1)  |
+------------+----------------+----------+----------------+
```

Figure 4.- Example allocation transaction

In turn, all the parties in the hierarchy allocate or delegate
address blocks following the current allocation hierarchy. When a
party wants to verify the allocation of a block of addresses, it
downloads the blockchain and verifies all the blocks and transactions
up to the genesis block, for which it has trust. Figure 5 presents
an example of allocation of one prefix to each of the RIRs.

```
+-------------+----------------+----------+---------------+
|             |                | Rest of  | Hash(IANA     |
|             |                | space    | Public Key 4) |
|             |                +----------+---------------+
|             |                | Allocate | Hash(RIPE     |
|             |                | 5/8      | Public Key 1) |
|             |                +----------+---------------+
|             |                | Allocate | Hash(APNIC    |
|    IANA     | Signature IANA | 14/8     | Public Key 2) |
| Public Key 3| Private Key 3  +----------+---------------+
|             |                | Allocate | Hash(ARIN     |
|             |                | 23/8     | Public Key 1) |
|             |                +----------+---------------+
|             |                | Allocate | Hash(AFRINIC  |
|             |                | 102/8    | Public Key 1) |
|             |                +----------+---------------+
|             |                | Allocate | Hash(LACNIC   |
|             |                | 200/8    | Public Key 1) |
+-------------+----------------+----------+---------------+
```

Figure 5.- Example multi-output allocation transaction

Inside the blockchain the typical operations to manage blocks of IP
addresses can be defined, such as the delegation of prefixes (figure
6). This helps to enforce the rules of IP addresses management. For
instance, since this transaction is marked as a delegation, if the
new owner created an allocation transaction it would be rejected by
the other nodes, because the parent transaction does not have the
privileges to perform it.


```
+-------------+----------------+----------+---------------+
|             |                | Rest of  | Hash(APNIC    |
|    APNIC    | Signature APNIC | space    | Public Key 3) |
| Public Key 1| Private Key 1  +----------+---------------+
|             |                | Delegate | Hash(ISP A    |
|             |                | 1.2/16   | Public Key 1) |
+-------------+----------------+----------+---------------+
```

Figure 6.- Example delegation transaction

Performing a key rollover is simple, because each transaction has its
associated public key, and only depends on the previous transaction.
In other words, rekeying means changing the public key only in the

holder's transaction. This can be done adding a new transaction with
the same data but transferring it to a new public key also controlled
by the initial holder (figure 7). This approach lets each entity
decide its rekeying policies independently.

```
+--------------+----------------+----------+---------------+
|   ISP A      | Signature ISP A | Delegate | Hash(ISP A    |
| Public Key 1 |  Private Key 1  | 1.2/16   | Public Key 2) |
+--------------+----------------+----------+---------------+
```

Figure 7.- Example key rollover of a prefix delegation

It is worth noting that this chain can define as many operations as
required, for instance storing the binding of AS numbers to the IP
prefixes they announce (figure 8).

```
+--------------+----------------+----------+---------------+
|              |   Signature    | Bind     |               |
|   ISP A      |    ISP A       | 1.2/16   | Hash(ISP A    |
| Public Key 2 | Private Key 2  | AS no.   | Public Key 3) |
|              |                | 12345    |               |
+--------------+----------------+----------+---------------+
```

Figure 8.- Example binding of AS number to prefix

Additional and more complex operations can be defined if the
management logic requires it. For instance, several signatures (from
different parties) can be required to consider a transaction valid,
etc.

5.1. Pros and cons

In this section we analyze the pros and cons of this architecture
compared to traditional PKI infraestructures:

Advantages:

o Decentralized: No central entity controls the blockchain, it is
   shared among all participants.

o No CAs, CRLs or certificates needed: No digital certificates,
   Certification Authorities or CRLs are needed.

   o Simplified rekeying: A key rollover can easily be performed by
      issuing a new transaction allocating the prefixes to a new keypair
      controlled by the same holder. This process can be performed
      without involving any third-party.

   o Censorship-resistant: since the control of a transaction is
      completely under the holder of the private key, the revocation of
      IP addresses without the legitimate holder's permission involves
      obtaining its private key. Even if the private key of the
      previous owner was compromised, ownership of the current
      transaction is still preserved, as opposed to the compromise of a
      CA's private key (or a misbehaving CA).

   o Limited prior trust: It is not required to trust other nodes.
      However, in PoS it is necessary to periodically authenticate the
      chain state out-of-band to prevent some attacks.

   o Simplified management: since CAs are not required, their
      management overhead is avoided.

   o Auditable: allocations and delegations can be tracked back in the
      blockchain to determine if they originate from the legitimate
      holder.

   Drawbacks:

   o PoS does not rely on strong cryptographic guarantees: As opposed
      to PKI-based systems that rely on strong and well-established
      cryptographic mechanisms, PoS-based infraestructures ultimately
      rely on the good behaviour of the high-stakers.

   o Costly bootstrapping: When a node is activated it has to download
      and verify the entire blockchain.

   o Large storage required: The blockchain grows forever as more
      blocks are added, blocks cannot be removed.

5.2. Security evaluation

5.2.1. Attacks against a PoS-based consensus algorithm

   This section presents a list of the most relevant attacks against a
   Proof of Stake algorithm and how to mitigate them.

5.2.1.1. Stake grinding

   Stake grinding refers to the manipulation of the consensus algorithm
   in order to progressively obtain more stake, with the goal of signing
   blocks more frequently with the ultimate goal of taking control of
   the blockchain. It proceeds as follows: when the attacker has to
   sign a block, it computes all the possible blocks (varying the data
   inside them) to find a combination that gives the highest possibility
   of signing another block in the future. It then signs this block and
   sends it to the network. This procedure is repeated for all the next
   signing opportunities. Over time, the attacker will sign more and
   more blocks until the consensus algorithm will always select the
   attacker to sign all blocks, thereby having taken control of the
   blockchain.

   To prevent this attack, the source of randomness used to select the
   signers has to be hard to alter or to predict.

5.2.1.2. Nothing at stake

   Nothing at stake is one of the fundamental drawbacks of Proof of
   Stake and requires careful design based on the incentives of the
   participants. In common PoS designs, the signers of the new block
   receive an economical incentive (e.g., Ethereum). However this does
   not hold in the IP address scenario, since participants should not
   receive any incentive. The incentive is, as stated before, achieving
   a consistent view of the IP address space and having a secure
   Internet.

5.2.1.3. Range attacks

   A range attack is performed by creating a fork some blocks back from
   the tip of the chain. It is conceptually similar to the attack named
   as 'Risk of overtaking' in Section 3.3.1. In this scenario, the
   attacker has privately fabricated a chain which (according to the
   consensus algorithm rules) will be selected over the original one.
   Benefits of this attack include gaining more stake on the blockchain
   (this attack could be part of a stake grinding attack) or rewriting
   the transaction history to erase a payment made in the original
   blockchain.

   The simplest solution to this attack is adding a revert limit to the
   blockchain, forbidding forks going back more than N blocks. This
   provides a means to solidify the blockchain. However, nodes that
   have been offline for more than N blocks will need an external source
   that indicates the correct chain. It has been proposed to do this
   out of band. This is why a PoS blockchain is not purely trustless
   and requires a small amount of trust.

5.2.1.4. Lack of participation

   Participants in a PoS algorithm will not always sign a block, since
   they might be offline when they are selected or lack incentives.
   Because of this, the final fraction of high-stakers that sign blocks
   can be very different from the full set of high-stakers. The direct
   consequence of this situation is that the portion of participants
   that decide what goes into the blockchain can be a small set of
   nodes. If this participation is low enough, it can leave the control
   of the blockchain to a small amount of people/oligarchy, thus rising
   security concerns.

5.2.2. Attacks against the P2P network

   This section presents attacks directed towards the underlying P2P
   network used to exchange information among the participants of the
   blockchain.

5.2.2.1. DDOS attacks

   Since blockchains are inherently based on P2P architectures, they
   present a higher degree of resistance to DDOS attacks than
   centralized server architectures, provided that the network has a
   significant number of participants. In addition, it is always
   possible to keep an offline copy of the blockchain.

5.2.2.2. Transaction flooding

   A special type of DDOS attack consists in creating a large amount of
   legit transactions that transfer a small amount of tokens (i.e.
   delegate a lot of small IP prefixes). If the number of transactions
   is large enough, the addition of new transactions can be
   significantly delayed because not all of them fit into a single
   block. The effectiveness of the attack also depends on the
   throughput of the blockchain (transactions/second). Simple solutions
   may be to limit the granularity upon which IP addresses can be split.
   Of course, only the legitimate holder of a large amount of IP address
   can perform this attack.

5.2.2.3. Routing attacks

   The underlying P2P network in blockchains does not typically use any
   security mechanism, e.g. node authentication or integrity of network
   protocol messages. This enables potentially disruptive attacks. For
   example, specially located rogue nodes could drop new transactions,
   which would block updates on the blockchain and leave legit nodes
   uncommunicated. The effectiveness of this kind of attacks depends on

how the P2P algorithm selects peers and the topology of the P2P
network.

However, the most potentially dangerous attack of this type are
network partitions, i.e. isolating a group of nodes from the rest of
the network so they cannot communicate each other (e.g.,
[Apostolaki2017]). The consequence of this attack is that two
versions of the blockchain are created, one at each network
partition. When the partition disappears and the nodes reconnect one
of the two chains will be discarded, causing a service disruption.
It is worth noting that Bitcoin has suffered similar attacks
[realrouteattack].

5.2.2.4. Transaction censorship

When a node adds a block it chooses arbitrarily which transactions
are added into it, i.e. no specific rules control how transactions
are added to a block. This enables a node to selectively add some
transactions and intentionally exclude others, with the consequence
that some transactions may be never added to the blockchain. In the
context of IP addresses, this may be performed by a competing ISP to
prevent another ISP from executing a certain modification. Possible
solutions revolve around:

o Giving more priority to older transactions (similarly to Bitcoin).

o Punishing nodes that exhibit this kind of behaviour, e.g. removing
   part of their block of IP addresses or lowering their chance of
   adding blocks.

6. Other Considerations

6.1. Revocation

Due to the irreversible nature of transactions, once a block of IP
addresses has been allocated to an entity it is not possible to
modify or remove it, as opposed to CRLs (Certificate Revocation
Lists). However, due to operational issues (compromised or lost
keys, human mistake, holder misbehaviour, etc) it is critical to
provide a way to recover a block of addresses. Moreover, since IP
addresses are a finite public good they cannot be lost. Taking into
account that a blockchain can enforce any rules its participants
agree upon, this section presents some possible approaches to
implement revocation, such approaches should not be considered as
mutually exclusive. The revocation procedure must be discussed among
the community to achieve consensus between the relevant players
(IANA, RIRs, ISPs, institutions, etc).

All these mechanisms present different balances of power between the current holder and the entity whose asset is being revoked. Behind all these mechanisms there is a fundamental tradeoff between trusting an upstream provider of the addresses and retaining full control of the block of addresses.

Regardless of the revocation policy and as opposed to traditional PKI systems, each IP prefix delegation only depends on the private key of the holder of such IP block. As such, it does not need to trust a CA or a chain of certificates. Only by means of this private key the IP delegation can be altered.

6.1.1. Expiration time

A simple approach to allow revocation is adding a lease time (i.e, time-to-live) to the blocks of addresses. After the lease ends, the new holder of the address block automatically becomes the previous one, or addresses are transferred to a default holder. As stated before, this revocation procedure should be enforced by the rules of the blockchain, this means that participants would not recognise expired allocations as valid.

6.1.2. Multi-signature transactions

A multi-signature transaction is a transaction that admits more than one authorized signer. In other words, a transaction is considered valid if it has, for instance, 2 out of 5 valid signatures. This way, 3 keys can be lost but with the renaming 2 keys the block of addresses can be recovered. This approach exemplifies the aforementioned tradeoff in trust, since the holder of the block of addresses must trust the owners of the keys participating in the multi-signature transaction.

6.1.3. Revocation transaction

A simpler approach than multi-signature transactions is creating a 'revocation' transaction. When a block of address is required to be reassigned without the consent of the current holder, a revocation transaction (specifying the new holder) is inserted in the blockchain. This transaction should be issued either by a consensuated authority or by a disputing entity. The revocation transaction should be resolved by either accepting the revocation transaction automatically when issued by the accepted authority or by means of out-of-band mechanisms when issued by a disputing party.

6.1.4. Out-of-band mechanisms

   Disputes regarding transactions can be resolved by means of out-of-
   band mechanisms, e.g, discussion, court, etc. In order to reflect
   the decision of this out-of-band mechanism the blockchain must be
   modified. Since this represents a deviation from the rules, it must
   be done through a hard blockchain fork. Although cumbersome and
   complex, this is feasible from a technical standpoint.

6.2. Storage management

   The never ending size of the blockchain presents a potential
   scalability issue. At the time of this writing, mature blockchains
   like Bitcoin require more than 100 GB of storage. Simply deleting or
   summarizing old transactions degrades the security of PoW-based
   chains, since their security relies on the computing power required
   to generate them. The longer they are, the harder they are to
   attack.

   However, PoS-based chains do not rely on computing power and hence,
   space-saving strategies do not degrade the security. For instance a
   simple solution could be to, once the PoS-based chain reaches a
   certain storage size, summarize a subset of the older transactions.
   In what follows we overview this strategy:

```
+-----+-----+-----+-----+---------------+-----+-----+-----+
|  0  |  1  |  2  |  3  |      .....    |47832|47833|47834|
+-----+-----+-----+-----+---------------+-----+-----+-----+
```

9.1 Old blockchain

```
+-----+-----+-----+-----+-----+
| r0  | r1  | r2  | r3  | r4  |
+-----+-----+-----+-----+-----+
```

9.2 Write present state to special reset blocks

```
+-----+-----+-----+-----+-----+-----+-----+
| r0  | r1  | r2  | r3  | r4  |  0  |  1  | ...
+-----+-----+-----+-----+-----+-----+-----+
```

9.3 Continue working after the reset blocks

Figure 9.- A simple technique to reduce blockchain storage

This approach reduces bootstrapping cost, it is worth noting that
this strategy requires trust on the reset blocks, such blocks can be
obtained with an out-of-band mechanism (see Section 5.2.1.3 for
further information).

6.3. Proof of Networking?

In this section we speculate how one could design an equivalent of
Proof-of-Work (PoW) for networks. Conceptually, PoW is a proof of
computational resources, can we devise a proof of networking
resources? It could be thought that a PoW equivalent may exist in
the context of networks, i.e., an equivalent to spending computer
cycles in a network. Some resources unique to networks are
bandwidth, computation of checksums, number of BGP peers, etc.
Hence, we could envisage a blockchain secured by the resources
inherent to its participating computer networks. As long as half of
the resources were controlled by honest members, security is
guaranteed. For example, bandwidth could be a potential candidate;
however it does not satisfy two key features present in PoW:

o Asymmetry: the proof has to be hard to generate but fast to
   verify.

o Verifiability: it has to be possible to embed the proof in the
   block in order to account for the spending of resources.

In this context, Proof-of-Networking is an open research issue .

6.4. Configuration parameters

Configuration parameters refer to a set of values:

o Block creation rate

o Maximum block size

o Other parameters related to the consensus algorithm

These parameters, beyond regulating the operation of the blockchain
also have an influence on its performance. For example, a small
block size increases propagation speed (thus consensus can be reached
faster) but reduces the number of transactions per second that the
blockchain can handle. As an example, in Bitcoin, the 10-minute
block creation rate seeks to balance fast confirmation times and
reduced probability of forks [Antonopoulos2015]. Experimental
deployments and operational requirements should help tuning such
parameters.

7. Security Considerations

This document aims to understand the security implications of using
the blockchain technology to secure IP addresses allocation.

8. IANA Considerations

This memo includes no request to IANA.

9. Acknowledgements

The authors wish to thank Jordi Herrera-Joancomarti, Andreu
Rodriguez-Donaire and Jordi Baylina for their helpful discussions
about Bitcoin and blockchain technology, as well as Leo Vegoda for
his insights regarding revocation mechanisms.

10. Informative References

[Antonopoulos2015]
          Antonopoulos, A., "Mastering Bitcoin, available online:
          http://chimera.labs.oreilly.com/books/1234000001802/
          index.html", 2015.

[Apostolaki2017]
          Apostolaki, M., Zohar, A., and L. Vanbever, "Hijacking
          Bitcoin: Routing Attacks on Cryptocurrencies. 2017 IEEE
          Symposium on Security and Privacy (SP).", 2017.

[Bitcoin] Nakamoto, S., "Bitcoin: A Peer-to-Peer Electronic Cash
          System. https://bitcoin.org/bitcoin.pdf", 2008.

[Blockstack]
          Ali, et al., M., "Blockstack : A Global Naming and Storage
          System Secured by Blockchains, USENIX Annual Technical
          Conference", 2016.

[Ethereum]
          The Ethereum project, "https://www.ethereum.org/", 2016.

[Hari2016]
          Hari, A. and T. Lakshman, "The Internet Blockchain: A
          Distributed, Tamper-Resistant Transaction Framework for
          the Internet. Fifteenth ACM Workshop on Hot Topics in
          Networks", 2016.

[miningfarm]
          Inside a mining farm, "http://www.bbc.com/future/
          story/20160504-we-looked-inside-a-secret-chinese-bitcoin-
          mine", 2016.

[Namecoin]
          Namecoin, "https://namecoin.org/", 2011.

[Peercoin]
          The Peercoin cryptocurrency, "https://peercoin.net/",
          2016.

[realrouteattack]
          Hacker Redirects Traffic From 19 Internet Providers to
          Steal Bitcoins, "https://www.wired.com/2014/08/isp-
          bitcoin-theft/", 2014.

Authors' Addresses

   Jordi Paillisse
   UPC-BarcelonaTech
   c/ Jordi Girona 1-3
   Barcelona, Catalonia 08034
   Spain

   Email: jordip@ac.upc.edu

   Alberto Rodriguez-Natal
   Cisco Systems
   170 Tasman Drive
   San Jose, CA
   USA

   Email: natal@cisco.com

   Vina Ermagan
   Cisco Systems
   170 Tasman Drive
   San Jose, CA
   USA

   Email: vermagan@cisco.com

   Fabio Maino
   Cisco Systems
   170 Tasman Drive
   San Jose, CA
   USA

   Email: fmaino@cisco.com

   Albert Cabellos
   UPC-BarcelonaTech
   c/ Jordi Girona 1-3
   Barcelona, Catalonia 08034
   Spain

   Email: acabello@ac.upc.edu

Network Working Group   J. Paillisse
Internet-Draft   UPC-BarcelonaTech
Intended status: Informational   A. Rodriguez-Natal
Expires: December 30, 2018   V. Ermagan
                                                            F. Maino
                                                        Cisco Systems
                                                           L. Vegoda
                                                          Individual
                                                         A. Cabellos
                                                    UPC-BarcelonaTech
                                                       June 28, 2018

          An analysis of the applicability of blockchain to secure IP addresses
                        allocation, delegation and bindings.
                     draft-paillisse-sidrops-blockchain-02

   Abstract

   This document analyzes how blockchain technology can be used to
   secure the allocation, delegation and binding to topological
   information of the IP address space. The main outcomes of the
   analysis are that blockchain is suitable in environments with
   multiple distrusting parties and that Proof of Stake is a potential
   candidate for a consensus algorithm.

Copyright Notice

Table of Contents

1. Introduction

   Blockchain [Bitcoin] is attracting a lot of attention among the
   security community since it provides means for exchanging information
   among a set of distrusting entities without the use of digital
   certificates and centralized control. Blockchain provides means for
   the distrusting parties to reach consensus in a distributed way.
   Formally, it is regarded as a new solution to the Byzantine Generals
   problem, well-known in fault-tolerant distributed systems
   [Byzantine].

   Although at the time of this writing the main application of
   blockchain are financial systems, their use in the field of
   networking is being explored (e.g., [Hari2016]). Some successful
   systems exist such as [Blockstack] and [Namecoin], which aim at
   building a secure naming system, providing a similar functionality to
   that of DNSSEC.

   The main goal of this document is to represent a first step towards
   the understanding of the properties of blockchains and their
   applicability in the Internet infrastructure, specifically securing
   the allocation, delegation and bindings of IP addresses. First, it
   introduces blockchain, then it analyzes how blockchain could be used
   to secure the delegation of IP addresses. Finally, it presents an
   initial design for such an infrastructure. This document also
   includes a preliminary security analysis of such system. It is
   important to note that the goal of this document is not to provide a
   complete architecture that secures IP address allocation, delegation
   and bindings.

2. Definition of Terms

   TBC

3. Blockchain in a nutshell

3.1. Overview

   Conceptually, a blockchain is a distributed, secure and trustless
   database. It can also be regarded as a state machine with rules that
   clearly state which transitions can be performed. Participants in
   the blockchain communicate through a P2P network. The smallest data
   unit of a blockchain is a transaction. Users attach data to a
   transaction along with its signature and their associated public key.
   Usually, the attached data is an asset or a token, something that is
   unique and should not be replicated (e.g., coins in Bitcoin). Then
   they broadcast this transaction to the other participants. The rest
   of the nodes in the network temporarily store this transaction. At
   some fixed intervals in time, one of the nodes takes a set of these
   transactions and groups them in a block. It then broadcasts this
   block back to the network. When the other nodes receive this block
   they verify it, remove the transactions contained in the block from
   the temporary storage and add it after the previous block, thus
   creating a chain of blocks. It should be noted that all nodes store
   the entire blockchain locally. In addition, most blockchains give
   some sort of reward to nodes that add new blocks, although this is
   not strictly necessary. Figure 1 presents an overview of the most
   common elements in a block.

```
+--------+---------------+----------------+---------------+
| Block  | Hash(Previous | Hash(All Block | Block Creator |
| Number |     Block)    |  Transactions) |   Signature   |
+--------+---------------+----------------+---------------+
|                      Transaction 1                      |
+---------------------------------------------------------+
|                      Transaction 2                      |
+---------------------------------------------------------+
+                          ...                            +
+                          ...                            +
+---------------------------------------------------------+
|                      Transaction N                      |
+---------------------------------------------------------+
```


                   Figure 1.- Common structure of a block

   Two basic mechanisms are used to protect the chained data: a chain of
   signatures and a consensus algorithm.

3.1.1. Chain of signatures

The chain of signatures operates at transaction level. Consider the
sender and receiver of a token, each with its public-private keypair.
To change the owner of a token, the sender signs the data and the
receiver's public key. It then puts together its public key, the
signature, the data and the hash of the receiver's public key (Figure
2) to form a transaction.

```
+-------------+------------------+--------+---------------+
|   Sender    | Signature Sender |  Data  | Hash(Receiver |
| Public Key  |   Private Key    |        |  Public Key)  |
+-------------+------------------+--------+---------------+
```

Figure 2.- Common transaction structure in a blockchain

In conclusion, the rules of the blockchain enforce that:

o The owner of the receiver private key has total control over the
   contents of the transaction. In Bitcoin this translates in a
   central property: only this owner can spend a coin.

o When an owner sends a token to the new owner, it irreversibly
   transfers the control of the contents to the new owner.

3.1.2. Consensus algorithm

The consensus algorithm is the central part of blockchain and it
controls the chaining of data blocks. The main role of the algorithm
is to provide a set of well-defined rules so that participants agree
on a consistent view of the database. For this it has the following
main functions. First, forks (multiple chains) can exist. This may
happen for instance due to varying network latency among
participants. In this case the participants must agree on which is
the valid chain. And second, another important function of the
consensus algorithm is to determine which participants are allowed to
add new data blocks. Section 3.3 contains more information regarding
available consensus algorithms.

It is important to note that regardless of the consensus algorithm,
in blockchain data blocks are always added, never deleted nor
modified. This creates a tamper-proof, shared ledger among all
participants. Transactions can be tracked back by inspecting past
blocks, thus enabling the verification of claims by certain parties.

3.2. Features

   The following list tries to briefly summarize the main
   characteristics of the blockchain technology:

   Decentralized: No central entity controls the blockchain, it is
      shared among all participants.

   No CAs: No digital certificates, Certification Authorities or CRLs
      are needed.

   Limited prior trust: It is not required to trust other nodes. It is
      worth noting that some consensus algorithms rely on some limited
      levels of trust.

   Tamper-proof: Since data can be only added but never modified,
      attempts to alter previous records are detected.

   Non-repudiation: All nodes share a common, immutable view on the
      status of the blockchain, and blockchain provides non-repudiation
      mechanisms.

   Censorship-resistant: Gaining control over a transaction involves
      having access to the associated private key.

   Append-only: Data is always added, but never modified nor deleted.

   Privacy: Entities participating in the blockchain can achieve
      privacy using anonymous keys, i.e. randomly-generated keys not
      related to their identity. In addition, a new keypair should be
      generated for each new transaction in order to prevent tracking
      [Bitcoin], section 10.

   Slow updates: New transactions have to be verified, added to a block
      and received by all nodes. This results in a delay since the
      transaction is created until it is finally available to all the
      nodes. This delay will depend on the consensus algorithm and the
      block creation rate.

   Large storage: The size of the blockchain keeps growing forever,
      because data blocks are always added. This may result in
      scalability issues.

3.3. Description of consensus algorithms

   The two more popular consensus algorithms are: Proof of Work and
   Proof of Stake.

3.3.1. Proof of Work (PoW)

   In Proof of Work nodes have to solve a complex mathematical problem
   to add a block, requiring some computational effort. This is
   commonly known as mining. For example in Bitcoin the problem is to
   find a hash starting with a fixed amount of zeroes, the only known
   way to solve this problem is by brute force. The valid chain is the
   one with most accumulated computing power, this chain is also the
   more expensive in terms of computing power to modify. This is
   because modifying a block going N blocks back from the tip of the
   chain would require redoing the computations for all these N blocks.
   As a result, an attacker should have more computational power than
   the power required to create the N blocks to be able to modify the
   chain. Overall, it is commonly assumed that if more than half of the
   nodes are honest, the blockchain is considered secure.

   PoW offers relevant features, adding new blocks requires an external
   resource (CPU power) that has an economical cost. However this also
   results in some relevant drawbacks:

   Risk of takeover: The security of PoW is entirely based on
      computation power. This means that if an entity has access to
      more than half of the total blockchain's computing power it can
      control the chain. As a result and in order to keep blockchain
      secure, the incentive of taking control of the chain must be lower
      than the cost of acquiring and operating the hardware that
      provides the equivalent to half of the participants computing
      power. This is hard to guarantee since the economy of the
      blockchain and the economy of the required hardware are
      independent. As an example an attacker can acquire the required
      hardware and operate it, take control of the blockchain to obtain
      an economical benefit and finally sell the hardware to reduce the
      final cost of the attack.

   Hardware dependency: Bitcoin automatically increases -over time- the
      complexity of the mathematical problem that needs to be solved in
      order to add a block. This is done to account for Moore's law.
      As a result the community has designed mining specific hardware
      (ASICs) that provides a competitive advantage. In this context
      blockchain becomes less democratic, since the cost of
      participating in it increases. On the other hand, several ASIC-
      resistant algorithms are in use in various cryptocurrencies. This
      is usually achieved with memory-intensive calculations or

frequently changing the mining algorithm. Although they appear to
be a promising alternative, vendors react by developing a silicon
implementation of the algorithm. In this situation, the
developers usually change the algorithm by means of a hard fork
[monero]. Ultimately, this becomes an arms-race.

Energy inefficiency: PoW requires large amounts of energy to perform
the computations (e.g., [miningfarm]).

3.3.2. Proof of Stake (PoS)

The main idea behind Proof of Stake is that participants with more
assets (or stake) in the blockchain are more likely to add blocks.
With this, the control of the chain is given to entities who own more
stake. For each new block, a signer is pseudo-randomly selected from
the list of participants typically weighted according to their stake.
A fundamental assumption behind PoS is that such entities have more
incentives for honest behaviour since they have more assets in the
chain.

Proof of Stake is seen as an alternative to PoW. At the time of this
writing, major players in the blockchain environment (such as
[Ethereum]) are preparing a shift towards PoS. Moreover, several
blockchains based on PoS already exist (eg. [Peercoin]). The main
reason behind this paradigm shift is that PoS addresses some of PoW's
main drawbacks:

o It does not require special hardware nor computationally or
   energy-expensive calculations.

o An attacker must get hold of a significant part of the assets in
   order to gain control of the blockchain. As opposed to PoS the
   investment required to gain control of the chain lies within the
   chain, and does not involve using external resources.

On the other side, Proof of Stake introduces new sources of attacks:

o In Proof of Stake the signer is selected randomly among the
   stakers. In this context attackers can manipulate the source of
   randomness to sign more blocks and ultimately gain control over
   the chain.

o As opposed to PoW, creating forks is very inexpensive, since no
   computational power is required. The PoS must provide means to
   select the valid chain, which is typically the longer one.

o Collusions of high-stakers can create alternate chains which can
   appear to be valid.

4. Blockchain for IP addresses

4.1. Problem statement

   The objective of this section is to analyze if an infrastructure
   using blockchain can provide a similar degree of security to
   traditional PKI-based architectures. Specifically we aim to secure:

   o Binding of IP address blocks to the holder (private key holder).

   o The allocations and delegations of IP address blocks among their
      holders.

   o Binding of IP address blocks to their topological locators (eg.
      AS numbers allocations).

   This information is public and shared among a set of distrusting
   entities over the Internet. The architecture must be able to:

   o Allow anyone to verify the legitimate holder of a block of
      addresses

   o Let participating entities allocate address blocks without
      requiring a trusted third party.

   o Restrict the allocation of a block of addresses to only its
      legitimate holder.

   o Prevent data modification without the consent of its holder.

4.2. Analysis

   The main rationale behind using blockchain to secure IP address
   allocations is that IPs can be understood as coins, both concepts
   share some fundamental characteristics:

   o They are unambiguously allocated to entities.

   o Can be transferred between them.

   o Cannot be assigned to two entities at the same time.

   o Can be divided up to a certain limit.

   Such similar properties make it possible to envisage a blockchain
   that allows its participants delegate IP address blocks, similarly to
   how Bitcoin transfers coins. For example, IANA could write a
   transaction allocating addresses to the RIRs, which in turn could

allocate them to the LIRs, etc. Complex management logic can be
defined as needed. (For example, rejecting a transaction that
allocates of a block of addresses originated by an entity that does
not hold that block.) In addition, transactions accept multiple
inputs and outputs, i.e. an arbitrary amount of public keys either
as senders or receivers. This means that it is possible to break and
merge blocks of addresses as required. Section 5 provides more
detailed information about this architecture.

4.3. A consensus algorithm for IP addresses

   As stated before, the consensus algorithm is a central part of a
   blockchain. The first consensus algorithm designed for blockchain
   was PoW, and it is a common choice for new blockchain
   implementations. However it presents several drawbacks
   (Section 3.3.1) for the IP address scenario.

   Using computing power as a means to secure blockchains has been
   proved to work in financial environments. However, the capability to
   add new blocks and the security of the chain itself depends on the
   computing power of the participants, which is not always aligned with
   their interest in the well-being of the blockchain. Depending on the
   objectives of the attacker, certain attacks can become profitable.
   Namely, buying a large quantity of hardware to be able to rewrite the
   blockchain with false data (e.g., incorrect delegations of IP
   addresses). This is because the incentives of the participants of
   the IP addresses blockchain are not linked with their computing
   power.

   In contrast, with Proof of Stake the capability to alter the
   blockchain remains within it. This aspect is of particular
   importance in the context of securing IP addresses: it would mean
   that entities holding large blocks of IP addresses are more likely to
   add blocks. These parties have a reduced incentive in tampering the
   blockchain because they would suffer the consequences: an insecure
   Internet. Typically entities that hold large blocks of the IP
   address space have their business within the Internet and as such,
   have clear incentives in the correct operation and security of the
   Internet.

   Furthermore, in such blockchain the risk of takeover is reduced
   compared to PoW. The reason is that accumulating a large amount of
   IP addresses is typically more complex than accumulating computing
   power. The risk of takeover is also mitigated compared to other PoS-
   based blockchains. In this blockchain an attacker would buy tokens
   from the other parties, who receive a monetary compensation to
   participate in the attack. However, in a blockchain for IP addresses
   this would mean buying IP addresses from other parties, who do not

have a clear incentive to sell their blocks of addresses to the
attacker. Because of this, PoS appears to be a firm candidate for a
consensus algorithm in a blockchain for securing IP addresses
allocations and delegations.

5. Overview of the architecture

This architecture mimics the hierarchy of IP address allocation
present in today's Internet, with IANA on top of it. All nodes trust
IANA's public key, which writes a genesis transaction assigning all
of the address space to itself (figure 3).

```
+--------------+----------------+----------+----------------+
|    IANA      | Signature IANA | Allocate |    Hash(IANA   |
| Public Key 1 | Private Key 1  | 0/0      |  Public Key 2) |
+--------------+----------------+----------+----------------+
```

Figure 3.- Genesis transaction

It then begins allocating each block of addresses to the IP address
holders. Each transaction allocates part of the address space to the
legitimate holder, and the rest of space is given back to IANA using
a new keypair (figure 4).

```
+--------------+----------------+----------+----------------+
|              |                | Rest of  |   Hash(IANA    |
|    IANA      | Signature IANA | space    |  Public Key 3) |
| Public Key 2 | Private Key 2  +----------+----------------+
|              |                | Allocate |   Hash(APNIC   |
|              |                | 1/8      |  Public Key 1) |
+--------------+----------------+----------+----------------+
```

Figure 4.- Example allocation transaction

In turn, all the parties in the hierarchy allocate or delegate
address blocks following the current allocation hierarchy. When a
party wants to verify the allocation of a block of addresses, it
downloads the blockchain and verifies all the blocks and transactions
up to the genesis block, for which it has trust. Figure 5 presents
an example of allocation of one prefix to each of the RIRs.

```
+--------------+----------------+----------+----------------+
|              |                | Rest of  |   Hash(IANA    |
|              |                | space    | Public Key 4)  |
|              |                +----------+----------------+
|              |                | Allocate |   Hash(RIPE    |
|              |                | 5/8      | Public Key 1)  |
|              |                +----------+----------------+
|              |                | Allocate |   Hash(APNIC   |
|     IANA     | Signature IANA | 14/8     | Public Key 2)  |
| Public Key 3 | Private Key 3  +----------+----------------+
|              |                | Allocate |   Hash(ARIN    |
|              |                | 23/8     | Public Key 1)  |
|              |                +----------+----------------+
|              |                | Allocate |  Hash(AFRINIC  |
|              |                | 102/8    | Public Key 1)  |
|              |                +----------+----------------+
|              |                | Allocate |  Hash(LACNIC   |
|              |                | 200/8    | Public Key 1)  |
+--------------+----------------+----------+----------------+
```

Figure 5.- Example multi-output allocation transaction

Inside the blockchain the typical operations to manage blocks of IP
addresses can be defined, such as the delegation of prefixes (figure
6). This helps to enforce the rules of IP addresses management. For
instance, since this transaction is marked as a delegation, if the
new owner created an allocation transaction it would be rejected by
the other nodes, because the parent transaction does not have the
privileges to perform it.

```
+--------------+----------------+----------+----------------+
|              |                | Rest of  |   Hash(APNIC   |
|    APNIC     | Signature APNIC | space   | Public Key 3)  |
| Public Key 1 | Private Key 1  +----------+----------------+
|              |                | Delegate |   Hash(ISP A   |
|              |                | 1.2/16   | Public Key 1)  |
+--------------+----------------+----------+----------------+
```

Figure 6.- Example delegation transaction

Performing a key rollover is simple, because each transaction has its associated public key, and only depends on the previous transaction. In other words, rekeying means changing the public key only in the holder's transaction. This can be done adding a new transaction with the same data but transferring it to a new public key also controlled by the initial holder (figure 7). This approach lets each entity decide its rekeying policies independently.

```
+--------------+----------------+----------+----------------+
|    ISP A     | Signature ISP A| Delegate | Hash(ISP A     |
| Public Key 1 |  Private Key 1 | 1.2/16   | Public Key 2)  |
+--------------+----------------+----------+----------------+
```

Figure 7.- Example key rollover of a prefix delegation

It is worth noting that this chain can define as many operations as required, for instance storing the binding of AS numbers to the IP prefixes they announce (figure 8).

```
+--------------+----------------+----------+----------------+
|              |   Signature    | Bind     |                |
|    ISP A     |     ISP A      | 1.2/16   | Hash(ISP A     |
| Public Key 2 |  Private Key 2 | AS no.   | Public Key 3)  |
|              |                | 12345    |                |
+--------------+----------------+----------+----------------+
```

Figure 8.- Example binding of AS number to prefix

Additional and more complex operations can be defined if the management logic requires it. For instance, several signatures (from different parties) can be required to consider a transaction valid, restrict permissions for customer sub-delegations, etc.

5.1. Support for IPv6 and AS numbers

The allocation and delegation of IPv6 addresses and AS numbers is equivalent to that of IPv4, maintaining the IANA -> RIR -> LIR hierarchy. For example, for IPv6:

```
+-------------+-----------------+----------+---------------+
|   IANA v6   |  Signature IANA | Allocate | Hash(IANA v6  |
| Public Key 1| v6 Private Key 1|   0::/0  | Public Key 2) |
+-------------+-----------------+----------+---------------+


+-------------+-----------------+----------+---------------+
|             |                 | Rest of  | Hash(IANA v6  |
|   IANA v6   |  Signature IANA | space    | Public Key 3) |
| Public Key 2| v6 Private Key 2 +---------+---------------+
|             |                 | Allocate | Hash(IANA v6  |
|             |                 | 2000::/3 | Public Key 4) |
+-------------+-----------------+----------+---------------+


+-------------+-----------------+----------+----------------+
|             |                 | Rest of  |  Hash(IANA v6  |
|   IANA v6   |  Signature IANA | space    |  Public Key 5) |
| Public Key 4| v6 Private Key 2 +---------+----------------+
|             |                 | Allocate | Hash(AFRINIC v6|
|             |                 | 2c00::/12|  Public Key 1) |
+-------------+-----------------+----------+----------------+


+-------------+-----------------+----------+----------------+
|             |                 | Rest of  | Hash(AFRINIC v6|
|  AFRINIC v6 | Signature AFRINIC| space   |  Public Key 2) |
| Public Key 1| v6 Private Key 1 +---------+----------------+
|             |                 | Allocate |  Hash(ISP A v6 |
|             |                 | 2c0c::/15|  Public Key 1) |
+-------------+-----------------+----------+----------------+
```

Figure 9.- IPv6 allocation transactions. From top to bottom:
genesis transaction, global unicast allocation, AFRINIC allocation
and LIR allocation.

The process is equivalent for AS numbers. Besides, in the context of
a multi-signature scheme, it is also possible to ask the holder of
the AS number to confirm the binding of its AS number to a particular
prefix.

5.2. Pros and cons

In this section we analyze the pros and cons of this architecture
compared to traditional PKI infraestructures:

Advantages:

o Decentralized: No central entity controls the blockchain, it is
   shared among all participants.

o No CAs, CRLs or certificates needed: No digital certificates,
   Certification Authorities or CRLs are needed.

o Simplified rekeying: A key rollover can easily be performed by
   issuing a new transaction allocating the prefixes to a new keypair
   controlled by the same holder. This process can be performed
   without involving any third-party.

o Censorship-resistant: since the control of a transaction is
   completely under the holder of the private key, the revocation of
   IP addresses without the legitimate holder's permission involves
   obtaining its private key. Even if the private key of the
   previous owner was compromised, ownership of the current
   transaction is still preserved, as opposed to the compromise of a
   CA's private key (or a misbehaving CA).

o Limited prior trust: It is not required to trust other nodes.
   However, in PoS it is necessary to periodically authenticate the
   chain state out-of-band to prevent some attacks.

o Simplified management: since CAs are not required, their
   management overhead is avoided.

o Auditable: allocations and delegations can be tracked back in the
   blockchain to determine if they originate from the legitimate
   holder.

o Limited legal liability: since users control their private keys,
   Internet Registries cannot be held legally responsible of their
   loss. In turn, this can foster the creation of a unified registry
   instead of the current five. Ultimately, this would ease cross-
   registry resource transfers.

o No single point of failure: again, due to the fact that each user
   controls its private key, the compromise of a user's key does not
   compromise the entire system. This starkly contrasts with the
   compromise of a CA, which can potentially invalidate all
   downstream certificates.

o Simplified state update: PKIs need specific subsystems to update
   its state (e.g. issue/revoke certificates). On the other hand,
   in a blockchain all these operations are embedded in it thanks to
   its transactional nature.

Drawbacks:

o PoS does not rely on strong cryptographic guarantees: As opposed
   to PKI-based systems that rely on strong and well-established
   cryptographic mechanisms, PoS-based infraestructures ultimately
   rely on the good behaviour of the high-stakers.

o Costly bootstrapping: When a node is activated it has to download
   and verify the entire blockchain.

o Large storage required: The blockchain grows forever as more
   blocks are added, blocks cannot be removed.

5.3. Security evaluation

5.3.1. Attacks against a PoS-based consensus algorithm

   This section presents a list of the most relevant attacks against a
   Proof of Stake algorithm and how to mitigate them.

5.3.1.1. Stake grinding

   Stake grinding refers to the manipulation of the consensus algorithm
   in order to progressively obtain more stake, with the goal of signing
   blocks more frequently with the ultimate goal of taking control of
   the blockchain. It proceeds as follows: when the attacker has to
   sign a block, it computes all the possible blocks (varying the data
   inside them) to find a combination that gives the highest possibility
   of signing another block in the future. It then signs this block and
   sends it to the network. This procedure is repeated for all the next
   signing opportunities. Over time, the attacker will sign more and
   more blocks until the consensus algorithm will always select the
   attacker to sign all blocks, thereby having taken control of the
   blockchain.

   To prevent this attack, the source of randomness used to select the
   signers has to be hard to alter or to predict.

5.3.1.2. Nothing at stake

   Nothing at stake is one of the fundamental drawbacks of Proof of
   Stake and requires careful design based on the incentives of the
   participants. In common PoS designs, the signers of the new block
   receive an economical incentive (e.g., Ethereum). However this does
   not hold in the IP address scenario, since participants should not
   receive any incentive. The incentive is, as stated before, achieving
   a consistent view of the IP address space and having a secure
   Internet.

5.3.1.3. Range attacks

   A range attack is performed by creating a fork some blocks back from
   the tip of the chain. It is conceptually similar to the attack named
   as 'Risk of takeover' in Section 3.3.1. In this scenario, the
   attacker has privately fabricated a chain which (according to the
   consensus algorithm rules) will be selected over the original one.
   Benefits of this attack include gaining more stake on the blockchain
   (this attack could be part of a stake grinding attack) or rewriting
   the transaction history to erase a payment made in the original
   blockchain.

   The simplest solution to this attack is adding a revert limit to the
   blockchain, forbidding forks going back more than N blocks. This
   provides a means to solidify the blockchain. However, nodes that
   have been offline for more than N blocks will need an external source
   that indicates the correct chain. It has been proposed to do this
   out of band. This is why a PoS blockchain is not purely trustless
   and requires a small amount of trust.

5.3.1.4. Monopolies

   A monopoly refers to a single party controlling enough IP addresses
   so it can sign a significant proportion of new blocks, thus being
   able to decide which information is written in the chain (e.g., a 51
   % attack in Bitcoin). However and in this use-case, this is of less
   concern since parties do not have a clear incentive to alter normal
   chain operation. In order to successfully launch this attack a party
   should control more than 50% of the IP blocks, while this is
   difficult to achieve and participants do not have a clear incentive
   to sell/give away blocks of IPs, the attacker would also impact its
   own infrastructure, making the Internet less secure. Section 7.4
   contains more details regarding monopolies.

5.3.1.5. Lack of participation

   Participants in a PoS algorithm will not always sign a block, since
   they might be offline when they are selected or lack incentives.
   Because of this, the final fraction of high-stakers that sign blocks
   can be very different from the full set of high-stakers. The direct
   consequence of this situation is that the portion of participants
   that decide what goes into the blockchain can be a small set of
   nodes. If this participation is low enough, it can leave the control
   of the blockchain to a small amount of people/oligarchy, thus rising
   security concerns.

5.3.2. Attacks against the P2P network

This section presents attacks directed towards the underlying P2P network used to exchange information among the participants of the blockchain.

5.3.2.1. DDOS attacks

Since blockchains are inherently based on P2P architectures, they present a higher degree of resistance to DDOS attacks than centralized server architectures, provided that the network has a significant number of participants. In addition, it is always possible to keep an offline copy of the blockchain.

5.3.2.2. Transaction flooding

A special type of DDOS attack consists in creating a large amount of legit transactions that transfer a small amount of tokens (i.e. delegate a lot of small IP prefixes). If the number of transactions is large enough, the addition of new transactions can be significantly delayed because not all of them fit into a single block. The effectiveness of the attack also depends on the throughput of the blockchain (transactions/second). Simple solutions may be to limit the granularity upon which IP addresses can be split. Of course, only the legitimate holder of a large amount of IP address can perform this attack.

5.3.2.3. Routing attacks

The underlying P2P network in blockchains does not typically use any security mechanism, e.g. node authentication or integrity of network protocol messages. This enables potentially disruptive attacks. For example, specially located rogue nodes could drop new transactions, which would block updates on the blockchain and leave legit nodes uncommunicated. The effectiveness of this kind of attacks depends on how the P2P algorithm selects peers and the topology of the P2P network.

However, the most potentially dangerous attack of this type are network partitions, i.e. isolating a group of nodes from the rest of the network so they cannot communicate each other (e.g., [Apostolaki2017]). The consequence of this attack is that two versions of the blockchain are created, one at each network partition. When the partition disappears and the nodes reconnect one of the two chains will be discarded, causing a service disruption. It is worth noting that Bitcoin has suffered similar attacks [realrouteattack].

5.3.2.4. Transaction censorship

When a node adds a block it chooses arbitrarily which transactions
are added into it, i.e. no specific rules control how transactions
are added to a block. This enables a node to selectively add some
transactions and intentionally exclude others, with the consequence
that some transactions may be never added to the blockchain. In the
context of IP addresses, this may be performed by a competing ISP to
prevent another ISP from executing a certain modification. Possible
solutions revolve around:

o Giving more priority to older transactions (similarly to Bitcoin).

o Punishing nodes that exhibit this kind of behaviour, e.g.
   removing part of their block of IP addresses or lowering their
   chance of adding blocks.

6. Revocation

Due to the irreversible nature of transactions, once a block of IP
addresses has been allocated to an entity it is not possible to
modify or remove it, as opposed to CRLs (Certificate Revocation
Lists). However, due to operational issues (compromised or lost
keys, human mistake, holder misbehaviour, etc) it is critical to
provide a way to recover a block of addresses. Moreover, since IP
addresses are a finite public good they cannot be lost. Taking into
account that a blockchain can enforce any rules its participants
agree upon, this section presents some possible approaches to
implement revocation, such approaches should not be considered as
mutually exclusive. The revocation procedure must be discussed among
the community to achieve consensus between the relevant players
(IANA, RIRs, ISPs, institutions, etc).

All these mechanisms present different balances of power between the
current holder and the entity whose asset is being revoked. Behind
all these mechanisms there is a fundamental tradeoff between trusting
an upstream provider of the addresses and retaining full control of
the block of addresses.

Regardless of the revocation policy and as opposed to traditional PKI
systems, each IP prefix delegation only depends on the private key of
the holder of such IP block. As such, it does not need to trust a CA
or a chain of certificates. Only by means of this private key the IP
delegation can be altered.

6.1. Expiration time

   A simple approach to allow revocation is adding a lease time (i.e,
   time-to-live) to the blocks of addresses. After the lease ends, the
   new holder of the address block automatically becomes the previous
   one, or addresses are transferred to a default holder. As stated
   before, this revocation procedure should be enforced by the rules of
   the blockchain, this means that participants would not recognise
   expired allocations as valid.

6.2. Multi-signature transactions

   A multi-signature transaction is a transaction with more than one
   associated public key. In other words, a transaction is considered
   valid if it has, for instance, 2 out of 5 valid signatures. This
   way, 3 keys can be lost but with the renaming 2 keys the block of
   addresses can be recovered. This approach exemplifies the
   aforementioned tradeoff in trust, since the holder of the block of
   addresses must trust the owners of the keys participating in the
   multi-signature transaction. For instance, if some of these keys are
   owned by IANA or an Internet Registry, we can return part of the
   control over the allocation to them.

6.3. Revocation transaction

   A simpler approach than multi-signature transactions is creating a
   'revocation' transaction. When a block of address is required to be
   reassigned without the consent of the current holder, a revocation
   transaction (specifying the new holder) is inserted in the
   blockchain. This transaction should be issued either by a
   consensuated authority or by a disputing entity. The revocation
   transaction should be resolved by either accepting the revocation
   transaction automatically when issued by the accepted authority or by
   means of out-of-band mechanisms when issued by a disputing party.

6.4. Heartbeat transaction

   Another approach involves issuing a heartbeat transaction every N
   days, signalling to the network that the holder still owns the key
   associated with that particular resource. If the holder fails to
   issue this transaction, the blockchain considers that the resource is
   automatically returned to the registry.

6.5. Out-of-band mechanisms

   Disputes regarding transactions can be resolved by means of out-of-
   band mechanisms, e.g, discussion, court, etc. In order to reflect
   the decision of this out-of-band mechanism the blockchain must be
   modified. Since this represents a deviation from the rules, it must
   be done through a hard blockchain fork. Although cumbersome and
   complex, this is feasible from a technical standpoint.

6.6. A simple revocation protocol

   Here we present a simple revocation protocol to handle accidental key
   loss:

   1. On detecting the key loss, the holder notifies the registry, e.g.
       via e-mail.

   2. The registry issues a revocation transaction, similar to the one
       in section Section 6.3.

   3. The current holder has a fixed period of time to issue a
       transaction re-claiming the resource. This transaction must be
       signed by the private key associated with the claimed resource.

   4. If the holder issues the transaction, it retains the resource.

   5. Otherwise, after the fixed time interval, the blockchain
       considers that the resource is returned to the registry, so it
       can be re-allocated.

   This protocol combines two of the aforementioned techniques, and
   allows to balance power between resource holders and registries.
   Registries can revoke lost or unclaimed resources, while address
   holders can retain them if the registry misbehaves or its key is
   compromised. However, it should be noted that this protocol does not
   protect from stolen keys.

   The time interval can be different depending on the nature of the
   revocating entity. For example, IANA -> RIR allocations could wait a
   couple of weeks, whereas RIR -> LIR allocations could go faster with
   a 72 hours notification period.

7. Other Considerations

7.1. Storage management

   The never ending size of the blockchain presents a potential
   scalability issue. At the time of this writing, mature blockchains

like Bitcoin require more than 100 GB of storage. Simply deleting or
summarizing old transactions degrades the security of PoW-based
chains, since their security relies on the computing power required
to generate them. The longer they are, the harder they are to
attack.

However, PoS-based chains do not rely on computing power and hence,
space-saving strategies do not degrade the security. For instance a
simple solution could be to, once the PoS-based chain reaches a
certain storage size, summarize a subset of the older transactions.
In what follows we overview this strategy:

```
+-----+-----+-----+-----+----------------+-----+-----+-----+
|  0  |  1  |  2  |  3  |      .....      |47832|47833|47834|
+-----+-----+-----+-----+----------------+-----+-----+-----+
```

9.1 Old blockchain

```
+-----+-----+-----+-----+-----+
| r0  | r1  | r2  | r3  | r4  |
+-----+-----+-----+-----+-----+
```

9.2 Write present state to special reset blocks

```
+-----+-----+-----+-----+-----+-----+-----+
| r0  | r1  | r2  | r3  | r4  |  0  |  1  | ...
+-----+-----+-----+-----+-----+-----+-----+
```

9.3 Continue working after the reset blocks

Figure 10.- A simple technique to reduce blockchain storage

This approach reduces bootstrapping cost, it is worth noting that
this strategy requires trust on the reset blocks, such blocks can be
obtained with an out-of-band mechanism (see Section 5.3.1.3 for
further information).

7.2. Proof of Networking?

In this section we speculate how one could design an equivalent of Proof-of-Work (PoW) for networks. Conceptually, PoW is a proof of computational resources, can we devise a proof of networking resources? It could be thought that a PoW equivalent may exist in the context of networks, i.e., an equivalent to spending computer cycles in a network. Some resources unique to networks are bandwidth, computation of checksums, number of BGP peers, etc. Hence, we could envisage a blockchain secured by the resources inherent to its participating computer networks. As long as half of the resources were controlled by honest members, security is guaranteed. For example, bandwidth could be a potential candidate; however it does not satisfy two key features present in PoW:

o Asymmetry: the proof has to be hard to generate but fast to verify.

o Verifiability: it has to be possible to embed the proof in the block in order to account for the spending of resources.

In this context, Proof-of-Networking is an open research issue .

7.3. Configuration parameters

Configuration parameters refer to a set of values:

o Block creation rate

o Maximum block size

o Other parameters related to the consensus algorithm

These parameters, beyond regulating the operation of the blockchain also have an influence on its performance. For example, a small block size increases propagation speed (thus consensus can be reached faster) but reduces the number of transactions per second that the blockchain can handle. As an example, in Bitcoin, the 10-minute block creation rate seeks to balance fast confirmation times and reduced probability of forks [Antonopoulos2015]. Experimental deployments and operational requirements should help tuning such parameters.

7.4. PoS algorithm design particularities

The particular use case of IP addresses presents some characteristics that the PoS algorithm should take into account:

Monopoly prevention: As described in section Section 5.3.1.4, monopolies can pose a threat to a PoS blockchain. In order to

prevent a small number of large-stakers from controlling the
chain, we can design the PoS algorithm to have a smart mapping of
IP prefixes to the weight of the random selection. A potential
solution could be fine-tuning the weighting of IP addresses
(slightly biasing the choice towards medium-sized holders), in
order to reduce the power of high stakers. This can provide an
upper-bound of the maximum number of addresses that a party can
hold to avoid monopolies (ideally as high as possible).

IPv6 support: Large parts of the IPv6 address space remain
   unallocated and still owned by IANA (at the time of writing this
   document, less than 0.5% of v6 address space has been allocated to
   the RIRs). The PoS algorithm should ignore this space (not count
   it) to avoid IANA signing nearly all v6 blocks and thus,
   preventing an IANA monopoly.

IPv4/v6 stake isolation: Since there are more IPv6 addresses than
   IPv4, this creates an imbalance of power in a PoS blockchain:
   randomly selecting from both pools of addresses naturally causes
   v6 holders signing more blocks than v4 holders. Thus, some kind
   of isolation between v4 and v6 stake is required. For example, we
   could alternatively generate blocks with only v4 or v6
   transactions, signed by a v4 or v6 holder, respectively.

7.5. Candidate PoS consensus algorithms

There are several existing PoS algorithms that could satisfy the
requirements of a blockchain for IP addresses. The following list
presents three of them that have been claimed by the authors to be
proven mathematically correct. A substantial difference among them
is the supported portion of offline participants. The list does not
pretend to be exhaustive.

Algorand: [Algorand] leverages a multi-step protocol to provide a
   verifiable random selection of the block signer. The most
   relevant features of Algorand are:

   * A cryptographic sortition mechanism to randomly select the
      participants in each step of the protocol, based on Verifiable
      Random Functions [I-D.irtf-cfrg-vrf].

   * The decoupling of block creation and block signing, to avoid
      stake grinding attacks.

   * A new Byzantine Agreement protocol (BA*) to achieve consensus.

        * Player replaceability: Algorand uses a different set of
           participants for each of its steps. Thus, malicious
           participants from one step cannot influence the following.

        * Upper bound of 1/3 of dishonest players.

   Ouroboros: [Ouroboros] presents a similar approach to Algorand:
        first, it selects a subset of users. Then, these users perform
        the random selection by means of a secure multiparty computation.
        As opposed to Algorand, however, this subset lasts for several
        blocks, called epochs. In addition, Ouroboros assumes that the
        majority of participants are online when they have to participate
        in the protocol, and that they remain offline only short periods
        of time.

   Snow White: The [SnowWhite] protocol improves on the previous two by
        supporting also large amounts of offline participants, as long as
        the majority of online members are honest. They call this
        property 'Robustly Reconfigurable Consensus'. Snow White also
        leverages a random function to decide if a participant has to sign
        a block, and defines epochs similarly to Ouroboros: after each
        epoch, participants for the new one are calculated.

7.6. Privacy concerns

   In order to protect privacy, the blockchain should not contain
   Personally Identifiable Information (PII). This is due to the fact
   that data in the blockchain cannot be removed and that it is a public
   ledger, accessible by anyone. Instead, PII like contact emails or
   postal addresses should be stored in the Registry's database (e.g.
   RIPE Database). Ideally, the blockchain should contain the minimum
   amount of data for correct operation, that is: public keys, blocks of
   IP addresses, AS numbers and their bindings (figure 11).

            Public   Private

   +---------------------+   +-----------------------+
Blockchain	+-----------+	Registry Database
	<---	Update	<---	
Pairs of (Prefix,	|prefix, key|	*Registry Policies
Public key)	|pair		*Contact data
	+-----------+	
   +---------------------+   +-----------------------+
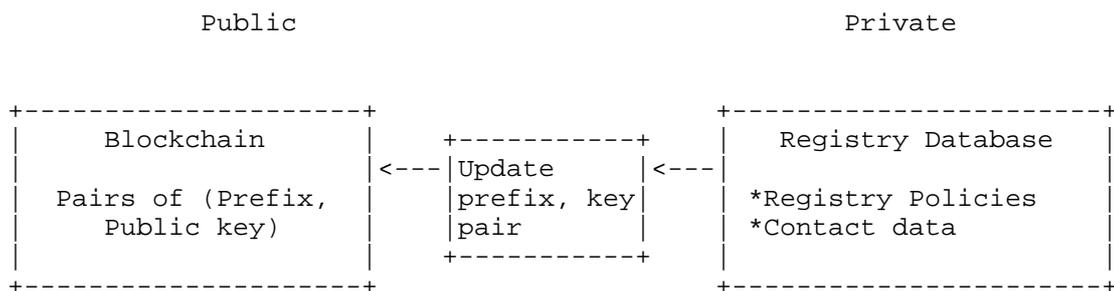
           Figure 11.- Data flow between Registry and blockchain

7.7. Governance

   Blockchain does not mean anarchy. In fact, any blockchain requires a
   governing entity that determines its rules and ensures that all of
   its participants agree on its operation. The need of governance is
   illustrated by the recent Bitcoin Cash fork [Bitcoincash]. Due to a
   disagreement among Bitcoin users, they created a Bitcoin hard fork
   called Bitcoin Cash. This split the Bitcoin blockchain in two,
   causing some confusion. Proper governance should avoid such
   situations.

   This particular use case is not an exception: all concerned parties
   (IANA, RIRs, ISPs, etc) should reach consensus regarding which rules
   are enforced in the blockchain. For example, dispute resolution or
   revocation procedures.

8. Implementations

   There are several implementations to secure the allocation of IP
   prefixes. They present different scopes and levels of maturity.

   o IPchain uses a Proof of Stake algorithm and is specifically
      tailored for the allocation of IP addresses. Its performance has
      been evaluated [IPchain], and has been open-sourced
      [IPchain-repo].

   o [BGPCoin] runs on top of the Ethereum blockchain and provides
      similar features to IPchain. It has not been possible to find
      open-source code.

   o Another project identically named BGPCoin is designed to allow
      ISPs to exchange peering agreements and route advertisements by
      means of a blockchain [BGPCoin-repo]. It uses a hybrid PoW/PoS
      algorithm and has its own cryptocurrency.

9. Security Considerations

   This document aims to understand the security implications of using
   the blockchain technology to secure IP addresses allocation.

10. IANA Considerations

   This memo includes no request to IANA.

11. Acknowledgements

   The authors wish to thank Jordi Herrera-Joancomarti, Andreu
   Rodriguez-Donaire and Jordi Baylina for their helpful discussions

about Bitcoin and blockchain technology, as well as Marco Chiesa for
the heartbeat transaction idea.

12. Informative References

[Algorand]
          Gilad, Y., Hemo, R., Micali, S., Vlachos, G., and N.
          Zeldovich, "Algorand: Scaling byzantine agreements for
          cryptocurrencies. Proceedings of the 26th Symposium on
          Operating Systems Principles (pp. 51-68). ACM.", 2017.

[Antonopoulos2015]
          Antonopoulos, A. M., "Mastering Bitcoin, available online:
          http://chimera.labs.oreilly.com/books/1234000001802/
          index.html", 2015.

[Apostolaki2017]
          Apostolaki, M., Zohar, A., and L. Vanbever, "Hijacking
          Bitcoin: Routing Attacks on Cryptocurrencies. 2017 IEEE
          Symposium on Security and Privacy (SP). ", 2017.

[BGPCoin-repo]
          BGPCoin GitHub repository, , "https://github.com/bgpcoin",
          2018.

[BGPCoin] Xing, Q., Wang, B., and X. Wang, "POSTER: BGPCoin: A
          Trustworthy Blockchain-based Resource Management Solution
          for BGP Security. ACM Conference on Computer and
          Communications Security (CCS) 2017", 2017.

[Bitcoin] Nakamoto, S., "Bitcoin: A Peer-to-Peer Electronic Cash
          System. https://bitcoin.org/bitcoin.pdf", 2008.

[Bitcoincash]
          Bitcoin split in two, here's what that means, , "http://
          money.cnn.com/2017/08/01/technology/business/bitcoin-cash-
          new-currency/index.html", 2017.

[Blockstack]
          Ali, et al., M., "Blockstack : A Global Naming and Storage
          System Secured by Blockchains, USENIX Annual Technical
          Conference", 2016.

[Byzantine]
          Lamport, L., Shostak, R., and M. Pease, "The Byzantine
          Generals Problem. ACM Transactions on Programming
          Languages and Systems", 1982.

   [Ethereum]
             The Ethereum project, , "https://www.ethereum.org/", 2016.

   [Hari2016]
             Hari, A. and T. Lakshman, "The Internet Blockchain: A
             Distributed, Tamper-Resistant Transaction Framework for
             the Internet. Fifteenth ACM Workshop on Hot Topics in
             Networks", 2016.

   [I-D.irtf-cfrg-vrf]
             Goldberg, S., Reyzin, L., Papadopoulos, D., and J. Vcelak,
             "Verifiable Random Functions (VRFs)", draft-irtf-cfrg-
             vrf-01 (work in progress), March 2018.

   [IPchain-repo]
             IPchain GitHub repository, , "https://github.com/
             OpenOverlayRouter/blockchain-mapping-system", 2018.

   [IPchain] Paillisse, J., Ferriol, M., Garcia, E., Latif, H., Piris,
             C., Lopez, A., Kuerbis, B., Rodriguez-Natal, A., Ermagan,
             V., Maino, Fabio., and A. Cabellos, "IPchain: Securing IP
             Prefix Allocation and Delegation with Blockchain, arXiv
             preprint: https://arxiv.org/abs/1805.04439", 2018.

   [Namecoin]
             Namecoin, , "https://namecoin.org/", 2011.

   [Ouroboros]
             Kiayias, A., Russell, A., David, B., and R. Oliynykov,
             "Ouroboros: A provably secure proof-of-stake blockchain
             protocol. Annual International Cryptology Conference (pp.
             357-388). Springer, Cham.", 2017.

   [Peercoin]
             The Peercoin cryptocurrency, , "https://peercoin.net/",
             2016.

   [SnowWhite]
             Bentov, I., Pass, R., and E. Shi, "Snow White: Provably
             Secure Proofs of Stake. IACR Cryptology ePrint Archive,
             2016, 919.", 2016.

   [miningfarm]
             Inside a mining farm, , "http://www.bbc.com/future/story/
             20160504-we-looked-inside-a-secret-chinese-bitcoin-mine",
             2016.

   [monero]   Monero PoW algorithm update, , "https://www.ethnews.com/
               monero-team-mulls-changing-pow-algorithm-to-preempt-asic-
               miners", 2018.

   [realrouteattack]
               Hacker Redirects Traffic From 19 Internet Providers to
               Steal Bitcoins, , "https://www.wired.com/2014/08/isp-
               bitcoin-theft/", 2014.

Authors' Addresses

   Jordi Paillisse
   UPC-BarcelonaTech
   c/ Jordi Girona 1-3
   Barcelona, Catalonia 08034
   Spain

   Email: jordip@ac.upc.edu

   Alberto Rodriguez-Natal
   Cisco Systems
   170 Tasman Drive
   San Jose, CA
   USA

   Email: natal@cisco.com

   Vina Ermagan
   Cisco Systems
   170 Tasman Drive
   San Jose, CA
   USA

   Email: vermagan@cisco.com

   Fabio Maino
   Cisco Systems
   170 Tasman Drive
   San Jose, CA
   USA

   Email: fmaino@cisco.com

Leo Vegoda
Individual
4712 Admiralty Way, #152
Marina del Rey, CA   90292
USA

Email: leo@vegoda.org

Albert Cabellos
UPC-BarcelonaTech
c/ Jordi Girona 1-3
Barcelona, Catalonia   08034
Spain

Email: acabello@ac.upc.edu

            Enhanced Feasible-Path Unicast Reverse Path Filtering
                  draft-sriram-opsec-urpf-improvements-03

Abstract

   This document identifies a need for improvement of the unicast
   Reverse Path Filtering techniques (uRPF) [BCP84] for source address
   validation (SAV) [BCP38]. The strict uRPF is inflexible about
   directionality, the loose uRPF is oblivious to directionality, and
   the current feasible-path uRPF attempts to strike a balance between
   the two [BCP84]. However, as shown in this draft, the existing
   feasible-path uRPF still has short comings. This document describes
   an enhanced feasible-path uRPF technique, which aims to be more
   flexible (in a meaningful way) about directionality than the
   feasible-path uRPF. It can potentially alleviate ISPs' concerns
   about the possibility of disrupting service for their customers, and
   encourage greater deployment of uRPF techniques.

Status of This Memo

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

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

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

   This Internet-Draft will expire on September 6, 2018.

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

   This internet draft identifies a need for improvement of the unicast
   Reverse Path Filtering (uRPF) techniques [RFC2827] for source address
   validation (SAV) [RFC3704]. The strict uRPF is inflexible about
   directionality, the loose uRPF is oblivious to directionality, and
   the current feasible-path uRPF attempts to strike a balance between
   the two [RFC3704]. However, as shown in this draft, the existing
   feasible-path uRPF still has short comings. Even with the feasible-
   path uRPF, ISPs are often apprehensive that they may be dropping
   customers' data packets with legitimate source addresses.

This document describes an enhanced feasible-path uRPF technique,
which aims to be more flexible (in a meaningful way) about
directionality than the feasible-path uRPF. It is based on the
principle that if BGP updates for multiple prefixes with the same
origin AS were received on different interfaces (at border routers),
then incoming data packets with source addresses in any of those
prefixes should be accepted on any of those interfaces (presented in
Section 3). For some challenging ISP-customer scenarios (see
Section 3.3), this document also describes a more relaxed version of
the enhanced feasible-path uRPF technique (presented in Section 3.4).
Implementation considerations are discussed in Section 3.5.

Note: Definition of Reverse Path Filtering (RPF) list: The list of
permissible source address prefixes for incoming data packets on a
given interface.

Note: Throughout this document, the routes in consideration are
assumed to have been vetted based on prefix filtering [RFC7454] and
possibly (in the future) origin validation [RFC6811].

The enhanced feasible-path uRPF methods described here are expected
to add greater operational robustness and efficacy to uRPF, while
minimizing ISPs' concerns about accidental service disruption for
their customers. It is expected that this will encourage more
deployment of uRPF to help realize its DDoS prevention benefits
network wide.

1.1. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT",
"SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this
document are to be interpreted as described in RFC 2119 [RFC2119].

2. Review of Existing Source Address Validation Techniques

There are various existing techniques for mitigation against DDoS
attacks with spoofed addresses [RFC2827] [RFC3704]. There are also
some techniques used for mitigating reflection attacks [RRL]
[TA14-017A], which are used to amplify the impact in DDoS attacks.
Employing a combination of these preventive techniques (as
applicable) in enterprise and ISP border routers, broadband and
wireless access network, data centers, and DNS servers provides
reasonably effective protection against DDoS attacks.

Source address validation (SAV) is performed in network edge devices
such as border routers, Cable Modem Termination Systems (CMTS),
Digital Subscriber Line Access Multiplexers (DSLAM), and Packet Data
Network (PDN) gateways in mobile networks. Ingress Access Control

   List (ACL) and unicast Reverse Path Filtering (uRPF) are techniques
   employed for implementing SAV [RFC2827] [RFC3704] [ISOC].

2.1. SAV using Access Control List

   Ingress/egress Access Control Lists (ACLs) are maintained which list
   acceptable (or alternatively, unacceptable) prefixes for the source
   addresses in the incoming Internet Protocol (IP) packets. Any packet
   with a source address that does not match the filter is dropped. The
   ACLs for the ingress/egress filters need to be maintained to keep
   them up to date. Updating the ACLs is an operator driven manual
   process, and hence operationally difficult or infeasible.

   Typically, the egress ACLs in access aggregation devices (e.g. CMTS,
   DSLAM) permit source addresses only from the address spaces
   (prefixes) that are associated with the interface on which the
   customer network is connected. Ingress ACLs are typically deployed
   on border routers, and drop ingress packets when the source address
   is spoofed (i.e. belongs to obviously disallowed prefix blocks, RFC
   1918 prefixes, or provider's own prefixes).

2.2. SAV using Strict Unicast Reverse Path Filtering

   In the strict unicast Reverse Path Filtering (uRPF) method, an
   ingress packet at border router is accepted only if the Forwarding
   Information Base (FIB) contains a prefix that encompasses the source
   address, and forwarding information for that prefix points back to
   the interface over which the packet was received. In other words,
   the reverse path for routing to the source address (if it were used
   as a destination address) should use the same interface over which
   the packet was received. It is well known that this method has
   limitations when networks are multi-homed and there is asymmetric
   routing of packets. Asymmetric routing occurs (see Figure 1) when a
   customer AS announces one prefix (P1) to one transit provider (ISP-a)
   and a different prefix (P2) to another transit provider (ISP-b), but
   routes data packets with source addresses in the second prefix (P2)
   to the first transit provider (ISP-a) or vice versa.

```
           +------------+ ---- P1[AS2 AS1] ---> +------------+
           | AS2(ISP-a) | <----P2[AS3 AS1] ---- |  AS3(ISP-b)|
           +------------+                       +------------+
                /\                                    /\
                  \                                   /
                   \                                 /
                    \                               /
             P1[AS1]\                              /P2[AS1]
                     \                            /
               +----------------------+
               |     AS1(customer)    |
               +----------------------+
                   P1, P2 (prefixes originated)
```

            Consider data packets received at AS2
            (1) from AS1 with source address in P2, or
            (2) from AS3 that originated from AS1
                  with source address in P1:
                       * Strict uRPF fails
                       * Feasible-path uRPF fails
                       * Loose uRPF works (but ineffective in IPv4)
                       * Enhanced Feasible-path uRPF works best

        Figure 1: Scenario 1 for illustration of efficacy of uRPF schemes.

2.3. SAV using Feasible-Path Unicast Reverse Path Filtering

   The feasible-path uRPF helps partially overcome the problem
   identified with the strict uRPF in the multi-homing case. The
   feasible-path uRPF is similar to the strict uRPF, but in addition to
   inserting the best-path prefix, additional prefixes from alternative
   announced routes are also included in the RPF table. This method
   relies on announcements for the same prefixes (albeit some may be
   prepended to effect lower preference) propagating to all routers
   performing feasible-path uRPF checks. Therefore, in the multi-homing
   scenario, if the customer AS announces routes for both prefixes (P1,
   P2) to both transit providers (with suitable prepends if needed for
   traffic engineering), then the feasible-path uRPF method works (see
   Figure 2). It should be mentioned that the feasible-path uRPF works
   in this scenario only if customer routes are preferred at AS2 and AS3
   over a shorter non-customer route.

```
          +------------+  routes for P1, P2   +-----------+
          | AS2(ISP-a) |<-------------------->| AS3(ISP-b)|
          +-----------+          (p2p)        +-----------+
               /\                                   /\
                \                                   /
         P1[AS1]\                                  /P2[AS1]
                 \                                /
     P2[AS1 AS1 AS1]\                            /P1[AS1 AS1 AS1]
                     \                          /
                +-----------------------+      /
                |     AS1(customer)     |
                +-----------------------+
                  P1, P2 (prefixes originated)
```

        Consider data packets received at AS2 via AS3
        that originated from AS1 and have source address in P1:
        * Feasible-path uRPF works (if customer route to P1
          is preferred at AS3 over shorter path)
        * Feasible-path uRPF fails (if shorter path to P1
          is preferred at AS3 over customer route)
        * Loose uRPF works (but ineffective in IPv4)
        * Enhanced Feasible-path uRPF works best

   Figure 2: Scenario 2 for illustration of efficacy of uRPF schemes.

   However, the feasible-path uRPF method has limitations as well. One
   form of limitation naturally occurs when the recommendation of
   propagating the same prefixes to all routers is not followed.
   Another form of limitation can be described as follows. In Scenario
   2 (described above, illustrated in Figure 2), it is possible that the
   second transit provider (ISP-b or AS3) does not propagate the
   prepended route for prefix P1 to the first transit provider (ISP-a or
   AS2). This is because AS3's decision policy permits giving priority
   to a shorter route to prefix P1 via a peer (AS2) over a longer route
   learned directly from the customer (AS1). In such a scenario, AS3
   would not send any route announcement for prefix P1 to AS2. Then a
   data packet with source address in prefix P1 that originates from AS1
   and traverses via AS3 to AS2 will get dropped at AS2.

2.4. SAV using Loose Unicast Reverse Path Filtering

   In the loose unicast Reverse Path Filtering (uRPF) method, an ingress
   packet at the border router is accepted only if the FIB has one or
   more prefixes that encompass the source address. That is, a packet
   is dropped if no route exists in the FIB for the source address.
   Loose uRPF sacrifices directionality. This method is not effective
   for prevention of address spoofing since there is little unrouted
   address space in IPv4. It only drops packets if the spoofed address

is unreachable in the current FIB (e.g. RFC 1918, unallocated, allocated but currently not routed).

3. SAV using Enhanced Feasible-Path uRPF

3.1. Description of the Method

Enhanced feasible-path uRPF adds greater operational robustness and efficacy to existing uRPF methods discussed in Section 2. The technique is based on the principle that if BGP updates for multiple prefixes with the same origin AS were received on different interfaces (at border routers), then incoming data packets with source addresses in any of those prefixes should be accepted on any of those interfaces. It can be best explained with an example as follows:

Let us say, a border router of ISP-A has in its Adj-RIB-Ins [RFC4271]. the set of prefixes {Q1, Q2, Q3} each of which has AS-x as its origin and AS-x is in ISP-A's customer cone. Further, the border router received a route for prefix Q1 over a customer facing interface, while it learned routes for prefixes Q2 and Q3 from a lateral peer and an upstream transit provider, respectively. In this example scenario, the enhanced feasible-path uRPF method requires Q1, Q2, and Q3 be included in the RPF list for the customer interface in consideration. Loose uRPF (see Section 2.4) is recommended to be applied to the peer and provider interfaces in consideration.

Thus, enhanced feasible-path uRPF defines feasible paths for customer interfaces in a more generalized but precise way (as compared to feasible-path uRPF).

Looking back at Scenarios 1 and 2 (Figure 1 and Figure 2), the enhanced feasible-path uRPF provides comparable or better performance than the other uRPF methods. Scenario 3 (Figure 3) further illustrates the enhanced feasible-path uRPF method with a more concrete example. In this scenario, the focus is on operation of the feasible-path uRPF at ISP4 (AS4). ISP4 learns a route for prefix P1 via a customer-to-provider (C2P) interface from customer ISP2 (AS2). This route for P1 has origin AS1. ISP4 also learns a route for P2 via another C2P interface from customer ISP3 (AS3). Additionally, AS4 learns a route for P3 via a peer-to-peer (p2p) interface from ISP5 (AS5). Routes for all three prefixes have the same origin AS (i.e. AS1). Using the enhanced feasible-path uRPF scheme, given the commonality of the origin AS across the routes for P1, P2 and P3, AS4 includes all of these prefixes to the RPF list for the customer interfaces (from AS2 and AS3).

```
              +----------+   P3[AS5 AS1]  +------------+
              | AS4(ISP4)|<--------------|  AS5(ISP5) |
              +---------+      (p2p)      +-----------+
               /\   /\                           /\
              /     \                           /
     P1[AS2 AS1]/     \P2[AS3 AS1]             /
        (C2P)/        \(C2P)                  /
           /            \                    /
      +----------+    +----------+          /
      | AS2(ISP2)|    | AS3(ISP3)|         /
      +---------+     +---------+         /
           /\            /\             /
            \           /             /
      P1[AS1]\        /P2[AS1]   /P3[AS1]
        (C2P)\      /(C2P)     /(C2P)
             \     /         /
         +----------------+ /
         |  AS1(customer) |/
         +---------------+
              P1, P2, P3 (prefixes originated)
```
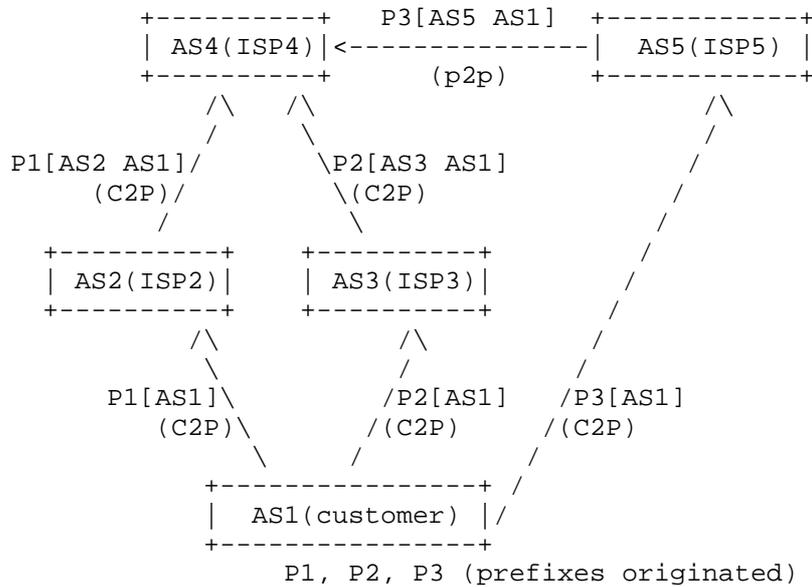
        Consider that data packets (sourced from AS1)
        may be received at AS4 with source address
        in P1, P2 or P3 via any of the neighbors (AS2, AS3, AS5):
        * Feasible-path uRPF fails
        * Loose uRPF works (but not desirable)
        * Enhanced Feasible-path uRPF works best

     Figure 3: Scenario 3 for illustration of efficacy of uRPF schemes.

3.1.1. Algorithm A: Enhanced Feasible-Path uRPF

   The underlying algorithm in the solution method described above can
   be specified as follows (to be implemented in a transit AS):

   1. Create the list of unique origin ASes considering only the routes
       in the Adj-RIB-Ins of customer interfaces. Call it Set A = {AS1,
       AS2, ..., ASn}.

   2. Considering all routes in Adj-RIB-Ins for all interfaces
       (customer, lateral peer, and provider), form the set of unique
       prefixes that have a common origin AS1. Call it Set X1.

   3. Include set X1 in Reverse Path Filter (RPF) list on all customer
       interfaces on which one or more of the prefixes in set X1 were
       received.

   4. Repeat Steps 2 and 3 for each of the remaining ASes in Set A
       (i.e., for ASi, where i = 2, ..., n).

3.2. Operational Recommendations

   The following operational recommendations will make the operation of
   the enhanced feasible-path uRPF robust:

   For multi-homed stub AS:

   o A multi-homed stub AS SHOULD announce at least one of the prefixes
      it originates to each of its transit provider ASes.

   For non-stub AS:

   o A non-stub AS SHOULD also announce at least one of the prefixes it
      originates to each of its transit provider ASes.

   o Additionally, from the routes it has learned from customers, a
      non-stub AS SHOULD announce at least one route per origin AS to
      each of its transit provider ASes.

   (Note: It is worth noting that in the above recommendations if "at
   least one" is replaced with "all", then even traditional feasible-
   path uRPF will work as effectively.)

3.3. A Challenging Scenario

   It should be observed that in the absence of ASes adhering the above
   recommendations, the following example scenario may be constructed
   which poses a challenge for the enhanced feasible-path uRPF (as well
   as for traditional feasible-path uRPF). In the scenario illustrated
   in Figure 4, since routes for neither P1 nor P2 are propagated on the
   AS2-AS4 interface, the enhanced feasible-path uRPF at AS4 will reject
   data packets received on that interface with source addresses in P1
   or P2.

```
                  +----------+
                  | AS4(ISP4)|
                  +----------+
                    /\    /\
                   /    \   P1[AS3 AS1]
       P1 and P2 not /      \ P2[AS3 AS1]
         propagated /        \ (C2P)
            (C2P)   /          \
            +----------+    +----------+
            | AS2(ISP2)|    | AS3(ISP3)|
            +----------+    +----------+
                   /\            /\
                    \          /  P1[AS1]
       P1[AS1] NO_EXPORT \        / P2[AS1]
       P2[AS1] NO_EXPORT  \      /  (C2P)
               (C2P)  \      /
            +----------------+
            |  AS1(customer) |
            +----------------+
                   P1, P2 (prefixes originated)
```
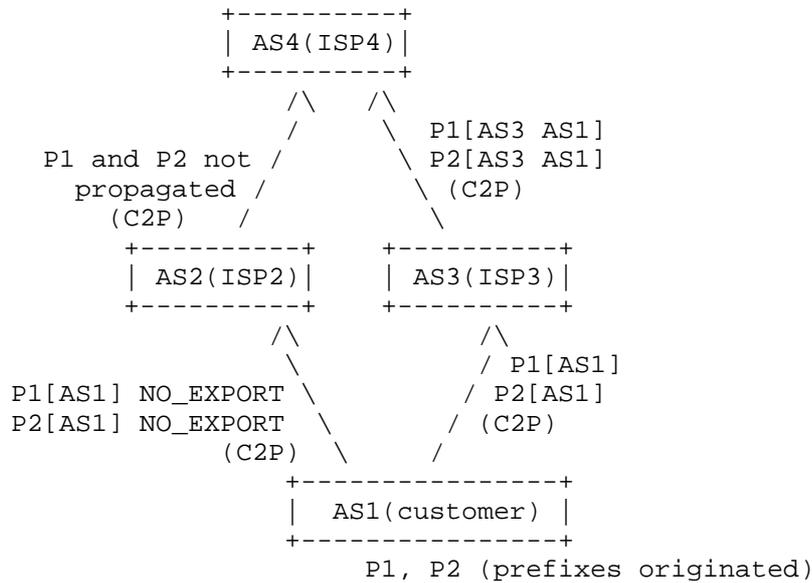
Figure 4: Illustration of a challenging scenario.

3.4. Algorithm B: Enhanced Feasible-Path uRPF with Additional
      Flexibility Across Customer Cone

   Adding further flexibility to the enhanced feasible-path uRPF method
   can help address the potential limitation identified above using the
   scenario in Figure 4 (Section 3.3). In the following, "route" refers
   to a route currently existing in the Adj-RIB-in. Including the
   additional degree of flexibility, the modified algorithm (implemented
   in a transit AS) can be described as follows (we call this Algorithm
   B):

   1. Create the set of all directly-connected customer interfaces.
       Call it Set I = {I1, I2, ..., Ik}.

   2. Create the set of all unique prefixes for which routes exist in
       Adj-RIB-Ins for the interfaces in Set I. Call it Set P = {P1,
       P2, ..., Pm}.

   3. Create the set of all unique origin ASes seen in the routes that
       exist in Adj-RIB-Ins for the interfaces in Set I. Call it Set A
       = {AS1, AS2, ..., ASn}.

   4. Create the set of all unique prefixes for which routes exist in
       Adj-RIB-Ins of all lateral peer and provider interfaces such that

each of the routes has its origin AS belonging in Set A. Call it
Set Q = {Q1, Q2, ..., Qj}.

5. Then, Set Z = Union(P,Q) represents the RPF list for every
    customer interface in Set I.

6. Apply loose uRPF method for SAV on all peer and provider
    interfaces.

When Algorithm B (which is more flexible than Algorithm A) is
employed, the type of limitation identified in Figure 4 (Section 3.3)
goes away.

3.5. Implementation Considerations

The existing RPF checks in edge routers take advantage of existing
line card implementations to perform the RPF functions. For
implementation of the enhanced feasible-path uRPF, the general
necessary feature would be to extend the line cards to take arbitrary
RPF lists that are not necessarily the same as the existing FIB
contents. In the algorithms (Section 3.1.1 and Section 3.4)
described here, the RPF lists are constructed by applying a set of
rules to all received BGP routes (not just those selected as best
path and installed in FIB).

3.5.1. Impact on FIB Memory Size Requirement

The techniques described here require that there should be FIB memory
(i.e., TCAM) available to store the RPF lists in line cards. For an
ISP's AS, the RPF list size for each line card will roughly and
conservatively equal the total number of prefixes in its customer
cone (assuming the algorithm in Section 3.4 is used). (Note: Most
ISP customer cone scenarios would not require the algorithm in
Section 3.4, but instead be served best by the algorithm in
Section 3.1.1, which requires much less FIB memory.) The following
table shows the measured customer cone sizes for various types of
ISPs [sriram-ripe63]:

+-------------------------------+-------------------------------+
Type of ISP	Measured Customer Cone Size in
	# Prefixes (in turn this is an
	estimate for RPF list size on
	line card)
+-------------------------------+-------------------------------+
Very Large Global ISP	32392
Very Large Global ISP	29528
Large Global ISP	20038
Mid-size Global ISP	8661
Regional ISP (in Asia)	1101
+-------------------------------+-------------------------------+

Table 1: Customer cone sizes (# prefixes) for various types of ISPs.

For some super large global ISPs that are at the core of the
Internet, the customer cone size (# prefixes) can be as high as a few
hundred thousand [CAIDA]. But uRPF is most effective when deployed
at ASes at the edges of the Internet where the customer cone sizes
are smaller as shown in Table 1.

A very large global ISP's router line card is likely to have a FIB
size large enough to accommodate 2 to 6 million routes [cisco1].
Similarly, the line cards in routers corresponding to a large global
ISP, a mid-size global ISP, and a regional ISP are likely to have FIB
sizes large enough to accommodate about 1 million, 0.5 million, and
100K routes, respectively [cisco2]. Comparing these FIB size numbers
with the corresponding RPF list size numbers in Table 1, it can be
surmised that the conservatively estimated RPF list size is only a
small fraction of the anticipated FIB memory size under relevant ISP
scenarios.

3.6. Summary of Recommendations

   Depending on the scenario, an ISP or enterprise AS operator should
   follow one of the following recommendations concerning uRPF/SAV:

   1. For directly connected networks, i.e., subnets directly connected
       to the AS and not multi-homed, the AS in consideration SHOULD
       perform ACL-based SAV.

   2. For a directly connected single-homed stub AS (customer), the AS
       in consideration SHOULD perform SAV based on the strict uRPF
       method.

   3. For all other scenarios:

       * If the scenario does not involve complexity such as NO_EXPORT
          of routes (see Section 3.3, Figure 4), then the enhanced
          feasible-path uRPF method in Algorithm A (see Section 3.1.1)
          SHOULD be applied.

       * Else, if the scenario involves the aforementioned complexity,
          then the enhanced feasible-path uRPF method in Algorithm B
          (see Section 3.4) SHOULD be applied.

4. Security Considerations

   The security considerations in BCP 38 [RFC2827] and BCP 84 [RFC3704]
   apply for this document as well. In addition, AS operator should
   apply the uRPF method that performs best (i.e., with zero or
   insignificant possibility of dropping legitimate data packets) for
   the type of peer (customer, provider, etc.) and the nature of
   customer cone scenario that apply (see Section 3.1.1 and
   Section 3.4).

5. IANA Considerations

   This document does not request new capabilities or attributes. It
   does not create any new IANA registries.

6. Acknowledgements

   The authors would like to thank Job Snijders, Marco Marzetti, Marco
   d'Itri, Nick Hilliard, Gert Doering, Igor Gashinsky, Barry Greene,
   and Joel Jaeggli for comments and suggestions.

7. Informative References

   [CAIDA]   "Information for AS 174 (COGENT-174)", CAIDA Spoofer
              Project , <https://spoofer.caida.org/as.php?asn=174>.

   [cisco1]   "Internet Routing Table Growth Causes ROUTING-FIB-
              4-RSRC_LOW Message on Trident-Based Line Cards", Cisco
              Trouble-shooting Tech-notes , January 2014,
              <https://www.cisco.com/c/en/us/support/docs/routers/asr-
              9000-series-aggregation-services-routers/116999-problem-
              line-card-00.html>.

   [cisco2]   "Cisco Nexus 7000 Series NX-OS Unicast Routing
              Configuration Guide, Release 5.x (Chapter: Managing the
              Unicast RIB and FIB)", Cisco Configuration Guides , June
              2018, <https://www.cisco.com/c/en/us/td/docs/switches/data
              center/sw/5_x/nx-
              os/unicast/configuration/guide/l3_cli_nxos/
              l3_manage-routes.html#22859>.

   [ISOC]   Vixie (Ed.), P., "Addressing the challenge of IP
              spoofing", ISOC report , September 2015,
              <https://www.us-cert.gov/ncas/alerts/TA14-017A>.

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

   [RFC2827] Ferguson, P. and D. Senie, "Network Ingress Filtering:
              Defeating Denial of Service Attacks which employ IP Source
              Address Spoofing", BCP 38, RFC 2827, DOI 10.17487/RFC2827,
              May 2000, <https://www.rfc-editor.org/info/rfc2827>.

   [RFC3704] Baker, F. and P. Savola, "Ingress Filtering for Multihomed
              Networks", BCP 84, RFC 3704, DOI 10.17487/RFC3704, March
              2004, <https://www.rfc-editor.org/info/rfc3704>.

   [RFC4271] Rekhter, Y., Ed., Li, T., Ed., and S. Hares, Ed., "A
              Border Gateway Protocol 4 (BGP-4)", RFC 4271,
              DOI 10.17487/RFC4271, January 2006,
              <https://www.rfc-editor.org/info/rfc4271>.

   [RFC6811] Mohapatra, P., Scudder, J., Ward, D., Bush, R., and R.
              Austein, "BGP Prefix Origin Validation", RFC 6811,
              DOI 10.17487/RFC6811, January 2013,
              <https://www.rfc-editor.org/info/rfc6811>.

   [RFC7454] Durand, J., Pepelnjak, I., and G. Doering, "BGP Operations
              and Security", BCP 194, RFC 7454, DOI 10.17487/RFC7454,
              February 2015, <https://www.rfc-editor.org/info/rfc7454>.

   [RRL]   "Response Rate Limiting in the Domain Name System",
              Redbarn blog , <http://www.redbarn.org/dns/ratelimits>.

   [sriram-ripe63]
            Sriram, K. and R. Bush, "Estimating CPU Cost of BGPSEC on
            a Router", Presented at RIPE-63; also at IETF-83 SIDR WG
            Meeting, March 2012,
            <http://www.ietf.org/proceedings/83/slides/
            slides-83-sidr-7.pdf>.

   [TA14-017A]
            "UDP-Based Amplification Attacks", US-CERT alert
            TA14-017A , January 2014,
            <https://www.us-cert.gov/ncas/alerts/TA14-017A>.

Authors' Addresses

   Kotikalapudi Sriram
   USA National Institute of Standards and Technology
   100 Bureau Drive
   Gaithersburg MD 20899
   USA

   Email: ksriram@nist.gov

   Doug Montgomery
   USA National Institute of Standards and Technology
   100 Bureau Drive
   Gaithersburg MD 20899
   USA

   Email: dougm@nist.gov

   Jeffrey Haas
   Juniper Networks, Inc.
   1133 Innovation Way
   Sunnyvale CA 94089
   USA

   Email: jhaas@juniper.net