           DHCPv6-Shield: Protecting against Rogue DHCPv6 Servers

Abstract

   This document specifies a mechanism for protecting hosts connected to
   a switched network against rogue DHCPv6 servers. It is based on
   DHCPv6 packet filtering at the layer 2 device at which the packets
   are received. A similar mechanism has been widely deployed in IPv4
   networks ('DHCP snooping'); hence, it is desirable that similar
   functionality be provided for IPv6 networks. This document specifies
   a Best Current Practice for the implementation of DHCPv6-Shield.

Table of Contents

1. Introduction

   This document specifies DHCPv6-Shield, a mechanism for protecting
   hosts connected to a switched network against rogue DHCPv6 servers
   [RFC3315]. The basic concept behind DHCPv6-Shield is that a layer 2
   device filters DHCPv6 messages intended for DHCPv6 clients
   (henceforth, "DHCPv6-server messages"), according to a number of
   different criteria. The most basic filtering criterion is that
   DHCPv6-server messages are discarded by the layer 2 device unless
   they are received on specific ports of the layer 2 device.

   Before the DHCPv6-Shield device is deployed, the administrator
   specifies the layer 2 port(s) on which DHCPv6-server messages are to
   be allowed. Only those ports to which a DHCPv6 server or relay is to
   be connected should be specified as such. Once deployed, the
   DHCPv6-Shield device inspects received packets and allows (i.e.,
   passes) DHCPv6-server messages only if they are received on layer 2
   ports that have been explicitly configured for such purpose.

   DHCPv6-Shield is analogous to the Router Advertisement Guard
   (RA-Guard) mechanism [RFC6104] [RFC6105] [RFC7113], intended for
   protection against rogue Router Advertisement [RFC4861] messages.

   We note that DHCPv6-Shield mitigates only DHCPv6-based attacks
   against hosts. Attack vectors based on other messages meant for
   network configuration (such as ICMPv6 Router Advertisements) are not
   addressed by DHCPv6-Shield itself. In a similar vein, DHCPv6-Shield
   does not mitigate attacks against DHCPv6 servers (e.g., Denial of
   Service).

2. Requirements Language

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

3. Terminology

   DHCPv6-Shield:

      The set of filtering rules specified in this document, meant to
      mitigate attacks that employ DHCPv6-server packets.

   DHCPv6-Shield device:

      A layer 2 device (typically a layer 2 switch) that enforces the
      filtering policy specified in this document.

For the purposes of this document, the terms "IPv6 Extension Header",
"First Fragment", "IPv6 Header Chain", and "Upper-Layer Header" are
used as specified in [RFC7112]:

IPv6 Extension Header:

   IPv6 Extension Headers are defined in Section 4 of [RFC2460]. As
   a result of [RFC7045], [IANA-PROTO] provides a list of assigned
   Internet Protocol Numbers and designates which of those protocol
   numbers also represent IPv6 Extension Headers.

First Fragment:

   An IPv6 fragment with a Fragment Offset equal to 0.

IPv6 Header Chain:

   The IPv6 Header Chain contains an initial IPv6 header, zero or
   more IPv6 Extension Headers, and optionally, a single Upper-Layer
   Header. If an Upper-Layer Header is present, it terminates the
   IPv6 Header Chain; otherwise, the "No Next Header" value (Next
   Header = 59) terminates it.

   The first member of the IPv6 Header Chain is always an IPv6
   header. For a subsequent header to qualify as a member of the
   IPv6 Header Chain, it must be referenced by the "Next Header"
   field of the previous member of the IPv6 Header Chain. However,
   if a second IPv6 header appears in the IPv6 Header Chain, as is
   the case when IPv6 is tunneled over IPv6, the second IPv6 header
   is considered to be an Upper-Layer Header and terminates the IPv6
   Header Chain. Likewise, if an Encapsulating Security Payload
   (ESP) header appears in the IPv6 Header Chain, it is considered to
   be an Upper-Layer Header, and it terminates the IPv6 Header Chain.

Upper-Layer Header:

   In the general case, the Upper-Layer Header is the first member of
   the Header Chain that is neither an IPv6 header nor an IPv6
   Extension Header. However, if either an ESP header or a second
   IPv6 header occurs in the IPv6 Header Chain, it is considered to
   be an Upper-Layer Header, and it terminates the IPv6 Header Chain.

   Neither the upper-layer payload nor any protocol data following
   the upper-layer payload is considered to be part of the IPv6
   Header Chain. In a simple example, if the Upper-Layer Header is a
   TCP header, the TCP payload is not part of the IPv6 Header Chain.
   In a more complex example, if the Upper-Layer Header is an ESP

       header, neither the payload data nor any of the fields that follow
       the payload data in the ESP header are part of the IPv6 Header
       Chain.

4. DHCPv6-Shield Configuration

   Before being deployed for production, the DHCPv6-Shield device is
   explicitly configured with respect to which layer 2 ports are allowed
   to receive DHCPv6 packets destined to DHCPv6 clients (i.e.,
   DHCPv6-server messages). Only those layer 2 ports explicitly
   configured for such purpose are allowed to receive DHCPv6 packets to
   pass to DHCPv6 clients.

5. DHCPv6-Shield Implementation Requirements

   Following are the filtering rules that are enforced as part of a
   DHCPv6-Shield implementation on those ports that are not allowed to
   receive DHCPv6 packets to DHCPv6 clients:

   1. DHCPv6-Shield implementations MUST parse the entire IPv6 Header
       Chain present in the packet to identify whether or not it is a
       DHCPv6 packet meant for a DHCPv6 client (i.e., a DHCPv6-server
       message).

           RATIONALE: DHCPv6-Shield implementations MUST NOT enforce a
           limit on the number of bytes they can inspect (starting from
           the beginning of the IPv6 packet), since this could introduce
           false negatives: DHCP6-server packets received on ports not
           allowed to receive such packets could be allowed simply
           because the DHCPv6-Shield device does not parse the entire
           IPv6 Header Chain present in the packet.

   2. When parsing the IPv6 Header Chain, if the packet is a First
       Fragment (i.e., a packet containing a Fragment Header with the
       Fragment Offset set to 0) and it fails to contain the entire IPv6
       Header Chain (i.e., all the headers starting from the IPv6 header
       up to, and including, the Upper-Layer Header), DHCPv6-Shield MUST
       drop the packet and ought to log the packet drop event in an
       implementation-specific manner as a security fault.

           RATIONALE: Packets that fail to contain the entire IPv6 Header
           Chain could otherwise be leveraged for circumventing
           DHCPv6-Shield. [RFC7112] requires that the First Fragment
           (i.e., the fragment with the Fragment Offset set to 0) contain
           the entire IPv6 Header Chain. [RFC7112] also allows
           intermediate systems such as routers to drop packets that fail
           to comply with this requirement.

NOTE: This rule should only be applied to IPv6 fragments with
a Fragment Offset of 0 (non-First Fragments can be safely
passed, since they will never reassemble into a complete
datagram if they are part of a DHCPv6 packet meant for a
DHCPv6 client received on a port where such packets are not
allowed).

3. DHCPv6-Shield MUST provide a configuration knob that controls
whether or not packets with unrecognized Next Header values are
dropped; this configuration knob MUST default to "drop". When
parsing the IPv6 Header Chain, if the packet contains an
unrecognized Next Header value and the configuration knob is
configured to "drop", DHCPv6-Shield MUST drop the packet and
ought to log the packet drop event in an implementation-specific
manner as a security fault.

RATIONALE: An unrecognized Next Header value could possibly
identify an IPv6 Extension Header and thus be leveraged to
conceal a DHCPv6-server packet (since there is no way for
DHCPv6-Shield to parse past unrecognized Next Header values
[IPV6-UEH]). [RFC7045] requires that nodes be configurable
with respect to whether or not packets with unrecognized
headers are forwarded and allows the default behavior to be
that such packets be dropped.

4. When parsing the IPv6 Header Chain, if the packet is identified
to be a DHCPv6 packet meant for a DHCPv6 client, DHCPv6-Shield
MUST drop the packet and SHOULD log the packet drop event in an
implementation-specific manner as a security alert.

RATIONALE: Ultimately, the goal of DHCPv6-Shield is to drop
DHCPv6 packets destined to DHCPv6 clients (i.e., DHCPv6-server
messages) that are received on ports that have not been
explicitly configured to allow the receipt of such packets.

5. In all other cases, DHCPv6-Shield MUST pass the packet as usual.

NOTE: For the purpose of enforcing the DHCPv6-Shield filtering
policy, an ESP header [RFC4303] should be considered to be an
"upper-layer protocol" (that is, it should be considered the last
header in the IPv6 Header Chain). This means that packets
employing ESP would be passed by the DHCPv6-Shield device to the
intended destination. If the destination host does not have a
security association with the sender of the aforementioned IPv6
packet, the packet would be dropped. Otherwise, if the packet is
considered valid by the IPsec implementation at the receiving host
and encapsulates a DHCPv6 message, what to do with such a packet
is up to the receiving host.

The rules above indicate that if a packet is dropped due to this
filtering policy, the packet drop event should be logged in an
implementation-specific manner as a security fault. It is useful for
the logging mechanism to include a per-port drop counter dedicated to
DHCPv6-Shield packet drops.

In order to protect current end-node IPv6 implementations, Rule #2
has been defined such that the default is for packets that cannot be
positively identified as not being DHCPv6-server packets (because the
packet is a fragment that fails to include the entire IPv6 Header
Chain) to be dropped. This means that, at least in theory,
DHCPv6-Shield could result in false-positive blocking of some
legitimate (non-DHCPv6-server) packets. However, as noted in
[RFC7112], IPv6 packets that fail to include the entire IPv6 Header
Chain are virtually impossible to police with stateless filters and
firewalls; hence, they are unlikely to survive in real networks.
[RFC7112] requires that hosts employing fragmentation include the
entire IPv6 Header Chain in the First Fragment (the fragment with the
Fragment Offset set to 0), thus eliminating the aforementioned false
positives.

The aforementioned filtering rules implicitly handle the case of
fragmented packets: if the DHCPv6-Shield device fails to identify the
upper-layer protocol as a result of the use of fragmentation, the
corresponding packets would be dropped.

Finally, we note that IPv6 implementations that allow overlapping
fragments (i.e., that do not comply with [RFC5722]) might still be
subject of DHCPv6-based attacks. However, a recent assessment of
IPv6 implementations [SI6-FRAG] with respect to their fragment
reassembly policy seems to indicate that most current implementations
comply with [RFC5722].

6. Security Considerations

The recommendations in this document represent the ideal behavior of
a DHCPv6-Shield device. However, in order to implement DHCPv6-Shield
on the fast path, it may be necessary to limit the depth into the
packet that can be scanned before giving up. In circumstances where
there is such a limitation, it is recommended that implementations
drop packets after attempting to find a protocol header up to that
limit, whatever it is. Ideally, such devices should be configurable
with a list of protocol header identifiers so that if new transport
protocols are standardized after the device is released, they can be
added to the list of protocol header types that the device
recognizes. Since any protocol header that is not a UDP header would
be passed by the DHCPv6-Shield algorithm, this would allow such
devices to avoid blocking the use of new transport protocols. When

an implementation must stop searching for recognizable header types
in a packet due to such limitations, the device SHOULD be
configurable to either pass or drop that packet.

The mechanism specified in this document can be used to mitigate
DHCPv6-based attacks against hosts. Attack vectors based on other
messages meant for network configuration (such as ICMPv6 Router
Advertisements) are out of the scope of this document. Additionally,
the mechanism specified in this document does not mitigate attacks
against DHCPv6 servers (e.g., Denial of Service).

If deployed in a layer 2 domain with several cascading switches,
there will be an ingress port on the host's local switch that will
need to be enabled for receiving DHCPv6-server messages. However,
this local switch will be reliant on the upstream devices filtering
out rogue DHCPv6-server messages, as the local switch has no way of
determining which upstream DHCP-server messages are valid.
Therefore, in order to be effective, DHCPv6-Shield should be deployed
and enabled on all layer 2 switches of a given layer 2 domain.

As noted in Section 5, IPv6 implementations that allow overlapping
fragments (i.e., that do not comply with [RFC5722]) might still be
subject to DHCPv6-based attacks. However, most current
implementations seem to comply with [RFC5722] and hence forbid IPv6
overlapping fragments.

We note that if an attacker sends a fragmented DHCPv6 packet on a
port not allowed to receive such packets, the First Fragment would be
dropped, and the rest of the fragments would be passed. This means
that the victim node would tie memory buffers for the aforementioned
fragments, which would never reassemble into a complete datagram. If
a large number of such packets were sent by an attacker, and the
victim node failed to implement proper resource management for the
fragment reassembly buffer, this could lead to a Denial of Service
(DoS). However, this does not really introduce a new attack vector,
since an attacker could always perform the same attack by sending a
forged fragmented datagram in which at least one of the fragments is
missing. [CPNI-IPv6] discusses some resource management strategies
that could be implemented for the fragment reassembly buffer.

Additionally, we note that the security of a site employing
DHCPv6-Shield could be further improved by deploying [RFC7513] to
mitigate IPv6 address spoofing attacks.

Finally, we note that other mechanisms for mitigating attacks based
on DHCPv6-server messages are available that have different
deployment considerations. For example, [SECURE-DHCPV6] allows for
authentication of DHCPv6-server packets if the IPv6 addresses of the
DHCPv6 servers can be pre-configured at the client nodes.

7. References

7.1. Normative References

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

   [RFC2460]   Deering, S. and R. Hinden, "Internet Protocol, Version 6
                (IPv6) Specification", RFC 2460, DOI 10.17487/RFC2460,
                December 1998, <http://www.rfc-editor.org/info/rfc2460>.

   [RFC3315]   Droms, R., Ed., Bound, J., Volz, B., Lemon, T., Perkins,
                C., and M. Carney, "Dynamic Host Configuration Protocol
                for IPv6 (DHCPv6)", RFC 3315, DOI 10.17487/RFC3315, July
                2003, <http://www.rfc-editor.org/info/rfc3315>.

   [RFC4303]   Kent, S., "IP Encapsulating Security Payload (ESP)", RFC
                4303, DOI 10.17487/RFC4303, December 2005,
                <http://www.rfc-editor.org/info/rfc4303>.

   [RFC4861]   Narten, T., Nordmark, E., Simpson, W., and H. Soliman,
                "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861,
                DOI 10.17487/RFC4861, September 2007,
                <http://www.rfc-editor.org/info/rfc4861>.

   [RFC5722]   Krishnan, S., "Handling of Overlapping IPv6 Fragments",
                RFC 5722, DOI 10.17487/RFC5722, December 2009,
                <http://www.rfc-editor.org/info/rfc5722>.

   [RFC7045]   Carpenter, B. and S. Jiang, "Transmission and Processing
                of IPv6 Extension Headers", RFC 7045,
                DOI 10.17487/RFC7045, December 2013,
                <http://www.rfc-editor.org/info/rfc7045>.

   [RFC7112]   Gont, F., Manral, V., and R. Bonica, "Implications of
                Oversized IPv6 Header Chains", RFC 7112,
                DOI 10.17487/RFC7112, January 2014,
                <http://www.rfc-editor.org/info/rfc7112>.

7.2. Informative References

   [CPNI-IPv6]   Gont, F., "Security Assessment of the Internet Protocol
                 version 6 (IPv6)", UK Centre for the Protection of
                 National Infrastructure, (available on request).

   [IANA-PROTO] IANA, "Protocol Numbers",
                 <http://www.iana.org/assignments/protocol-numbers>.

   [IPV6-UEH]   Gont, F., Liu, W., Krishnan, S., and H. Pfeifer, "IPv6
                 Universal Extension Header", Work in Progress,
                 draft-gont-6man-rfc6564bis-00, April 2014.

   [RFC6104]   Chown, T. and S. Venaas, "Rogue IPv6 Router
                 Advertisement Problem Statement", RFC 6104,
                 DOI 10.17487/RFC6104, February 2011,
                 <http://www.rfc-editor.org/info/rfc6104>.

   [RFC6105]   Levy-Abegnoli, E., Van de Velde, G., Popoviciu, C., and
                 J. Mohacsi, "IPv6 Router Advertisement Guard", RFC
                 6105, DOI 10.17487/RFC6105, February 2011,
                 <http://www.rfc-editor.org/info/rfc6105>.

   [RFC7113]   Gont, F., "Implementation Advice for IPv6 Router
                 Advertisement Guard (RA-Guard)", RFC 7113,
                 DOI 10.17487/RFC7113, February 2014,
                 <http://www.rfc-editor.org/info/rfc7113>.

   [RFC7513]   Bi, J., Wu, J., Yao, G., and F. Baker, "Source Address
                 Validation Improvement (SAVI) Solution for DHCP", RFC
                 7513, DOI 10.17487/RFC7513, May 2015,
                 <http://www.rfc-editor.org/info/rfc7513>.

   [SECURE-DHCPV6]
                 Jiang, S. and S. Shen, "Secure DHCPv6 Using CGAs", Work
                 in Progress, draft-ietf-dhc-secure-dhcpv6-07, September
                 2012.

   [SI6-FRAG]   SI6 Networks, "IPv6 NIDS evasion and improvements in
                 IPv6 fragmentation/reassembly", 2012,
                 <http://blog.si6networks.com/2012/02/
                 ipv6-nids-evasion-and-improvements-in.html>.

Acknowledgements

   The authors would like to thank Mike Heard, who provided detailed
   feedback on earlier draft versions of this document and helped a lot
   in producing a technically sound document throughout the whole
   publication process.

   The authors would like to thank (in alphabetical order) Ben Campbell,
   Jean-Michel Combes, Sheng Jiang, Ted Lemon, Pete Resnick, Carsten
   Schmoll, Juergen Schoenwaelder, Robert Sleigh, Donald Smith, Mark
   Smith, Hannes Tschofenig, Eric Vyncke, and Qin Wu for providing
   valuable comments on earlier draft versions of this document.

   Part of Section 3 of this document was borrowed from [RFC7112],
   authored by Fernando Gont, Vishwas Manral, and Ron Bonica.

   This document is heavily based on [RFC7113], authored by Fernando
   Gont. Thus, the authors would like to thank the following
   individuals for providing valuable comments on [RFC7113]: Ran
   Atkinson, Karl Auer, Robert Downie, Washam Fan, David Farmer, Mike
   Heard, Marc Heuse, Nick Hilliard, Ray Hunter, Joel Jaeggli, Simon
   Perreault, Arturo Servin, Gunter Van de Velde, James Woodyatt, and
   Bjoern A. Zeeb.

   The authors would like to thank Joel Jaeggli for his advice and
   guidance throughout the IETF process.

   Fernando Gont would like to thank Diego Armando Maradona for his
   magic and inspiration.

Authors' Addresses

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

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

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

   Email: liushucheng@huawei.com

   Gunter Van de Velde
   Alcatel-Lucent
   Copernicuslaan 50
   Antwerp, Antwerp 2018
   Belgium

   Phone: +32 476 476 022
   Email: gunter.van_de_velde@alcatel-lucent.com