IPv6 Security for Enterprise Organizations

Enno Rey @enno_insinuator
Christopher Werny @bcp38_

#whoarewe

- Old-school networking guys, with a special focus on security (www.ernw.de)
- Doing quite some stuff in the IPv6 space
  - https://insinuator.net/2019/01/ipv6-talks-publications
- Operating a (medium-size) conference network with v6-only+NAT64 in the default SSID since 2016

Agenda

- Some Discussion: Why IPv6 Is Different, Security-wise
- Traffic Filtering in IPv6 Networks
- (short break)
- IPv6 Security in L2 Networks / First Hop Security et al.
- Conclusions

$SECURITY_OF_A_PROTOCOL / Factors

- Properties of $PROTOCOL
- Attack surface / “exposure to incidents”
- State of security controls
  - Availability (of controls)
  - Feature effectiveness & maturity
  - Operational feasibility
- Experience of operators, and vendors ;-)

See also: https://insinuator.net/2014/11/protocol-properties-attack-vectors/

Recent Sample

Cisco Nexus 9000 Series Fabric Switches Application Centi
Mode Default SSH Key Vulnerability

Critical

Advisory ID:	cisco-sa-20190501-nexus9k-sshkey	CVE-2019-1804	Download CVRF
First Published:	2019 May 1 16:00 GMT	CWE-310	Download PDF
Last Updated:	2019 May 2 17:09 GMT		Email
Version 1.1:	Interim		
Workarounds:	No workarounds available		
Cisco Bug IDs:	CSCvn80686		
CVSS Score:	Base 9.8		

Summary

A vulnerability in the SSH key management for the Cisco Nexus 9000 Series Application Centric Infrastructure (ACI) Mode Switch Software could allow an unauthenticated, remote attacker to connect to the affected system with the privileges of the *root* user.

The vulnerability is due to the presence of a default SSH key pair that is present in all devices. An attacker could exploit this vulnerability by opening an SSH connection via IPv6 to a targeted device using the extracted key materials. An exploit could allow the attacker to access the system with the privileges of the *root* user. This vulnerability is only exploitable over IPv6; IPv4 is not vulnerable.

See also:
https://tools.cisco.com/security/center/content/CiscoSecurityAdvisory/cisco-sa-20190501-nexus9k-sshkey

ERNW
providing security.

Differences

- Increased complexity
  - This mostly applies to the *local link*
  - See also:
    - https://insinuator.net/2015/05/ipv6-complexity/
    - https://ripe74.ripe.net/archives/video/58/ [from 7:10]
- Parameter provisioning & trust model
  - Again this mostly applies to Ethernet networks
- Extension headers
- Multiple addresses per interface
  - Impact on filtering approach/rules

See also:
https://insinuator.net/2015/06/is-ipv6-more-secure-than-ipv4-or-less/

https://www.ernw.de/download/Enno_Rey_RIPE74_Structural_Deficits_IPv6.pdf

What’s a Router? (I)

- Wikipedia:
  - router = “a **router** is a device that forwards *data packets* between *computer networks*”

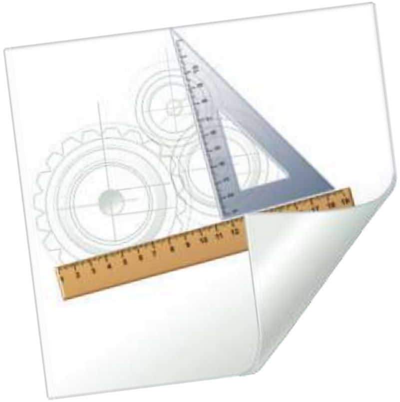
- RFC 2460:
  - router: “router - a node that forwards IPv6 packets not explicitly addressed to itself.”

What's a *Router*, in IPv6?

Looking Closer

- RFC 2461: "Routers advertise their presence together **with various link and Internet parameters** either periodically, or in response to a Router Solicitation message".

- In the end of the day, in IPv6 a router is not just a forwarding device but a provisioning system as well.



IPv6's Trust Model

On the *local link* we're all brothers & sisters.

Do It Like Jim

What an IPv6 Datagrams Looks Like...

Problem

- Variable types
- Variable sizes
- Variable order
- Variable number of occurrences of each one.
- Variable fields

IPv6 = f(v,w,x,y,z)

Security Problems Due to EHs

- Heavily increased parsing complexity
- Evasion of blacklist-based security controls
  - IDPS systems.
  - First Hop Security (FHS) features
  - Insufficient ACL/filtering implementations.
- For the record
  - "EHs" in the terminology of most sec ppl encompass: HBH, DestOptions, RH, FragHdr
  - AH &ESP have their (legitimate) role.
  - But nothing else…

https://www.ernw.de/download/eu-14-Atlasis-Rey-Schaefer-briefings-Evasion-of-HighEnd-IPS-Devices-wp.pdf

To Give You an Idea

For more details see also https://ripe76.ripe.net/wp-content/uploads/presentations/67-RIPE76_JHammer_RFC6980.pdf

Test Case No.	Description	Chiron Options Used (in addition to baseline cmd)	Impact on Target OS' IPv6 Config (without RA Guard)	What was observed in Wireshark on Target OS? (without RA Guard)	What still got through with RA Guard enabled?	Overall Result With RA Guard Enabled
13	Two fragments, with two DestOptions in fragmentable part	-lfE 60,60 -nf 2	Added 2nd default gw, created additional address	One fragment plus RA packet which contains two DestOptions EHs	1st fragment, but *not* the RA	No impact
14	Four fragments, with two DestOptions in fragmentable part	-lfE 60,60 -nf 4	Added 2nd default gw, created additional address	Three fragments plus RA packet which contains two DestOptions	Three fragments, plus RA containing two DestOptions EHs. Nothing logged on the switch.	Successful attack
15	Two fragments, with two RoutingHdr EHs in fragmentable part	-lfE 43,43 -nf 2	Added 2nd default gw, created additional address	One fragment plus RA packet which contains two RoutingHdr EHs	Two fragments, plus RA containing EHs. "traceback" on switch console when running 15.0(2)SE2	Successful attack when switch runs 15.0(2)SE2, no impact when switch runs 15.0(2)SE10a
16	Two fragments, with two RHs and two DestOptions, in mixed order	-lfE 60,43,60,43 -nf 2	Added 2nd default gw, created additional address	One fragment plus RA packet which contains the four EHs	1st fragment, but *not* RA	No impact
17	Same as 16 but four fragments	-lfE 60,43,60,43 -nf 4	none	1st three segments only, but not RA	1st three fragments, but not RA	No impact
18	Same as 16 but three fragments	-lfE 60,43,60,43 -nf 3	Added 2nd default gw, created additional address	Two fragments, then RA containing all EHs	1st two fragments plus RA	Successful attack

CVE 2019-5597

Packet Filter is OpenBSD's service for filtering network traffic and performing Network Address Translation. Packet Filter is also capable of normalizing and conditioning TCP/IP traffic, as well as providing bandwidth control and packet prioritization. Packet Filter has been a part of the GENERIC kernel since OpenBSD 5.0.

Because other BSD variants import part of OpenBSD code, Packet Filter is also shipped with at least the following distributions that are affected in a lesser extent:

- FreeBSD
- pfSense
- OPNSense
- Solaris

Note that other distributions may also contain Packet Filter but due to the imported version they might not be vulnerable. This advisory covers the latest OpenBSD's Packet Filter. For specific details about other distributions, please refer to the advisory of the affected product.

The issue

Unless IPv6 reassembly is explicitly disabled, Packet Filter reassembles IPv6 fragments to perform the filtering based on its configuration. The packets are then re-fragmented to comply with the end-to-end nature of the IPv6 fragmentation.

When dealing with malicious fragmented IPv6 packets, the functions *pf_reassemble6()* and *pf_refragment6()*, may use an improper offset to apply a transformation on the packets. This behavior can have the following impacts:

- A kernel panic can happen, effectively stopping the system;
- An unexpected modification of the packets before and after the application of the filtering rules can occur. This may be leveraged to bypass the rules under some circumstances (see Rule bypass p.10).

See also:
https://www.synacktiv.com/ressources/Synacktiv_OpenBSD_PacketFilter_CVE-2019-5597_ipv6_frag.pdf

Properties of Enterprise Networks

- Lots of Ethernet ;-)
  - Data centers
  - Campus networks
    - WiFi
    - Wired
- Security models heavily rely on
  - Filtering (firewalls, ACLs, host-level)
  - Segmentation (?)
  - Hardening (?)

ERNW
providing security.

IPv6 in Town

- Understand what you have/rely on (security-wise)

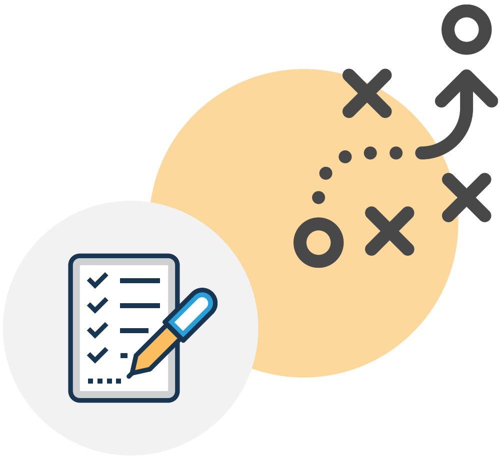
- Understand implications of IPv6
  - Can we do the same (sec) stuff as before?
    Would that make sense? ;-)
    - From protocol design perspective
    - Vendor support (of features)

- Adapt where needed
  - This is what we're going to cover in this tutorial...

Let’s get practical

Areas to Be Considered

- Addressing & Routing
- Server Configuration Approaches & Implications
- Filtering
  - In transit
  - Host level (filtering & hardening)
- The Local Link / First Hop Security



Dual-Stack vs. v6-only

- Strictly speaking not a security topic
- Still there are implications, e.g. in the space of
  - Troubleshooting connectivity issues, namely when traffic passes security controls
  - Increased (double?) effort for filtering rules
  - Logging & analysis & correlation (!)

Address Planning & Security Implications (I)

- We've seen organizations who try to bake a security element into their addressing plans.
  - E.g. by the definition of special bits which then can be specifically considered in firewall rules.

- Interesting idea ;-) ... but we're very skeptical re: (namely long-term) real-life feasibility of such an approach.
  - Don't.

See also:
https://www.ernw.de/download/TR18_NGI_IPv6-Addr-Mgmt-First-5-Years.pdf

https://insinuator.net/2019/02/ipv6-address-management-the-external-flag/

Address Planning & Security (II)

- Some organizations consider substituting the "[inbound] reachability-inhibiting" property of (IPv4) NAT by an approach of "network isolation on the routing layer"
  - *Selective route propagation*
  - *Null-routing of selected prefixes*
- From many perspectives this *can* be a quite elegant and efficient security control, BUT
  - You should really know what you do. More important: all parties involved in operations of your network infrastructure must know and understand this...
  - All usual doubts re: overloading the address plan (semantics-wise) apply...

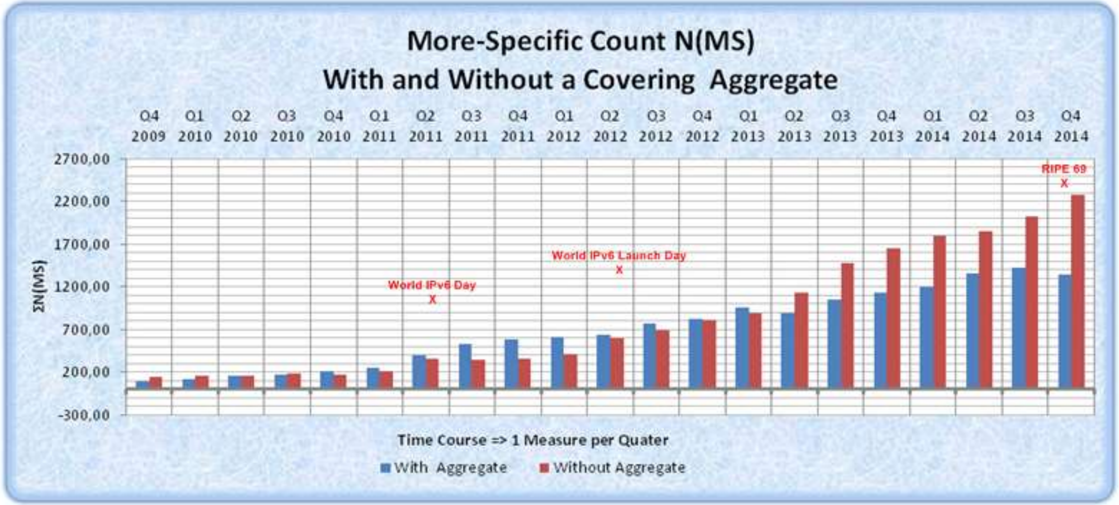
Isolation on Routing Layer

- Selective announcements
  - Keep "strict filtering" in mind
  - See also:
    - RIPE69 AP WG “/48 Considered Harmful”
- Null-routing/blackholing of (to-be) protected prefixes at network borders
  - E.g. prefix used for loopback addresses of network devices
  - This is what we see most often (planned).
- Reduced *hop limit* in specific segments

See also:
https://www.insinuator.net/2015/12/developing-an-enterprise-ipv6-security-strategy-part-2-network-isolation-on-the-routing-layer/

Strict Filtering

Some Numbers (2015)



See also:
https://www.troopers.de/media/filer_public/8a/6c/8a6c1e42-f486-46d7-8161-9cfef4101ecc/tr15_ipv6secsummit_langner_rey_schaetzle_slash48_considered_harmful_update.pdf

Evaluate Carriers Sample

Number	Category	Requirement	XY Expectation	Weight	Provider's Answer	Comment
1	General	IPv6 service level agreements (SLAs) meet or exceed existing/IPv4 SLAs.	Yes	Very high	No	
2	General	IPv6 circuit bandwidth, latency, packet loss, and jitter specifications meet or exceed existing/IPv4 specifications/properties.	Yes	Very high	No	
3	QoS	The QoS policies (queuing/discard) applicable to both IPv4 and IPv6 traffic are identical.	Yes	Very high	No	
4	Metrics	IPv6 performance metrics of $PROVIDER's network will be made available.	Yes	Medium	No	
5	Monitoring	$PROVIDER hosts and provides access to a "looking glass" IPv6 BGP router and/or similar functionality (e.g. an access-controlled monitoring portal) for troubleshooting purposes.	Yes	High	No	
6	MPLS	Full support of MPLS 6VPE (RFC 4659) throughout $PROVIDER's MPLS network.	Yes	High	No	
7	Internet Access	$PROVIDER is willing to accept IPv6 prefix advertisements from XY's RIPE PA space allocation up to /48 _without_ a covering aggregate, provided appropriate route6 objects exist.	Yes	Very high	No	
8	Internet Access	In case answer to previous question is "No", what would be the maximum prefix length that XY can advertise without a covering aggregate?	/48	Very high	No	
9	Internet Access	$PROVIDER does not impose any restrictions on IPv6 prefixes accepted as long as their length is shorter or equal /48 and appropriate route6 objects have been created (that means: "strict filtering" like described in http://www.space.net/~gert/RIPE/ipv6-filters.html will not be applied to XY's IPv6 prefixes).	TRUE	Very high	No	
10	Internet Access	XY's IPv6 own address space can be used in the transit network between $PROVIDER's and XY's BGP router(s)?	Yes	Medium	No	
11	MTU	What is the maximum MTU of IPv6 packets that can be transported without fragmentation through $PROVIDER's network? Different for MPLS network?	Pls specify	Very high	No	
12	MTU	All network devices/hosts under $PROVIDER's control originate ICMPv6 PTB messages when needed.	Yes	Very high	No	
13	MTU	All network devices under $PROVIDER's control pass any ICMPv6 PTB messages in transit which are originated from other devices/hosts.	Yes!	Very high	No	

See also:
https://insinuator.net/2015/01/ipv6-related-requirements-for-the-internet-uplink-or-mpls-networks/

Addressing & Security Implications (III)

- Some people think that going with/ implementing a fully static (IP parameter) configuration approach protects their systems from ND/RA-related attacks.
  - This is not fully correct.
  - The intended security stance is only achieved by additionally disabling the (system-) local processing of RAs.
    - Which in turn has to be carefully evaluated from an operations perspective.

See also:
https://blog.apnic.net/2017/01/16/ipv6-configuration-approaches-servers/

https://www.troopers.de/media/filer_public/ff/9b/ff9b181d-a2f5-4444-9481-73384950094f/ernw_tr16_ipv6secsummit_protectinghosts_final.pdf

Traffic Filtering in the Age of IPv6

ERNW
providing security.

Traffic Filtering

- Variants
  - In transit
    - Internet uplink(s)
    - Network intersection points within corpnet
  - Host based / local

- Main question
  - Differences re: IPv4

Filtering IPv6 / Main Differences

- Do! Extension headers and/or fragments
- Filtering of specific address ranges (multicast and un-assigned by IANA)
- Apply specific rules wrt filtering ICMPv6.

- For Internet uplinks: keep performance impact (in particular from logging) in mind

ERNW
providing security.

Filtering on Internet Uplinks

- Balance between
  - Visibility (of "bad stuff")
  - Speed
- ACL processing in itself shouldn't have too much performance impact on ASR 1K platforms.
  - Disable sending ICMPv6 Type1 might be required for hardware-only processing.
    - Better rate-limit.
  - Protocol type-code access lists always on RP?
- Logging desired/required? – For high speed Internet facing devices going with "drop only" might be preferable.

See also: https://www.insinuator.net/2015/12/developing-an-enterprise-ipv6-security-strategy-part-3-traffic-filtering-in-ipv6-networks-i/

Filtering ICMPv6

- Our recommendation for Internet border gateways

```
permit icmp any any unreachable
permit icmp any any packet-too-big
permit icmp any any hop-limit
permit icmp any any parameter-problem
permit icmp any any echo-request
permit icmp any any echo-reply
permit icmp any any nd-ns
permit icmp any any nd-na
deny icmp any any log-input (?)
```

See also: https://www.insinuator.net/2015/12/developing-an-enterprise-ipv6-security-strategy-part-4-traffic-filtering-in-ipv6-networks-ii/

Infrastructure Controls

- Filtering Extension Headers, Cisco

```
deny ipv6 any any routing
deny ipv6 any any hbh
deny ipv6 any any dest-option
deny ipv6 any any mobility

[allow udp any eq domain $OWN_DNS_SYSTEMS]
deny ipv6 any any fragments              [monitor this!]
[deny ipv6 any any undetermined-transport]
```

Infrastructure Controls

- Commercial Firewalls / Sample
- From: sk39374

By default, Check Point Security Gateway drops all extension headers, except fragmentation. This can be adjusted by editing the `allowed_ipv6_extension_headers` section of `$FWDIR/lib/table.def` file on the Security Management Server.

Furthermore, as of R75.40 there is an option to block type zero even if Routing header is allowed. It is configurable via a kernel parameter `fw6_allow_rh_type_zero`. The default of 0 means it is always blocked. If the value is set to 1, then the action is according to `allowed_ipv6_extension_headers`.

See also:
https://www.troopers.de/wp-content/uploads/2014/01/TROOPERS14-Overview_of_the_Real-World_Capabilities_of_Major_Commercial_Security_Products-Christopher_Werny+Antonios_Atlasis-Part2_2.pdf

Infrastructure Controls

- Filtering unallocated space, Approach (I)

```
deny 0400::/6 any
deny 0800::/5 any
deny 1000::/4 any
deny 2d00::/8 any
deny 2e00::/7 any
deny 3000::/4 any
deny 4000::/3 any
deny 6000::/3 any
deny 8000::/3 any
deny a000::/3 any
deny c000::/3 any
deny e000::/4 any
deny f000::/5 any
deny f800::/6 any
deny fe00::/9 any
```

See also:
http://www.iana.org/assignments/ipv6-address-space/ipv6-address-space.xhtml

http://www.iana.org/assignments/ipv6-unicast-address-assignments/ipv6-unicast-address-assignments.xhtml

Infrastructure Controls

- Filtering *Martians*

```
deny ipv6 host ::1 any log-input
deny ipv6 fc00::/7 any
deny ipv6 fec0::/10 any
deny ipv6 2001:db8::/32 any
deny ipv6 2001:2::/48 any
```

See also:
https://tools.ietf.org/rfc/rfc6890.txt

Infrastructure Controls

- Alternative (better!) approach wrt address space filtering

```
deny ipv6 2001:db8::/32 any
permit ipv6 2000::/3 any
permit ipv6 fe80::/10 any
[permit ipv6 :: any]
deny ipv6 any any
```

ACLs (within Corpnet) / Considerations

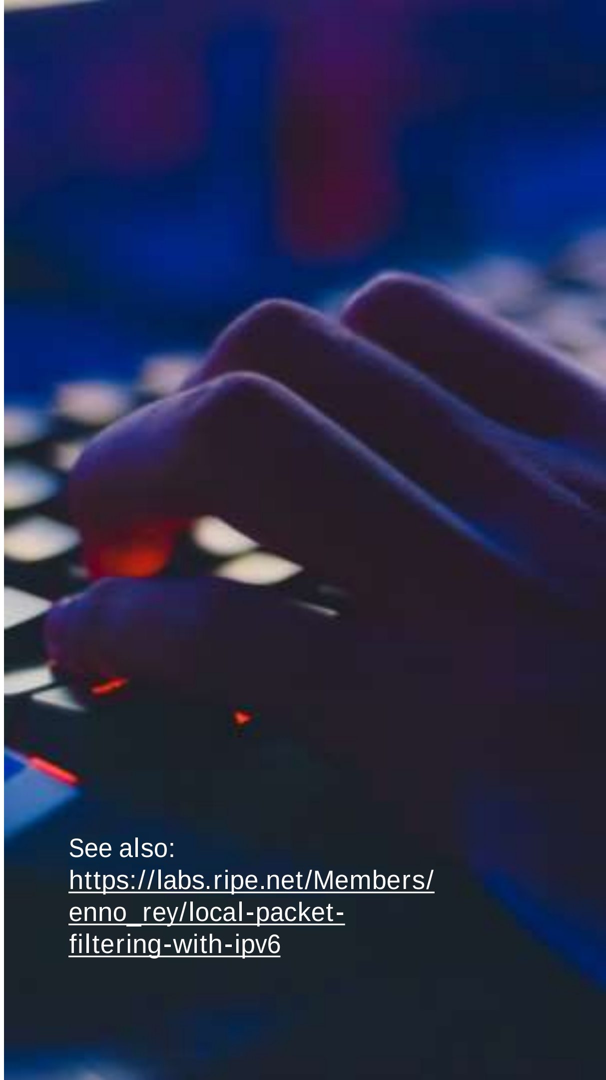
- Re-create vs. “translate”

- Re-creation allows for review of rules
  (re: their necessity) and/or clean-up of unused rules

- Translation (when created automatically)
  - Evidently only works with a well thought-out & universally followed approach
    - Which is what you have, right? ;-)
  - You’ll carry on “the sh*t that had grown over years”…

Host Based Filtering

- Apply with caution, and keep operations implications/efforts in mind.



See also:
https://labs.ripe.net/Members/enno_rey/local-packet-filtering-with-ipv6

Extension Headers

- The term "IPv6 extension headers" denotes the "standard" ones as of RFC 2460 except for AH & ESP, which then leaves: HBH, Routing Header, Fragment Hdr, DestOptions.
- Two main reasons to include them in the filter list:
  - EHs can be abused for nefarious things on the local link/ RFC 6980 might only provide limited protection against RA Guard evasion attacks.
  - Some security products/components might expose a different default stance as for filtering EHs.
- Packets with EHs but otherwise permitted upper layer protocols might not be blocked by a final "default deny" rule.

See also:
https://www.ernw.de/download/Enno_Rey_RIPE74_Structural_Deficits_IPv6.pdf

Extension Headers Recommendation

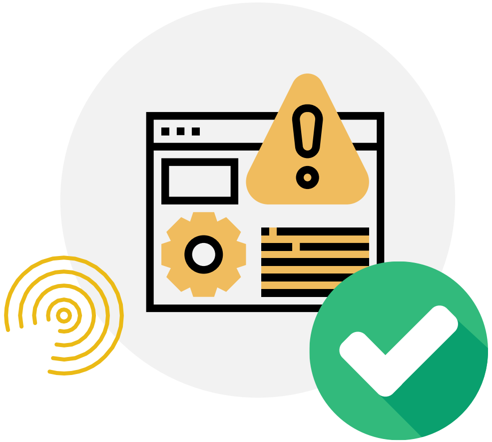
- Allow AH & ESP in case IPsec is needed towards the host.
- Allow HBH in case MLD is needed (see also below).
- Allow fragment header in case you consider it possible that legitimate fragmented packets come in.
  - If you do so, reflect on explicitly denying fragmented RA/ND traffic but this might not be supported configuration-wise and it might be debatable from a rule-set complexity/operational effort perspective.
- Explicitly deny other EHs, namely routing header (type 43) and Destination Options (type 60).

See also:
https://insinuator.net/2015/11/some-notes-on-the-drop-ipv6-fragments-vs-this-will-break-dnssec-debate/

ICMPv6 Types 1–4

- All of these are diagnostic/error messages and hence considered vital for the proper functioning of network communications (in particular type 2 [PTB]).
- Not many (publicly known) security issues with/of these packets.

- Recommendation: allow ("don't touch") them.



Ping

- Except for very specific circumstances (tenant isolation in cloud environments comes to mind) you'll want to allow inbound Ping (Echo Request – ICMPv6 type 128) to a system.
- The operational benefits of Ping are far greater than the real [usually even: perceived] negative security impact.

- Recommendation: allow.

Router Advertisements

- From an overall architecture perspective RAs are/can be considered the most important IPv6 packets at all.

- Recommendation: allow.
- In “fully static configuration“ scenario one might deny/block them, but should do so only after diligent testing.

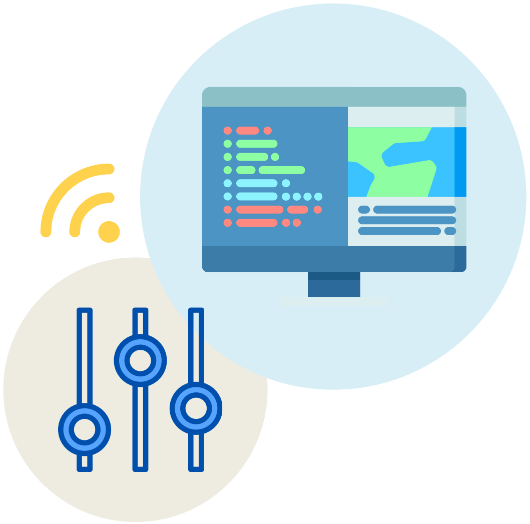
Neighbor Solicitations & Advertisements

- In most cases blocking NS/NA packets (on an Ethernet link at least) will break something.

- Recommendation: allow.
- In case you're concerned about NDP spoofing attacks a local packet filter would be the wrong control anyway.

ICMPv6 Redirects

- Since many years there have been security discussions around ICMP(v6) redirect messages (ICMPv6 type 137).
- Those are packets with a fully valid purpose and maybe even needed in some cases.
- They can easily be abused for malicious purposes (traffic redirection).



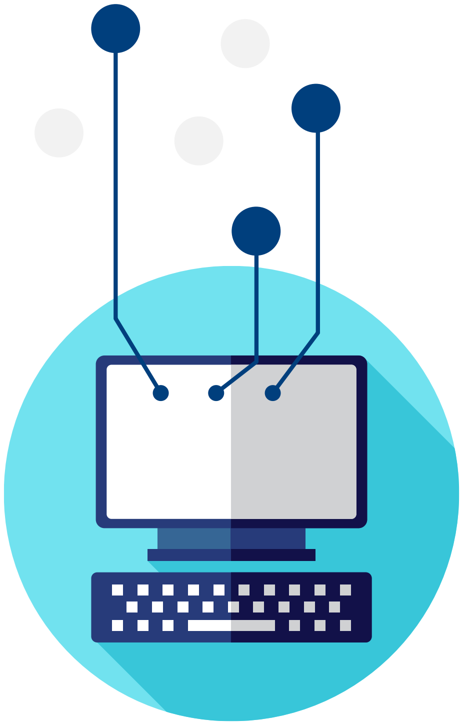
ICMPv6 Redirects Recommendation

- No action needed in a white-list rule set.
- If really really needed, allow them (ICMPv6 type 137).
- Probably a good idea to block them (from an operational impact vs. associated security risk ratio perspective).

ERNW
providing security.

MLD

- As long as no inter-subnet multicast communication is actually needed/in place you probably won’t need MLD.
- This can be expected for the vast majority of networks where the type of filtering we discuss here is applied at all.



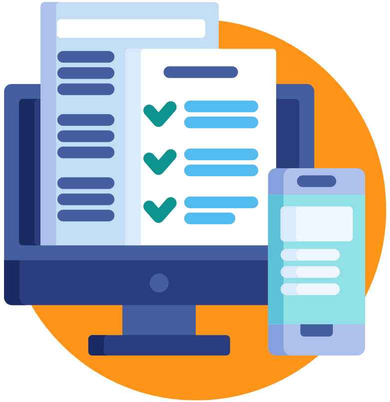
See also:
https://insinuator.net/2014/09/mld-and-neighbor-discovery-are-they-related/

MLD Recommendation

- No action needed in a white-list rule set.
- If really needed, allow ICMPv6 types 130–132 and maybe 143 (depending on MLD versions in use).
- You can subsequently block MLD (as opposed to entirely disabling it which on Windows breaks ND, but not on Linux).

DHCPv6

- In case DHCPv6 is involved in parameter provisioning to the systems in question you'll need (to allow) it.
- In all other scenarios it won't be needed.
- From a host/server perspective, inbound UDP 546 is needed.
  - Probably the client port of server-side packets is not always deterministic → do not include a source port in the rule.
- Disabling a local DHCPv6 client might yield unintended results on Windows systems.
  - Depending on the method chosen for the task so blocking those packets might be the best way of getting rid of DHCPv6 interactions.



See also: https://insinuator.net/2017/01/ipv6-properties-of-windows-server-2016-windows-10/

DHCPv6 Recommendation

- No action needed in a white-list rule set.
- Explicitly allow inbound UDP 546 once a system needs to receive DHCPv6 messages.

Hardening

- This encompasses all steps applied to the (IPv6 stack) of the local host.
- tl;dr: there’s not much to do in this space.

See also:
https://www.troopers.de/media/filer_public/ff/9b/ff9b181d-a2f5-4444-9481-73384950094f/ernw_tr16_ipv6secsummit_protectinghosts_final.pdf

For Reference

- ERNW's IPv6 Hardening Guides, developed by Antonios Atlasis
- Linux [Hard_Linux]
    - https://www.ernw.de/download/ERNW_Guide_to_Securely_Configure_Linux_Servers_For_IPv6_v1_0.pdf
- Windows [Hard_Windows]
    - https://www.ernw.de/download/ERNW_Guide_to_Configure_Securely_Windows_Servers_For_IPv6_v1_0.pdf
- OS X [Hard_OSX]
    - https://www.ernw.de/download/ERNW_Hardening_IPv6_MacOS-X_v1_0.pdf

Host Level Perspective

- Main (additional) protection strategies
- "Minimal machine" approach
  - Remove un-needed (IPv6) functionality (not the full IPv6 stack!), e.g. MLD.
- Static configuration of IPv6 parameters
  - Keep operational effort & concept of "deviation from default" in mind.
- Tweaking of IPv6-parameters/ behavior
  - ND parameters, MLD, RFC 6980 et al.
- Local packet filtering
  - See above. Keep operations in mind.

Minimal Machine

- Main potential measures
- On Linux systems MLD can be disabled (or just not be enabled?).
- On Windows systems disabling MLD (via `netsh` command) creates a state where *Neighbor Discovery* does not work correctly anymore
  → not recommended.
- If systems are provisioned with static IPv6 addresses, DHCPv6 should be disabled as a service (Windows and Linux).
  - Maybe do the same in SLAAC-only networks?
  - In general might/have to be done per address family.
- On systems with static IPv6 addresses, the processing of router advertisements can be disabled. We already discussed this ;-)
  - [Hard_Linux], Sect. 5.2 or [Hard_Windows], Sect. 5.4.

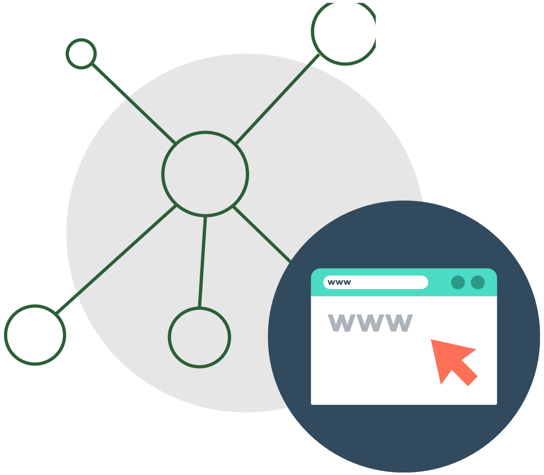
See also
https://www.insinuator.net/2014/11/mld-considered-harmful/

https://www.insinuator.net/2014/09/mld-and-neighbor-discovery-are-they-related/.

Layer 2

IPv6 Security on the Local Link / L2 Networks

- In many environments this is the most discussed area.



IPv6 Sec on the Local Link / Quick Recap

- By design, all systems are considered to be trustworthy
  - Main exchanges are not authenticated, integrity-controlled or the like
- Specific messages can heavily influence the behavior of other nodes on the link.
- There‘s a variety of messages which bring their own complexity.
  - What happens exactly might depend on the OSs present on the link.

- In short: it‘s a mess

ERNW
providing security.

Quick Overview of Mitigation Approaches

- *First Hop Security* (FHS) features of switches
  - Very limited availability in virtual environments
  - Can often be circumvented via EHs
  - → Basic network hygiene but not bulletproof
- ACLs (usually port-based)
  - In general better security stance than FHS, but different ops implications
- Don't use ND at all (L3-only with / 64s for servers)
  - Can usually only be done in IPv6-only networks

In Case You Want to Do Your Own Testing

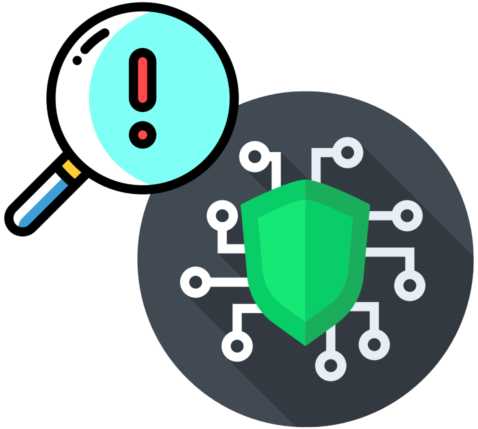
- The main IPv6 specific (attack) toolkits are
  - Antonios Atlasis‘ Chiron
  - Marc Heuse’s THC-IPV6
  - Fernando Gont‘s IPv6 Toolkit
  - Scapy (whose IPv6 capabilities are mainly maintained by Guillaume Valadon)
- Each has specific strenghts & limits.
- We usually prefer to use Chiron because of the powerful options in the space of extension Headers and fragmentation.

First Hop Security / Overview

- Collective name, initially coined by Cisco (?), for a number of IPv6 security features which are implemented on switches.
- Exists in several “generations” since 2011
  - 1st gen: mainly RA Guard (RFC 6105)
    - Basic network hygiene as of 2019
  - 2nd/3rd gen: more complex features
    - We don’t know any org using this stuff
- Several (all?) implementations can be evaded
  - Inherent conflict between flexibility & speed (ASICs)

Attacks / Security Issues on LL

- Rogue Router Advertisements
  - By accident
  - As attack, in order to redirect/blackhole traffic
- Neighbor Spoofing
  - Similar ARP spoofing in IPv4 networks
  - Why would one want to do that?
- All types of DoS scenarios
  - Somewhat classic against RA & ND.
  - Potentially also quite a few possible via MLD.



See also:
https://www.troopers.de/media/filer_public/7c/35/7c35967a-d0d4-46fb-8a3b-4c16df37ce59/troopers15_ipv6secsummit_atlasis_rey_salazar_mld_considered_harmful_final.pdf

But Can‘t We just Filter the Bad Stuff?

There‘s RA Guard et al., right?

- Hmm… like most other *blacklist-based* security features RA Guard can be circumvented.
  - There‘s no (easy) cure for this. Choose two out of (function|speed|cost).
- Hey, we have RFC 6980 for this.
  - We for ones consider this one of the most important IPv6 RFCs from the last years.
  - But it seems not easy to implement…
    - Which in turn might not be surprising.

RFC 6980

Internet Engineering Task Force (IETF) — F. Gont
Request for Comments: 6980 — SI6 Networks / UTN-FRH
Updates: 3971, 4861 — August 2013
Category: Standards Track
ISSN: 2070-1721

Security Implications of IPv6 Fragmentation with IPv6 Neighbor Discovery

Abstract

This document analyzes the security implications of employing IPv6 fragmentation with Neighbor Discovery (ND) messages. It updates RFC 4861 such that use of the IPv6 Fragmentation Header is forbidden in all Neighbor Discovery messages, thus allowing for simple and effective countermeasures for Neighbor Discovery attacks. Finally, it discusses the security implications of using IPv6 fragmentation with SEcure Neighbor Discovery (SEND) and formally updates RFC 3971 to provide advice regarding how the aforementioned security implications can be mitigated.

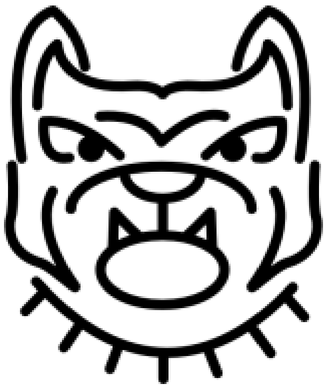
RA Spoofing, thc-ipv6

- `fake_router26` *`interface`*
  - Announce (only) new router with attacker's link-local address

- Main options
  - `-A` *`network/prefix`*
  - `-s source_ip`
  - `-l router_lifetime` ("0" → delete [legitimate] router, with -s)
  - `-E evasion_type` (see below)
  - `-m mac_address` (send to specific destination MAC address; not sure about real benefit of this one, as sent to ff02::1 → everybody sees it)

thc-ipv6 / Evasion of RA Guard

- `fake_router26` has some predefined evasion options
  - `-E D` is “the classic one“
- None of them reliably work against current implementations of *RA Guard*!
  - → not really useful nowadays.
- Marc added “F” option implementing evasion after our blogposts on RFC 6980 testing.



RA Spoofing / Chiron

- Basic variant:
  - `chiron_local_link.py eth0 -ra –s` *`spoofed_address`*
  - -s is pretty much always needed.
    Common use: attacker's ll address
- Main options
  - `-pr` *`prefix`*
  - *`-rl router_lifetime`* (e.g. "0" ;-)

Source: Chiron, Colum, Padraic,
1881-1972 Pogány, Willy, 1882-1955, ill

Chiron / RA Guard Evasion

- Chiron has extensive capabilities with regard to extension headers and fragmentation, for all modules.

- Main approaches:
  - Fragmentation (only) – usually not too helpful
  - Extension headers (only) – usually not too helpful
  - Fragmentation + ext_hdrs in unfragmentable part – might work
  - Fragmentation + ext_hdrs in fragmentable part – usually best results
  - Number & type of ext_hdrs might play a role, too.

- Be creative ;-)
  - E.g. https://insinuator.net/2015/01/dhcpv6-guard-do-it-like-ra-guard-evasion/

Fun with Chiron (II)

- Baseline
  - `chiron_local_link.py eth0 -ra -s fe80::2`
- Fragment + add ext_hdr to unfragmentable part (1st frag)
  - `chiron_local_link.py eth0 -ra -s fe80::2 -nf 2 -luE 60`
  - `chiron_local_link.py eth0 -ra -s fe80::2 -nf 2 -luE 43`
  - `chiron_local_link.py eth0 -ra -s fe80::2 -nf 4 -luE 60`
- Fragment + add ext_hdr(s) to fragmentable part (consecutive frag.)
  - `chiron_local_link.py eth0 -ra -s fe80::2 -nf 2 -lfE 60`
  - `chiron_local_link.py eth0 -ra -s fe80::2 -nf 2 -lfE 43`
  - `chiron_local_link.py eth0 -ra -s fe80::2 -nf 2 -lfE 60,43`
  - `chiron_local_link.py eth0 -ra -s fe80::2 -nf 4 -lfE 60`

Now this Slide Makes More Sense ;-)

Test Case No.	Description	Chiron Options Used (in addition to baseline cmd)	Impact on Target OS' IPv6 Config (without RA Guard)	What was observed in Wireshark on Target OS? (without RA Guard)	What still got through with RA Guard enabled?	Overall Result With RA Guard Enabled
13	Two fragments, with two DestOptions in fragmentable part	-lfE 60,60 -nf 2	Added 2nd default gw, created additional address	One fragment plus RA packet which contains two DestOptions EHs	1st fragment, but *not* the RA	No impact
14	Four fragments, with two DestOptions in fragmentable part	-lfE 60,60 -nf 4	Added 2nd default gw, created additional address	Three fragments plus RA packet which contains two DestOptions	Three fragments, plus RA containing two DestOptions EHs. Nothing logged on the switch.	Successful attack
15	Two fragments, with two RoutingHdr EHs in fragmentable part	-lfE 43,43 -nf 2	Added 2nd default gw, created additional address	One fragment plus RA packet which contains two RoutingHdr EHs	Two fragments, plus RA containing EHs. "traceback" on switch console when running 15.0(2)SE2	Successful attack when switch runs 15.0(2)SE2, no impact when switch runs 15.0(2)SE10a
16	Two fragments, with two RHs and two DestOptions, in mixed order	-lfE 60,43,60,43 -nf 2	Added 2nd default gw, created additional address	One fragment plus RA packet which contains the four EHs	1st fragment, but *not* RA	No impact
17	Same as 16 but four fragments	-lfE 60,43,60,43 -nf 4	none	1st three segments only, but not RA	1st three fragments, but not RA	No impact
18	Same as 16 but three fragments	-lfE 60,43,60,43 -nf 3	Added 2nd default gw, created additional address	Two fragments, then RA containing all EHs	1st two fragments plus RA	Successful attack

ACL-based Approach / Sample

```
deny icmp any any router-advertisement
deny ipv6 any host FF02::1 fragments
deny ipv6 any host FF02::C fragments
deny ipv6 any host FF02::FB fragment
deny ipv6 any host FF02::1:3 fragments
deny ipv6 any FF02::1:FF00:0/104 fragments
deny ipv6 any FE80::/64 fragments
permit ipv6 any any
```

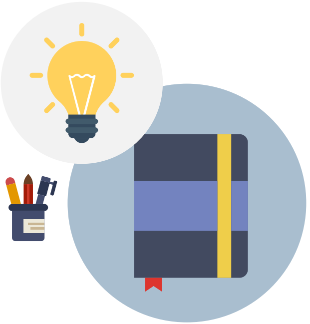
See also:
https://static.ernw.de/whitepaper/ERNW_Whitepaper62_RA_Guard_Evasion_Revisited_v1.0.signed.pdf

IPv6 L2 Is a Mess Unfortunately

- One course of action to avoid all the problems on the local link is:
- Provide each server a dedicated /64
  - The only neighbor each server has is the default gateway
- Could be realised with a routed port on the ToR switch.
  - Scalability should not be an issue for the “typical” enterprise DC.
- Unfortunately, this can not be reasonably done in a dual-stack implementation.

Summary/Checklist of Recommendations

- Reflect on the security controls in your org
  - Which ones to {keep,adapt}.
  - Consider state.
- Traffic filtering
  - Will need some slight modifications (EHs et al.)
  - Think about conversion approach.
- Layer 2
  - Define risk appetite & strategy (e.g. FHS vs. ACLs)
  - RA Guard = basic network hygiene, everywhere



Conclusions

- IPv6 is different than IPv4
  - Namely in enterprise organizations this can have some security implications.
- As so often *operational feasibility* should be strongly considered ;-)

- Enjoy #RIPE78

There's never enough time...

THANK YOU... ...for yours!

@Enno_Insinuator

erey@ernw.de

ernw.de

insinuator.net

Slides available soon.

Sources

As indicated on slides.

Image Sources

Icons made by Freepik
from www.flaticon.com
https://unsplash.com
https://www.pexels.com/