TROOPERS

ERNW
providing security.

Protecting Hosts in IPv6 Networks

A discussion of security controls on the host level

Enno Rey
erey@ernw.de

Who Am I

- Founder (2001) and head of ERNW, a company providing vendor-independent security assessment & consulting services.

ERNW RESEARCH
pursuing knowledge.

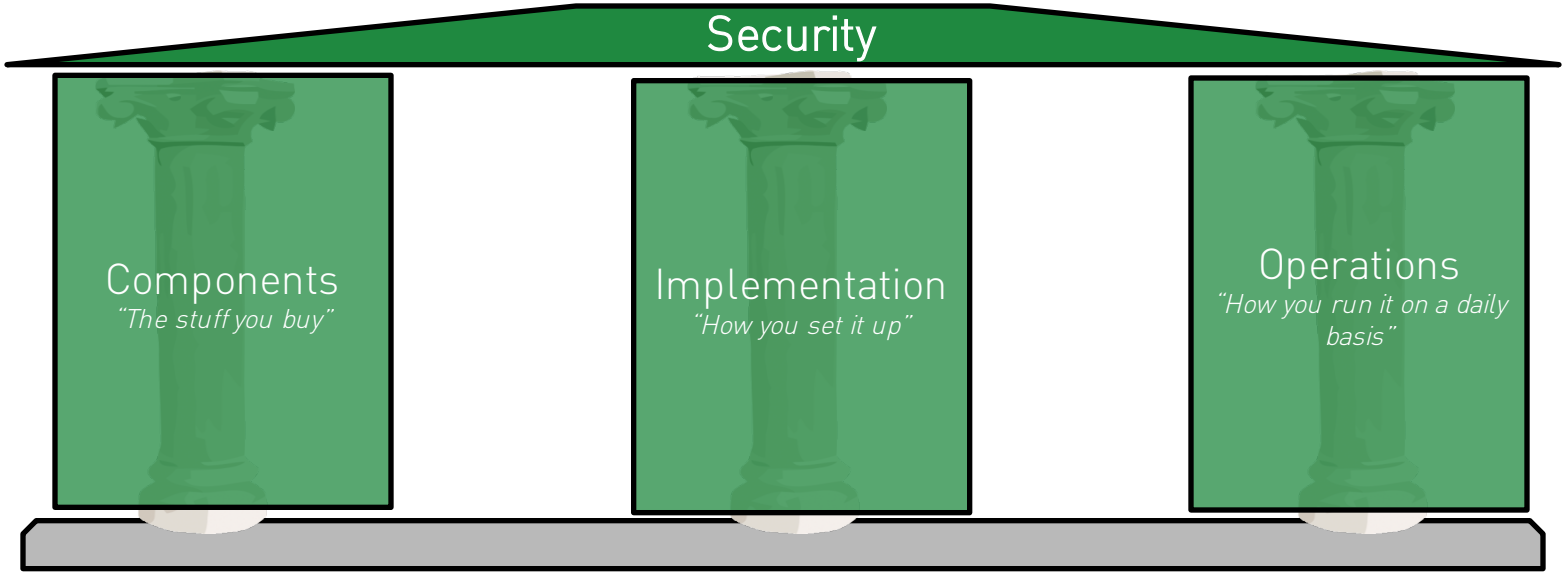
ERNW INSIGHT
expanding your knowledge.

- Old-school network guy involved with IPv6 since 1999.

Agenda

- Some preliminary remarks on the operations perspective
- Protection of IPv6 is a matter of network infrastructure (controls), mostly.
- Discussion of specific controls

Keep in Mind...



Please identify the most important pillar!

So, when thinking about security controls…

- Two essential factors must be evaluated:
  - *Security benefit*
    - "How much do we gain, security-wise?"
    - "What's the risk reduction of this control?"
  - Operational feasibility
    - "What's the **operational** effort to do it?"
    - Pls note: *opex*, not *capex*, counts!





- For some more discussion on these see also:
  - http://www.insinuator.net/2011/05/evaluating-operational-feasibility/
  - http://www.insinuator.net/2010/12/security-benefit-operational-impact-or-the-illusion-of-infinite-resources/

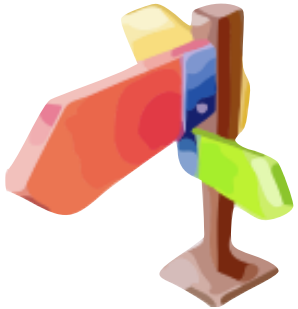
Evaluating *operational effort*

- For each potential control the following points should be taken into account
  - How many lines of code/configuration does it need?
    - Can it be implemented by means of templates or scripts? Effort needed for this?
  - To what degree does the implementation differ in different scenarios?
    - Per system/subnet/site?
    - Can "the difference" be scripted?
      - Taken from another source (e.g. central database)
      - "Calculated" (e.g. neighboring routers on local link)

  - How much additional configuration is needed for previous functionality?
    - E.g. to pass legitimate traffic in case of ("new") application of ACLs?
  - "Business impact" incl. number of associated support/helpdesk calls.
  - Cost for deployment of additional hardware/licenses.
    - Cost for their initial procurement is *CAPEX* (=> *not* relevant here).

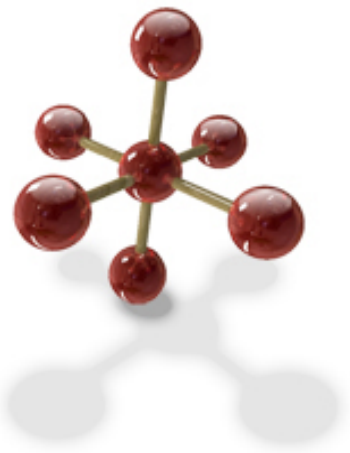
The Concept of "Deviation from Default"

- By this term we designate any deviation from a default setting of any IT system which happens by means of some configuration step(s).
  - Change some parameter from "red" to "black" or 0 to 1 or …
- *Deviation from default* always requires OPEX.
  - In particular if to be maintained through affected systems' lifecycle.
  - Even more so if affected system base is heterogeneous.
  - By its very nature, OPEX is limited. You knew that, right? ;-)
- *Deviation from default* doesn't scale.
  - $SEGMENT might have 20 systems today. And tomorrow?
- *Deviation from default* adds complexity.
  - In particular if it's "just some small modifications" combined…
    - Remember RFC 3439's *Coupling Principle*?



IPv6 Security Controls on the Host Level

Protection of IPv6 is a Matter of Network Infrastructure (controls), Mostly.



- In the following we assume that some (IPv6 specific) security controls have already been applied.
- This leaves two main questions
  - What's the residual risk from a host's perspective?
  - How to address that?

Evaluation of IPv6 Risks *After* NW Layer Controls

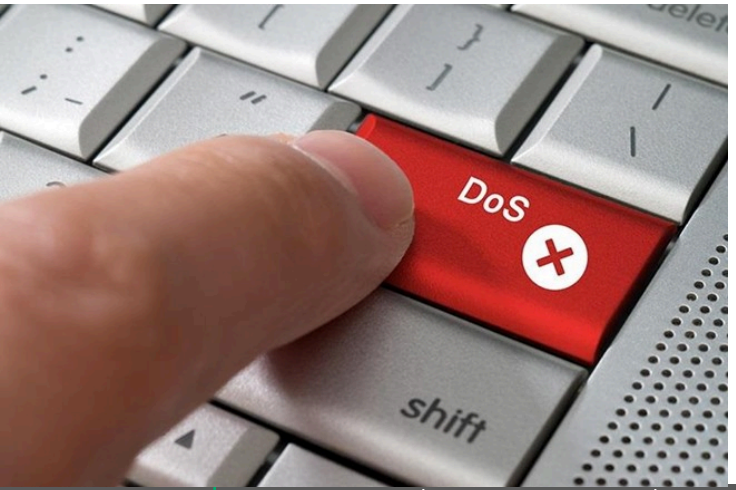
From a case study

Class	Specific Threat	Risk Rating in IPv4 Networks	Risk Delta via IPv6, after Implementation of Controls	Comments
Traffic Redirection	ARP/NA Spoofing	high risk	equal risk	no dedicated infrastructure controls planned
Traffic Redirection	DNS Spoofing	medium risk	equal risk	no dedicated infrastructure controls planned
Traffic Redirection	Spoofing of Default GW through DHCP	high risk	significantly lower risk	no dedicated infrastructure controls planned as attack no longer possible
Traffic Redirection	Route Injection	medium risk	equal risk	addressed partially by "isolation on routing layer" approach
Traffic Redirection	Attacks against FHRP	medium risk	equal risk	no dedicated infrastructure controls planned
Traffic Redirection	Rogue RAs	high risk	slightly increased risk	RA Guard & DHCPv6 Guard, risk expected to decrease over time due to RFC 6980
Attacks against Provisioning	Modification of DNS resolver through DHCP	high risk	equal risk	DHCPv6 Guard
Denial-of-Service	Resource Depletion	medium risk	equal risk	addressed partially by "isolation on routing layer" approach and by "IPv6 specific filtering"
Denial-of-Service	Flooding of Helper Protocols	low risk	slightly increased risk	addressed (only) partially by RA Guard and DHCPv6 Guard
Denial-of-Service	Traffic blackholing	high risk	equal risk	RA Guard & DHCPv6 Guard
Unauthorized Access over Network	Capability to establish undesired connections	medium risk	slightly increased risk	addressed by "isolation on routing layer" approach and by "IPv6 specific filtering"

For initial table (without controls) see:
https://www.ernw.de/download/ERNW_TR16_IPv6SecSummit_Enterprise_Security_Strategy.pdf

Host Level Perspective

Main residual risks (sample/case study)



- Denial-of-Service originating from the *local-link*.
  - Increased exposure wrt malformed pkts.
  - Flooding of helper protocols.
- Unauthorized access
  - Less isolation/separation of address space assumed.
  - Less protection from security controls on the network infrastructure level.

For Reference

ERNW's IPv6 Hardening Guides,
developed together with 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 config. 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
  - Be cautious & keep operations in mind.

Minimal Machine

Main potential measures

See also
https://www.insinuator.net/2014/11/mld-considered-harmful/
https://www.insinuator.net/2014/09/mld-and-neighbor-discovery-are-they-related/.

- 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?
- On systems with static IPv6 addresses, the processing of router advertisements can be disabled
  - [Hard_Linux], Sect. 5.2 or [Hard_Windows], Sect. 5.4.

Static Configuration

Main measures

- Usually this encompasses
  - IP address(es)
  - Default gateway(s)
  - DNS resolver(s)
  - NTP server(s)
- BUT: to work properly/as expected all dynamic mechanisms have to be disabled also.

Disable Dynamic Stuff

This might include

- Disable local processing of RAs
- Disable local processing of ICMPv6 type 137 (*redirects*).
- Disable DHCP(v6) service

Suppress RA Processing on Hosts

- Operationally expensive & severe *deviation from default*.
- Note: just assigning a static IP address might not suffice.
  - E.g. MS Windows systems can still generate additional addresses/interface identifiers.
- Still we know and – somewhat – understand that most of you have a strong affinity to this approach
  - Human (and in particular: sysadmin) nature wants to *control* things…

Overview for Different OS

- MS Windows

  - `netsh int ipv6 set int [index] routerdiscovery=disabled`

- FreeBSD

  - `sysctl net.inet6.ip6.accept_rtadv=0`
  - Do not run/invoke `rtsold`. (but the above prevents this anyway).

- Linux

  - Sth like: `echo 0 > /proc/sys/net/ipv6/conf/*/accept_ra`
  - See also IPv6 sect. of https://www.kernel.org/doc/Documentation/networking/ip-sysctl.txt

Disable IPCMPv6 137

- Linux
  - `net.ipv6.conf.default.accept_redirects = 0`
- Windows
  - `netsh interface ipv6 set global icmpredirects=disabled`

Tweaking Parameters

Main potential measures

- Use of MLDv2 only
  - E.g. see [Hard_Linux], Sect. 5.4.
- Enabling/configuration of a behavior that follows RFC 6980, if that is not default state of an OS (for example, it actually *is* the default for Linux).
-
- Additional measures as described in [Hard_Linux], Sect. 5.4

MLDv2 Only

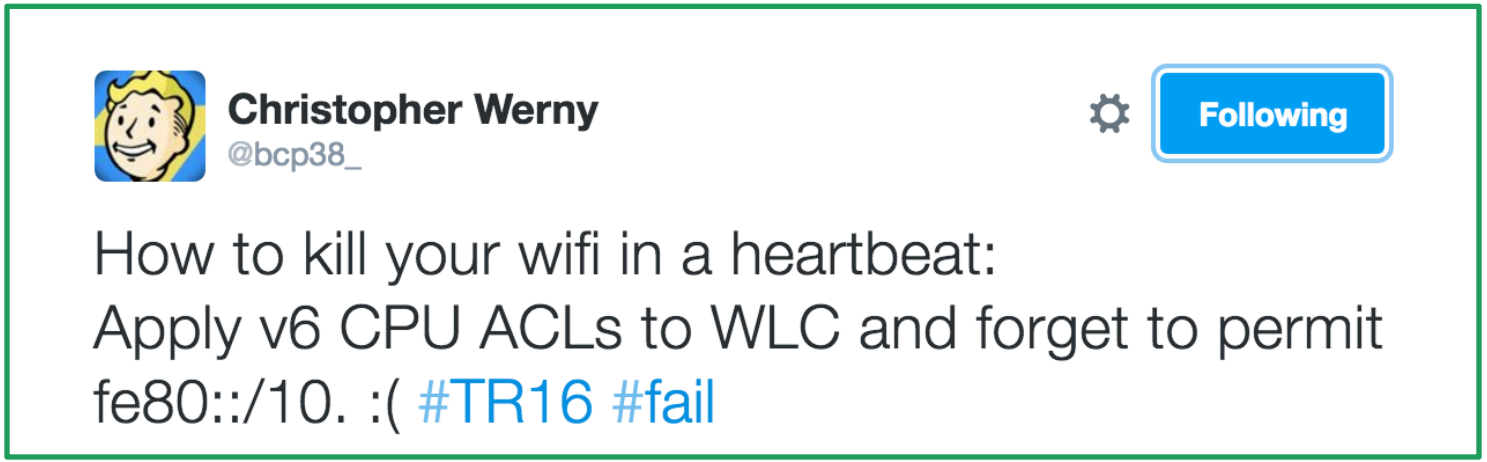
- Linux:
  - net.ipv6.conf.all.force_mld_version = 2

Local Packet Filtering

Some warning

- This should be an *ultima ratio* approach.
- Be very careful
  - Look at mailing list archives for people who shot themselves in the foot (e.g. by filtering ND/RA messages).

Case Study


Christopher Werny
@bcp38_
Following
How to kill your wifi in a heartbeat:
Apply v6 CPU ACLs to WLC and forget to permit fe80::/10. :(#TR16 #fail

Local Packet Filtering

- Sources
  - RFC 4890 *Recommendations for Filtering ICMPv6 Messages in Firewalls*
  - [Hard_Linux] & [Hard_Windows]
- Use $TECH available anyway on (or highly integrated with) $PLATFORM
  - BSD: pf/ipfw6
  - Linux: nftables/ip6tables
  - Windows: Windows Firewall

Conclusions & Summary

- Let me repeat this: IPv6 security SHOULD be addressed on the infrastructure level.
- There's some additional stuff which can be done on the host level.
  - Usually in segments with very high security requirements (and a low number of systems).
- Keep operational impact of these measures in mind!
  - Going with a "static" approach quickly becomes complicated & cumbersome...

There's never enough time...

THANK YOU...

...for yours!

@Enno_Insinuator

erey@ernw.de

Slides & further information:
https://www.troopers.de
https://www.insinuator.net
(..soon)

Questions?

Image Credits

- Icons made by Freepik from www.flaticon.com are licensed by CC 3.0 BY.