ERNW
providing security.

RIPE 74
BUDAPEST, HUNGARY
8 - 12 May 2017

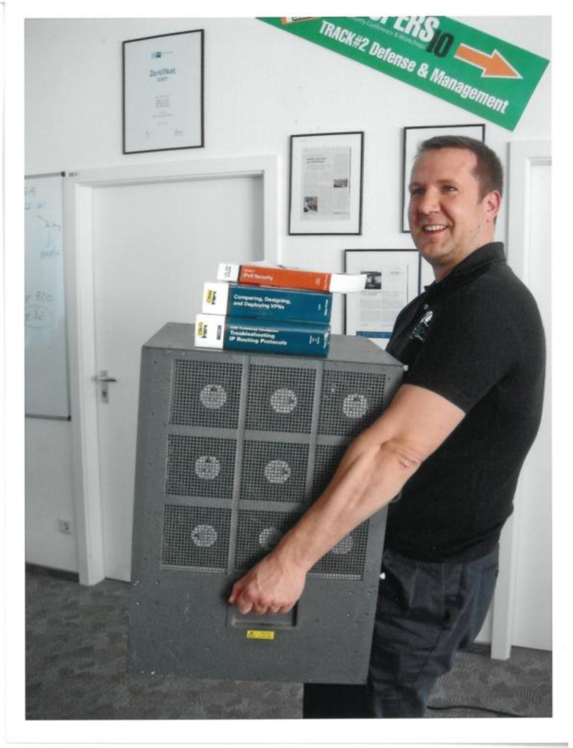
IPv6 Address Selection – A Look from the Lab

Omar Eissa
Jan Kwiotek
Enno Rey

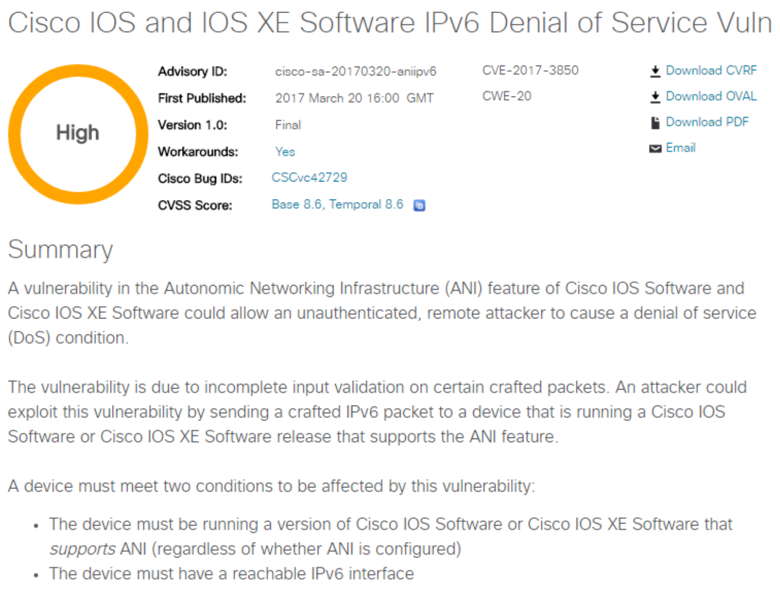
oeissa@ernw.de
erey@ernw.de
@Enno_Insinuator

#whoami

- Networking background, doing security as a full-time profession since 1997
- Taking care of LIR stuff at some enterprise LIRs
  - Including the one with probably the coolest org handle: ORG-HACK1-RIPE
- Blogging about IPv6 & other pieces at `https://insinuator.net/tag/ipv6/`



Other Stuff Omar Does



Cisco IOS and IOS XE Software IPv6 Denial of Service Vuln

High

Advisory ID:	cisco-sa-20170320-aniipv6	CVE-2017-3850	Download CVRF
First Published:	2017 March 20 16:00 GMT	CWE-20	Download OVAL
Version 1.0:	Final		Download PDF
Workarounds:	Yes		Email
Cisco Bug IDs:	CSCvc42729		
CVSS Score:	Base 8.6, Temporal 8.6		

Summary

A vulnerability in the Autonomic Networking Infrastructure (ANI) feature of Cisco IOS Software and Cisco IOS XE Software could allow an unauthenticated, remote attacker to cause a denial of service (DoS) condition.

The vulnerability is due to incomplete input validation on certain crafted packets. An attacker could exploit this vulnerability by sending a crafted IPv6 packet to a device that is running a Cisco IOS Software or Cisco IOS XE Software release that supports the ANI feature.

A device must meet two conditions to be affected by this vulnerability:

- The device must be running a version of Cisco IOS Software or Cisco IOS XE Software that *supports* ANI (regardless of whether ANI is configured)
- The device must have a reachable IPv6 interface

ERNW providing security.

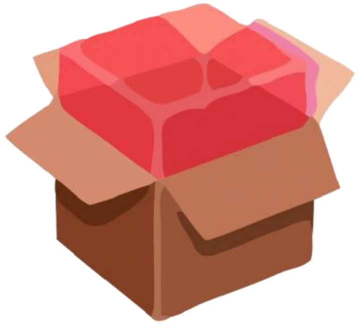
RIPE 74 BUDAPEST, HUNGARY 8 - 12 May 2017

Agenda

- Introduction
- What the specs say
- Lab Setup & Results
- Summary and Conclusions

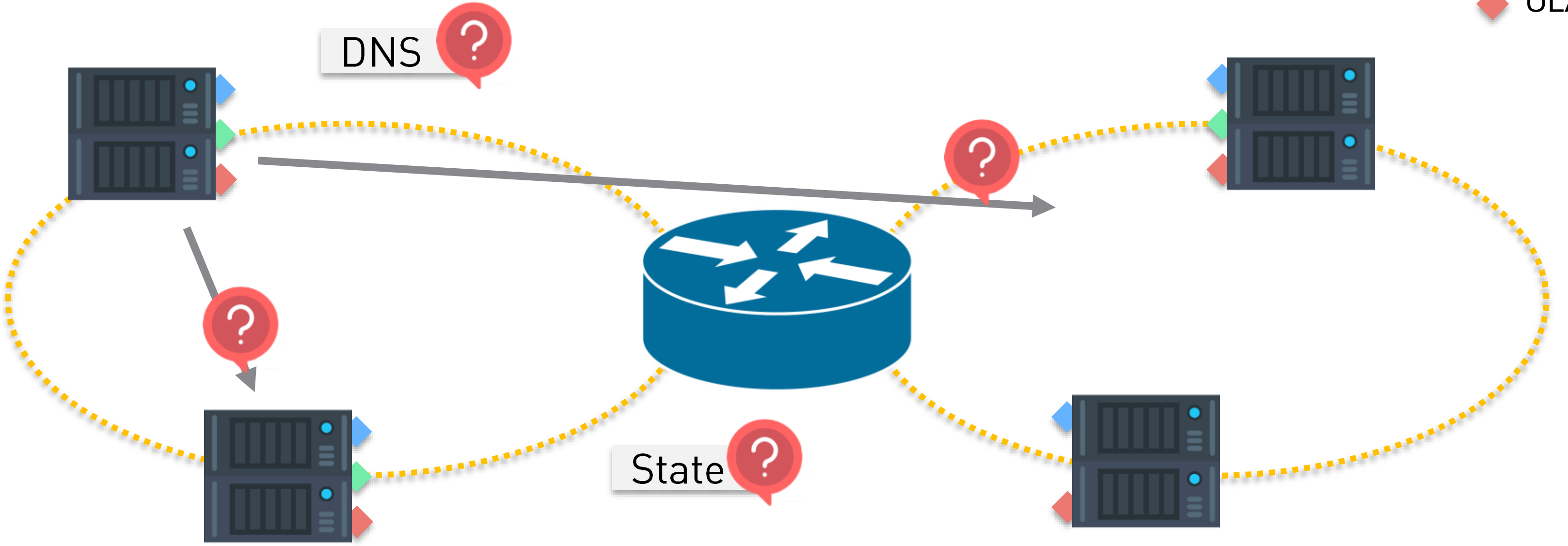
Intro

- IPv6's source address selection mechanism is currently defined in RFC6724 (which updates RFC 3484).
- We looked at the behavior wrt RFC6724 in different operating systems.
- Based on multiple scenarios with different network configurations.
- Lab setup included OSs at latest patch level as of Aug 2016.



The Problem to Be Solved

🔷 LLA
🟩 GUA
🟥 ULA

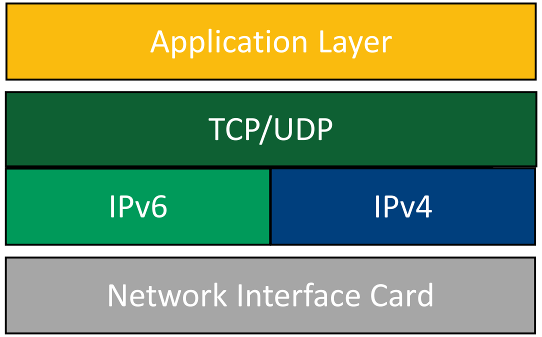


Keep in Mind

- Source Address Selection as of RFC 6724 is only one element in a chain of decisions an OS/stack has to take
  - IPv4 vs. IPv6 on a *dual stack* system
    - HE/RFC 6555 and/or NCSI (Windows)
  - Which information to extract from a DNS response with multiple records?
    - Change of default behavior between Windows versions, e.g. see https://support.microsoft.com/en-us/help/968920/windows-vista-and-windows-server-2008-dns-clients-do-not-honor-dns-round-robin-by-default

- Applications can override OS behavior
  - Many years ago tweaking of parameter needed for JAVA apps

```
-Djava.net.preferIPv6Addresses=true
```

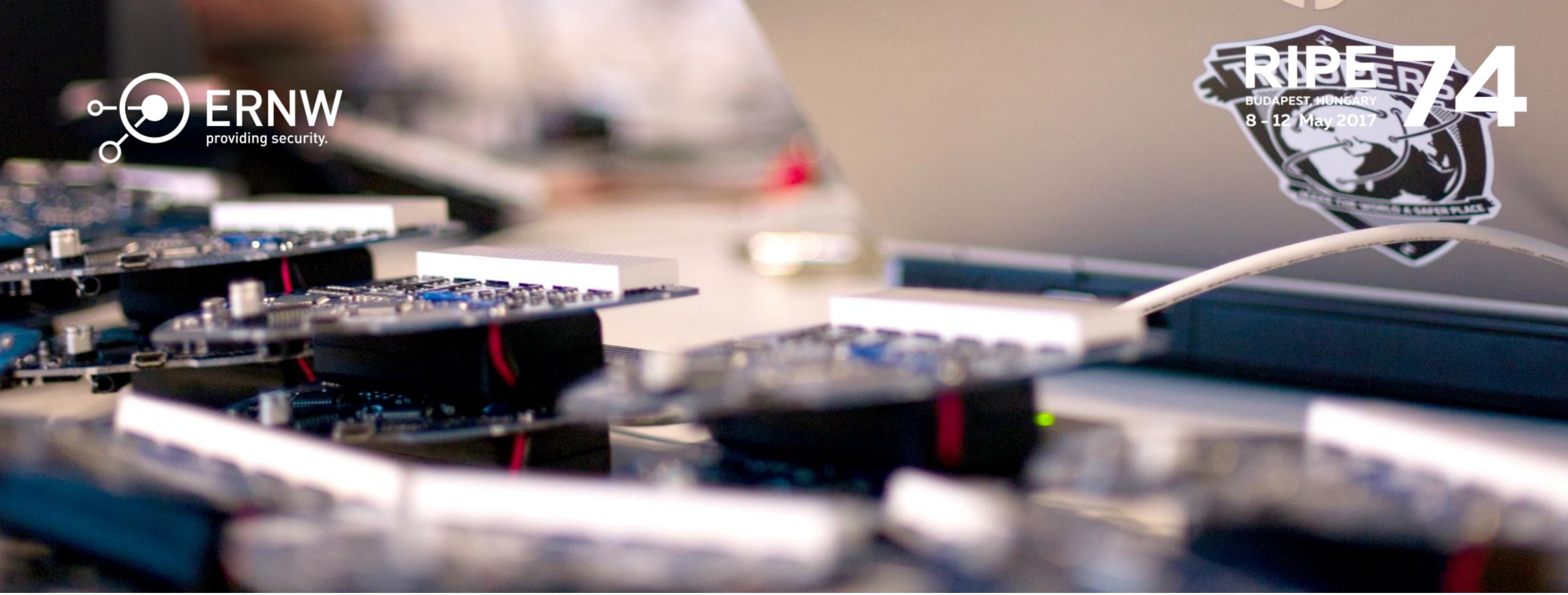


Scenarios We Looked at in the Lab

- *Source IPv6 address selection* functionality in default state/config in IPv6 protocol stack:

- Global Unicast Addresses (GUAs)
  - Configured in static manner
  - obtained via DHCPv6
  - or IPv6 Stateless Address Autoconfiguration (SLAAC)

- Unique Local IPv6 Unicast Addresses (ULAs)



What the Specs Say

Relevant details of the
IPv6 source address selection algorithm.

ERNW providing security.

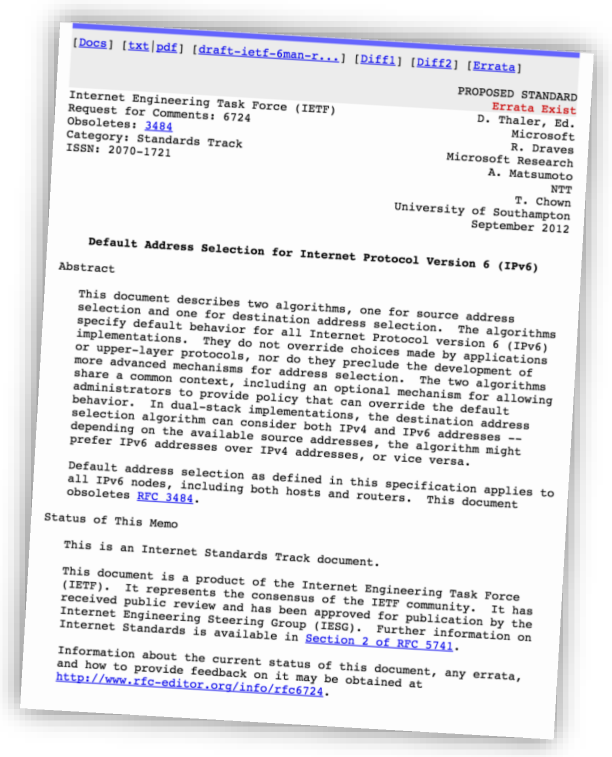
RIPE 74 BUDAPEST, HUNGARY 8 - 12 May 2017

Relevant Specs

- RF 3484, obsoleted by
- RFC 6724
  - Jen suggests a specific update in draft-linkova-6man-default-addr-selection-update
- RFC 7078 Distributing Address Selection Policy Using DHCPv6
  - Any implementations?

Revisiting RFC 6724

- Multiple ways how the source address can be selected in IPv6
  - Applications can always choose the source address and override operating system selection.
  - If the source address is not set, the source address selection algorithm of the OS comes into play.
    - This is what we looked at.



[Docs] [txt|pdf] [draft-ietf-6man-r...] [Diff1] [Diff2] [Errata]

PROPOSED STANDARD
Errata Exist

Internet Engineering Task Force (IETF)
Request for Comments: 6724
Obsoletes: 3484
Category: Standards Track
ISSN: 2070-1721

D. Thaler, Ed.
Microsoft
R. Draves
Microsoft Research
A. Matsumoto
NTT
T. Chown
University of Southampton
September 2012

Default Address Selection for Internet Protocol Version 6 (IPv6)

Abstract

This document describes two algorithms, one for source address selection and one for destination address selection. The algorithms specify default behavior for all Internet Protocol version 6 (IPv6) implementations. They do not override choices made by applications or upper-layer protocols, nor do they preclude the development of more advanced mechanisms for address selection. The two algorithms share a common context, including an optional mechanism for allowing administrators to provide policy that can override the default behavior. In dual-stack implementations, the destination address selection algorithm can consider both IPv4 and IPv6 addresses -- depending on the available source addresses, the algorithm might prefer IPv6 addresses over IPv4 addresses, or vice versa.

Default address selection as defined in this specification applies to all IPv6 nodes, including both hosts and routers. This document obsoletes RFC 3484.

Status of This Memo

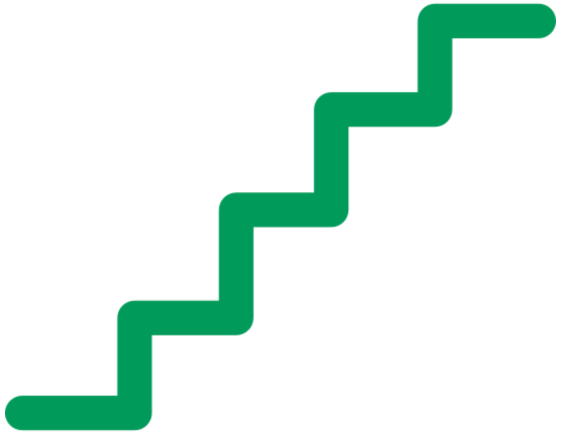
This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in Section 2 of RFC 5741.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at http://www.rfc-editor.org/info/rfc6724.

Revisiting RFC 6724 (II)

- Step 1: all available addresses are sorted into a list
    - Including IPv4 addresses
- Step 2: up to 9 different rules are applied that reorder the sequence of the list
- Step 3: this algorithm compares two elements (A,B) of the list and put either A or B on top
- Step 4: when algorithm terminates, the topmost element is returned and used as source address for an outgoing packet

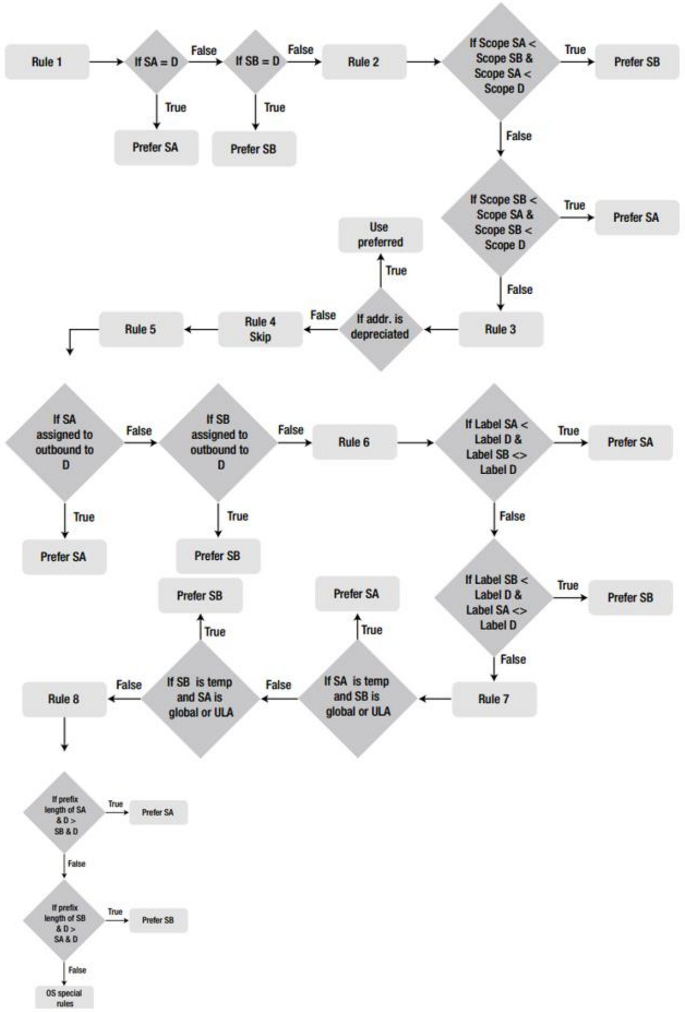


The Rules

according to RFC 6724:

- Rule 1: Prefer same address.
- Rule 2: Prefer appropriate scope.
- Rule 3: Avoid deprecated addresses
- Rule 4: Prefer home addresses.
- Rule 5: Prefer outgoing interface.
- Rule 5.5: Prefer addresses in a prefix advertised by the next-hop
- Rule 6: Prefer matching label.
- Rule 7: Prefer temporary addresses.
- Rule 8: Use longest matching prefix.

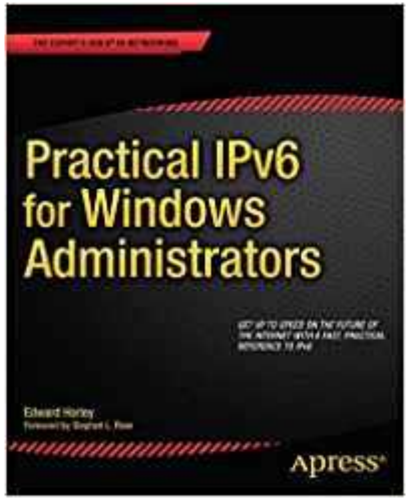
ERNW providing security.



Rule 1
If SA = D
False
True
Prefer SA
If SB = D
False
True
Prefer SB
Rule 2
If Scope SA < Scope SB & Scope SA < Scope D
True
Prefer SB
False
If Scope SB < Scope SA & Scope SB < Scope D
True
Prefer SA
False
Rule 3
If addr. is depreciated
True
Use preferred
False
Rule 4 Skip
Rule 5
If SA assigned to outbound to D
True
Prefer SA
False
If SB assigned to outbound to D
True
Prefer SB
False
Rule 6
If Label SA < Label D & Label SB <> Label D
True
Prefer SA
False
If Label SB < Label D & Label SA <> Label D
True
Prefer SB
False
Rule 7
If SA is temp and SB is global or ULA
True
Prefer SA
False
If SB is temp and SA is global or ULA
True
Prefer SB
False
Rule 8
If prefix length of SA & D > SB & D
True
Prefer SA
False
If prefix length of SB & D > SA & D
True
Prefer SB
False
OS special rules

RIPE 74
BUDAPEST, HUNGARY
8 - 12 May 2017

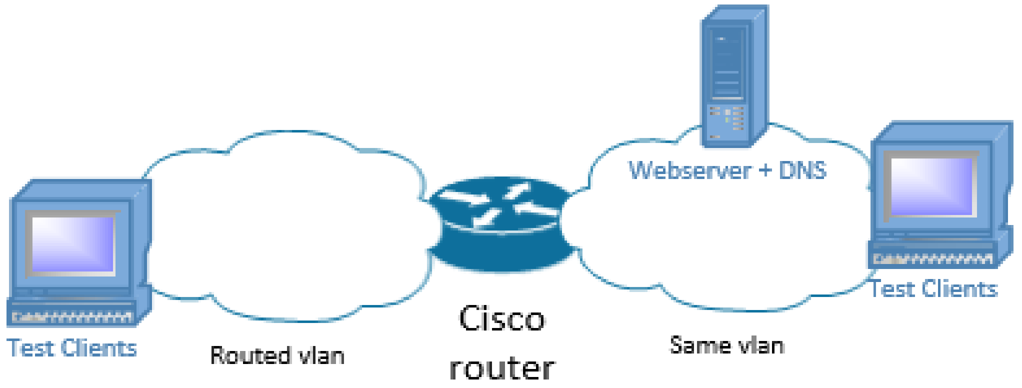
Kudos to Ed Horley for permission to use this here!



Lab Setup

Scenarios and test operating systems

Lab Setup



- One webserver
- + clients that connect to the server, which are connected either:
  - in the same subnet
  - or a routed subnet
- IPv6 parameters are distributed via Router Advertisements to the clients which perform SLAAC.
- Unix and Linux based systems can configure the DNS settings via a Router Advertisement Option for DNS.
- For the Windows based operating systems additionally the o-flag is set with a stateless DHCP server only for distributing the DNS server address.

Details on Setup

- Main part of the experiment is the Cisco router that provides the Router Advertisements to network and gets reconfigured for each test.
- The webserver serves a static http web page, which shows the actual IPv6 address of the client.
- This server answers also the DNS requests from the clients and gets configured as DNS server on them via Router Advertisements or a stateless DHCP.
- The mobile devices are connected via WiFi over an Access Point that is bridged into the actual network segment of the tests, which has enabled no services or IP addresses that could interfere with our setup.

Basic Settings

The following settings on the router are not changed during the tests:

```
ipv6 nd reachable-time 500
ipv6 nd other-config-flag
ipv6 nd ra lifetime 60
ipv6 nd ra interval 10
ipv6 nd ra dns server 2001:DB8:0:1::2
ipv6 dhcp server test rapid-commit
```

Selection of OS

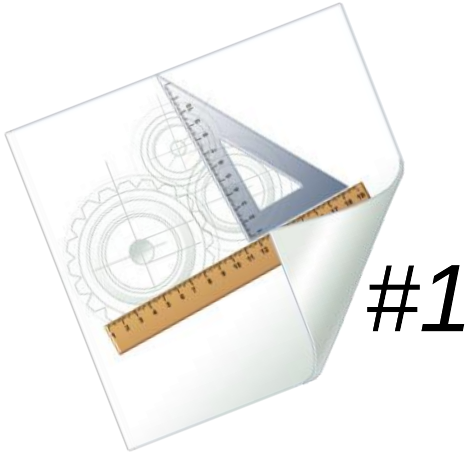
and their patch levels

Operating System	Version
Windows 7 Pro / VM	SP1, update 27.06.2016
Windows 8.1 Pro / VM	update 27.06.2016
Windows 10 Pro / VM	update 27.06.2016
Ubuntu 16.4 LTS / VM	Kernel 4.4.0-21-generic
Debian 8 / VM	Kernel 3.16.0-4-amd64
Iphone 6s	IOS 9.3.2
Windows 10 Mobile / Lumia 550	update 25.06.2016

Test Scenario #1

- The clients are on the **same subnet** as the webserver and get only a link-local IPv6 address, the GUA prefix is not advertised via Router Advertisements but configured on the webserver as well as on the router.

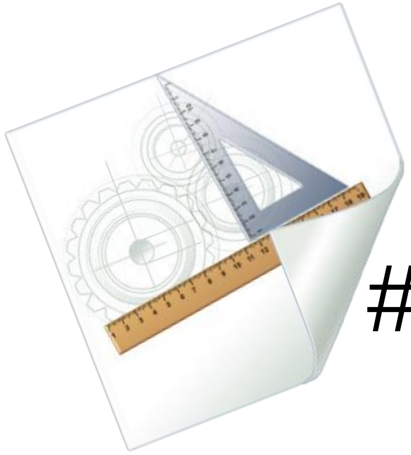
- Relevant router configuration of the client subnet:
  - ipv6 address FE80::1 link-local
  - ipv6 address 2001:DB8:0:1::1/64
  - 2001:db8:0:1::/64 no-advertise



Test Scenario #2

- The clients are on the **same subnet** as the webserver and get a GUA IPv6 address which is advertised via Router.

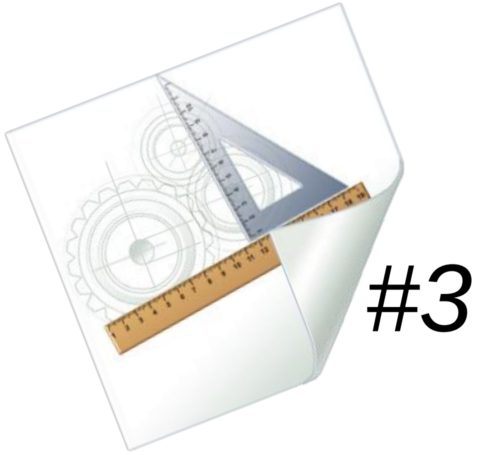
- Relevant router configuration of the client subnet:
  - ipv6 address 2001:DB8:0:1::1/64
  - ipv6 nd prefix 2001:db8:0:1::/64



#2

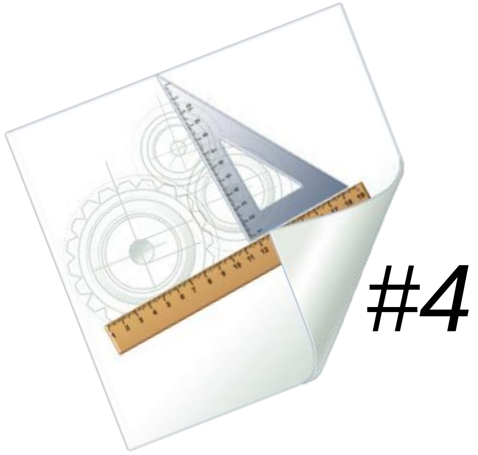
Test Scenario #3

- The clients are on the **same subnet** as the webserver and get a GUA and a ULA address via Router Advertisements. The webserver is still statically configured to the GUA address and the DNS serves only the GUA address of the webserver.
- Relevant router configuration of the client subnet:
  - ipv6 address 2001:DB8:0:1::1/64
  - ipv6 address FC00:DEAD:BEEF:CAFE::1/64
  - ipv6 nd prefix 2001:DB8:0:1::/64
  - ipv6 nd prefix FC00:DEAD:BEEF:CAFE::/64



Test Scenario #4

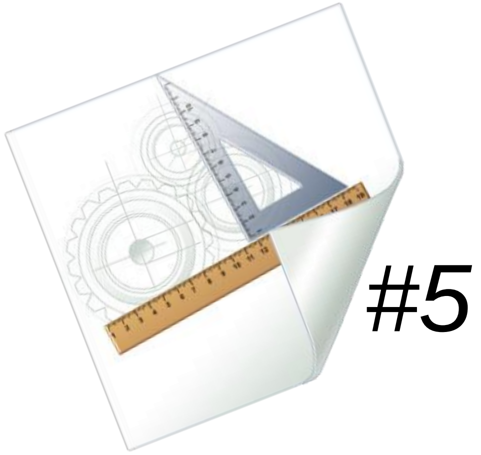
- The clients are on the **same subnet** as the webserver and get a GUA address via Router Advertisements, while the ULA prefix is not advertised but present on the router. The webserver is still statically configured to the GUA address and the DNS serves only the GUA address of the webserver.
- Relevant router configuration of the client subnet:
  - ipv6 address 2001:DB8:0:1::1/64
  - ipv6 address FC00:DEAD:BEEF:CAFE::1/64
  - ipv6 nd prefix 2001:DB8:0:1::/64 no-advertise
  - ipv6 nd prefix FC00:DEAD:BEEF:CAFE::/64



Test Scenario #5

- The clients are on the **same subnet** as the webserver and get a GUA and a ULA address via Router Advertisements. The webserver is still statically configured to the GUA address but has additionally an ULA address configured. The DNS server holds in this case both addresses of the webserver as AAAA records (to be used) in round robin mode.

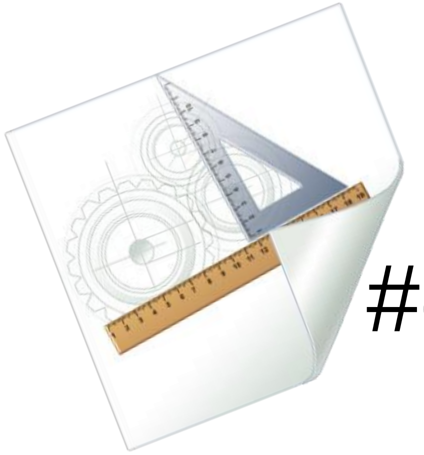
- Relevant router configuration of the client subnet:
  - ipv6 address 2001:DB8:0:1::1/64
  - ipv6 address FC00:DEAD:BEEF:CAFE::1/64
  - ipv6 nd prefix 2001:DB8:0:1::/64
  - ipv6 nd prefix FC00:DEAD:BEEF:CAFE::/64



Test Scenario #6

- The clients are on the **same subnet** as the webserver and get 2 GUA addresses via Router Advertisements. The webserver is statically configured to the GUA address and the DNS serves only the GUA address of the webserver.

- Relevant router configuration of the client subnet:
    - ipv6 address 2001:DB7:0:1::1/64
    - ipv6 address 2001:DB8:0:1::1/64
    - ipv6 nd prefix 2001:DB7:0:1::/64
    - ipv6 nd prefix 2001:DB8:0:1::/64

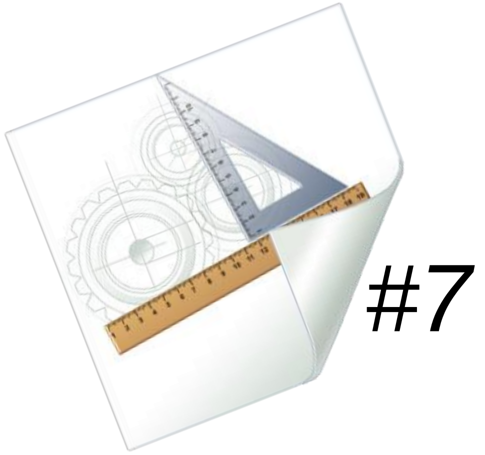


#6

Test Scenario #7

- The clients are on a **different subnet** as the webserver and gets a GUA address via Router Advertisements. The webserver is statically configured to the GUA address in the first subnet and the DNS serves only the GUA address of the webserver.

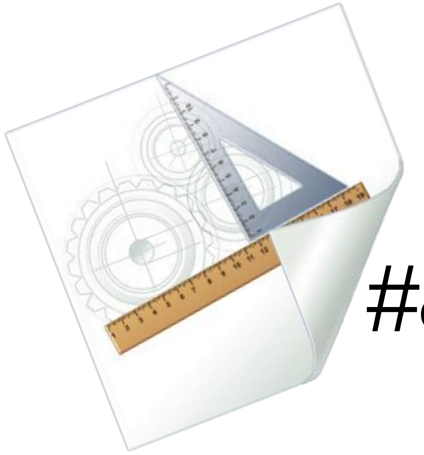
- Relevant router configuration of the client subnet:
  - ipv6 address 2001:DB9:0:1::1/64
  - ipv6 nd prefix 2001:DB9:0:1::/64



Test Scenario #8

- The clients are on a **different subnet** as the webserver and get GUA as well as a ULA address via Router Advertisements. The webserver is statically configured to the GUA address in the first subnet but has an additional ULA address. The DNS serves the GUA as well as the ULA address of the webserver.

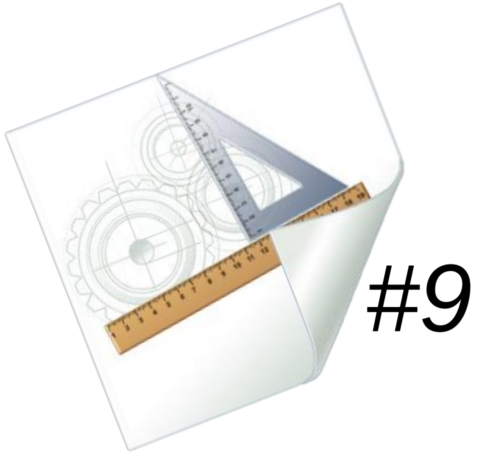
- Relevant router configuration of the client subnet:
  - ipv6 address 2001:DB9:0:1::1/64
  - ipv6 address FC00:DEAD:BEEF:CCCC::1/64
  - ipv6 nd prefix FC00:DEAD:BEEF:CCCC::/64
  - ipv6 nd prefix 2001:DB9:0:1::/64



#8

Test Scenario #9

- The clients are on a **different subnet** as the webserver and get 2 GUA addresses via Router Advertisements. The webserver is statically configured to the GUA address in the first subnet and the DNS serves only the GUA address of the webserver.
- Relevant router configuration of the client subnet:
  - ipv6 address 2001:DB9:0:1::1/64
  - ipv6 address 2001:DB10:0:1::1/64
  - ipv6 nd prefix 2001:DB9:0:1::/64
  - ipv6 nd prefix 2001:DB10:0:1::/64



ERNW
providing security.

RIPE 74
BUDAPEST, HUNGARY
8 - 12 May 2017

Results

Scenarios and test operating systems

Expected Results

according to RFC6724.

Case	Result	Rule
1	Link-local	Preferred outgoing interface
2	GUA TMP	Prefer temporary addresses, else longest matching prefix
3	GUA TMP	Prefer temporary addresses, else longest matching prefix
4	ULA TMP	Prefer temporary addresses
5	GUA TMP	Prefer temporary addresses, else longest matching prefix
6	GUA TMP (db8)	Prefer temporary addresses, else longest matching prefix
7	GUA TMP	Prefer temporary addresses
8	GUA TMP	Prefer temporary addresses, else longest matching prefix
9	GUA TMP (db9)	Prefer temporary addresses, else longest matching prefix

Let’s go…

- We tested each operating system according to our methodology and verified the results. The following table shows the results of our experiment after the tests are finished.

- As we can see quickly, there are differences between the operating systems for the source address selection and not all operating systems select the same source address in our test setup.

Initial Observations

- For the **first** test case:
  - Only the two Linux systems are able to resolve the IPv6 address via DNS and reach the server correctly.
  - Windows systems have problems with DNS resolution but can access the server directly via IP address while Mac OS-X based systems and are not able to connect to the server at all.

- The **second** test case:
  - Has an expected result for all windows systems.
  - Mac OS-X based systems show a different behavior and tend to skip the usage of the temporary address in this scenario but still have them configured on the interfaces while Debian does not have a temporary address by default as this is disabled in the operating system.

- The result of the **third** test case:
  - Is identical to the second.
  - However, since the ULA address of the webserver is not distributed via DNS this is expected.

Initial Observations II

- The **fourth** test case:
  - Shows the expected result for all operating systems. Here the Mac based systems also use the temporary address, again only Debian not, because of the default setting.

- The **fifth** test case:
  - Shows an interesting result for Windows 7 and the Linux based systems. In the DNS both ULA and GUA address of the server is published and those systems tend to use this information in a round-robin behavior.
  - The other systems always preferred the GUA address, even if they get both via DNS information.
  - Again the Mac based systems do not use the temporary address.

- The **sixth** test case shows again a result that is expected with respect to the RFC.

Initial Observations III

- The **seventh** test case:
  - Shows also an expected result.
  - However, this test case is very similar to the second case with only one GUA address but in this case the OSX based systems make use the temporary address.

- The **eighth** test case:
  - Can be compared to the fifth case and shows the very same results with respect to the changed subnet addresses.

- The **ninth** test case shows again an expected result with respect to the RFC.

ERNW providing security.

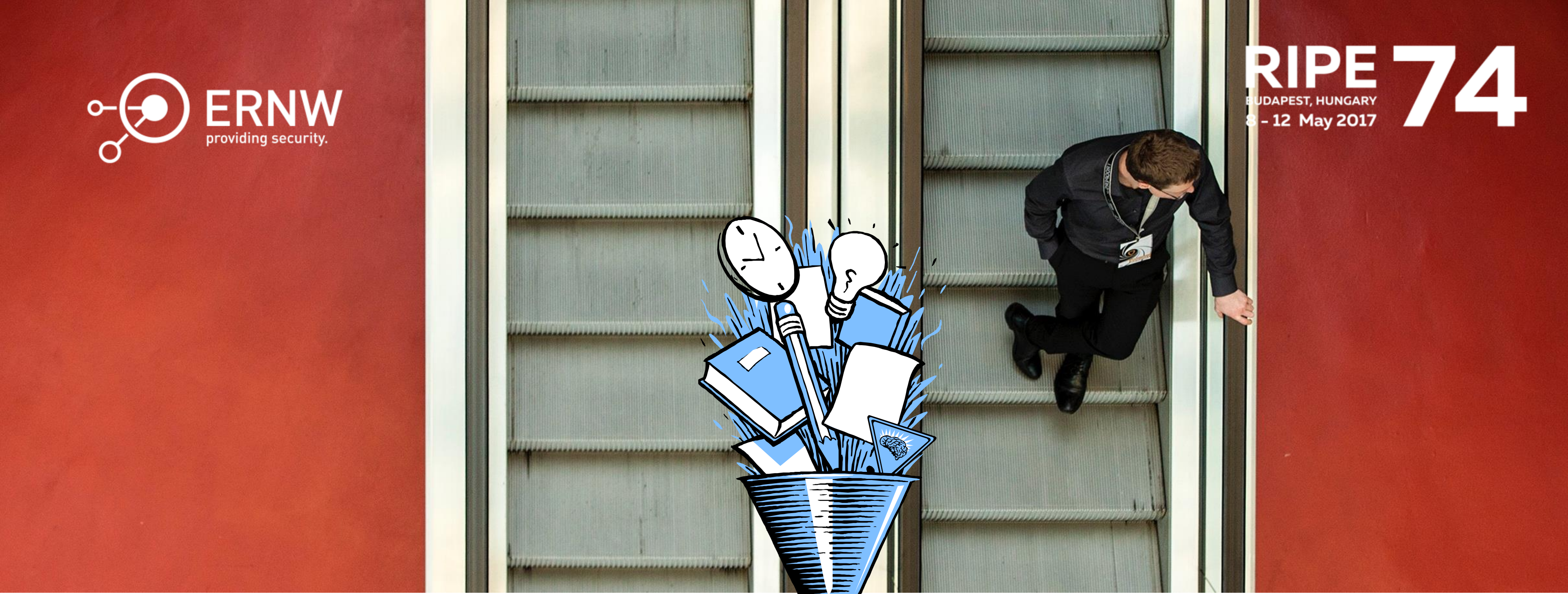
Results / Overview

RIPE 74 BUDAPEST, HUNGARY 8 - 12 May 2017

Same Subnet													
Test	LL	GUA	ULA	Client Prefix	Server IP	Win 7	Win 8.1	Win 10	Ubuntu 16.4 LTS	Debian 8	Mac OSX	Win 10 Mobile	Iphone
1	x	su		2001:db8:0:1::/64 no-advertise	2001:db8:0:1::2	LL (2)	LL (2)	LL (2)	LL	LL	not reachable	LL (2)	not reachable
2	x	x		2001:db8:0:1::/64	2001:db8:0:1::2	GUA TMP	GUA TMP	GUA TMP	GUA TMP	GUA	GUA	GUA TMP	GUA
3	x	x	x	2001:db8:0:1::/64 FC00:DEAD:BEEF:CAFE::/64	2001:db8:0:1::2	GUA TMP	GUA TMP	GUA TMP	GUA TMP	GUA	GUA	GUA TMP	GUA
4	x		x	2001:db8:0:1::/64 no-advertise FC00:DEAD:BEEF:CAFE::/64	2001:db8:0:1::2	ula tmp	ula tmp	ula tmp	ula tmp	ula	ula tmp	ula tmp	ula tmp
5	x	x	x	2001:db8:0:1::/64 FC00:DEAD:BEEF:CAFE::/64	2001:db8:0:1::2 FC00:DEAD:BEEF:CAFE::2	GUA TMP or ULA TMP (1)	gua tmp	gua tmp	GUA TMP or ULA TMP (1)	GUA or ULA (1)	GUA	GUA TMP	GUA
6	x	2		2001:db8:0:1::/64 2001:db7:0:1::/64	2001:db8:0:1::2	GUA TMP (db8)	GUA TMP (db8)	GUA TMP (db8)	GUA TMP (db8)	GUA (db8)	GUA TMP (db8)	GUA TMP (db8)	GUA TMP (db8)
Routed Subnet													
7	x	x		2001:db9:0:1::/64	2001:db8:0:1::2	GUA TMP	GUA TMP	GUA TMP	GUA TMP	GUA	GUA TMP	GUA TMP	GUA TMP
8	x	x	x	2001:db9:0:1::/64 FC00:DEAD:BEEF:CCCC::/64	2001:db8:0:1::2 FC00:DEAD:BEEF:CAFE::2	DNS: ULA TMP IP (GUA): GUA TMP	GUA TMP	GUA TMP	DNS: ULA TMP IP (GUA): GUA TMP	DNS: ULA IP (GUA): GUA	GUA TMP	GUA TMP	GUA TMP
9	x	2		2001:db9:0:1::/64 2001:db10:0:1::/64	2001:db8:0:1::2	GUA TMP (db9)	GUA TMP (db9)	GUA TMP (db9)	GUA TMP (db9)	GUA (db9)	GUA TMP (db9)	GUA TMP (db9)	GUA TMP (db9)

Further Interesting Observations

- As we expected, some of the operating systems do not select the source address as we predicted them before.
  - Interesting here is the fact, that Windows 7 as well as Ubuntu and Debian seem to use the two AAAA records in the DNS in round-robin mode and prefer sometimes the ULA address and another time the GUA address while newer versions of Windows use always the GUA address as well as MAC OSX and iPhones.
- Another observation is, that MAC OSX and iPhones do not use privacy extensions in every test case.
  - In some cases, they prefer the GUA address over the GUA temporary address, which is not expected if we look at the rules of the RFC.



ERNW providing security.

RIPE 74 BUDAPEST, HUNGARY 8 - 12 May 2017

Summary & Conclusion

Conclusions

- Overall (somewhat to our surprise) all tested OSs seem to follow RFC 6724 in a consistent manner.

- Differences as for their behavior arise from
  - OS specific properties re: privacy extensions.
  - OS specific behavior re: handling DNS (responses).

- Interactions with other mechanisms have to be kept in mind.

There's never enough time…

THANK YOU…

…for yours!

@Enno_Insinuator

erey@ernw.de

ernw.de

insinuator.net

Slides available soon.

ERNW
providing security.

RIPE 74
BUDAPEST, HUNGARY
8 - 12 May 2017

Sources

- Antonios Atlasis, E. R. (03. March 2015). MLD Considered Harmful.
  - https://www.troopers.de/media/filer_public/7c/35/7c35967a-d0d4-46fb-8a3b-4c16df37ce59/troopers15_ipv6secsummit_atlasis_rey_salazar_mld_considered_harmful_final.pdf
- Auer, K. (21. June 2012). Controlling IPv6 source address selection.
  - http://biplane.com.au/blog/?p=30
- Auer, K. (21. June 2012). IPv6 Source Address Selection – what, why, how.
  - http://biplane.com.au/blog/?p=22
- C. Huitema, B. C. (September 2004). Deprecating Site Local Addresses.
  - https://tools.ietf.org/html/rfc3879
- C. Perkins, D. J. (July 2011). Mobility Support in IPv6.
  - https://tools.ietf.org/html/rfc6275

ERNW
providing security.

RIPE 74
BUDAPEST, HUNGARY
8 - 12 May 2017

Image Sources

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