IPv6 Router Advertisement Flags, RDNSS and DHCPv6 Conflicting Configurations

Operational & Security Implications – Second Iteration – July 2017

Version:	1.1
Date:	17/07/2017
Classification:	Public
Author(s):	Antonios Atlasis;Enno Rey;Omar Eissa;Christopher Werny

Table of Content

List of Tables

1 Handling

The present document is classified as Public.

1.1 Document Status and Owner

As the owner of this report, the document owner has exclusive authority to decide on the dissemination of this document and responsibility for the distribution of the applicable version in each case to the places defined in the respective section.

The possible entries for the status of the document are "Initial Draft", "Draft", "Effective" (currently applicable) and "Obsolete".

Title:	IPv6 Router Advertisement Flags, RDNSS and DHCP Conflicting Configurations – Operational & Security Implications – Second Iteration – July 2017
Document Owner:	ERNW GmbH
Version:	1.1
Status:	Effective
Classification:	Public
Author(s):	Antonios Atlasis;Enno Rey;Omar Eissa;Christopher Werny

1.2 Possible Classifications:

Public:	Everyone
Internal:	All employees and customers
Confidential:	Only employees
Secret:	Only specific employees

2 Introduction

The IPv6 world is a complicated one, there is no doubt about that. The numerous new features and possibilities added to the protocol, either from its very first specification, RFC 2460 [1], to its later enhancements, certainly increase flexibility and capabilities, but they may also introduce operational issues, at least in "contradicting scenarios" which may lead even to security implications.

One of the several capabilities provided by IPv6 is that hosts are provided two options to configure their IPv6 address(es) and other parameters like the *Recursive DNS Server* (RDNSS) information [2]; that is either by using the stateless address autoconfiguration [3] or by obtaining the information from a DHCPv6 server[4]. IPv6 hosts are advised as for the environment options by some specific flags included in the *Router Advertisement* (RA) messages sent by the local router(s). These flags are the following [2]:

- The "*Managed address configuration*" flag (M). When set, this flag indicates that IPv6 addresses are available via DHCPv6.
- The "*Other configuration*" flag (O). When set, this flag indicates that other configuration information, like DNS-related one, is available via DHCPv6.

As [2] clearly states:

1. If neither M nor O flags are set, this indicates that there is no information via DHCPv6.
2. If the M flag is set, the O flag is redundant and it can be ignored.

Moreover, one of the options that can be used by RA messages is the "*Prefix Information*" one. This option includes, among else, yet another flag relevant for our purposes:

- The "*Autonomous address configuration*" (A) flag. When set, this flag indicates that this prefix can be used for stateless address autoconfiguration, as specified in [3].

Initially, no option was defined for configuring DNS information by using the RA messages only. Later, some more options were added to the RA messages which provide the IPv6 clients the capability to configure their RDNSS and / or DNS Search List (DNSSL) records directly from RAs without the need of DHCPv6-provided information [5]. As stated, "*this is a useful alternative in networks where an IPv6 host's address is autoconfigured through IPv6 stateless address autoconfiguration and where there is either no DHCPv6 infrastructure at all, or some hosts do not have a DHCPv6 client.*"

3 Problem Statement

Let's discuss now some interesting cases where some information is either not available, or contradictory, or even provided by more than one source.

As section 5.5.2 RFC 4862 [3] states, in absence of RAs, "*even if a link has no routers, the DHCPv6 service to obtain addresses may still be available, and hosts may want to use the service*".

Regarding the order of the RDNSS, according to section 5.3.1 of RFC 6106 [5] the host SHOULD copy the values of the options into the Resolver Repository in order. Specifically, as it is also explained in section 6.2 of the same RFC, for each RDNSS address, if it does not exist in the DNS Server List, it should be inserted in at the top of the Resolver Repository. When the IPv6 host has gathered a sufficient number (e.g. three) of RDNSS addresses, the latest received RDNSS SHOULD be more preferred than the old ones. Specifically, the new one replaces the old one that will expire first in terms of Lifetime.

What should happen if DNS information such as RDNSS can be obtained from multiple sources, such as RA and DHCPv6? According to RFC 6106, section 5.3.1 [5] "*the IPv6 host SHOULD keep some DNS options from all sources*". However, as it is also stated, "*the DNS information from DHCP takes precedence over that from RA for DNS queries*".

If we look carefully to the wording used by the aforementioned RFCs ("indicates", "can be used", "may", etc.), it is clear that all these flags are advisory, and not prescriptive (or "normative" as of IETF terminology). To make things more complicated, some vendors (Microsoft) have explicitly declared publicly that they do not intend to develop any support for RFC 6106 into their OS (Windows) [7] [8]. As of May 2017, this statement is no longer true as Microsoft has implemented RFC 8106 (6106) beginning with the Windows 10 Creators Update[1]

So, the question that arise, is: what will happen if there is contradictory configuration information provided by either one source or from more than one source? What if, for instance, the M-flag is set and a prefix information is also advertised with the A flag set? What if RDNSS information is provided from both RAs and a DHCPv6 server? Do hosts follow the aforementioned recommendation? And, what if the contradictory information is provided by RAs sent by two different routers? In all these cases, are there any operational implications, or even worse, can an attacker take advantage of such cases in order to launch (potentially very subtle) attacks?

There is not a lot of work in the literature examining such contradictory scenarios. Answers to some of these questions were given in [6], where the behaviour of popular Operating Systems (OS) was examined under various combinations of the M, O and A flags. However, the RDNSS option was not set at the tested RA messages, while scenarios where there are two routers simultaneously in the same local link providing contradictory information, were not examined. This paper attempts to advance the study of the topic one step further by including scenarios where a) an RDNSS information is provided by RA messages and b) there is either one router and a DHCPv6 server on the same local link, or two router providing contradictory information. As it will be shown, not only there is a divergent behaviour between the different OS belonging even to the same OS family, but also the operational consequences as well as the security implications of at least some of them can be severe. The final goal of the paper is to offer a better understanding of the IPv6 world at the local link so as to avoid unwanted surprises in your environments.

[1] *https://insinuator.net/2017/05/one-step-closer-rdnss-rfc-8106-support-in-windows-10-creators-update/*

4 Lab Set-Up and Testing Methodology

4.1 Lab Set Up

Our lab consists of the following devices:

- A DHCPv6 Server and specifically, a DHCP ISC Version 4.3.1 installed in CentOs 6.6. The DHCPv6 server is configured to provide both IPv6 addresses and RDNSS information.
- Two (2) routers Cisco 4321 using Cisco IOS Software version 15.5(1)S.
- The following OS as clients:
  - Fedora 21, kernel version 3.18.3-201 x64
  - Ubuntu 14.04.1 LTS, kernel version 3.13.0-44-generic
  - CentOS 7, kernel version 3.10.0-123.13.2.el7
  - Mac OS-X 10.10.2 Yosemite 14.0.0 Darwin
  - Windows 7
  - Windows 8.1

NOTES:

1. To enable the Ubuntu system to get a RDNSS from an IPv6 router directly (so as to implement RFC 6106 [5]), it needs to have installed the *rdnssd* packet (it is not installed by default). This is not the case for the other tested Linux systems, which get RDNSS information by default.
2. All the devices are located on the same link.
3. In all tests, all host start from a clean state (they have neither IPv6 addresses nor RDNSS information).

4.2 Testing Methodology

For our testing purposes we create a list of specific scenarios with conflicting advertised parameters and we examine which IPv6 addresses and which RDNSS are assigned to each host. We also examine the precedence of the RDNSS entries if more than one are assigned. The scenarios described below are not an exhaustive list of all combinations of the values of the various RA parameters (A, M, O). Instead, the most interesting cases are examined which can create unexpected behaviour from the clients. Actually, the expected behaviour of all the other potential cases can be derived by observing the results of the presented scenarios.

We split the tests in two major categories; first, when one IPv6 router and a DHCPv6 server are on the same link and secondly, when two routers with conflicting parameters and a DHCPv6 server co-exist. In each tested scenario, immediately after the results we examine any operational or security considerations for the most interesting cases.

5 FIRST ITERATION (FEBRUARY 2015) - PERFORMED TESTS AND RESULTS

5.1 Scenarios Using Only One IPv6 Router and a DHCPv6 Server

In these scenarios there is two one router and, unless otherwise specified, one DHCPv6 server on the same link. The behaviour of the router and of the DHCPv6 server remain unchanged during the tests.

The results of this section are summarised at the end of it in Table 1.

5.1.1 Case 1: One Router with the Management Flag not Set and a DHCPv6 Server

This is the only scenario from the ones presented in this document where there are no conflicting parameters. It is used to set a baseline and examine which OS implement RFC 6106.

5.1.1.1 Set Up

In this test case there are:

- One IPv6 Router with the following settings:
  - M=0, A=1, O=0 and an RDNSS is advertised.
- A DHCPv6 server on the same link advertising IPv6 addresses and RDNSS.

In this test, the questions are the following:

a) Which OS accept RDNSS information from the RAs (implementing RFC 6106)?

b) Although neither M nor O flags are set, which OS (if any) obtain information (IPv6 address and / or RDNSS) from the DHCPv6 server?

5.1.1.2 Results

The results of these tests are briefly described below:

- Fedora 21, MAC OS-X, CentOS 7 and Ubuntu 14.04 get an IPv6 address and an RDNSS from the IPv6 router only.
- Windows 7 get an IPv6 address from the router only, but they do not get any DNS information, neither from the router nor from the DHCPv6 server. They also do not get IPv6 address from the DHCPv6 server.
- Windows 8.1 get an IPv6 address from both the IPv6 router and the DHCPv6 server, despite the fact that the Management flag (M) is not set. They get RDNSS information from the DHCPv6 only.

5.1.1.3 Comments

- Fedora 21, MAC OS-X, CentOS 7 and Ubuntu 14.04 have what we could call an RFC compliant behaviour. This would be the case regarding Windows 7 too with the exception that they do not get RDNSS information from the router.
- Windows (7/8.1) do not implement RFC 6106.
- Windows 8.1 seeks for DHCPv6 server information in any case (so, it seems that they do not examine the M and O flags of the RAs in this scenario).
- Windows 7 and Windows 8.1 exhibit a different behaviour.

5.1.1.4 Security Considerations

An attacker, without having to install a rogue router, can install a rogue DHCPv6 server and provide IPv6 addresses to Windows 8.1 systems. This can allow her to interact with these systems in a different scope, which, for instance, is not monitored by an IDPS system.

5.1.2 Case 2: One Router with Conflicting Parameters and a DHCPv6 Server

5.1.2.1 Set Up

In this test case, we have:

- An IPv6 Router with the following settings:
  - M=0, A=1, O=1, and an RDNSS is advertised.
- A DHCPv6 server on the same link advertising IPv6 addresses and RDNSS. .

NOTE: The conflicting parameters are the O flag which is set (advising the clients to get other information like DNS from a DHCPv6 server) and the RDNSS information advertised by the RAs.

5.1.2.2 Results

- Fedora 21, Centos 7 and Ubuntu 14.04 get IPv6 address using SLAAC only (no address from the DHCPv6 server).
  - Fedora 21, Centos 7 get RDNSS from both the RAs and the DHCPv6 server. The RDNSS obtained from the router has a higher priority though.
  - Ubuntu 14.04 gets an RDNSS from the router, and a "domain search list" information from the DHCPv6 server – but not RDNSS information.
- MAC OS -X also gets RDNSS from both, IPv6 address using SLAAC (no IPv6 address from the DHCPv6 server) but the RDNSS obtained from the DHCPv6 server is first (it has a higher priority). However, the other obtained from the RAs is also present.
- Windows 7 and Windows 8.1 obtain IPv6 addresses using SLAAC and RDNSS from the DHCPv6 server. They do not get IPv6 address from the DHCPv6 server. Compare the Windows 8.1 behaviour with the previous case.

5.1.2.3 Comments

- MAC OS-X gives higher priority to the RDNSS obtained from the DHCPv6 server.
- Fedora and Centos give higher priority to the RDNSS information obtained from the router.
- Windows 8.1:
  - When M=0, O=0 and A=1 (Case 1) get an IPv6 address both from the DHCPv6 server and using SLAAC from the RAs.
  - When M=0, O=1 and A=1 (Case 2) they get IPv6 address using SLAAC only, and not from the DHCPv6 server.
  - When M=1, A=1, O=1 (see Case 4 later) they also get an IPv6 address from both the router and the DHCPv6 server. But this is expected since the M flag is set.

There is a kind of contradictory behaviour between the cases where O=0, M=0 and O=1 and M=0.

5.1.2.4 Security Considerations

If you want to perform MiTM using a rogue DNS while legitimate RAs with the O flag set are sent to enforce the use of a DHCPv6 server, you can spoof RAs with the same settings with the legitimate prefix (in order to remain undetectable) but advertising YOUR (attacker's) DNS using RDNSS. In this case, Fedora 21, Centos 7 and Ubuntu 14.04 will use the rogue RDNSS (advertised by the RAs) as a first option.

5.1.3 Case 3: Same as Case 2 but Without a DHCPv6 Server

5.1.3.1 Set Up

In this test case, we have:

- An IPv6 Router with the following settings:
  - M=0, A=1, O=1, and an RDNSS is advertised.
- There is no DHCPv6 present.

5.1.3.2 Results

- Windows 7 and Windows 8.1 get an IPv6 address using SLAAC but they do not get RDNSS information.
- MAC OS-X, Fedora 21, Centos 7 and Ubuntu 14.04 get an IPv6 address using SLAAC and RDNSS from the RAs.

5.1.3.3 Comments

This is not an interesting case at all. All results are as expected.

5.1.4 Case 4: All Flags are Set and a DHCPv6 Server is Present

5.1.4.1 Set Up

In this test case, we have:

- An IPv6 Router with the following settings:
  - M=1, A=1, O=1, and an RDNSS is advertised.
- A DHCPv6 server on the same link advertising IPv6 addresses and RDNSS. .

5.1.4.2 Results

- Fedora 21 and Centos 7:
  - They get IPv6 address both from SLAAC and DHCPv6 server.
  - They get RDNSS both from RAs and DHCPv6 server.
  - The DNS of the RAs has higher priority.
- Ubuntu 14.04:
  - It gets IPv6 address both using SLAAC and from the DHCPv6 server.
  - It gets RDNSS from RAs only.
  - From the DHCPv6 server it only gets "Domain Search List" information, no RDNSS.
- MAC OS-X:
  - It gets IPv6 addresses both using SLAAC and from the DHCPv6 server.
  - It also gets RDNSS both from RAs and the DHCPv6 server.
  - The DNS server of the DHCPv6 has higher priority.
- Windows 7 and Windows 8.1:
  - They get IPv6 address both from SLAAC and DHCPv6 server.

  - They get RDNSS only from the DHCPv6 server.

5.1.4.3 Comments

- Ubuntu, when there is RDNSS information from both the router and a DHCPv6 server, they only get this information from the router.
- Fedora 21 and Centos 7 show a preference to the RDNSS obtained from the RAs once more.

5.1.5 Case 5: All Flags are Set and There is No DHCPv6 Server is Present

5.1.5.1 Set Up

In this test case, we have:

- An IPv6 Router with the following settings:
  - M=1, A=1, O=1, and an RDNSS is advertised.
  - There is no DHCPv6 is present.

5.1.5.2 Results

- Windows 7 and Windows 8.1 get an IPv6 address using SLAAC but no RDNSS information.
- MAC OS-X, Fedora 21, Centos 7, Ubuntu 14.04 get an IPv6 address using SLAAC and RDNSS from the RAs.

5.1.5.3 Comments

This is not an interesting case either. All results are as expected.

5.1.6 Case 6: A Prefix is Advertised by RAs but the 'A' flag is not Set

5.1.6.1 Set Up

In this test case, we have:

- An IPv6 Router with the following settings:
  - M=0, A=0 (while a prefix information is advertised), O=0 and an RDNSS is advertised.
  - DHCPv6 is present

5.1.6.2 Results

- Fedora 21, Centos 7, Ubuntu 14.04 and MAC OS-X:
  - They do not get any IPv6 address (neither from the RAs, nor from the DHCPv6).
  - They get a RDNSS from the router only (not from DHCPv6).
- Windows 8.1
  - They get IPv6 address and RDNSS from the DHCPv6 server ("last resort" behaviour).
  - They do not get any information (neither IPv6 address not RDNSS) from the router.
- Windows 7:
  - They get nothing (neither IPv6 address nor RDNSS) from any source (RA or DHCPv6).

5.1.6.3 Comments

- There is a different behaviour between Windows 8.1 and Windows 7. Windows 8.1 implement the "last resort" capability.
- Strictly speaking, the Fedora, Centos, Ubuntu and MAC OS-X could be considered displaying an RFC compliant behaviour.

Table 1: One IPv6 Router in the Environment

	Scenario	Collected Information	Windows 7	Windows 8.1	Ubuntu 14	Centos 7	Fedora 21	MAC OS-X
1	A=1, M=0, O=0 DHCPv6 present	IPv6 address	router	both	router	router	router	router
		RDNSS	-	DHCPv6	router	router	router	router
2	A=1, M=0, O=1 DHCPv6 present	IPv6 address	router	router	router	router	router	router
		RDNSS	DHCPv6	DHCPv6	router	router/DHCPv6	router/DHCPv6	DHCPv6/router
3	A=1, M=0, O=1 no DHCPv6 present	IPv6 address	router	router	router	router	router	router
		RDNSS	-	-	router	router	router	router
4	A=1, M=1,O=1 DHCPv6 present	IPv6 address	Both	both	both	both	both	both
		RDNSS	DHCPv6	DHCPv6	router	router/DHCPv6	router/DHCPv6	DHCPv6/router
5	A=1, M=1,O=1 no DHCPv6 present	IPv6 address	router	router	router	router	router	router
		RDNSS	-	-	router	router	router	router
6	A=0, M=0, O=0 DHCPv6 present	IPv6 address	-	DHCPv6	-	-	-	-
		RDNSS	-	DHCPv6	router	router	router	Router

NOTES:

1. **In all cases, RDNSS is advertised from the RAs sent by the IPv6 router.**
2. **Router/DHCPv6 means that RDNSS is obtained from both the IPv6 Router and the DHCPv6 server, but the one obtained from the IPv6 router has a higher priority.**
3. **DHCPv6/Router means that RDNSS is obtained from both the IPv6 Router and the DHCPv6 server, but the one obtained from the DHCPv6 server has a higher priority.**
4. **Both (regarding addresses) means that an IPv6 address is obtained both by using Stateless Address AutoConfiguration and the DHCPv6 Server.**

5.2 Scenarios Using Two IPv6 Router and a DHCPv6 Server

In these scenarios there are two routers on the same link. At first, only one router is present (resembling the "legitimate router)", while the second one joins the link after the clients first configured by the RAs of the first router. Our goal is to examine the behaviour of the clients during the interchange of the RAs from the two different routers.

The results of this section are summarised at the end of it, in Table 2.

5.2.1 Case 7: Router 1 Advertising M=0, O=0 and RDNSS, and then Router 2 advertising M=1, O=1 while DHCPv6 is Present

5.2.1.1 Set Up

In this test case, we have:

Initially:

- One IPv6 router with the following settings:
  - M=0, O=0, A=1 and RDNSS advertised and 15 seconds time interval of the RAs.

After a while (when clients are configured by the RAs of the above router):

- Another IPv6 router with the following settings:
  - M=1, O=1, no advertised prefix information, and 30 seconds time interval of the RAs.
  - A DHCPv6 server on the same link advertising IPv6 addresses and RDNSS.

Assuming that the second router and the DHCPv6 server are rogue ones (since they joined the network later), the best approach for the clients from a security perspective would be to retain only the information from the initial router. Still, this is probably against the RFCs which presumably prescribe to incorporate the information heard most recently, not least for failover scenarios and to allow for flexible changes in live networks.

5.2.1.2 Results

- MAC OS-X and Ubuntu 14.04:
  - Initially they get address and RDNSS from the first router.
  - When they receive RAs from the second router, they never get any information (IPv6 address or RDNSS) from the DHCPv6 server.
- Windows 7:
  - Initially they get address from the first router – no RDNSS.
  - When they receive RAs from the second router, they never get any information (IPv6 address or RDNSS) from the DHCPv6 server.
- Fedora 21 and Centos 7:
  - Initially they get IPv6 address and RDNSS from the RAs of the first router.
  - When they receive an RA from router 2, they also get an IPv6 address and RDNSS from the DHCPv6 server while retaining the ones (IPv6 address and RDNSS) obtained from the RAs of the first router. The RDNSS obtained from the first router has a higher priority than the one obtained from the DHCPv6 server (probably because it was received first).
  - When they receive again RAs from the first router, they lose/forget the information (IPv6 address and RDNSS) obtained from the DHCPv6 server.

- Windows 8.1:
  - Initially, they get just an IPv6 address from the first router 1 - no RDNSS information (since they do not implement RFC 6106).
  - When they receive RAs from the second router, then they also get an IPv6 address from the DHCPv6 server, as well as RDNSS from it. They do not lose the IPv6 address obtained by the first router using SLAAC.
  - When they receive RA from the first router, they retain all the obtained so far information (there isn't any change).

5.2.1.3 Security Considerations

MAC OS-X behaviour seems to display the best approach from a security perspective.

Fedora 21 and Centos 7 behaviour cannot be explored for a MiTM attack using a rogue DNS information either, since the one obtained by the RAs of the first router has a higher priority.

5.2.2 Case 8: (Router 2) Initially M=1, O=1 and DHCPv6, then 2nd Router (Router 1) Rogue RAs Using M=0, O=0 and RDNSS Provided

5.2.2.1 Set Up

In this test case, we have:

Initially:

- One IPv6 router with the following settings:
  - M=1, O=1, no advertised prefix information, and 30 seconds time interval of the RAs.
  - A DHCPv6 server on the same link advertising IPv6 addresses and RDNSS.

After a while (when clients are configured by the RAs of the above router):

- Another IPv6 router with the following settings:
  - M=0, O=0, A=1, RDNSS advertised and 15 seconds time interval of the RAs.

Assuming that the second router is rogue one (since it joined the network later), the best approach for the clients from a security perspective would be to retain only the information from the DHCPv6 server. Still, this is probably against the RFCs which presumably prescribe to incorporate the information heard most recently, not least for failover scenarios and to allow for flexible changes in live networks.

5.2.2.2 Results

- Fedora 21 and Centos 7:
  - At first, they get information (IPv6 address and RDNSS) from the DHCPv6 server.
  - When they receive RAs from the second router, they get address(es) and RDNSS from these RAs. At the same time, the IPv6 address and the RDNSS obtained from the DHCPv6 server are gone.
  - When they receives again an RA from the first router, they perform the DHCPv6 Confirm/Reply procedure and they get an IPv6 address and RDNSS from the DHCPv6 server while retaining the ones obtained from the RAs of the second router. Moreover, the RDNSS from router 1 has higher priority than the one from DHCPv6.

- Ubuntu 14.04:
  - At first, it gets information (IPv6 address and RDNSS) from the DHCPv6 server.
  - When it receives RAs from the second router, it also gets information from it, but it does not lose the information obtained from the DHCPv6 server. It retains both. It only gets "Domain Search list" from the DHCPv6 server – no RDNSS information.
  - When it receives RAs from the first router, there is no change; it retains all the obtained information.
- Windows 7:
  - Initially they get IPv6 address and RDNSS from the DHCPv6 server.
  - When they get RAs from the second router, they lose this information (IPv6 address and RDNSS obtained from the DHCPv6 server) and they get only SLAAC addresses using the RAs of the second router – no RDNSS.
  - When they receive RAs from the first router again, they get RDNSS and IPv6 address from the DHCPv6 server, but they also keep the SLAAC addresses.
- Windows 8.1:
  - Initially they get information (IPv6 address and RDNSS) from the DHCPv6 server.
  - When they receive RAs from the second router, they never get any information from them.
- MAC OS-X:
  - Initially it gets information (IPv6 address and RDNSS) from the DHPCv6 server.
  - When it gets RAs from the second router, it also gets a SLAAC IPv6 address but no RDNSS information from the RAs of this router. It also does not lose any information obtained from DHCPv6.
  - When it gets RAs from the first router again, the situation does not change (IPv6 addresses from both the DHCPv6 and SLAAC process are retained, but RDNSS information only from the DHCPv6 server).

5.2.2.3 Security Considerations

The behaviour of Fedora 21, Centos 7 and Windows 7 can be exploited for DoS purposes. A rogue IPv6 router not only provides its own information to the clients, but it also removes the previous obtained (legitimate) information.

The Fedora and Centos behaviour can also be exploited for MiTM purposes by advertising rogue RDNSS by RAs which include RDNSS information.

Table 2: Two IPv6 Routers in the Environment

	Scenario		Collected Information	Windows 7	Windows 8.1	Ubuntu 14	Centos 7	Fedora 21	MAC OS-X
7	Initial Situation	Router 1: A=1, M=0, O=0, RDNSS, 15 sec. RA interval	IPv6 address	router	router	router	router	router	router
			RDNSS	-	-	router	router	router	router
	Later addition	Router 2: M=1, O=1, no advertised prefix, 30 sec. RA interval DHCPv6 server.	IPv6 address	router	both	router	both	both	router
			RDNSS	-	DHCPv6	router	Router/DHCPv6	Router/DHCPv6	router
	Router 1 RAs received again		IPv6 address	router	both	router	router	router	router
			RDNSS	-	DHCPv6DHCPv6	router	router	router	router
8	Initial Situation	Router 2: M=1, O=1, no advertised prefix, 30 sec. RA interval DHCPv6 server	IPv6 address	DHCPv6	DHCPv6	DHCPv6	DHCPv6	DHCPv6	DHCPv6
			RDNSS	DHCPv6	DHCPv6	DHCPv6	DHCPv6	DHCPv6	DHCPv6
	Later addition	Router 1: A=1, M=0, O=0, RDNSS, 15 sec. RA interval	IPv6 address	Router 1	DHCPv6	both	Router 1	Router 1	both
			RDNSS	-	DHCPv6	Router 1	Router 1	Router 1	DHCPv6
	Router 2 RAs received again		IPv6 address	Both		both	both	both	both
			RDNSS	DHCPv6		Router 1	Router1/DHCPv6	Router1/DHCPv6	DHCPv6

NOTES: 1. Router 1/DHCPv6 means that RDNSS is obtained from both the IPv6 Router and the DHCPv6 server, but the one obtained from the IPv6 router has a higher priority.

2. Both (regarding addresses) means that an IPv6 address is obtained both by using Stateless Address AutoConfiguration and the DHCPv6 Server.

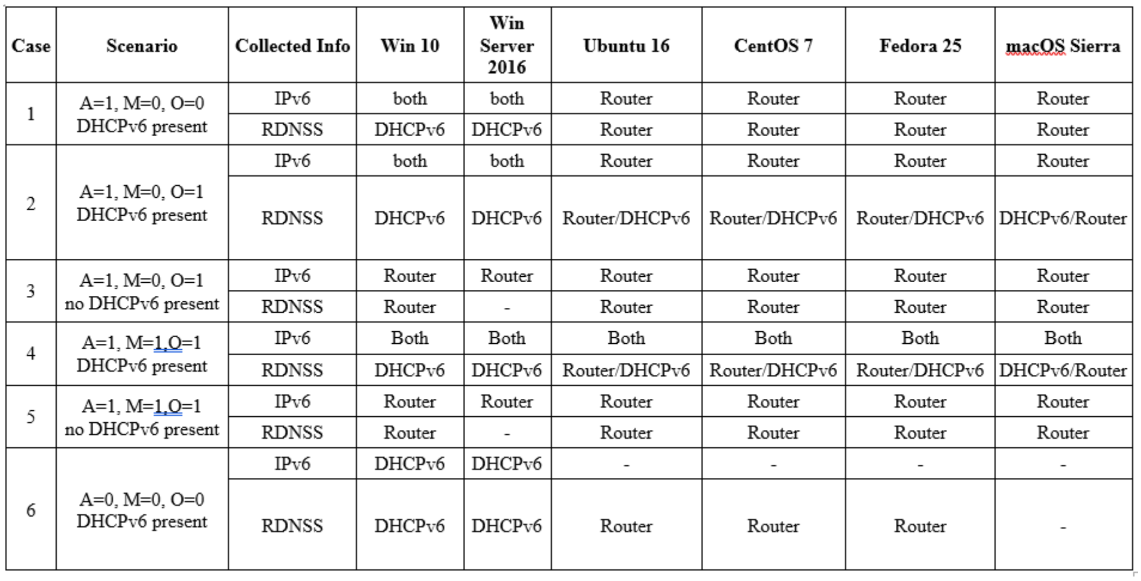
6 Second Iteration (July 2017) - Performed Tests and Results

Given that our initial tests were performed two years ago, and the several new operating system versions (and Linux kernels) were released in the meantime, we decided to perform the tests again to include the updated operating systems. The lab setup is analogue to our previous tests but used the following operating systems and DHCPv6 Server:

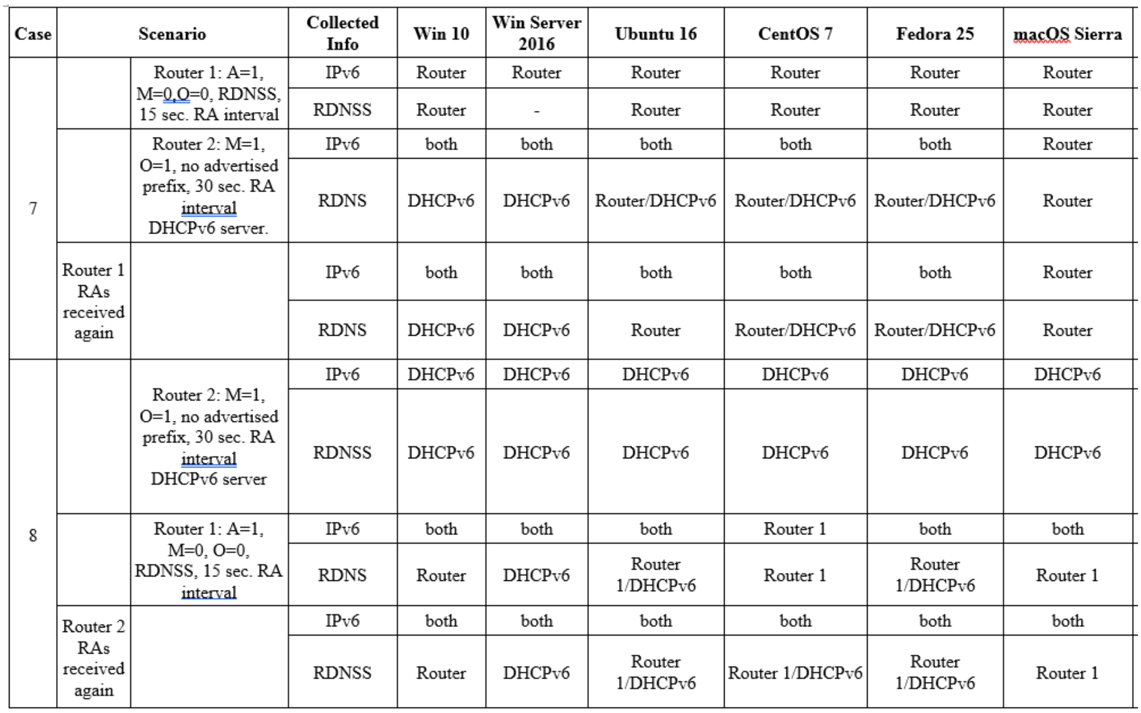
- ISC DHCPv6 Server 4.3.5 running on CentOS 7
- CentOS 7 Kernel 4.10.11-1.el7.elrepo.x86_64
- Fedora 25 Kernel 4.10.10-200.fc25.x86_64
- Ubuntu 16.04 LTS Kernel 4.10.11-041011-generic
- Windows 10 Build 1703 (Creators Update)
- Windows Server 2016 Build 1607
- macOS Sierra

6.1 Results

The following table outlines the results of the test cases:



Case	Scenario	Collected Info	Win 10	Win Server 2016	Ubuntu 16	CentOS 7	Fedora 25	macOS Sierra
1	A=1, M=0, O=0 DHCPv6 present	IPv6	both	both	Router	Router	Router	Router
		RDNSS	DHCPv6	DHCPv6	Router	Router	Router	Router
2	A=1, M=0, O=1 DHCPv6 present	IPv6	both	both	Router	Router	Router	Router
		RDNSS	DHCPv6	DHCPv6	Router/DHCPv6	Router/DHCPv6	Router/DHCPv6	DHCPv6/Router
3	A=1, M=0, O=1 no DHCPv6 present	IPv6	Router	Router	Router	Router	Router	Router
		RDNSS	Router	-	Router	Router	Router	Router
4	A=1, M=1,O=1 DHCPv6 present	IPv6	Both	Both	Both	Both	Both	Both
		RDNSS	DHCPv6	DHCPv6	Router/DHCPv6	Router/DHCPv6	Router/DHCPv6	DHCPv6/Router
5	A=1, M=1,O=1 no DHCPv6 present	IPv6	Router	Router	Router	Router	Router	Router
		RDNSS	Router	-	Router	Router	Router	Router
6	A=0, M=0, O=0 DHCPv6 present	IPv6	DHCPv6	DHCPv6	-	-	-	-
		RDNSS	DHCPv6	DHCPv6	Router	Router	Router	-



Case	Scenario		Collected Info	Win 10	Win Server 2016	Ubuntu 16	CentOS 7	Fedora 25	macOS Sierra
7		Router 1: A=1, M=0,O=0, RDNSS, 15 sec. RA interval	IPv6	Router	Router	Router	Router	Router	Router
			RDNSS	Router	-	Router	Router	Router	Router
		Router 2: M=1, O=1, no advertised prefix, 30 sec. RA interval DHCPv6 server.	IPv6	both	both	both	both	both	Router
			RDNS	DHCPv6	DHCPv6	Router/DHCPv6	Router/DHCPv6	Router/DHCPv6	Router
	Router 1 RAs received again		IPv6	both	both	both	both	both	Router
			RDNS	DHCPv6	DHCPv6	Router	Router/DHCPv6	Router/DHCPv6	Router
8		Router 2: M=1, O=1, no advertised prefix, 30 sec. RA interval DHCPv6 server	IPv6	DHCPv6	DHCPv6	DHCPv6	DHCPv6	DHCPv6	DHCPv6
			RDNSS	DHCPv6	DHCPv6	DHCPv6	DHCPv6	DHCPv6	DHCPv6
		Router 1: A=1, M=0, O=0, RDNSS, 15 sec. RA interval	IPv6	both	both	both	Router 1	both	both
			RDNS	Router	DHCPv6	Router 1/DHCPv6	Router 1	Router 1/DHCPv6	Router 1
	Router 2 RAs received again		IPv6	both	both	both	both	both	both
			RDNSS	Router	DHCPv6	Router 1/DHCPv6	Router 1/DHCPv6	Router 1/DHCPv6	Router 1

6.2 Observations

6.2.1 Test Case 1

- Even though neither the M nor the O Flag is set both Windows 10 and Windows Server 2016 get IPv6 addresses over SLAAC and DHCPv6. This is because of the implicit „Managed Address Configuration" Flag configured on the Interface. This mimics the behavior from the IPv4 world where Windows machines (as one of the first packets) sends a DHCPv6 Solicit message. The DNS server configured on the interface was taken from DHCPv6 (even though Windows 10 support RDNSS in the meantime[2]).
- All Linux derivate and MAC OS only use SLAAC for IPv6 address generation and acquire the DNS server from the router advertisement (we constitute this as RFC Compliant).

6.2.2 Test Case 2

- We couldn't observe anything special within this test case. As expected, both Windows 10 and Windows Server 2016 get an address over DHCPv6 and SLAAC and have received the DNS Server from DHCPv6
- CentOS, Fedora and MAC OS Sierra generated addresses by means of SLAAC and had both supplied DNS Server configured. The only difference here is that macOS Sierra seems to prefer the DHCPv6 supplied DNS server (compared to the Linux systems that prefer the SLAAC provided)

[2] *https://insinuator.net/2017/05/one-step-closer-rdnss-rfc-8106-support-in-windows-10-creators-update/*

6.2.3 Test Case 3

- Every operating system behaved like expected. All tested operating systems generated an IPv6 address by means of SLAAC and installed the DNS server supplied within the router advertisement with the only exception being Windows Server 2016 (due to the lack of RDNSS support).

6.2.4 Test Case 4

- As with test case 3, every operating systems behaved like expected. Windows based systems and macOS preferred the DHCPv6 provided DNS server while Linux systems preferred the SLAAC provided.

6.2.5 Test Case 5

- Every operating system behaved like expected. All operating systems generated an IPv6 address by means of SLAAC and installed the DNS server supplied within the router advertisement. Only exception was Windows Server 2016 (due to the lack of RDNSS support).

6.2.6 Test Case 6

- In this test case Windows based systems received an IPv6 via DHCPv6 and installed the supplied DNS server as well. Cent OS, Fedora and macOS Sierra did not generate any IPv6 address but installed the DNS server from the RA. The only exception here is macOS Sierra. We currently do not know why macOS behaves that way, and we weren't able to find any information/documents from Apple describing the behavior.

6.2.7 Test Case 7

- To reiterate, in this test case we have mnitially:
  - One IPv6 router with the following settings:
    - M=0, O=0, A=1 and RDNSS advertised and 15 second time interval of the RAs.
- After a while (when clients are configured by the RAs of the above router):
  - Another IPv6 router with the following settings:
    - M=1, O=1, no advertised prefix information, and 30 second time interval of the RAs.
    - A DHCPv6 server on the same link advertising IPv6 addresses and RDNSS.

- Assuming that the second router and the DHCPv6 server are rogue ones (since they joined the network later), the best approach for the clients from a security perspective would be to retain only the information from the initial router. Still, this is probably against the RFCs which presumably prescribe to incorporate the information heard most recently, not least for failover scenarios and to allow for flexible changes in live networks.

6.2.7.1 Results

- The result for the initial router advertisement is consistent across all platforms. IPv6 addresses got generated by means of SLAAC and DNS was installed from the RA (except for Windows Server 2016).

- After the second router and DHCPv6 server got into the network, all operating systems acquired an additional address over DHCPv6 except for macOS Sierra which ignored DHCPv6 completely and retained the configuration parameters from the first (legitimate router).

    - Windows 10 and Server 2016 installed the DHCPv6 provided DNS server as primary while all Linux variants installed the DHCPv6 provided additionally.

6.2.8 Test Case 8

- To reiterate, in this test case we have Initially:
  - One IPv6 router with the following settings:
    - M=1, O=1, A=0 and RDNSS advertised and 30 second interval of the RAs.
    - A DHCPv6 server on the same link advertising IPv6 addresses and RDNSS.
- After a while (when clients are configured by the RAs/DHCPv6 of the above router):
  - Another IPv6 router with the following settings:
    - M=0, O=0, A=1, RDNSS advertised and 15 seconds time interval of the RAs.

Assuming that the second router is rogue one (since it joined the network later), the best approach for the clients from a security perspective would be to retain only the information from the DHCPv6 server. Still, this is probably against the RFCs which presumably prescribe to incorporate the information heard most recently, not least for failover scenarios and to allow for flexible changes in live networks.

6.2.8.1 Results

- The initial result was for the most part the expected. All operating systems received an IPv6 as well as the DNS server information from the DHCPv6 server.

- After the second router got introduced, all operating systems generated an IPv6 address based on the PIO within the RA. For reasons unknown at the moment, CentOS 7 completely forgot about the DHCPv6 provided configuration and had only the SLAAC address and RA provided DNS configured. Nearly the same behavior could be observed for macOS Sierra. The only exception was that macOS still had the DHCPv6 provided address configured on the interface. Fedora and Ubuntu installed the RA provided DNS Server as primary on the interface (the same could be observed for Windows 10.

- When the operating systems received the RA from the initial router again, CentOS 7 has performed DHCPv6 again and the address and DNS server were visible on the system again. No further differences could be observed.

7 Conclusions and Future Work

Looking at the results of the tested scenarios, we can easily come to some specific conclusions. First, the behaviour of the OS in case of contradicting or additional information provided from routers and a DHCPv6 server cannot be taken for granted. Not only there are different behaviours between OS of the same family (e.g. Centos, Fedora and Ubuntu), but moreover, even between different versions of the same OS (Windows 7 and Windows 8.1). In an operational environment where more than OS and several versions are used, the expected behaviour cannot always be predicted and troubleshooting, in case of accidental misconfiguration, can be a pain for the system administrators. Furthermore, in some specific cases the OS state can change even between seconds while receiving RAs with contradictory information from different routers. Such a situation can be a real nightmare for system administrators. Of course, in several cases, as it was also discussed, such scenarios can also be exploited by attackers residing at the local link for causing DoS or even performing MiTM attacks. The argument that such attacks can take place at the local link anyway by other means is not a reason for ignoring them because, not only some more attack vectors are added, but also, some cases, if exploited properly, can be detected less easily.

Nevertheless, there is still room for more research on the topic. For instance, what is the maximum number of RDNSS information stored in each OS? When reaching this maximum number, can such information be overridden by newly provided one as the RFCs suggest? If so, which are the operational and security consequences? When different IPv6 addresses are provided by router and DHCPv6 servers, which ones are going to be used, etc.?

However, for the time being we can agree that all this additional functionalities and flexibility provided by IPv6 does not come without price; and this is not only the additional complexity but also the sometimes unexpected behaviour.

7.1 Conclusion from the Second Iteration

On the positive side, we could observe that at least for the test cases where only one router was present within a segment, the results for most of the test cases were consistent across most tested operating systems. Things have changed positively. Only minor differences could be identified.

Things start to get unpredictable as soon as a second router gets introduced into the network after the initial provision of the systems has been performed. This leads us to the conclusion that there is still room for improvement and the topic is still worth further research as already indicated in our first conclusions.

8 References

[1] S. Deering, R. Hinden, *Internet Protocol, Version 6 (IPv6) Specification*, RFC 2460, December 1998.

[2] T. Narten, E. Nordmak, W. Simpson, H. Soliman, *Neighbor Discovery for IP version 6 (IPv6)*, RFC 4861, September 2007.

[3] S. Thomson, T. Narten, T. Jinmei, *IPv6 Stateless Address Autoconfiguration*, RFC 4862, September 2007.

[4] R. Droms, J. Bound, B. Volz, T. Lemon, C. Perkins, M. Carney, *Dynamic Host Configuration Protocol for IPv6 (DHCPv6)*, RFC 3315, July 2003.

[5] J. Jeong, S. Park, L. Beloeil, S. Madanapalli, *IPv6 Router Advertisement Options of DNS Configuration*, RFC 6106, November 2010.

[6] B. Liu, S. Jiang, R.Bonica, X. Gong, W. Wang, *DHCPv6/SLAAC Address Configuration Interaction Problem Statement*, draft-ietf-v6ops-dhcpv6-slaac-problem-07, August 17, 2016.

[7] Ed. Horley, *Practical IPv6 for Windows Administrators (RFC 6106 – Ipv6 Router Advertisement Options for DNS Configuration)*, p.12, Apress; 1 edition (December 23, 2013).

[8] Technet Microsoft, *Does Win7 or W2K8 server support RFC 6106?*, January 22 2012, *https://social.technet.microsoft.com/Forums/en-US/5757980a-5983-4efc-a5f3-27687b90fe41/does-win7-or-w2k8-server-support-rfc-6106*, (retrieved in 8th March 2015).