IPv6 Maintenance (6man) Working Group
Internet Draft
Updates: 4861, 4862 (if approved)
Intended status: Standards Track
Expires: March 2021

E. Vasilenko
X. Xiao
Huawei Technologies

September 24, 2020

ND improvement to prevent Man-in-the-middle attack
draft-vasilenko-6man-nd-mitm-protection-00

Abstract

Privacy protection is the bigger and bigger concern of many governments and public in general. ND has a few open man-in-the-middle attack vectors. MITM is considered among the most dangerous attack types because of information leakage. This document proposes minimal modifications for ND to protect IPv6 nodes against still open MITM attacks. It could be implemented gradually on any nodes, with the biggest benefit from support on routers.

Status of this Memo

Copyright Notice

Table of Contents

1. Terminology and pre-requisite

Good knowledge and frequent references to [ND] is assumed. Many terms are inherited from [ND]. Additional terms are introduced:

Security DAD: Duplicated Address Detection for security check
at the time to write or rewrite for Link Layer Address

Intruder: The Node under control of malicious 3rd party

Intercepted Victim: The node that could lose the privacy of communication

Poisoned Victim: The node that could suffer an unauthorized modification of Neighbor Cache entry; depending on the scenario, it could additionally lose the privacy of communication

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here

2. Introduction

Cyber Security is one of the biggest concern for many governments: CPNI, HIPAA, FCRA, ECPA in US; GDPR in Europe; CSL in China; DPB in India; Data Protection Act 2018 in the UK. Many regulations have been refreshed or fully redeveloped in recent years.

[ND Trust Model] clearly states: "it is desirable to limit the amount of potential damage in the case a node becomes compromised. For example, it might still be acceptable that a compromised node is able to launch a denial-of-service attack, but it is undesirable if it is able to hijack existing connections or establish man-in-the-middle attacks on new connections."

The most dangerous and easy way to organize a MITM attack is based on rogue Router Advertisements. Fortunately, rogue router is easy to block on link level device by filtering ICMPv6 message type 134 ([RA-Guard]).

This document discusses MITM attacks that are more difficult to block, because it is a challenge for link layer to clarify IP address ownership in the plain (distributed) ND architecture. The Internet has open tools to exploit this vulnerability (parasite6, scapy). Attacks are explained in [RIPE-Training] and [IEEE ND Security]. Last reference has very big collection of ND security improvements proposed (50+ solutions), but the problem related to information leakage is still open.

This document proposes a simple modification for ND protocol that could be briefly explained as:

- Link Layer Address rewrite or initial write for any Neighbor Cache entry MUST be the result of only *multicast* Neighbor Solicitation. Multicast would give a chance for Poisoned Victim to understand that many nodes are going to use the same IP address.

- "Security DAD" is mandatory: it SHALL be checked that not more than expected number of NA replies have been received in the response to *multicast* NS. It means that for the majority of cases (default DuplicityLevel): no more than 1 reply.

- Any node SHOULD override neighbor cache entry on routers by sending unsolicited NA message after respective node addresses would progress to preferred state.

3. Security vulnerabilities

3.1. Rewrite by unsolicited NA

Let's consider the case when all nodes support [ND] without any extensions. Additionally, assume that Intruder gains access to one node on the link by exploiting some other vulnerability or Intruder has connected his host physically to the link.

3 nodes involved in the attack: Intruder (typically a host), Intercepted Victim (typically another host) and Poisoned Victim (typically a router).

Neighbor Cache poisoning could be organized (by the procedure below) for any node (host or router). Router is typically the more probable target for the Poisoned Victim role because Intercepted Victim has a high probability to communicate with a Router.

The algorithm for attack:

- Intruder has many ways to get Intercepted Victim IP address - see [IPv6 Reconnaissance]. The IP address could not be considered hidden information.

- Intruder starts sending non-legitimate unicast NA to Poisoned Victim with the "target IP address" of Intercepted Victim, but link layer address of the Intruder, with flags "Override" and "Solicited" set.

- If Neighbor Cache entry on Poisoned Victim for reachability to Intercepted Victim is already created (by NS or return traffic - the second one is highly probable for routers) - then the Neighbor Cache entry would be overwritten, because "Override" flag was set. Neighbor cache entry would be put to STALE state at a minimum. If a particular implementation does not track correspondence between NS and NA (not required for state machine in [ND]) then cache entry would be put directly to REACHABLE, because "Solicited" flag was set (section 7.2.5 of [ND]).

- Intercepted Victim would not see unicast NA - it would not have chance to recognize "duplicate address". Upstream traffic would flow normally from Intercepted Victim to Poisoned Victim, but downstream traffic would go from Poisoned Victim to Intruder. An Intruder could build a much bigger set of attacks on this basement, for example: it could edit DNS answers and redirect all Intercepted Victim's traffic to itself (both directions).

An intruder could send poisoning NA periodically and intercept traffic as soon as it would become available from Poisoned Victim. Initial traffic missed for Intruder could be very small (tens of milliseconds).

3.2. Be the first and suppress DAD

An Intruder could claim an IP address (and establish communication) while Intercepted Victim is disconnected from the link. Intruder just needs to suppress DAD on itself - Poisoned Victim would never understand that address duplication did happen for the basic [ND] environment.

It is a more difficult attack vector for an Intruder, because it needs to wait Intercepted Victim reconnection to the link.
It would be difficult to push the router to reinitialize the link.
It is potentially possible to influence the host to re-initialize the link by DoS, social engineering or other vulnerability. It is definitely possible to wait till host or router would reload after upgrade (predicted event for host).
This attack vector does not make sense for the Intruder under the case of availability attack vector #1. But it could become next primary attack vector if Intruder would be prevented from attack vector #1. Hence, solution should block it too.

3.3. Win the race just after DAD

The Intruder could be silent and monitor DAD. As soon as the Intruder would see DAD - it would have RetransTimer (1sec by default) to persuade the Poisoned Victim that it is the only one on the link that claims this IP address. The Intruder could pass any multicast or unicast check, because Intercepted Victim would be silent inside the DAD procedure for the basic [ND] environment.

It is a more difficult attack vector for an Intruder, but could become primary if other attack vectors would be blocked. See more detailed explanation in previous section.

3.4. Implications for off-link nodes

All 3 attack vectors above have additional strong vulnerability consequence that make sense to discuss separately: Intruder impersonates Intercepted Victim for traffic from off-link (for connections initiated from any node behind the router). It is the strongest form of 2-way Man-in-the-middle attack.
It is not very important for ordinary hosts, because not many remote nodes would try to contact Intercepted Victim. There is the possibility for unlucky exception when admin would come to Intercepted Victim for remote monitoring, then Intruder could ask him for password hash. Admin password is extremely valuable for the Intruder.
It is really big leakage of information if Intruder impersonates some important corporate server. Many users of this company would connect to the Intruder. The majority of internal Enterprise applications do not have mutual authentication properly activated. The Intruder would be capable to collect many passwords of this company. It is easy for Intruder to emulate the original application because it could proxy user requests to Intercepted Victim (on the same link) and get proper responses that fully satisfy users. Particular application could have additional vulnerabilities that is easy to exploit in the situation when Intruder is on the server side.

3.5. Speed up by [Gratuitous ND]

This is not separate attack vector. It is just a little improvement for the Intruder for attack vectors #1.

Let's assume that [Gratuitous ND] is implemented at least on Poisoned Victim.

[Gratuitous ND] has changed 1st paragraph of [ND] section 7.2.5 - it permits creation of new Neighbor Cache entry on Poisoned Victim, including the situation when there was no NS before and no record has been created in any state.

Let's assume that garbage collector deleted stale record for seldom speaking node. Then consider situation of attack vector #1. Basic [ND] pushes the Intruder to send unsolicited NA frequently (up to the rate limit on the Poisoned Victim). In contrast, [Gratuitous ND] permits to create the neighbor cache record by one message much in advance of traffic flow. As a result, Intruder could get all traffic from Poisoned Victim to Intercepted Victim, no any packet would be missed. The improvement for the Intruder is very small - just tens of milliseconds of additional traffic would be intercepted. Attack vector is very effective in both cases.

Let's look to the attack vector #2. Intruder had plenty of time to establish communication by normal traffic flow. No additional value from [Gratuitous ND].

Let's look to the attack vector #3. Intruder have RetransTimer (1sec by default) to create a record on Poisoned Victim. Normally generated traffic should be fast enough. No additional value from [Gratuitous ND] again.

4. Solution - Security DAD

The general idea is very easy to explain:

o Link Layer Address rewrite or initial write for any Neighbor Cache entry MUST be the result of only *multicast* Neighbor Solicitation. Multicast would give a chance for Poisoned Victim to understand that many nodes are going to use the same IP address.

o "Security DAD" is mandatory: it SHALL be checked that not more than expected number of NA replies have been received in the response to *multicast* NS. It means that for the majority of cases (default DuplicityLevel): no more than 1 reply.

o Only fastest NA response to multicast NS SHOULD be used for write or rewrite of LLA. This rule would be useful only for the case of DuplicityLevel more than 1 (not default configuration). It is needed to support many anycast address on the same link.

- o Any node SHOULD override neighbor cache entry on routers by sending unsolicited NA message after respective node addresses would progress to preferred state.

The implementation is more complex, because it needs to be put into [ND] context - see next sections.

4.1. Standards modifications

4.1.1. Modifications to [ND]

* Section 5.1 and section 7.3.2 of [ND],
add new state of Neighbor Cache entry:

DUPLICATE - Target address has been put into dampening state for the time defined by DuplicateTimer, because SDAD (additional security check) has found duplicate address. The Neighbor Cache entry MUST not be used for traffic forwarding. All NS and NA for this target address SHALL be ignored. DUPLICATE entry SHOULD be deleted when DuplicateTimer expire.

* Sections 6.2.1 and 6.3.2 of [ND],
add to router and host variables:

DuplicateTimer - measured in seconds, dampening time for duplicate IP address discovered by SDAD procedure. It SHOULD be copied from "Reachable Time" advertised by router (on the router itself, it is known as AdvReachableTime - section 6.2.1 of [ND]). Future versions of [ND] should be extended to advertise this timer from router in RA messages.

* Sections 6.2.1 and 6.3.1 of [ND],
add to router and host configurable variables:

DuplicityLevel - number of NA responses that is accepted as legitimate for multicast NS solicitation. It permits to proper handling of anycast addresses present on the link if the maximum number of anycast addresses is known in advance. The node MUST have the capability to provision DuplicityLevel on a per-link granularity. 32-bit unsigned integer SHOULD be reserved for this counter.

- * Section 10 of [ND], add to Node constants:

DuplicityLevel 1 (no duplicate addresses permitted)

- * Section 5.3 (Garbage Collection) of [ND], add at the end:

Neighbor Cache entries with DuplicateTimer expired SHOULD be deleted. It is possible to delete entries before timer expiration, if there is a need to free space for new entries.

- * Section 7.2.2 (Sending NS) of [ND], add at the end:

Prepare state machine on every transmission of multicast NS:

o Initialize DUPLICITY counter by DuplicityLevel

o Restart RetransTimer of this Neighbor Cache entry

Do not change RetransTimer for unicast NS - reuse expired one.

- * Section 7.2.5 (Receipt NA) of [ND], replace at the beginning:

This modification is equivalent to [Gratuitous ND]. You could have this change already if [Gratuitous ND] is implemented on your router.

OLD TEXT:

When a valid Neighbor Advertisement is received (either solicited or unsolicited), the Neighbor Cache is searched for the target's entry. If no entry exists, the advertisement SHOULD be silently discarded. There is no need to create an entry if none exists, since the recipient has apparently not initiated any communication with the target.

NEW TEXT:

When a valid Neighbor Advertisement is received (either solicited or unsolicited), check first for the target's IP address entry in the Neighbor Cache.

If no entry exists on the host, it SHOULD silently discard this advertisement. There is no need to create an entry if none exists, since the recipient has apparently not initiated any communication with the target.

If no entry exists on the router, it should additionally check for Target link-layer address option.
If the received Neighbor Advertisement does not contain the Target link-layer address option, then the router SHOULD silently discard this advertisement. Else router SHOULD create a new entry for the target address with the link-layer address set to the Target link-layer address option. The neighbor cache entry state MUST be set to STALE.

* Section 7.2.5 (Receipt NA) of [ND], add at the end:

I. If NA received is inside RetransTimer window, then it is considered as a response to NS (solicited):

o If DUPLICITY counter does not yet reach 0: Process NA according to all other rules of this section, except write or rewrite of link layer addresses is permitted only if DUPLICITY is equal to DuplicityLevel (only fastest anycast is used for neighbor reachability cache). Additionally, decrement DUPLICITY.

o If DUPLICITY counter is 0: Change Neighbor Cache entry to DUPLICATE state. Clear all packets in the queue related to this target address. Log duplicate address event with target address and both Link Layer addresses (from received NA and from cache entry). Start DuplicateTimer for this Neighbor Cache entry.

II. If RetransTimer is expired or Neighbor Cache entry does not exist yet for this target address then NA is considered unsolicited disregard of any flags set. Process NA according to all other rules of this section, except additional check: if processing would need to initial write or rewrite Link Level Address (i.e. any change of LLA) then change Neighbor Cache entry state to INCOMPLETE and initiate *multicast* Neighbor Solicitation process against this target IP address.

* Section 7.2.6 (Sending unsolicited NA) of [ND], add at the end:

A node SHOULD send Neighbor Advertisement message with Override flag set to the all-routers multicast address after respective address would change state to preferred.

* It MAY be added after 4th paragraph of section 5.2 (Conceptual Sending Algorithm) of [ND]:

Traffic resolved to be sent through Neighbor Cache entry in DUPLICATE state MUST be dropped.

4.1.2. Modifications to [SLAAC]

* Neighbor Cache maintenance is pretty much restricted to [ND]. Hence, it is not mandatory to change [SLAAC] - the latter is more concerned on initialization and DAD. One optimization is possible specifically for [Gratuitous ND] environment. It is stated in paragraph 4 of section 5.4.2 [SLAAC] that messages sent in response to multicast announcements should be delayed for random time between 0 and MAX_RTR_SOLICITATION_DELAY, but the type of multicast message is mentioned in clear "router advertisement message". It MAY be reasonable to replace it to "any message", because [Gratuitous ND] does send unsolicited NA to all-routers multicast and this document proposes response by NS that could synchronize from all routers.

4.2. Interoperability analysis

Proposed solution does not need any additional functionality on link layer technology (no layer 2 features needed).

It does not block unsolicited NA - Link Local Addresses write or rewrite is still possible, including compatibility to [Gratuitous ND].

[Optimistic DAD] creates possibility for Intruder to intercept RetransTimer (1sec by default) of traffic in the case of attack vector #2 and #3, because optimistic address could transmit traffic without the possibility to override Poisoned Victim neighbor cache - NA should have override flag cleared. [Gratuitous ND] does not change this rule for the good reason: it would be not a good idea to strongly claim address that has not passed DAD check yet.

Hence, SDAD would not be activated up to the time the address would progress to preferred, that initiate NA with override flag set. You could disable [Optimistic DAD] if your concern about traffic intercepted for RetransTimer seconds is bigger than your concern of interface initialization delay for the same time.

[Gratuitous ND] operates as intended in the combination with [Optimistic DAD]. [Gratuitous ND] is effectively replaced by this document for the case when address is progressed to preferred, because stronger form of unsolicited NA (with override flag set) is proposed.

It should be no problem for multi-prefix and multi-homing environment discussed in [Multi-Homing], because Neighbor Cache population does not influence forwarding directions (Destination Cache does). [Gratuitous ND] could create unnecessary states on routers advertising address spaces from different ISP for the case of Provider-aggregatable address space. Mechanism specified in this document would confirm it to REACHABLE state, then it would be probably not used later (freeze in STALE state up to garbage collection). It is very small deficiency originated by [Gratuitous ND] that does not make sense to fix.

This document does not create any problem for standard update proposed in [Subnet Model], because determination of "on-link" addresses is on Destination Cache level that Neighbor Cache is not capable to influence.

[FCFS SAVI] deep security assistance from L2 switches would not benefit from this document - it is not possible to poison cache in SAVI architecture. It is important to mention that this document would not create any interoperability issue with [FCFS SAVI]. There would be no illegal write or rewrite requests (with duplicate addresses) that SDAD needs to check.

The same consequence is under SeND presence: cryptographically protected addresses does not need additional protection, but again current document would not create any interoperability problem by additional Security DAD check.

DHCP support is not affected by this document - it is irrelevant how node has got an IP address. Furthermore, it is not important whether it has been done in a legal way - if Intruders would try to hijack traffic of each other - they would be both blocked, because they would create duplicate addresses.

[NUD improvement] has the natural synergy with current document, because it converts unicast NS to multicast NS (when neighbor cache state would change to UNREACHABLE) that permits to save on additional multicast NS for SDAD check.

SDAD (as well as ordinary DAD) is not fully compatible to anycast IP address. DuplicityLevel permits to properly handle anycast addresses present on a link if the maximum number of anycast addresses is known in advance (see DuplicityLevel variable).

Active-Standby clustering solutions (including VRRP) should not be affected by this standard extension, because properly functioning cluster should respond to target IP only from one LLA. If any concern on cluster behavior exist - SDAD could be effectively disabled by setting bigger number for DuplicityLevel.

The document proposes minimal changes for ND protocol without changing the basic principles. Moreover, changes in nodes behavior are local - it is not needed to wait for formal standard update - new behavior would be fully compliant to [ND] and [SLAAC], as well as all other extensions mentioned above in this section.

5. Applicability analysis

5.1. Performance analysis

[Gratuitous ND] cache initialization improvement is not affected by this document, because algorithm above assumes immediate check, no need to wait for return traffic to be buffered.

Additional Security DAD (with associated multicast solicitation) should happen only for Link Layer Address write or rewrite that ideally should happen only one time for every address. Control Plane load should not be considerably increased. Moreover, in some cases SDAD check coincide with multicast NS that results in no additional messages. See later more detailed analysis of multicast performance for 3 primary scenarios. It has been analyzed per 1 address space, other Global Unicast Addresses or Unique Local Address would have exactly the same calculations.
LLA would generate additional multicast traffic that is omitted in calculations.

A reminder: according to section 5.1 of [SLAAC] - ordinary DAD has 1 multicast check by default (DupAddrDetectTransmits is 1).

We could assume that the host would not wait scheduled RA, it would probably initiate RS that would generate multicast RA.

1. Host to host communication (general scenario).
   The basic neighbor reachability detection (by [RFC 4861]) should use one multicast for DAD and one multicast solicitation to discover other node. It is not important for performance analysis that only originating node would check 2-way communication (and would promote cache entry to REACHABLE) as a result of initial multicast solicitation, because other node would record proper LLA too - it would use unicast solicitation later to prove 2-way communication. Hence, this document would generate an additional multicast request per every pair of communicating nodes. Depending on the mesh communication richness between nodes - it could increase multicast traffic from 0 up to 100% (depends on the number of communicating pairs): (2*c*(c-1)+h)/(c*(c-1)+h).

2. Router cache population in the basic [ND] scenario.
   Initial DAD is omitted here, because it was calculated in host-to-host scenario. The host would generate multicast RS and would receive multicast RA. Multicast NS is assumed as a result of one router receiving traffic to a particular host. This document reuses any multicast NS for SDAD, hence performance is not affected: (h*(2*r+1))/(h*(2*r+1))=1.

3. Router cache population in the scenario of [Gratuitous ND].
   Initial DAD is omitted here, because it was calculated in host-to-host scenario. The host would generate multicast RS and would receive multicast RA. One multicast unsolicited NA is proposed by [Gratuitous ND], it could not be used for SDAD, because it is going to different multicast group (routers). Hence, this document creates 33% more multicast traffic for 1 router on this link, 40% more multicast for 2 routers on this link, up to 50% more multicast for many routers: (h*(2*r+1+r))/(h*(2*r+1)).

It makes sense to remind, that normal neighbor cache entry refreshment should not generate additional multicast, because entry refreshment is assumed in unicast to known MAC address - see PROBE definition in [ND] section 5.1. The Garbage collector may completely delete long stale neighbor cache entry (good example of such host is a printer), then multicast would be needed anyway - this document does not increase multicast for that case.

5.2. Usability analysis

Host's Unsolicited NA (after address would go to preferred state) is proposed only in the direction of all-routers multicast (router to host communication). It would invoke SDAD check and reveal duplicity.
Attack vector #2 has been left unprotected by this document for the case of host to host communication. MITM for traffic inside link is still possible, because SDAD check is useless at the time when Intercepted Victim is not connected yet to the link, but later would be no rewrite request that would not initiate SDAD check.

Some technologies are prone to multicast loss. The mechanism in this document may not give security protection as intended if multicast NS would be lost only in the direction of Intercepted Victim - SDAD would not detect address duplicity.

The Intruder could potentially answer to multicast NS a few milliseconds faster than the Intercepted Victim. Poisoned Victim could update neighbor cache in the favor of Intruder and release the small portion of the traffic before Poisoned Victim would receive excessive number of NA responses and block both (Intruder and Intercepted Victim).

Security solution discussed in this document does not need any coordination on introduction in production. It could be done in any order or ad-hoc: it is not restricted which one particular node (router or host) would start additional SDAD check first.

There are 2 primary use cases envisioned (with the 3rd mixing both):

o Campus environment: the majority of traffic are going to default router on every link. The intruder would have the priority to poison router cache. Hence, router protection is priority

o Data Center environment: the big proportion of the East-West traffic. Hosts protection has bigger priority for this use case

DuplicityLevel is the common information for the link - it is better to advertise it from router in the future versions of ND. Current design choice (keep this information local) is justified to simplify transition - possibility to introduce SDAD support on any one node or node sets in any order.

5.3. DoS level analysis

MITM attack vector #1 was very effective and easy to organize, hence it was very probable. It has been converted by this document into the same probable DoS (Intercepted Victim and Intruder - both blocked). It could be the temptation to minimize Intercepted Victim disruption by blocking only the second node (first-come/first-serve approach). The Intruder would be the second in the majority of cases that would minimize disruption for Intercepted Victim. Unfortunately, there are less probable cases when Intruder would be capable of catching server with popular application in reload or upgrade state - then legal node would be blocked. The Intruder would be capable of pretending to be the particular application and start collection of valuable information from users that would visit this application. See section 3.4 (Implications for off-link nodes) for details. It is the leakage of information again. The leakage of information is considered much more serious security problem than DoS, hence such conversion of DoS back into leakage is not acceptable. Vulnerability harm is more important than the probability of vulnerability. It is especially true for [ND] where we have other DoS vulnerabilities anyway.

Only cryptographically based authentication does permit to really minimize DoS disruption for legal node.

6. Security Considerations

This document describes the conversion of most common MITM attack vectors into less dangerous DoS. It does not introduce new vulnerabilities.

One corner case is left unprotected by this document:
attack vector #2 for the situation when all involved nodes are hosts and at the same time Intercepted Victim has been connected to the link later after Intruder.

7. IANA Considerations

This document has no any request to IANA.

8. References

8.1. Normative References

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

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

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

[SLAAC] Thomson, S., Narten, T., and T. Jinmei, "IPv6 Stateless Address Autoconfiguration", RFC 4862, DOI 10.17487/RFC4862, September 2007, <https://www.rfc-editor.org/info/rfc4862>.

[Gratuitous ND] Linkova, J., "Gratuitous Neighbor Discovery: Creating Neighbor Cache Entries on First-Hop Routers", draft-ietf-6man-grand-01 (work in progress), July 2020.

[Optimistic DAD] N. Moore, "Optimistic Duplicate Address Detection (DAD) for IPv6", RFC 4429, DOI 10.17487/RFC4429, April 2006, <https://www.rfc-editor.org/info/rfc4429>.

[Multi-Homing] F. Baker, B. Carpenter, "First-Hop Router Selection by Hosts in a Multi-Prefix Network", RFC 8028, DOI 10.17487/RFC8028, November 2016, <https://www.rfc-editor.org/info/rfc8028>.

[ND Trust Model] P. Nikander, J. Kempf, E. Nordmark, "IPv6 Neighbor Discovery (ND) Trust Models and Threats", RFC 3756, DOI 10.17487/RFC3756, May 2004, <https://www.rfc-editor.org/info/rfc3756>.

[NUD improvement] E. Nordmark, I. Gashinsky, "Neighbor Unreachability Detection Is Too Impatient", RFC 7048, DOI 10.17487/RFC7048, July 2010, <https://www.rfc-editor.org/info/rfc7048>.

[Subnet Model] H. Singh, W. Beebee, E. Nordmark, "IPv6 Subnet Model: The Relationship between Links and Subnet Prefixes", RFC 5942, DOI 10.17487/RFC5942, July 2010, <https://www.rfc-editor.org/info/rfc5942>.

[FCFS SAVI] E. Nordmark, M. Bagnulo, E. Levy-Abegnoli, "FCFS SAVI: First-Come, First-Served Source Address Validation Improvement for Locally Assigned IPv6 Addresses", RFC 6620, DOI 10.17487/RFC6620, May 2012, <https://www.rfc-editor.org/info/rfc6620>.

8.2. Informative References

[RFC8200] S. Deering, R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 8200, DOI 10.17487/RFC8200, July 2017, <https://www.rfc-editor.org/info/rfc8200>.

[RIPE-Training] RIPE NCC, "IPv6 Security Training", February 2019, <https://www.ripe.net/support/training/material/ipv6-security/ipv6security-slides.pdf>.

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

[IPv6 Reconnaissance] F. Gont, T. Chown, "Network Reconnaissance in IPv6 Networks", RFC 7707, DOI 0.17487/RFC7707, March 2016, <https://www.rfc-editor.org/info/rfc7707>.

[IEEE ND Security] Amjed Sid Ahmed Mohamed Sid Ahmed, Rosilah Hassan, Nor Effendy Othman, "IPv6 Neighbor Discovery Protocol Specifications, Threats and Countermeasures: A Survey", DOI 10.1109/ACCESS.2017.2737524, August 2017, <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8022867>.

9. Acknowledgments

Thanks to 6man working group for problem discussion

Authors' Addresses

Eduard Vasilenko
Huawei Technologies
17/4 Krylatskaya st, Moscow, Russia 121614

Email: vasilenko.eduard@huawei.com

Xiao Xipeng
Huawei Technologies
205 Hansaallee, 40549 Dusseldorf, Germany

Email: xipengxiao@huawei.com