UNIVERSITEIT VAN AMSTERDAM

Master System and Network Engineering

Research Project

\- - -

Security of IPv6 and DNSSEC for penetration testers

\- - -

Author:
Vesselin Hadjitodorov

Supervisor:
J.A. (Jaap) van Ginkel
Institute of Informatics at University of Amsterdam

July 2011

Abstract

The wide adoption of DNSSEC and IPv6 by the general public is only a question of time, because of vulnerabilities found in DNS and the depletion of IPv4 addresses. Currently both DNSSEC and IPv6 are deployed mostly for testing purposes and few organizations and end users use them in actual production environments. These relatively new technologies started a lot of discussions. Most of the discussions are focused on their deployment, rather than the security issues, which DNSSEC and IPv6 can introduce.

Since the moment when DNSSEC and IPv6 are deployed widely is getting closer, I decided to research the security of these two protocols. The overall purpose of this research is to make a summary of the known security issues of DNSSEC and IPv6, methods and tools that can be used for detection of these issues.

This research concludes that the DNSSEC protocol itself can be considered secure and most of the known vulnerabilities are due to poor implementations rather than flaws in the design of the protocol. The research covers the DNS Denial of Service (DoS) amplification attack, the DNSSEC zone walking and implementation specific issues.

Based on the experience from my research I consider the security of IPv6 and IPv4 comparable. The two protocols have several common features and mechanisms, but they are also different in several aspects. Security issues, which are IPv6 specific, are described in this report.

Enumeration of IPv6 hosts can be a more challenging task for a penetration tester compared to enumeration of IPv4 hosts, due to the larger search space in IPv6. Still there are feasible methods for IPv6 host enumeration, which are described in the report.

During the research I did not encounter new vulnerabilities in DNSSEC and IPv6.

Acknowledgments

I would like to thank J.A. (Jaap) van Ginkel from the Institute of Informatics at University of Amsterdam for his assistance, feedback and tips during this research project.

Contents

1 Introduction

The goal of this project is to provide overview of known security issues in DNSSEC and IPv6 and techniques that can be used by penetration testers to detect them. The main research question is:

What are the security issues of DNSSEC and IPv6 and how to perform penetration tests in order to identify them?

The main research question has been divided into sub questions in order to put focus on specific topics:

1. What are the known security issues for DNSSEC and IPv6?

1.1. Are these new issues, or are they based on vulnerabilities of the old technologies?

1.2. Are there security issues during the transition period, caused when the old and new technologies are used in parallel?

2. How can the identified security issues be mitigated?

3. How can a penetration tester check for these known security issues?

3.1. How can these security issues be recognized?

3.2. What tooling can be used for performing the penetration tests?

3.3. How to perform tests on the large IPv6 scopes?

4. What not yet discovered protocol insecurities was I able to identify ?

This report is divided into a DNSSEC part and an IPv6 part. Each part covers the following topics regarding the two protocols:

- Known security issues
- Detection of security issues
- Mitigation of security issues
- Techniques and tools for penetration testing

At the end of each part there is a summary of the most important findings.

This report does not cover all possible vulnerabilities related to the protocols. It includes problems, which came to my attention during the research. Certain vulnerabilities might not be documented or well known and thus are not included in the report.

1.1 Approach

The activities on this research were divided into theoretical and practical parts.

The research began with familiarization with DNSSEC and IPv6. Information, regarding the two researched protocols, was collected using results from Internet search engines. The differences between IPv6 and DNSSEC compared to IPv4 and DNS were analyzed. Additionally Requests for Comments (RFCs), which provide information related to the two protocols, were studied. Few of the RFCs address possible issues and propose detection and mitigation techniques. The information from these RFCs was included in the report, because the RFCs represent the official recommendations of the Internet Engineering Task Force (IETF).

The theoretical part of the research also included study of documented vulnerabilities listed in the databases of National Vulnerability Database (NVD) and United States Computer Emergency Readiness Team (US-CERT). These databases provide an overview of specific issues, but usually do not include detailed description of the vulnerabilities. Further details about these issues were looked up using Internet search engines. Some of the found information sources (e.g. forum posts, mail list archives) were not suitable for referencing in scientific papers and were excluded from the report, although the information from them was considered. If several sources which address the same issue were found, the one with the highest quality was used as a reference.

The statistical information presented in this report was gathered from the web pages of Internet Corporation for Assigned Names and Numbers (ICANN) [1], DNSSEC Deployment Initiative [2] and European Registry for Internet Domains [3].

The practical part of the research was focused on testing techniques and tools reviewed in the report. The tools used for penetration testing, described in this report were chosen based on qualities such as DNSSEC / IPv6 capabilities, ability to detect specific issues, functionality and popularity. Although other tools with similar functionality exist, the recommended tools can perform the necessary tasks with no shortcomings. The approach and results from the tests are included in the report. Not all tools reviewed in the report were tested, because of the relatively short time period of the research.

Using the information and experience gained during the theoretical and practical parts of this research, the research questions were answered and conclusions were derived.

The time restrictions did not allow me to look for possible new undiscovered vulnerabilities. This is the reason the proposed research question "*4. What not yet discovered protocol insecurities was I able to identify* ?" to stay unanswered.

1.2 Related work

The research was based on previous work related to DNSSEC and IPv6 security. The referenced materials were chosen, because they provide detailed explanation and practical approach to the security of the researched protocols.

Suranjith Ariyapperuma and Chris J. Mitchell from the Information Security Group of the University of London published the report **Security vulnerabilities in DNS and DNSSEC** [4]. In the report they present analysis of security vulnerabilities in the two protocols and recommend improvements in their design.

Scott Hogg and Eric Vyncke are authors of the book "**IPv6 security**" (published by Cisco Press) [5]. The book describes insecurities in IPv6 networks and suggests mitigation techniques.

Marc Heuse is IT security expert in the field of IPv6. He discovered several vulnerabilities in IPv6 and presented talks regarding the security of the protocol. Marc Heuse developed the **THC IPv6 toolkit** [6]. The toolkit contains tools for scanning, testing and exploiting IPv6 vulnerabilities. The tools from the toolkit are designed for penetration testers and are licensed under GPLv3.

This report refers to several RFCs which describe possible security issues, specify or improve DNSSEC and IPv6.

Although publications and books related to the DNSSEC and IPv6 security are available, they don't analyze and evaluate the two protocols from penetration tester's point of view.

2 DNSSEC security

This section features information regarding DNSSEC related security issues, techniques and tools for detection and mitigation of those issues

DNSSEC is a security extension build on top of DNS, which is backward compatible with the existing DNS infrastructure. According to RFC4033 [7], DNSSEC adds to DNS origin authentication and data integrity by cryptographically signing of the DNS Resource Records (RRs). A DNS resolver can obtain the public key from the public / private key pair and validate the authenticity of the responses. The resolver must be configured with a trust anchor for the signed zone or for a parent of the signed zone.

Figure 1 illustrates a simplified example of DNS resolving + DNSSEC validation. DNSSEC uses the same mechanisms for resolving as DNS, but includes additional zone signing.

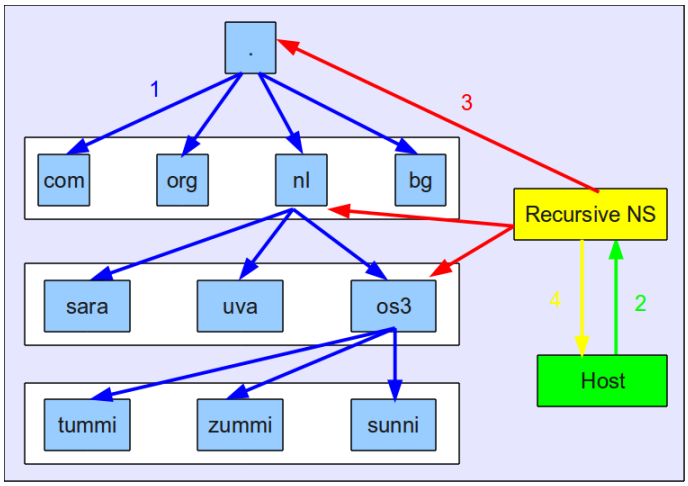

Figure 1: DNS resolving with DNSSEC

1. The **root** signs the zone containing **com**, **org**, **nl** and **bg** domains. **nl** signs the zone containing **uva**, **sara** and **os3** domains. **os3** signs the zone containing **tummi**, **zummi** and **sunni** domains.

2. A host wants to look up **sunni.os3.nl** and sends a query to a recursive name server.

3. The recursive name server walks down the DNS tree starting from the **root** to fetch the resource records, queried by the host along with their signatures. The recursive name server could validate the responses using DNSSEC and decide if the data should be send to the host.

4. The recursive name server sends the resolved data for **sunni.os3.nl** to the host. The host can perform DNSSEC validation in order to check the data for tampering.

RFC4034 [8] defines the following resource records (RRs) used by DNSSEC:

- DNSKEY - public key
- DS - delegation signer
- RRSIG - resource record digital signature
- NSEC - authenticated denial of existence

Figure 2 illustrates the chain of trust build upon the DNSSEC RRs.

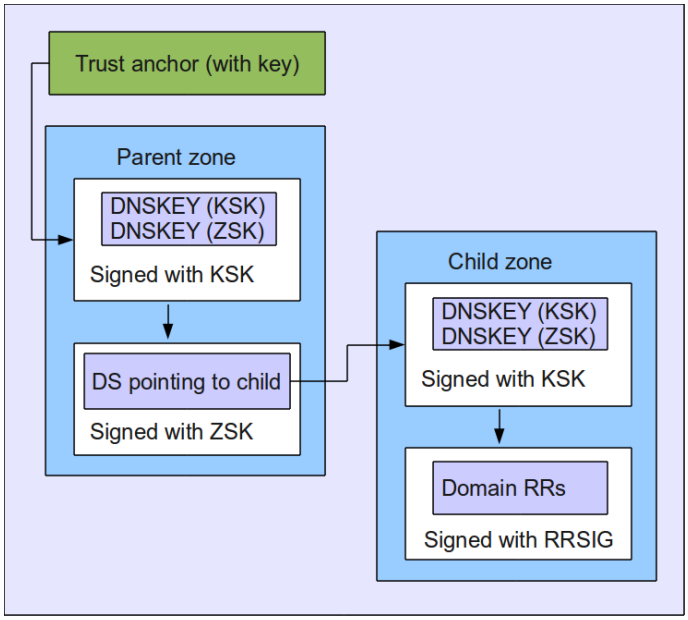

Figure 2: DNSSEC chain of trust

- The DNSKEY RRs contains Key Signing Key (KSK) and Zone Signing Key (ZSK) used for signing the keys and the zone.
- The DS RR is used to establish trusted relation between the parent and the child zone. It couples with the KSK of the child zone.
- The RRSIG RR is used for signing the rest RRs.

DNSSEC provides "authenticated denial of existence", that allows a resolver to authenticate a negative reply from a name server. A resolver can verify it by fetching a NSEC RR, which contains the next owner name in the zone (in the canonical order) that is authenticated.

2.1 DNSSEC security issues

This section covers vulnerabilities related to DNSSEC that came to my attention during the research. These DNSSEC issues are compared to DNS issues where possible. The following issues are included:

- DNSSEC DoS amplification attack
- DNSSEC zone walking
- Implementation issues of DNSSEC
- Lack of DNSSEC validation

I have proposed detection and mitigation methods for the issues, when applicable.

Issues caused by side factors such as cryptographic key generation, key storage, rollover procedures and etc. were not researched. An example of an issue that causes generation of weak cryptographic keys used by DNSSEC is described by US-CERT [9]:

> *A weakness exists in the random number generator used by the OpenSSL package included with the Debian GNU/Linux operating system and derivative systems that causes the generated numbers to be predictable. As a result of this weakness, certain encryption keys are much more common than they should be.*
> ...
> *A remote, unauthenticated attacker with minimal knowledge of the vulnerable system and the ability to conduct a brute force attack against an affected application may be able to guess secret key material.*

2.1.1 DNSSEC DoS amplification attack

DNSSEC DoS amplification attack description

"*The amplification effect in a recursive DNS attack is based on the fact that small queries can generate larger UDP packets in response*" (Randal Vaughn, Gadi Evron: DNS Amplification Attacks, 2006 [10]). An attacker can spoof a query and cause the reply from the DNS servers to be delivered to another host - the victim. The attack is illustrated on Figure 3:

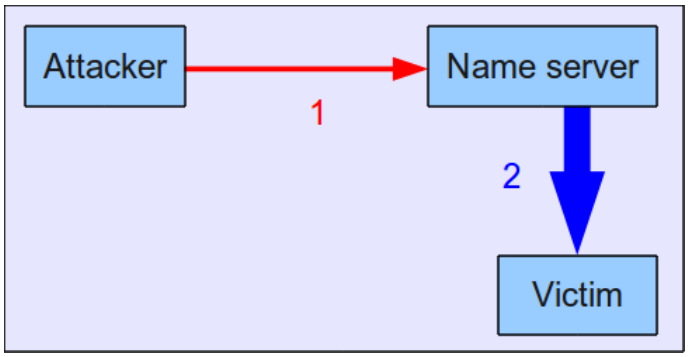

Figure 3: DNS DoS amplification attack

1. An attacker sends a query to the name server with spoofed source address, pretending to be the victim.
2. The name server sends the response to the victim, generating unexpected network load on the victim's network.

The attack can be distributed among several recursive and authoritative name servers in order to be more difficult to block compared to an attack from a single source. Although the victim can use a security policy that distinguishes and ignores the unwanted DNS traffic, still if the traffic is more than the available bandwidth before the point where it is blocked, a bottleneck can be created.

The DNS DoS amplification attack can be performed using either DNS or DNSSEC. I consider DNSSEC's responses more suitable for the attack, because of the embedded public keys and signatures in the DNSSEC resource records. The larger response size allows DNSSEC to have higher amplification ratio for this attack compared to DNS. In an open letter Dean Anderson [11] gives as an example a 126 times amplification factor attack using DNSSEC. From these figures one can predict that such an attack can have serious impact on the DNS infrastructure and the targets of the attack.

Detection of name servers vulnerable to DNSSEC DoS amplification attack

A penetration tester can send queries with spoofed source address to name servers, while monitoring the host that acts as a victim of the attack. If the attack is blocked, the name servers are protected by a security mechanism.

Mitigation of DNSSEC DoS amplification attack

I suggest recursive lookup on authoritative servers to be disabled by name server administrators. Clients can use internally available recursive name servers. In this way an attacker will not be able to abuse the recursive name servers for the DNS amplification attack.

RFC5358 [12] suggests limiting the DNS DoS amplification attacks by denying traffic from spoofed source IP addresses. Possible solution is implementing ingress filtering (as described in RFC3704 [13]) - a technique for checking if

packets originate from the networks they claim to be from. In RFC3704 the following implementation of ingress filtering are described:

- Ingress Access Lists - a filter that checks the source address of every message received on a network interface against a list of acceptable prefixes.
- Strict Reverse Path Forwarding (RPF) - the source address is looked up in the Forwarding Information Base and only packets, which would be forwarded using the source interface, are allowed.
- Feasible Path Reverse Path Forwarding - similar to strict RPF, but allows alternative routes and interfaces to be used.
- Loose Reverse Path Forwarding - similar to strict RPF, but differs in that it checks only for the existence of a route.
- Loose Reverse Path Forwarding ignoring default routes - it is an add-on on top of loose RPF, which ignores the default routes and accepts only explicit.

2.1.2 DNSSEC zone walking

The NSEC RRs were specified in RFC4034 [8] from March 2005 in order to add robust resistance against spoofing. The NSEC RR contains the owner name of the next RR set and the RR types present at the NSEC RR's owner name. The complete set of NSEC RRs in a zone indicates which RRs exist in the zone by forming a chain of consecutive domain names. "*This introduces the ability for a hostile party to enumerate all the names in a zone by following the NSEC chain*" (RFC4033 [7]). An example of three consecutive NSEC records:

```
smtp.ipv6.os3.nl.    3600    IN    NSEC
    sunni.ipv6.os3.nl.  AAAA NSEC RRSIG

sunni.ipv6.os3.nl.    3600    IN    NSEC
    tummi.ipv6.os3.nl.  AAAA NSEC RRSIG

tummi.ipv6.os3.nl.    3600    IN    NSEC
    vpnsmurf.ipv6.os3.nl.  AAAA NSEC RRSIG
```

DNS walking can be performed with *DNSSEC Walker* or a similar tool. Although the tool is a proof of concept (released in 2001), it was updated over the years and the current version performs DNSSEC zone walking without shortcomings. The tool requires as input parameters a name server address and a start name, from where the enumeration will begin. This is an example of a command, used to enumerate the os3.nl zone:

```
walker -y @ns1.os3.nl os3.nl
```

Further examples with the DNSSEC Walker can be found in the enumeration of IPv6 hosts section.

RFC5155 [15] from March 2008 specifies an extension for NSEC - "DNSSEC Hashed Authenticated Denial of Existence" (informally called "NSEC3") in order to mitigate the zone walking problem. NSEC3 records contain a cryptographically hashed value of the domain name instead of the name itself. Based

on data presented by the DNSSEC Deployment Initiative one can conclude that by the time NSEC3 was developed, several Top Level Domains (TLDs) have already implemented DNSSEC with NSEC [16]. Most of these TLDs haven't switched later to NSEC3. A report by EURid [17] from October 2010 shows that 10 of the 37 TLDs with DNSSEC use NSEC:

```
.arpa
.br
.bg
.biz
.pr
.se
.th
.us
.xnfzc2c9e2c (Sri Lanka)
.xnxkc2al3hye2a (Sri Lanka)
```

The remaining 27 TLDs have implemented NSEC3.

The scope of NSEC (NSEC3) RRs extends only to the zone, where they are located. Thus it is possible a child zone to have a different implementation from its parent zone. This allows the use of NSEC3 for a DNSSEC signed domain located in domain with NSEC and vice versa.

The NSEC3 RR indicates which hash function and salt were used to construct the hash and how many iterations of the hash function were performed over the original owner name. The salt is appended to the original owner name before hashing in order to defend against pre-calculated dictionary attacks.

An example of NSEC3 RR:

```
b4um86eghhds6nea196smvmlo4ors995.example. NSEC3 1 1 12 aabbccdd
    gjeqe526plbf1g8mklp59enfd789njgi MX RRSIG
```

The owner name for the NSEC3 RR is the base32 encoding of the hashed owner name prepended as a single label to the name of the zone [15]. It is followed by TTL, RR type and flags field. After the flags field is located the number of iterations, salt length and salt value, represented as a sequence of case-insensitive hexadecimal digits. The salt is followed by the Next Hashed Owner Name. Unlike the owner name of the NSEC3 RR, the value of the Next Hashed Owner Name field does not contain the appended zone name.

However the zone walking is not limited only for NSEC RRs. D. J. Bernstein [18] suggest that the hashes contained in NSEC3 RRs (in most cases SHA-1 as defined in RFC5155 [15]) can be cracked and used in the same way as the URLs in NSEC RR. According to him cracking of NSEC3 hashes can scale better compared to online brute-forcing with DNS queries. Online brute-forcing requires querying a name server for every attempt. Cracking hashes requires the same number of queries as the number of domains in a zone. After a NSEC3 RR is fetched, its hash can be cracked offline using CPUs, GPUs, FPGAs or cloud computing. Depending on the utilized computation power, this approach can be faster in order of magnitudes compared to online brute-forcing, in which the performance and connectivity of the name server can be a bottleneck for online

attacks. Bernstein performed tests on a cluster equipped with 9 x 2.4GHz Core 2 Quad CPUs. The cluster generated 5,800,000,000,000 hash guesses per day (with 2 hash iteration). According to him, similar performance is possible with a single Nvidia GeForce GTX 295 GPU [18].

The zone walking can be performed only in DNSSEC signed zones. A penetration tester can check whether NSEC RRs or NSEC3 RRs are used in a DNSSEC signed zone by querying the name server of the zone for these resource records. A domain can be queried for NSEC RR using a simple tool such as *dig*. An example for querying the os3.nl domain:

```
dig -t NSEC os3.nl.
```

The DNSSEC zone walking allows enumeration of domains, but I don't consider this being a vulnerability. "*It is part of the design philosophy of the DNS that the data in it is public and that the DNS gives the same answers to all inquirers*" (RFC2535 [19]). One can consider the DNSSEC zone walking an issue only for organizations that rely on hiding their hosts rather than securing them. Hiding hosts doesn't increase the security, because an attacker can use alternative techniques for host enumeration.

2.1.3 Implementation issues of DNSSEC

Based on the vulnerabilities listed in the databases of NVD and US-CERT one can conclude that most of the vulnerabilities related to DNSSEC are bugs in the implementations, rather than vulnerabilities of the protocol design.

The known DNSSEC implementation problems can result in:

- Cache-poisoning
- DoS of the DNS server

Bugs in ISC BIND 9.0.x listed in NVD [20] [21] can allow a remote attacker to conduct DNS cache poisoning attacks. A vulnerable name server with DNSSEC support may add unauthenticated records to its cache, received during the resolution of a recursive client query.

Crafted DNS packets and special queries are the main cause of name server DoS.

Both cache-poisoning and DoS issues were present in DNS, before the introduction of DNSSEC. A penetration tester can detect them by identifying vulnerable versions of the DNS servers using security scanning tool such as *Nessus*. *Nessus* is able to detect also opened recursive DNS resolvers and misconfigurations in the name server. *Nessus* has an open source alternative - *OpenVAS*, which started as a fork of *Nessus*.

Updating or patching to a non-vulnerable version usually mitigates implementation specific problems. When a problem is not fixed by the vendor it is necessary for the administrator to look for mitigation techniques adapted for the specific issue.

2.1.4 Lack of DNSSEC validation

Without support on the client side, DNSSEC validation can be performed by recursive DNS resolvers. The resolved addresses can be either send to the client or dropped if they don't validate. According to The Number Resource Organization [22], when validation is done by a recursive resolver, the resolver has to be compatible and configured to work with DNSSEC.

Kevin Murphy commented on the lack of native DNSSEC support of the browsers Microsoft Internet Explorer, Mozilla Firefox, Opera, Safari and Google Chrome in an article [23] from July 2010. Browser plugins are required for performing DNSSEC validation on the client side. In practice few users, who are aware of the DNS issues, install such plugins. Thus it is necessary that DNSSEC validation is integrated in the browsers by default and users are notified when it fails.

A penetration tester can perform DNSSEC validation with the command line using the tool *dig*. *dig* can do both "bottom-up" and "top-down" DNS validation. The top-down validation starts from the root towards the domain, while bottom-up validation starts from the domain towards the root.

An example of bottom-up validation with Dig:

```
dig +sigchase +trusted-key=root.keys www.os3.nl. A
```

An example of top-down validation with Dig:

```
dig +sigchase +topdown +trusted-key=root.keys www.os3.nl. A
```

Both methods require a file containing the root key. The key can be obtained with Dig and saved in a file:

```
dig +multiline nl. DNSKEY
```

I consider the lack of client side DNSSEC validation as one of the main reasons for the slow deployed of DNSSEC on end-user domains. Though the path between name servers and recursive resolvers could be secured with DNSSEC, if the client's stub resolver does not validate the DNS data, the client is still vulnerable to tampering. When DNSSEC validation is enabled by default in major browsers and other software products relying on DNS, the deployment speed is most likely going to improve, because of the fully secured path between the client and the authoritative name servers.

2.2 DNSSEC summary

During this research I did not encounter insecurities in the DNSSEC protocol itself. The design of DNSSEC allows an amplification attacks and possibility for zone enumeration known as zone walking. Although an extension, which should mitigate zone walking was developed, zone walking is still feasible by brute-forcing the NSEC3 hashes. Specific name server implementations can be vulnerable to cache-poisoning and DoS attacks. Both the amplification attack and implementation vulnerabilities were present before the introduction of DNSSEC.

DNSSEC is not natively supported by the popular web browsers. Clients need to install browser plugins or validation tools in order to be protected by

DNSSEC. Apart from that, recursive resolvers have to be updated and configured to understand DNSSEC. The lack of DNSSEC validation can also slows down the deployment of the protocol.

The tools *dig*, *DNSSEC Walker* and *Nessus* were found useful for performing penetration tests on DNSSEC enabled name servers.

3 IPv6 security

This section is divided into two parts. The first one features IPv6 related security issues. The second part is overview of techniques that can be used for enumeration of IPv6 hosts.

IPv6 was developed by the IETF to deal with the expected IPv4 address exhaustion. IPv6 was specified in RFC1883 [24] from December 1995, though it was updated over the years. While IPv4 allows 32 bits for an IP address, IPv6 uses 128-bit addresses, which provides 2^{96} times more address space. The standard IPv6 subnet size has been fixed to 64 bits, thus allowing the MAC address to be embedded in the rest 64 bits forming the host identifier.

Although one can view IPv6 as an extension of IPv4, the two protocols are incompatible. Most transport and application-layer protocols can function over IPv6 networks without any need of modifications. IPv6 is designed with simpler packet header format compared to IPv4 to minimize processing by routers. The rarely used header fields are moved to a separate optional header extensions, while the checksum was removed and the TTL was renamed to Hop Limit.

IPv6 does not provide broadcasting known from IPv4, where packets are delivered to all hosts on an attached link. It has been superseded by multicasting as specified in the "*IP Version 6 Addressing Architecture*" memo (latest version is RFC4291 [25]). Packets sent to a multicast address are received by all hosts, which are part of the multicast group.

The specifications of IPv6 mandate the support of IPsec - an end-to-end network security mechanisms, which provides encryption and authentication of IP traffic. IPv6 enabled hosts can auto-configure, when connected to a routed IPv6 network using the Neighbor Discovery Protocol (NDP).

3.1 IPv6 security issues

This section covers IPv6 related security issues, which came to my attention during the research. The following issues are included:

- Neighbor Discovery Protocol issues
  - Neighbor Solicitation / Neighbor Advertisement spoofing
  - Redirect spoofing
  - Router Solicitation / Router Advertisement spoofing
  - Duplicate Address Detection attack
  - Neighbor Advertisement flooding
  - Router Advertisement flooding
- IPv6 smurfing
- Routing header type 0
- Implementation issues of IPv6
- Transition techniques issues
  - Dual-stack network issues

– Tunneled IPv6 network issues
– Low user and administration awareness of IPv6 autoconfiguration

In this report I have proposed detection and mitigation methods for the issues, when applicable. Along with some of the covered issues I have included examples using the *THC IPv6 toolkit*.

3.1.1 Neighbor Discovery Protocol issues

RFC3756 [26] describes the IPv6 NDP as a mechanism used by nodes in an IPv6 network to learn the local topology. This includes the IP to MAC address mappings for the local nodes, the IP and MAC addresses of the routers present in the local network, and the routing prefixes served by the local routers. NDP uses five Internet Control Message Protocol version 6 (ICMPv6) packet types:

- Neighbor Solicitation (NS)
- Neighbor Advertisement (NA)
- Router Solicitation (RS)
- Router Advertisement (RA)
- Redirect

According to Scott Hogg and Eric Vyncke ("IPv6 Security" [5]) there is no authentication mechanism built into ICMPv6 and those packets can be spoofed. This is a flaw, which can allow an attacker to perform malicious activities such as traffic redirection and DoS.

NDP related issues have only local impact, because routers don't forward NDP messages. However I consider them a serious threat, because flat IPv6 networks can be much larger compared to IPv4 networks.

Details about the NDP packet types and possible attacks are described below.

Neighbor Solicitation / Neighbor Advertisement spoofing

NS / NA packets function in a similar way as ARP in IPv4. The basic mechanisms of the attack are described by US-CERT [27]:

> *After receiving a neighbor solicitation request from a system that is on-link and is using a spoofed IPv6 address as the source address, a router will create a neighbor cache entry. When this entry is made, some IPv6 implementations will create a Forwarding Information Base (FIB) entry. This FIB entry may cause the router to incorrectly forward traffic to the device that sent original spoofed neighbor solicitation request.*

I consider the NS / NA spoofing attack similar to the well known ARP spoofing. The attack is illustrated on Figure 4:

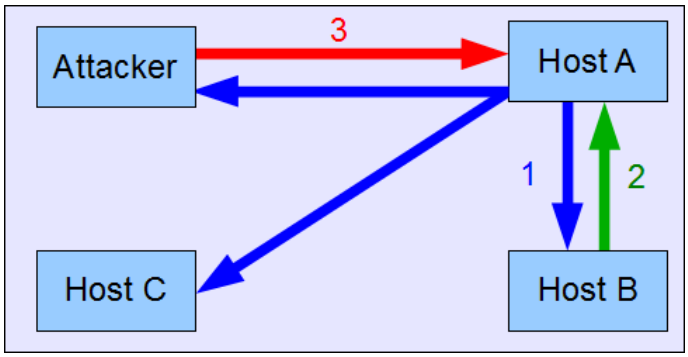

Figure 4: Neighbor Solicitation / Neighbor Advertisement spoofing

1. Host A knows the IPv6 address of host B, but doesn't know the MAC of host B needed for reaching it. Host A sends a NS message to the solicited-node multicast address corresponding to host B.

2. Host B receives the NS and sends a NA with its MAC address to host A.

3. The attacker also receives the NS and claims to be host B by sending a spoofed NA with its MAC address to host A. If at that moment host B is not on the network the attack can stay unnoticed.

I consider that NDP has improved security compared to ARP, because the NS requests are not sent to the broadcast address, but to the solicited node multicast address. The solicited node multicast address is explained by GOGO6 as follows [28]:

> *This address is created using a global multicast prefix, a "1" in the sixth hexadectet and "FF" as the most significant bits of the seventh hexadectet. The rest of the address is a duplication of the last 24 bits of the unicast address.*

Based on the above one can conclude that all hosts, which have the same last 24 bits in their unicast address, listen to the same multicast address. Because the NS request is not broadcast to everyone in the subnet, only one host out of 16,777,216 (2^{24}) is affected.

The Neighbor Solicitation / Neighbor Advertisement spoofing vulnerability has been discussed in RFC3756 [26]. The *parasite6* tool from *THC IPv6 toolkit* redirects all local traffic to the attacker's system by answering falsely to NS requests. The following command is used to perform the attack on interface eth0:

```
parasite6 eth0
```

IP forwarding should be enabled on the attacking machine or the redirected traffic can cause DoS.

Router Solicitation / Router Advertisement spoofing

IPv6 hosts can auto-configure when connected to a routed IPv6 network using Stateless Address Autoconfiguration (SLAAC). SLAAC is stateless compared to DHCP, because a DHCP server stores a state - the leased IP addresses. SLAAC is based on RS / RA messages exchanged between the router and hosts. RFC1256 [29] describes the method used for router discovery:

> *Each router periodically multicasts a Router Advertisement from each of its multicast interfaces, announcing the IP address(es) of that interface. Hosts discover the addresses of their neighboring routers simply by listening for advertisements. When a host attached to a multicast link starts up, it may multicast a Router Solicitation to ask for immediate advertisements, rather than waiting for the next periodic ones to arrive.*

RS / RA messages enable hosts to discover the existence of local routers, but don't provide data which router is a better choice for reaching a particular destination. If a host chooses not the optimal first-hop router for that destination, it should receive an NDP Redirect from the first chosen router, suggesting a better first-hop.

Some of the fields that a RA packet contains are the local network prefix, link-local address of the router and router priority. An attacker can advertise a fake router by sending spoofed RA packets. As a result a host can receive a RA for a router different from the one expected by the host or for a non existing router. The attack is illustrated on Figure 5.

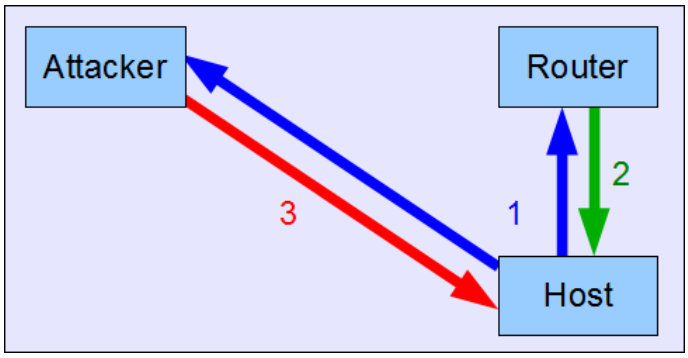

Figure 5: Router Solicitation / Router Advertisement spoofing

1. The host doesn't have a default router. In order to learn one, the host sends a RS packet.

2. The router replies to the host with a RA packet.

3. The attacker also replies and claims to be a router, with higher priority.

4. The host chooses the attacker as a default router.

One can compare this attack to a rogue DHCP server in IPv4, because of the similar outcome. The Router Solicitation / Router Advertisement spoofing attack is covered in details in RFC3971 [30].

The *THC IPv6 toolkit* contains the tool *fake_router6*, which can set any IP address as a default router, define network prefixes and DNS servers. An example of router advertisement with *fake_router6* on interface eth0:

```
fake_router6 eth0 2001:610:158:1020:226:55ff:fecd:8f84/64
```

Redirect spoofing

Redirect is an NDP mechanism used by routers to inform a host for a better route to a particular destination. Routers should detect if a host on the local network has made an inefficient first-hop routing decision and then recommend a better first-hop.

The NDP Redirect has a simple security mechanism - a copy of the packet, which caused the redirection, must be included in the NDP Redirect message. An attacker cannot blindly spoof a Redirect message, because the victim will not accept it. This can be bypassed by sending a spoofed ICMPv6 echo request to the victim, where the source IP address is set to the IP address of the router the victim is using. The victim will send an ICMPv6 echo reply to the router. The attacker can predict the content of the reply message and use it to craft an NDP Redirect message, which advertises another system as a better first-hop to the router. This attack vector is not possible in IPv4. Figures 6 and 7 illustrate the attack.

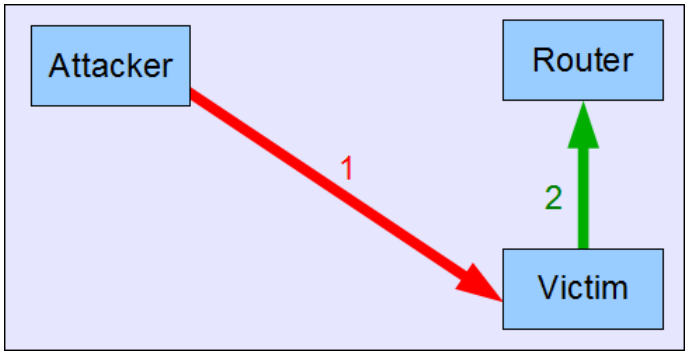

Figure 6: Redirect spoofing - The attacker sends ICMP echo request message

1. The attacker sends to the victim an ICMPv6 echo request, with a spoofed source address claiming to be originating from the router.
2. The victim sends the router an ICMPv6 echo reply.

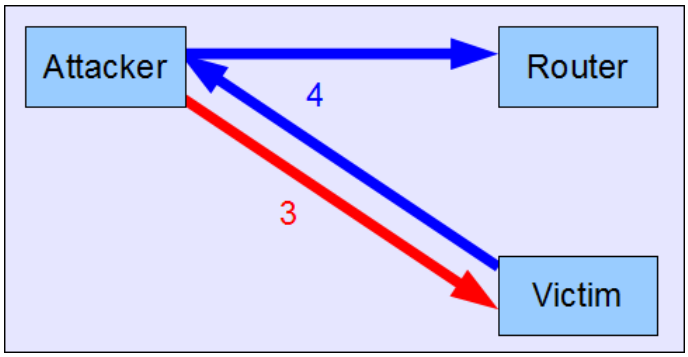

Figure 7: Redirect spoofing - The attacker sends NDP Redirect message

3. The attacker knows that the victim will reply and can use a predicted reply message to craft the NDP redirect packet, that advertises the attacker as a better route to the router.

4. Now all the traffic from the victim to the router is redirected to the attacker. The attacker can sniff the packets and forward them to the router in order to stay undetected.

The *redir6* tool from the *THC IPv6 toolkit* is an implementation of NDP Redirect spoofing. The tool accepts the following parameters:

```
redir6 <interface> <source-ip> <target-ip> <original-router>
<new-router> [new-router-mac]
```

Duplicate Address Detection attack

In IPv6 networks it is not allowed several hosts to share an IP address. To prevent duplicate IPv6 addresses, a host must check whether an IPv6 address it intends to use is free or already used by another host. This procedure is called Duplicate Address Detection (DAD). Since no higher layer traffic is allowed until a host has obtained an IP address, DAD relies on NS and NA messages in order to check if an IP address is in use.

An attacker could launch a DoS attack by responding to duplicate address detection attempts made by a newly connected host. If the attacker claims every IP address, then the host will not be able to obtain an address at all.

The DAD attack doesn't have an analogue in IPv4. The attack is described in RFC3756 [26]. *dos-new-ipv6* is the implementation of this attack in the *THC IPv6 toolkit*. The tool accepts only one parameter - the network interface:

```
dos-new-ip6 eth0
```

Neighbor Advertisement flooding

Routers can store a limited number of ND cache entries. In case a router is flooded with NA packets, the flood can result in exhaustion of the resources causing the router to crash or become slower and eventually filling up the entry table. When the table is full, a router cannot learn new ND entries or can even

cause old (legitimate) entries to be overwritten. This attack is evaluated by Jeff S. Wheeler in his presentation "IPv6 NDP Table Exhaustion Attack" [31].

The same effect as the described NA flooding can be achieved unintentionally, in case a host is using a random IPv6 address for every outgoing TCP connection as a privacy and security mechanism. This aspect was published in the draft of RFC3041 [32].

NA flooding is comparable to MAC flooding of a network switch, where the content addressable memory table of the switch is overfilled and the switch starts operating as a hub. The two attacks are not similar, because of their different outcome.

The *THC IPv6 toolkit* includes the *flood_advertise6*, which floods a target network with random NA messages. The tool requires only a network interface as a parameter:

```
flood_advertise6 eth0
```

Router Advertisement flooding

When receiving a RA messages from different routers and announcing different network prefixes, hosts and routers update their network knowledge according to the content of the messages. Although this activity is computation intensive, it is not likely that many routers will be sending RA messages in an average network. But if an attacker floods the local network with random RA messages, this will result in consumption of the available resources of the systems in the local network. RA flooding will make the systems unusable and unresponsive.

Marc Heuse listed several operating systems with IPv6 and SLAAC enabled by default, which are known for being vulnerable to this issue [33]. The most notable one is the Microsoft Windows series, including the latest version - Windows 7. According to the document, where Marc Heuse lists the vulnerable systems, Microsoft are aware of the RA flooding security issue, but they do not plan to release a fix for the issue.

The router advertisement flooding is IPv6 specific vulnerability. This vulnerability can be exploited using the *flood_router6* tool from the *THC IPv6 toolkit*. The following command sends router advertisements on interface eth0:

```
flood_router6 eth0
```

Detection of Neighbor Discovery Protocol attacks

The ICMPv6 based attacks are local to a subnet. This implies that detection mechanisms cannot be centralized in a single IDS responsible for a large network. Decentralized solution with access to every subnet in a network is necessary to detect NDP attacks.

NDPMon is an application, which monitors NDP traffic and can notify a network administrator if a host on the network spoofs NDP packets. The program is similar to *arpwatch* used for detection of ARP spoofing in IPv4. *NDPMon* can monitor NS and NA packets and detect if a new NA message is conflicting with a previous one, which is a sign of possible spoofed NA message.

In order to detect a fake Router Advertisement, an IDS can compare the source IP address and MAC address of the router, which sent the RA, to a list of known routers.

A draft for RFC [34], published in 2005, suggests changes to RFC2461 [35]. They include a method for detection of "exploitation of inherent vulnerabilities in the Neighbor Discovery processes". This method forces NDP packets to be multicast only to the host's Solicited Node Multicast group, thus allowing a security device to detect attacks. The draft proposes a solution for the NA / NS spoofing and host Redirect issue, but solution for RA / RS problems are not discussed.

Proposed method for detection of Neighbor Discovery spoofing:

- Neighbor Advertisements must be sent to the recipient's Solicited-node Multicast Address
- Require that a node shall silently discard Neighbor Advertisements that are not addressed to the node's SNA.

Proposed method for detection of host Redirect:

- Require host Redirect messages to be sent to the destination node's SNA.
- Require that a node shall silently discard Host Redirection packets that are not addressed to the node's SNA.

Apart from the above, I suggest the following additional measures for detection of Neighbor Advertisement flooding:

- The NDP entry table of the router can be monitored. In case the table is filling up faster, than its entries are expiring, a notification can be sent to the network administrator.
- If possible a list of trusted devices can be implemented on the router, which gives them a priority over the rest hosts, that send ND messages in a network. When a certain limit in the NDP entry cache table is reached, only messages from those trusted devices will be processed and the rest messages will be ignored.

RA flooding can be detected in a similar way as NS flooding. I suggest monitoring the number of new RA messages. If unusual large numbers of routers advertise new prefixes, most likely the network is being flooded. The network administrator can make a list of trusted routers and when one or several not listed routers advertise themselves, a script can notify the administrator.

Mitigation of Neighbor Discovery Protocol attacks

RFC3756 [26] recommends the use of IPsec for authenticating NDP message. However the RFC doesn't provide detailed implementation instructions. Due to the requirement of manual configuration of IPsec, SEcure Neighbor Discovery (SEND) was developed (specified in RFC3971 [30]). SEND adds new options to NDP that make it more secure. The security of SEND is based on signing the

NDP messages using RSA Public key signatures and the use of cryptographically generated addresses. NDP packets, which are not signed, are treated as unsecured.

During the research I studied possible ways to reduce the likelihood of IPv6 NDP vulnerabilities from being exploited. One of the effective solutions is segmenting the network by assigning a unique prefix to every router interface or by implementing Virtual LANs. In this way an attacker won't be able to affect a large number of hosts.

Another possible solution is reducing the subnet size. This can limit the number of possible hosts under the maximum capacity of the router's entry table, which will mitigate NA flooding. However this solution is not compatible with SLAAC, because SLAAC requires at least a /64 subnet. Additionally smaller subnets can allow an attacker to enumerate hosts easier.

To improve the robustness against Man in the Middle attacks, the administrator can configure application and transport layer encryption (TLS, SSH tunnels, etc.), because the encryption can prevent third parties from viewing the intercepted network traffic.

For mitigation of Neighbor Advertisement flooding I suggest the size of IPv6 entry cache table to be increased to a value allowing reasonable time for reaction. This should be implemented along with other measures such as throttling the host learning speed during the attack. The throttling will allow a router to continue serving the old, known hosts and learn a limited amount of new hosts.

Router Advertisement flooding can be mitigated by throttling the learning speed of hosts.

3.1.2 IPv6 smurfing

The IPv4 smurf attack is a way of generating significant traffic on the victim's network. It is an amplified attack, in which an attacker sends an ICMP echo request with spoofed source address to the broadcast address. All hosts, which receive the request, will reply to the source IP, thus generating traffic and possibly cause a DoS.

IPv6 does not use broadcasting as a form of communication. However, IPv6 relies on multicasting, and multicast addresses might also be used for a smurf attack. This makes the differences between IPv4 and IPv6 smurfing small. A simplified illustration of IPv6 smurfing can be seen on Figure 8.

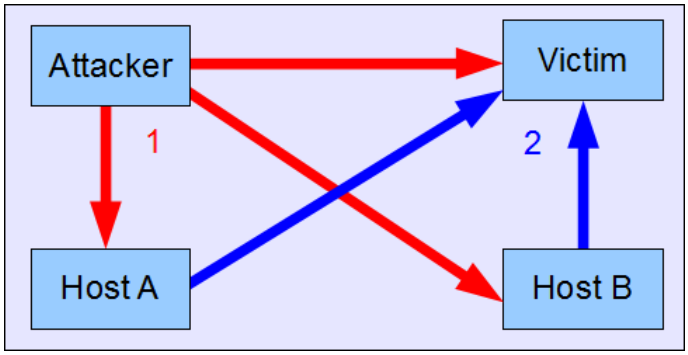

Figure 8: Smurf attack

1. The attacker sends an ICMPv6 echo request packet with spoofed source address to a multicast address.

2. The hosts, which received the request send an reply to the victim, which can overload the victim's network connection.

The attacker can send packets to the link-local all nodes multicast address (FF02::1) and the link-local all routers multicast address (FF02::2) for performing the smurf attack on IPv6. These two addresses identify the group of all nodes and routers in the scope of the local subnetwork.

The *THC IPv6 toolkit* features the *smurf6* and *rsmurf6* tools. The differences between the two tools are listed below.

The *smurf6* tool sends ICMPv6 echo request packets with spoofed source (using the victim's IP address) to the multicast address FF02::1. The hosts on the LAN that are vulnerable to the attack send ICMPv6 echo reply packets, which flood the victim. The victim of *smurf6* can be on the local subnet with the attacker or on a remote subnet.

```
smurf6 eth0 2001:610:158:960::100
```

rsmurf6 uses a different approach. It sends ICMPv6 echo reply packets that are sourced from FF02::1 and destined for remote computers. If the destination system is allowed to respond to packets sourced from a multicast address, the response causes a traffic flood on the remote LAN. This attack has stronger amplification, because each packet generated by *rsmurf6* can generate large amount of packets on the remote LAN.

```
rsmurf6 eth0 2001:610:158:960::100
```

Most modern IPv6 implementations are protected against this vulnerability and drop multicast packets, which can cause smurfing. Scott Hogg and Eric Vyncke recommend that IPv6 hosts should not be responding to echo request packets destined to a multicast group address [5].

Possible protection against remote smurf attacks can be ingress filtering, which rejects the attacking packets on the basis of the forged source address.

3.1.3 Routing header type 0

RFC2460 [36] defines an IPv6 extension header called Routing Header. The value of this header can be set to a specific type as defined in the RFC. The header type "0" (known as RH0) forces a packet to follow a strictly predefined path between network nodes. This feature allows RH0 to be used for amplification attack. RFC5095 [37] explains the attack:

> *A single RH0 may contain multiple intermediate node addresses, and the same address may be included more than once in the same RH0. This allows a packet to be constructed such that it will oscillate between two RH0-processing hosts or routers many times. This allows a stream of packets from an attacker to be amplified along the path between two remote routers, which could be used to cause congestion along arbitrary remote paths and hence act as a denial-of-service mechanism.*

...
This attack is particularly serious in that it affects the entire path between the two exploited nodes, not only the nodes themselves or their local networks. Analogous functionality may be found in the IPv4 source route option, but the opportunities for abuse are greater with RH0 due to the ability to specify many more intermediate node addresses in each packet.

According to information collected after the CanSecWest/core07 talk, several major operating systems and network vendors are vulnerable to this issue [38]. The operating systems, which are not vulnerable to the RH0 amplification attack did not implement RH0 or implemented it not according to the IETF standard.

Another possible malicious use of RH0 is to bypass firewalls that prohibit outside access to a host in the internal network. An attacker can send a packet with RH0 through the firewall to a router, which will redirect it to the target host.

The possible security problems that RH0 can cause were considered by IETF. RH0 was deprecated with RFC5095 [37] from December 2007. The RFC recommends using ingress filtering until routers are updated. The ingress filtering should be applied as recommended in RFC2827 [39] and RFC3704 [13]. If a whole network has to be protected, the ingress filtering should be implemented on the border, where the network is connected to the outside world.

3.1.4 Implementation issues of IPv6

A large percentage of the IPv6 issues listed in the National Vulnerability Database are not related to the design of the protocol, but are a result of insecure implementations [40]. Vulnerabilities, which are exploited using flaws in the IPv6 protocol, were not considered implementation specific during the research.

Based on information collected from the 98 vulnerabilities listed by NVD between October 2002 and June 2011, one can conclude that most of the implementation issues can result in:

- DoS
- Security policies bypassing
- Buffer overflow

These vulnerabilities can occur in the network stack of the device OS / firmware or in a specific piece of software installed on the device. The vulnerabilities, which allow bypassing of the security policies, are caused by none or insufficient filtering of the IPv6 packets compared to IPv4 [40].

Two vulnerabilities listed by NVD are not IPv6 specific. They apply also to IPv4 and are caused by bugs in the products.

When implementation specific vulnerabilities are discovered, usually they are fixed by the vendors in the newer versions of their products. IPv6 implementations are relatively new and are not tested in production environments as thoroughly as the IPv4 implementations. With the wider adoption of IPv6 it is likely that the bugs will be fixed and IPv6 implementation will be as good as IPv4.

Operating systems with IPv6 enabled by default can be considered vulnerable, because an attacker can advertise a rogue router, which will be automatically configured on the host. According to SixXS these operating systems include the latest versions of Windows, Mac OS and most Linux distributions [41].

A penetration tester can check for implementation specific issues by identifying an affected product by its fingerprint. *nmap* and *Nessus* are tools with large fingerprint databases, which can recognize vulnerable versions of software. *Nessus* is able to detect IPv6 specific issues listed in the Common Vulnerabilities and Exposures (CVE) database.

Along with IPv6 specific implementation issues, there can be issues which apply only when IPv6 is used along with IPv4 during the transition period.

3.1.5 Transition techniques related issues

The switch between IPv4 and IPv6 cannot happen instantaneously. A migration period is necessary, during which the two protocols will coexist allowing users to be connected to both IPv4 and IPv6 networks. During this phase, transition techniques like dual-stack, tunneling and translation will be used. However these transition mechanisms can introduce security issues, discussed in this section. During the transition phase, users and administrators have to consider both, IPv4 and IPv6 issues and combination of attacks using both protocols. For example an attacker can compromise a remote system through IPv6 vulnerability and perform ARP spoofing on the IPv4 network connection of the compromised system.

Dual-stack network issues

A dual-stack system can be less secure compared to a single stack (either IPv4 or IPv6), because an attacker has more possible attack vectors to exploit. Also it is more difficult for a system administrator to secure both IPv4 and IPv6 networks on adequate level.

Firewalls may not be enforcing the same policy for IPv4 as for IPv6 traffic, which could be due to misconfiguration of the firewalls. It is possibility for firewalls to have more relaxed policy for IPv4 or IPv6, thus allowing unfiltered traffic to pass through. In 2007 ICANN did a survey on the of IPv6 support in commercial firewalls [42]. The results show that the support of IPv6 was low at that time and traffic could go through unnoticed. A new survey was conducted by ICANN in 2010, but the results are not published yet.

These issues can be individual for every system and configuration. They can be detected by scanning the hosts and firewalls for opened port and if the same rules are enforced for both IPv4 and IPv6 networks.

Tunneled IPv6 network issues

A host using a tunneled IPv6 connection over a native IPv4 connection can be more vulnerable compared to a dual-stack host. Ryan Giobbi [43] shows examples how the encapsulated IPv6 traffic can pass unnoticed by firewalls creating security vulnerability. The tunneling software requires opening a port in the firewall that can be used for attacks, unless tunnel-aware firewall is in

place.

According to the RFC draft "Issues with Dual Stack IPv6 on by Default" [44] a poorly configured or implemented VPN may redirect traffic from a protected VPN network to an unprotected IPv6 interface, causing security issues.

A penetration tester can detect if data, which is normally blocked by the firewall, will pass through a tunnel in the firewall.

Low user and administration awareness of IPv6 autoconfiguration

The operating systems, which have as a feature IPv6 enabled by default (e.g. Windows 7 and Linux distributions with kernel version higher than 2.6 [41]), can autoconfigure without the knowledge of their user or system administrator. If security mechanisms and policy are not in place to protect against IPv6 based attacks, a host might get compromised through an IPv6 network.

Mitigation of transition techniques related issues

I believe that administrators (and maybe users as well) should be educated about the features and required security policies of IPv6. I recommend that IPv6 is disabled if administrators or users are not planning to use it or haven't implemented protection from threats originating from their IPv6 networks.

The security policy implemented in firewalls, VPN software, or other devices, should take a stance whether it applies equally to IPv4 and IPv6 traffic. The "Issues with Dual Stack IPv6 on by Default" RFC draft [44] covers such issues and recommends the implementation of more complex techniques for mitigation:

> *There is still a risk that IPv6 packets could be tunneled over a transport layer such as UDP, implicitly bypassing the security policy. Some more complex mechanisms could be implemented to apply the correct policy to such packets. This could be easy to do if tunnel endpoints are co-located with a firewall, but more difficult if internal nodes do their own IPv6 tunneling.*

A shorter transition period will minimize the time during which systems could be vulnerable to transition technique specific issues. I consider important that IPv6 is deployed fast so it can become the most used version of IP and minimize the transition period.

3.2 Enumeration of IPv6 hosts

With the adoption of IPv6, more often penetration testers will encounter systems, which are using the new version of the protocol. Some of the systems will have only IPv6 connection, while other will be using IPv4 and IPv6 simultaneously. Research performed by Hidde van der Heide and Roy Duisters on the real-life IPv4 and IPv6 network security policies shows that there can be differences in the opened ports on the IPv4 and IPv6 connection of the same system [45]. The research shows that IPv6 can have less strict restrictions and allow access to services, which are not intended to be on the Internet. These differences in the policies can allow a penetration tester to find many more vulnerabilities in a system.

Scanning an IPv6 subnet requires different approach from scanning an IPv4 subnet, because of the bigger address space of IPv6. A standard IPv4 /24 subnet has $2^8 = 256$ addresses, whereas a standard IPv6 /64 subnet has $2^{64} =$ 18,446,744,073,709,551,616 (more than 18 quintillion) addresses. Several methods that can reduce the search space for IPv6 hosts are described in RFC5157 [46]. Additionally DNS can be useful to resolve IPv6 addresses from the domains of the target.

3.2.1 Reducing the address space by analyzing the numbering scheme

The huge amount of address in an IPv6 subnet can be reduced in several ways. This reduction can be achieved due to possible patterns in the distribution of IPv6 addresses over the IPv6 subnet. Depending on the implementation, the IPv6 address might not be random, but generated using conventional, well known methods. This can allow a penetration tester to analyze them and limit the search to a more specific IP address range.

Consecutive ordered IP addresses

For convenience a system administrator can use IPv6 addresses, which are consecutive ordered, easy to write and remember (for example [prefix]::1). If a single address from the subnet is known by a penetration tester, retrieving the rest of the addresses is a trivial task.

Autoconfigured hosts with embedded MAC address

Autoconfigured IPv6 addresses, which contain an embedded MAC address can be recognized by the value "FFFE" inserted between 3^{rd} and 4^{th} bytes of the MAC [47]. This by itself reduces the search space to 2^{48} hosts. If a penetration tester is testing an organization, where most of the devices are manufactured by a single vendor it is possible to reduce the search space even more. In case the devices are similar and purchased together there is a chance that they have consecutive or close MAC addresses.

Autoconfigured hosts with embedded IPv4 address

According to RFC4291 [25], an IPv6 address can carry an IPv4 address embedded in the low-order 32 bits. This practice is done due to compatibility concerns and simpler association of the IPv6 to IPv4 addresses in dual-stack systems. If a penetration tester knows the IPv4 address of a host it is possible to use it to derive the IPv6 address to perform tests on it.

3.2.2 DNS resolving

IPv6 address are longer and use a larger character set compared to IPv4, which makes an IPv6 address harder to remember and type. Because of this, more users and system administrators will prefer to use DNS entries instead of the IPv6 address.

A couple of approaches involving DNS can be used by a penetration tester to enumerate IPv6 hosts.

Reverse DNS of IPv4

Reverse DNS resolution is used to determine the domain name that is associated with a given IP address. Dual-stack hosts can have a domain name associated to both their IPv4 and IPv6 address. A Penetration tester can scan for active hosts in the target IPv4 network. Next the tester can use reverse DNS to lookup the domain names associated with the IPv4 addresses. After the domains are known, they can be resolved in order to get the AAAA RRs associated to the IPv6 addresses.

Searching the Web for subdomains

Search engines have databases, in which they store web pages indexed by their bots. These pages can contain links to subdomains, which might be interesting for a penetration tester. With a simple query most of the pages containing the target domains can be retrieved from a search engine. An example using Google:

```
inurl:".os3.nl"
```

When dealing with high profile domain names it is highly likely for a penetration tester to find too many results. I suggest parsing the results by submitting the query to the search engine with a command-line tool like *cURL* and filtering with additional scripts.

Enumeration of DNS entries using brute-force

Brute-forcing DNS entries can be done using a dictionary attack or by automated generation of possible domain labels. RFC1034 [48], which specifies the domain names, allows only alphabet letters (Latin), numbers and hyphen in the domain labels. This limits the possible characters to 37.

dnsmap is a tool for subdomain brute-forcing [49]. The tool reads a file containing keywords and sends DNS request for each entry to the targeted name server asking for a specific domain. The word list can consist of popular subdomains like “ftp”, “www” and “admin” or random strings. The brute-forcing can be detected by the name server administrator, because it generates unusual amount of traffic and load.

Enumeration of DNS zones using DNSSEC zone walking

The DNSSEC zone walking and the *DNSSEC walker* tool are covered in the DNSSEC section of the report.

When all the domains in a zone are enumerated, the A (for IPv4) and AAAA (for IPv6) RRs can be fetched. The DNSSEC walking and fetching can be done with a simple bash pipe:

```
walker -y @ns1.os3.nl os3.nl | awk -F: /'IN\tAAAA/ {print $0}'
```

A sample output of the bash pipe, ordered by IPv6 address:

```
tummi.ipv6.os3.nl.   86400  IN  AAAA  2001:610:158:960:0:0:0:50
zummi.ipv6.os3.nl.   86400  IN  AAAA  2001:610:158:960:0:0:0:51
sunni.ipv6.os3.nl.   86400  IN  AAAA  2001:610:158:960:0:0:0:52
grammi.ipv6.os3.nl.  86400  IN  AAAA  2001:610:158:960:0:0:0:53
gruffi.ipv6.os3.nl.  86400  IN  AAAA  2001:610:158:960:0:0:0:54
```

Zone walking tests and results

During the research I performed tests on several ICT-related DNS zones in order to compare the number of domains pointing to IPv4 and IPv6 hosts in a production environment. The following bash pipe was used for fetching the AAAA RRs:

```
walker -y @ns1.os3.nl os3.nl | awk -F: /'IN\tAAAA/ {print $0}' |
awk -F: /'IN\tAAAA\t/ {print $0}' | awk /'\t/ {print $5}' | uniq |
wc -l
```

domain	**A**	**AAAA**
iana.org	39	14
ripe.net	11,069	10,705
os3.nl	432	44
digsys.bg	24,155	0

From the results is visible that IPv6 is still far from being widely adopted compared to IPv4, although the core activities of the tested subjects are ICT-related. RIPE NCC is the only exception, where the amount of IPv4 and IPv6 addresses is almost equal.

3.3 IPv6 summary

IPv6 and IPv4 have several common features and mechanisms, but they are also different in some aspects. Based on the experience from my research I consider the security of IPv6 and IPv4 comparable, because most of the known attacks against the two protocols use similar attack vectors.

Some of the security issues of IPv6 are caused by flaws in the design and other are due to poor implementation of the protocol. The design issues can result in Denial of Service and Man in the Middle attacks. The outcome of IPv6 implementation specific issues is DoS, security policies bypassing and buffer overflow. Mitigation techniques for both design and implementation issues are known.

During the transition period a security issue can be the use of double security policy, which can allow traffic from one of the networks to be less strictly inspected compared to other network connections. IPv6 connections tunneled over an IPv4 should be monitored by firewall, which is aware of the encapsulated traffic.

Tools such as the *THC IPv6 toolkit* offer a variety of options for exploitation and penetration testing of IPv6 networks. Vulnerable implementations can be detected with *Nessus* or *nmap*.

IPv6 networks can be enumerated by analyzing the used numbering scheme, resolving DNS and using results from search engines. If the target is using

DNSSEC, zone walking can be performed to enumerate all sub domains in the target's DNS zone.

4 Conclusion

This section contains the answers of the research questions and a summary of the most significant findings during the research. In order to provide clear and specific answer to the research questions, I answered the questions regarding each protocol separately. The main research question is answered once.

DNSSEC

1. What are the known security issues for DNSSEC?

DNS can be used for performing an amplification attack due to the larger in size responses compared to the queries that are send to name servers. The attack can be more effective when the name servers support DNSSEC, because of the embedded public keys and signatures in the DNSSEC resource records.

DNSSEC allows enumeration of DNS zones known as the "DNSSEC zone walking", which is based on the chain of NSEC / NSEC3 resource records.

Several implementation specific vulnerabilities are known in name servers with DNSSEC support, which might result in cache-poisoning or DoS.

The lack of DNSSEC validation on the client side can be considered an issue, because the path between authoritative name servers and end user is not yet secured. This is going to be a problem until DNSSEC is integrated in end user products, which rely on DNS.

1.1. Are these new issues, or are they based on vulnerabilities of the old technologies?

The DNS amplification attack is an old issue, present also in the DNS protocol.

The zone walking is possible only in zones, which are DNSSEC signed, thus it is a new issue.

Implementation specific vulnerabilities were present in DNS before DNSSEC was introduced.

1.2. Are there security issues during the transition period, caused when the old and new technologies are used in parallel?

DNSSEC is a security extension of DNS and can't exist as a standalone protocol. Thus there is no transition period in the sense that DNSSEC will replace DNS. However DNSSEC is not widely adopted yet and many software products don't support it. The lack of validation can be considered a transition issue, because a DNS zone can be DNSSEC signed, but the path between authoritative servers and most users is not yet secure.

2. How can the identified security issues be mitigated?

The amplification attack can be mitigated by denying traffic from spoofed source addresses. The recommended solution by IETF is implementing ingress filtering.

The DNSSEC zone walking can be partially mitigated by implementing

NSEC3 instead of NSEC resource records. NSEC3 RRs contain hashes of domain names instead of the domain names themselves. This solution is partial, because an attacker can brute-force the hashes and reveal the domain names.

Implementation specific issues can be mitigated by updating a vulnerable version of a name server to a non-vulnerable one.

3. How can a penetration tester check for these known security issues?

The DNS amplification attack is based on the normal functions of DNS and its design. A penetration tester can check whether a security mechanism, which detects spoofed DNS queries, is implemented on the tested name server.

The zone walking can be checked by querying a name server for NSEC (NSEC3) resource records in a certain zone. If they are present, the zone can be walked. The query can be sent with the command line tool *dig*.

The implementation specific issues can be detected by identifying vulnerable name server versions. Security scanners with large fingerprint databases can detect vulnerable implementations.

3.1. How can these security issues be recognized?

Name servers, which allow the DNS amplification attack, can be recognized by performing the attack against a host, which is monitored by a penetration tester. A penetration tester can check if the name server is responding to the queries from the spoofed source address.

NSEC and NSEC3 resource records are mandatory part of DNSSEC. A penetration tester can identify which one of them is implemented on the tested zone.

Implementation specific issues can be recognized by identifying the version of the name server and checking if that version is vulnerable to attacks.

3.2. What tooling can be used for performing the penetration tests?

dig can be used to query a name server for specific RRs and perform DNSSEC validation. With *dig* a penetration tester can determine if NSEC or NSEC3 are implemented in a DNSSEC signed zone.

DNSSEC zone walking over NSEC resource records can be performed with the proof of concept tool *DNSSEC Walker*.

The security scanner *Nessus* has a large fingerprint database with vulnerable systems. A penetration tester can use it to recognize problematic versions and insecure configurations of name servers.

IPv6

1. What are the known security issues for IPv6?

The following security issues were found during the research:

- Neighbor Discovery Protocol issues
  - Neighbor Solicitation / Neighbor Advertisement spoofing

  - Redirect spoofing
  - Router Solicitation / Router Advertisement spoofing
  - Duplicate Address Detection attack
  - Neighbor Advertisement flooding
  - Router Advertisement flooding
- IPv6 smurfing
- Routing header type 0
- Implementation issues
- Transition techniques issues
  - Dual-stack network issues
  - Tunneled IPv6 network issues
  - Low user and administration awareness of IPv6 autoconfiguration

The outcome from these vulnerabilities can be DoS and Man in the Middle attacks, buffer overflow and security policy bypassing.

1.1. Are these new issues, or are they based on vulnerabilities of the old technologies?

The following issues are new for IPv6.

- Neighbor Discovery Protocol issues
  - Redirect spoofing
  - Duplicate Address Detection attack
  - Neighbor Advertisement flooding
  - Router Advertisement flooding
- Implementation issues
- Transition techniques issues
  - Dual-stack network issues
  - Tunneled IPv6 network issues
  - Low user and administration awareness of IPv6 autoconfiguration

The NDP issues are based on new features and mechanisms introduced by IPv6.

Transition techniques issues are related to the coexistence of IPv4 and IPv6, poor IPv6 support in firewalls as well as the low user and administration awareness of IPv6.

The following issues were available in some form, before the introduction of IPv6:

- Neighbor Discovery Protocol issues
  - Neighbor Solicitation / Neighbor Advertisement spoofing
  - Router Solicitation / Router Advertisement spoofing
- Smurfing
- Routing header type 0

The outcome of NDP spoofing is similar to well known attacks such as ARP spoofing and rogue DHCP server.

The IPv6 smurfing attack uses multicast address instead of broadcast (as used in IPv4 smurfing).

Router header type 0 has similar effect as IPv4 source route option, but allows more effective amplification attacks. This header type is currently deprecated.

1.2. Are there security issues during the transition period, caused when the old and new technologies are used in parallel?

Dual-stack networks are vulnerable to both IPv4 and IPv6 issues. Firewalls may not enforce the same policy for IPv4 as for IPv6 traffic, resulting in one network connection being less secure than the others. Additionally IPv6 traffic, tunneled over IPv4 connection, might pass through firewalls unnoticed.

In several operating systems autoconfiguration of IPv6 connections is enabled by default. Although this is not a bug, but a feature, when the user is not aware, this can result in compromising the host through the IPv6 network connection.

2. How can the identified security issues be mitigated?

The Neighbor Discovery Protocol issues can be mitigated using IPsec or SEND. SEND is considered easier to implement and was developed to add robustness to the NDP. Solutions for specific NDP issues are also available.

Ingress filtering is a recommended measure against IPv6 smurfing and routing header type 0 attacks.

Most implementation specific issues can be mitigated by applying updates or patches. In certain cases, like the Router Advertisement flooding vulnerability in Microsoft Windows, the vendor doesn't provide updates which fix the issue.

Most of the transition technique related issues can be mitigated with firewalls, which should enforce the same rules for IPv4 and IPv6 networks. When traffic is tunneled, the firewalls should be aware of the tunneling and should be able to monitor the traffic through the tunnel.

3. How can a penetration tester check for these known security issues?

A penetration tester can identify RH0, IPv6 smurfing and NDP vulnerabilities by exploiting them and monitoring one or many hosts for the outcome of the vulnerabilities.

Vulnerable implementations can be detected by identifying the version of a product. Security scanners such as *Nessus* or tools like *nmap* are able to recognize the product version based on their fingerprint databases.

The IPv6 capabilities of a firewall can be tested by performing scans on a target system. The presence of unnecessary opened ports on a IPv6 network connection might be a sign of an insecure firewall configuration or the lack of firewall.

3.1. How can these security issues be recognized?

For certain IPv6 issues a penetration tester can try to exploit them and monitor for the expected outcome of the vulnerabilities.

Implementation specific issues can be recognized by identifying the vulnerable versions of software products.

A penetration tester can detect some transition technique issues by scanning the hosts and firewalls for opened port and if the same rules are enforced for both IPv4 and IPv6 networks. The ability of a firewall to monitor tunneled traffic can be checked by sending data, which is usually blocked by the firewall. If the data passes through the tunnel in the firewall, the firewall is not capable of monitoring tunneled traffic.

3.2. What tooling can be used for performing the penetration tests?

The tools from the *THC IPv6 toolkit* can be used to perform NDP related vulnerabilities and IPv6 smurfing.

Scanners such as *Nessus* and *nmap* can be used to identify implementation specific issues.

3.3. How to perform tests on the large IPv6 scopes?

There are several known techniques for enumeration of IPv6 hosts. Analyzing the numbering scheme in the targeted network can be used to predict the IP address of the hosts. If there are DNS entries associated to the hosts, reverse DNS, brute-forcing and DNSSEC zone walking can be used for enumeration. Additionally with search engines a penetration tester could be able to find domain names in the zone of the targeted organization.

What are the security issues of DNSSEC and IPv6 and how to perform penetration tests in order to identify them?

The design of the DNS protocol allows an amplification attack, which could be more effective if DNSSEC is used, due to the larger size of the resource records. A penetration tester can check for vulnerable name servers, by sending queries with spoofed source address and monitoring the traffic of the target host.

Zone walking is the possibility to enumerate a zone by walking the chain of DNSSEC NSEC records. Zone walking is not an actual vulnerability, because DNS data is public. The NSEC3 resource record, containing hashes instead of the domain names, was added to mitigate the zone walking. However brute-forcing the hashes makes the zone walking feasible again. A Penetration tester

can check for the possibility to perform zone walking by querying a name server for an NSEC / NSEC3 RR.

DNSSEC and IPv6 can be vulnerable to implementation specific issues. The vulnerable implementation versions can be detected by scanners such as *Nessus* and *nmap*.

The IPv6 Neighbor Discovery Protocol has numerous insecurities, which can be exploited. They have a local attack vector and a penetration tester needs access to the network to identify vulnerable systems. The *THC IPv6 toolkit* can be used for performing tests and identifying them.

The IPv6 smurf attack is similar to the IPv4 smurf attack. A penetration tester can detect vulnerable systems by monitoring the network traffic of the target host. Again the *THC IPv6 toolkit* provides the necessary tools for performing test.

Transition techniques can be a subject to different attacks. Dual-stack networks are vulnerable to both IPv4 and IPv6 issues, thus administrators should secure both on the adequate level. Differences in the security policy for the two protocols (or lack of such) might leave security holes, which can be exploited by attackers.

Tunneled IPv6 traffic might be able to pass through firewall unnoticed, thus compromising the security of the host. The installed firewalls should be able to enforce the same security policy for both IPv4 and IPv6 and monitor tunneled traffic.

When a penetration tester needs to enumerate IPv6 hosts, methods such as analyzing the used IPv6 numbering scheme, resolving DNS and making use of search engines can be used. If DNSSEC is present on the target's DNS zone, the zone can be walked with tools such as *DNS Walker*.

Performing penetration tests on the large IPv6 scopes is still feasible using several known techniques. They are based on analyzing the numbering scheme of the hosts and reducing the search space or by resolving the DNS RRs associated with the host (if present). The DNS based enumeration includes reverse DNS resolving, DNS brute-forcing and DNSSEC zone walking.

5 Summary

The DNSSEC protocol itself can be considered secure and it is not likely that serious flaws are found in the future. One of the few design problems is the zone walking, which allows enumeration of hosts.

DNSSEC makes the DNS amplification attack more feasible, due to the larger size of the DNSSEC RR. This attack is not new for DNSSEC and there are known methods for detection and mitigation.

The most significant security issues for DNSSEC are caused by bad implementations of the protocols. The implementation problems are also not new and DNS had the same type of vulnerabilities as DNSSEC.

Some of the problems in IPv6 are well known from IPv4, because the two protocols share several common mechanisms. However there are several new vulnerabilities, which are possible only in IPv6. Most of the IPv6 issues are caused by bad implementations, which are being fixed with time. IPv6 issues can result in Man in the Middle and DoS attacks, security policy bypassing and buffer overflow.

There are tools and techniques, which penetration testers can use to detect IPv6 issues. The *THC IPv6 toolkit* is able to exploit most of the design flaws of IPv6, thus enabling penetration testers to detect vulnerable systems. Vulnerable implementations can be detected with security scanners.

The large IPv6 address space, makes the traditional scanning approach used in IPv4 not feasible. During the research I proposed techniques for enumeration of IPv6 networks, which can be used by penetration testers.

Although DNSSEC and IPv6 have security issues, I believe that the protocols are secure enough and will be widely deployed. In the near future penetration testers will encounter the two protocols more often and will need to use these or similar techniques and tools as the one researched during this project.

Glossary

Attack amplification

Attack amplification is a situation, in which an attacker sends relatively small amount of traffic to a host, which causes the host to generate higher volume of traffic compared to the initial. By spoofing the source address of traffic, the attack can be targeted to a specific system, causing a DoS.

DNS cache-poisoning

DNS cache-poisoning is a security or data integrity compromise in the Domain Name System (DNS). The compromise occurs when data is introduced into a DNS name server's cache database that did not originate from authoritative DNS sources. The cache poisoning may be a deliberate attempt of a maliciously crafted attack on a name server or it may also be an unintended result of a misconfiguration of a DNS cache. An improper software design of DNS applications can also result in cache-poisoning.

Denial-of-service (DoS) attack

A DoS attack is an attempt to make a computer resource unavailable to its intended users. One common method of attack involves saturating the target machine with communications requests, such that it cannot respond to legitimate traffic, or responds so slowly as to be rendered effectively unavailable.

Ingress filtering

Ingress filtering is a technique, which checks if the source address of packets is being spoofed or they originate from the address they claim to be.

Packet spoofing

In the context of this paper a spoofing attack is an malicious activity, in which an attacker modifies the content of a network packet in order to gain an illegitimate advantage. This attack is used to hijack traffic, impersonate someone else and cause DoS.

References

[1] Internet Corporation for Assigned Names and Numbers http://www.icann.org/

[2] DNSSEC Deployment Initiative http://www.dnssec-deployment.org/

[3] European Registry for Internet Domains http://www.eurid.eu/

[4] Suranjith Ariyapperuma, Chris J. Mitchell: *Security vulnerabilities in DNS and DNSSEC* http://isg.rhbnc.ac.uk/cjm/svidad.pdf, Royal Holloway, University of London

[5] Scott Hogg, Eric Vyncke: *IPv6 Security* Cisco Press, 2009,

[6] *THC-IPV6 - attacking the IPV6 protocol suite* http://www.thc.org/thc-ipv6/ [Accessed: July 03, 2011]

[7] RFC4033: *DNS Security Introduction and Requirements*

[8] RFC4034: *Resource Records for the DNS Security Extensions*

[9] US-CERT: *Debian and Ubuntu OpenSSL packages contain a predictable random number generator* http://www.kb.cert.org/vuls/id/925211

[10] Randal Vaughn, Gadi Evron: *DNS Amplification Attacks* March 17, 2006

[11] Dean Anderson: *NTIA DNSSEC Comments* http://www.ntia.doc.gov/legacy/DNS/comments/comment027.pdf

[12] RFC5358: *Preventing Use of Recursive Nameservers in Reflector Attacks*

[13] RFC3704: *Ingress Filtering for Multihomed Networks*

[14] *DNSSEC Walker* http://josefsson.org/walker/ [Accessed: July 03, 2011]

[15] RFC5155: *DNS Security (DNSSEC) Hashed Authenticated Denial of Existence*

[16] DNSSEC Deployment Initiative: *TLD deployment table* https://www.dnssec-deployment.org/wp-content/uploads/2010/08/TLD-deployment-Table-8_30_10.pdf August 30, 2010

[17] EURiD: *Overview of DNSSEC deployment worldwide* http://www.eurid.eu/files/Insights_DNSSEC1.pdf October, 2010

[18] D. J. Bernstein: *Breaking DNSSEC* http://cr.yp.to/talks/2009.08.10/slides.pdf

[19] RFC2535: *Domain Name System Security Extensions*

[20] National Vulnerability Database: *Vulnerability Summary for CVE-2010-0290* http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2010-0290

[21] National Vulnerability Database: *Vulnerability Summary for CVE-2009-4022* http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2009-4022

[22] The Number Resource Organization: *DNSSEC* http://www.nro.net/technical-coordination/dnssec [Accessed: July 06, 2011]

[23] Kevin Murphy: *Browser makers brush me off on DNSSEC support* http://domainincite.com/browser-makers-brush-me-off-on-dnssec-support/ July 29, 2010

[24] RFC1883: *Internet Protocol, Version 6 (IPv6) Specification*

[25] RFC4291: *IP Version 6 Addressing Architecture*

[26] RFC3756: *IPv6 Neighbor Discovery (ND) Trust Models and Threats*

[27] US-CERT: *IPv6 implementations insecurely update Forwarding Information Base* http://www.kb.cert.org/vuls/id/472363

[28] Jeremy Church: *NDP and the Solicited node multicast address* http://gogonet.gogo6.com/forum/topics/ndp-and-the-solicited-node November 6, 2010

[29] RFC1256: *ICMP Router Discovery Messages*

[30] RFC3971: *SEcure Neighbor Discovery (SEND)*

[31] Jeff S. Wheeler: *IPv6 NDP Table Exhaustion Attack* http://inconcepts.biz/~jsw/IPv6_NDP_Exhaustion.pdf

[32] *RFC3041 - DRAFT* http://tools.ietf.org/html/draft-dupont-ipv6-rfc3041harmful-05

[33] *ICMPv6 Router Announcement flooding denial of service affecting multiple systems* http://www.mh-sec.de/downloads/mh-RA_flooding_CVE-2010-multiple.txt 15 April, 2011

[34] *RFC DRAFT* http://tools.ietf.org/html/draft-pashby-ipv6-detecting-spoofing-00

[35] RFC2461: *Neighbor Discovery for IP Version 6 (IPv6)*

[36] RFC2460: *Internet Protocol, Version 6 (IPv6) Specification*

[37] RFC5095: *Deprecation of Type 0 Routing Headers in IPv6*

[38] *Links on the "IPv6 Type 0 Routing Header issue" following CanSecWest/core07 talk.* http://natisbad.org/RH0/ [Accessed: July 03, 2011]

[39] RFC2827: *Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing*

[40] National Vulnerability Database: *Search Results* http://web.nvd.nist.gov/view/vuln/search-results?query=ipv6&search_type=all&cves=on

[41] SixXS Wiki: *IPv6 Capable Operating Systems* http://www.sixxs.net/wiki/IPv6_Capable_Operating_Systems [Accessed: July 03, 2011]

[42] ICAAN: *Survey of IPv6 Support in Commercial Firewalls* http://www.icann.org/en/committees/security/sac021.pdf

[43] Ryan Giobbi: *Bypassing firewalls with IPv6 tunnels* http://www.cert.org/blogs/certcc/2009/04/bypassing_firewalls_with_ipv6.html CERT, April 2, 2009

[44] RFC DRAFT: *Issues with Dual Stack IPv6 on by Defaul* http://tools.ietf.org/html/draft-ietf-v6ops-v6onbydefault-03

[45] Hidde van der Heide, Roy Duisters: *A comparison of real-life IPv4 and IPv6 network security policies* May 29, 2011

[46] RFC5157: *IPv6 Implications for Network Scanning*

[47] *IPv6 Interface Identifiers* http://msdn.microsoft.com/en-us/library/aa915616.aspx

[48] RFC1034: *DOMAIN NAMES - CONCEPTS AND FACILITIES*

[49] *Passive DNS network mapper a.k.a. subdomains bruteforcer* http://code.google.com/p/dnsmap/ [Accessed: July 03, 2011]