IPv6 Security: Attacks and Countermeasures in a Nutshell

Johanna Ullrich, Katharina Krombholz, Heidelinde Hobel, Adrian Dabrowski, Edgar Weippl
SBA Research
Vienna, Austria
Email: (1stletterfirstname)(lastname)@sba-research.org

***Abstract*—The history of computers is full of underestimation: 640 kilobyte, 2-digit years, and 32-bit Internet addresses. IPv6 was invented to overcome the latter as well as to revise other drawbacks and security vulnerabilities of its predecessor IPv4. Initially considered the savior in terms of security because of its mandatory IPsec support, it turned out not to be the panacea it was thought to be. Outsourcing security to IPsec but eventually removing it as well as other design decisions led to a number of vulnerabilities. They range from the already known spoofing of answers to link-layer address requests to novel possibilities regarding node tracking. In an effort to fix them, a vast amount of updates have been introduced.**

In this paper, we discuss security and privacy vulnerabilities with regard to IPv6 and their current countermeasures. In a second step, vulnerabilities and countermeasures are systematized by the appliance of an extendible common language for computer security incidents. Our evaluation shows that a large part of vulnerabilities can be mitigated but several security challenges remain. We deduce three main research challenges for IPv6 security, namely address assignment and structure, securing local network discovery, and address selection for reconnaissance.

I. Introduction

The Internet Protocol (IP) is the principal communication protocol of the Internet. Its fast expansion led to a shortage of IPv4 addresses and triggered the current transformation process to the revised version IPv6 with an address range of 2^{128}. Even though the new version was updated multiple times, the basic security and privacy design was made in 1998. However, a full deployment in the 2010s means distinct security vulnerabilities. In 2011, the Internet Assigning Number Authority (IANA) distributed its last IPv4 addresses to the Regional Internet Registries [1], and some of them have already run out of addresses. This way, the prolonged transformation to IPv6 gains momentum.

In the narrower sense, IPv6 is only a new transport layer header. However, this is accompanied by a long list of upgrades and revisions of related technologies, which were closely tied to IPv4. This includes new entry types for the Domain Name System (DNS), the Internet Control Message Protocol (ICMP) version 6 or a redefinition of the pseudo-header for checksum calculation. As a result, some known IPv4 vulnerabilities are not relevant for IPv6, while other flaws still remain. Certainly, the enhancement of functionalities implies new security vulnerabilities.

For the successful worldwide adoption of IPv6, security and privacy aspects in the protocol suite have been examined thoroughly in recent years. The results have been published in various scientific papers, *Requests for Comments* (RFCs), videos and blogs. It is, therefore, a time-consuming and tedious task to collect all the findings and to obtain a comprehensive understanding of this topic. In addition to scientific work, we included non-scientific contributions from hacker blogs to complete our systematization with security challenges that were detected in the wild. The overall goal of this paper is to summarize and systematize the IPv6 vulnerabilities as well as the associated countermeasures in a nutshell. In the following, we assemble IPv6 vulnerabilities and evaluate appropriate countermeasures to provide a complete and comprehensive checklist for researchers, developers and administrators. Furthermore, we deduce major future research challenges, namely address assignment and structure, securing local network discovery, and address selection for reconnaissance.

The remainder of this paper is structured as follows: Section II introduces IPv6 and related technologies. Section III summarizes currently known security vulnerabilities, while Section IV considers privacy in relation to IPv6. Section V presents excerpts of the systematization, providing two tables describing vulnerabilities/countermeasures according to a common language to describe computer security incidents and a matrix showing their adequacy. Finally, Section VI discusses the main research challenge related to IPv6, Section VII compares IPv4 to IPv6 in a number of aspects, and Section VIII concludes this work.

II. Background on IPv6

In comparison to IPv4, its successor IPv6 encompasses four major modifications: (1) The address length has been quadrupled to 128 bit, providing $3.4 \cdot 10^{38}$ unique addresses. These contain a subnet prefix and an interface identifier, and are represented by 8 quadruples of hexadecimal values separated by colons [2]. (2) Regarding the amount of receivers, three types of addresses are distinguished: *unicast*, *anycast* and *multicast* addresses. There are no *broadcast* addresses in IPv6. (3) The header format has been simplified and fixed to 40 byte, as shown in Table I. Fragmentation and other optional functionality has been shifted to optional *extension headers*, which are inserted between the IP and the upper-layer protocol header. (4) Fragmentation has further been limited to end nodes with the objective of router offloading. (5) Formerly mandatory IPsec [3]–[5] is seen as its fifth major modification before being released as optional [6].

With IP being the Internet's main protocol, many constitutive Internet technologies are heavily tied to it and the change to version 6 resulted in updates of related protocols. One of

TABLE I: IPv6 Header Format [7]

Size in Bits	Field Name	Comment
4	Version	set to *6*
8	Traffic Class	replaces *Type of Services*
20	Flow Label	for packet flow marking
16	Payload Length	incl. *IPv6 Extension Headers*
8	Next Header	
8	Hop Limit	replaces *Time to Live*
128	Source Address	
128	Destination Address	

them is the *Internet Control Message Protocol* (ICMPv6) [8]. In spite of a reduced number of message types, its scope has increased beyond error and diagnostic messages. Performing now also address resolution by means of the *Neighbor Discovery Protocol* (NDP) [9], it is also the successor of the *Address Resolution Protocol* (ARP) and responsible for router discovery.

IPv6 addresses are either configured manually, statefully (such as by *Dynamic Host Configuration Protocol* (DHCPv6) [10][1]), or by the newly introduced *Stateless Autoconfiguration* (SLAAC) [12], [13], providing plug-and-play connectivity. With SLAAC, the host first creates a link-local address on its own. After receiving a *router advertisement*, the node generates global addresses with the announced network prefixes. Recommended network prefix sizes for end sites are between /48 and /64 [14], [15].

Due to the increasing number of mobile nodes, mobility support [16] has gained importance. It allows nodes to remain transparently reachable via the same address while wandering through the network. In case the mobile node is in a foreign network, it provides its actual address to its router by means of a binding update. This provides two possibilities for correspondent nodes to communicate with the mobile node: The communication can be passed on to the home agent, which tunnels the traffic on to the mobile node. Alternatively, route optimization allows direct communication without the home agent by using a certain routing header.

The transformation from version 4 to 6 takes time and is accompanied by a phase of co-existence. Some nodes are capable of both protocols, while others are limited to one or the other. Therefore, transition technologies that bridge this gap have been developed, which can be divided into two main types: (1) Tunneling delivers a packet as another packet's payload. [17] provides a general description on tunneling IPv6 over IPv4, while [18] is a specification for tunneling other protocols over IPv6. Currently, there are a high number of different technologies tunneling IPv6 over IPv4: *6to4* [19], [20], *IPv6 rapid deployment* [21], [22], *6over4* [23], [24], *ISATAP* [25] and Teredo [26], [27]. (2) Alternatively, protocol translation, i. e., the translation of IPv4 into IPv6 headers and vice versa, can be used. Due to being tightly connected, IP translation also includes ICMP translation. The first specification *Network Address Translation - Protocol Translation (NAT-PT)* has been criticized by [28], [29] for numerous reasons, e.g. lacking support of DNSSEC. Its successor is standardized in [30]–[34]. However, tunneling is currently preferred.

[1]The stateless DHCP approach is technically speaking not a means of address assignment because it does not maintain a client state [11].

III. Security Vulnerabilities

In the course of the development of the new Internet Protocol version, changes in and supplements to functionality were made. These enhancements, however, yield different behavior and therefore often result in novel security vulnerabilities. In this section, we summarize fundamental security vulnerabilities in IPv6 and present feasible countermeasures. We organize them by intended functionality, starting with extension headers, fragmentation and other native header fields. Subsequently, *Neighbor* and *Multicast Listener Discovery* are discussed, followed by tunneling and mobility support.

A. Extension Headers

extension headers provide optional functionality and are inserted before the next-layer protocol header. Two of them are of further interest for security: (1) The *routing header type 0* holds a list of addresses that have to be visited en route to the receiver. By alternating the two addresses, the packet cycles between two nodes, causing traffic amplification on a remote path and possibly resulting in denial of service [35]. This extension header was more harmful than beneficial and was finally deprecated [35][2].

Offloading routers was a major focus during development. *IPv6 extension headers* are, therefore, only allowed to be processed at the end nodes. The only exception is the *Hop-by-Hop header* and its *Router Alert option*, which may be used for updating in the future. However, this option may also cause a decrease in router performance when many packets are sent [37].

Initially, extension headers and options did not have to follow a certain format, therefore, middleboxes are not necessarily able to process new extension headers. Later, a uniform format for extension headers was standardized [38].

B. Fragmentation

IPv6 did not explicitly prohibit the reassembly of overlapping fragments initially despite this being a well-known security threat that can be used, e. g., to evade firewalls [39]. The best-known way of doing so is overwriting the TCP SYN flag. The countermeasure in IPv4 was dropping fragments with an offset of one byte [40]. But this is no appropriate mitigation for IPv6 because an arbitrary number of *extension headers* can be inserted prior to the next-layer protocol header and cause any offset.

Such insertions are also able to shift flags or port numbers to succeeding fragments. Common firewalls collect incoming packet fragments and reassemble them in any case, but reassembly implementations differ, making IPv6 vulnerable to the same attack scenarios as IPv4 [40], [41]. These differences in reassembly can also be used to fingerprint operating systems [42].

[2]*Routing header type 0* differs from the benign *type 2* [36] used for mobile applications.

As a consequence, overlapping fragments are now explicitly forbidden because benign nodes do not have any need of sending overlaps [43]. Further, deep packet inspection should treat initial fragments without flags or port numbers with suspicion as there is a guaranteed MTU in IPv6. Finally, fragmentation is still a stateful process within a stateless protocol with the risk of memory overflow.

Specific to IPv6 are *atomic fragments*. These packets consist of only one fragment and are used in protocol translation to deliver an identifier for fragmentation in IPv4 [44]. Unfortunately, these fragments can cause dropping of benign fragments that have the same identifier. Thus, the two types of fragments should be handled in isolation from each other.

C. Mandatory IPv6 Header Fields

Similar to the *Router Alert option*, a high number of different *flow labels* is able to decrease router performance because the latter has to store a state for every label value. A malicious attacker can also gain access someone else's quality of service by using the same *flow label* [45].

D. Neighbor Discovery

Neighbor discovery has many security implications due to its philosophy of trusting everybody on the local network. Assuming an attacker has managed to reach the local network, they can perform a variety of malicious actions.

Address Resolution: Spoofing attacks that provide wrong link-layer addresses are still possible (Figure 1a). Attackers are further able to prevent victims from address assignment by answering to *duplicate neighbor detection*. One applied countermeasure is *Optimistic Duplicate Address Detection*. Here, the node assumes that its address is unique in any case [46].

Router Advertisement Spoofing: Any node on the local network is able to announce itself as a router (see Figure 1b), or spoof a router's announcement. A number of variations of this attack are known: (1) Setting the router's lifetime to zero kicks the reminder from the client's configuration. (2) Announcing an arbitrary prefix lets the clients assume this prefix is local [47], [48].

(3) Flooding the network with *router advertisements* with various prefixes causes clients to configure one address per announcement and may lead to denial of service. These problems are not fully solved by using DHCP, as the attacker can force the node to abandon DHCP. As a countermeasure, the router advertisement guard – a middlebox filtering illegitimate announcements – is proposed [49], [50].

Advertisements may also be sent unintentionally due to misconfiguration. Preferences of benign announcements should therefore be high to guarantee service even in such a case [51].

Redirects: An attacker may redirect traffic by sending *redirects* and change the sender's configuration this way.

Smurf Attacks: An attacker sends a request to a multicast address, spoofing the victim's source address. Responses are returned to the victim, causing a denial of service. Adequate request types are *echo requests* or IP packets with an unknown extension header option of type 10. *echo requests* to multicast addresses must not be answered, but some implementations do. In contrast, the alternative containing an unknown option has to be answered [52]. Considering the latter, non-answering has been proposed [53], but even in case of becoming a standard, an exception remains for *Packet too Big* messages for path MTU discovery.

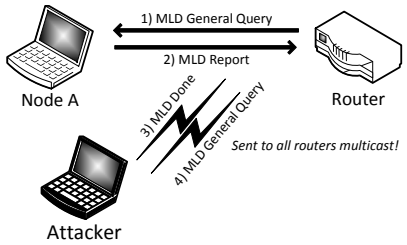


Fig. 2: Multicast Listener

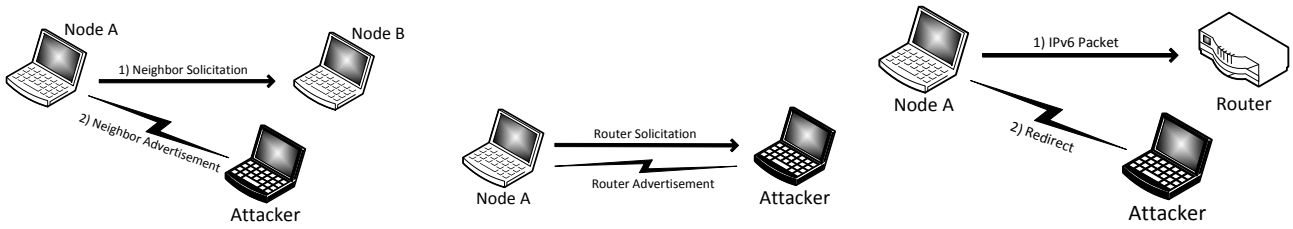
General security mechanisms tackling all vulnerabilities together have been targeted. With IPsec being initially mandatory, neighbor discovery seemed adequately secure, but it suffered from bootstrapping problems. Securing it would require manual key exchange, and therefore, unacceptable effort. As a consequence, *Secure Neighbor Discovery* (SeND) was introduced [54]. With this technology, cryptographically generated addresses enable the association of addresses to a public key [55], and signing messages with the private key prevents spoofing. However, RSA is calculation intensive and the overhead makes the systems more prone to denial-of-service attacks. Even more limiting is the low support. For example, there is only one proof-of-concept implementation for Microsoft operating systems [56]. Therefore, the only option remains to prevent attackers from joining the local network through physical protection or link-layer access control.

E. Multicast Listener Discovery

Multicast Listener Discovery (MLD) is a protocol maintaining information on nodes listening to multicast addresses. This allows the forwarding of packets destined for these addresses. A query router in charge of maintaining this information regularly sends general query messages asking for listening nodes. The latter answer with report messages. A malicious node can abort this forwarding of multicast-destined packets by sending a spoofed done message. The effect, however, would last only until the next general query message that is answered by the victim, initializing forwarding again.

Thus, the attacker has to attempt to itself become the query router. The query router is determined by having the lowest address. Although routers are frequently assigned ascending addresses, the lowest IPv6 interface identifier `::` (all zeros) is typically unused and addressing starts with `::1` [57] – possibly an IPv4 legacy.

After becoming the query router, it stops sending query requests, causing an MLD denial of service. However, the old query router will start querying again if it does not see MLD requests. However, if it sends such queries only to the all-router multicast address, the other routers are satisfied while the nodes face deteriorated service (see Figure 2). Assigning the lowest address `::` to the legitimate router is an adequate countermeasure, as explained above.



(a) Neighbor Discovery Spoofing (b) Router Advertisement Misuse (c) Redirects

Fig. 1: Basic Attacks with Neighbor Discovery

F. Tunneling

At the beginning of IPv6 deployment, tunneling illegitimate content over IPv6 was easy because many firewalls let any IPv6 traffic pass. While this has changed drastically, special threats arise from transition technologies due to the combination of the two IP versions.

Routing loops are an issue of automatic tunneling mechanisms, e. g. *Teredo* or *ISATAP* [58], [59]. Starting with a native IPv6 packet with a spoofed source address, this packet is forwarded to a tunnel ingress point. There it is encapsulated into an IPv4 packet and forwarded. At the egress point, the packet is decapsulated and equals the first, which is forwarded again to the ingress point. This causes traffic amplification because the *hop count* is only reduced on native IPv6 routers. Mitigation methods may include the general avoidance of multiple tunnels and border routers, a list of other tunnel routers' addresses to drop their packets, and checking IPv4 and IPv6 addresses for consistency [58], [60].

Special attacks are known for *Teredo*: (1) Cycling is possible between an end node and a cone NAT supporting hair-pin routing. (2) Even endless looping is possible with a bubble request. Originally intended to open another NAT via the server, the request to open the server address causes the server to send bubbles endlessly.

Nested encapsulation means the encapsulation of tunnel packets in packets of another tunnel, causing additional overhead through another packet header or even fragmentation. To counter this, a *Tunnel Encapsulation Limit option* limiting the number of nested tunnels has been introduced [61].

G. Address Space Size

The massive expansion of address space returns vulnerabilities known from the Internet's early days. Simplistic implementations of neighbor discovery may hold too many still unanswered neighbor address requests caused by network scanning. To mitigate this denial of service, filtering unused address space and minimal subnet sizing is proposed [62]. There is even discussion of minimizing subnets down to e. g. a /124, but then it is likely that implementations fail due to assuming minimum subnetworks of /64.

Point-to-point links encounter the threat of ping-pong packets in case a router forwards a packet back over the incoming interface and causes packet cycling. As above, taking smaller subnets, e. g. /127 would mitigate the risk [63]. Alternatively, the latest ICMPv6 specification [8] mitigates this by returning an ICMPv6 *Destination Unreachable* message.

H. Mobile IPv6

Binding updates inform the home agent of a mobile node's current address and enable it to stay reachable via its home address. Spoofing binding updates may inform the agent of a wrong address and can be used for man-in-the-middle, hijacking, passive wire tapping or denial-of-service attacks. In order to prevent these attacks in mobile IPv6 networks, the use of IPsec is recommended [64].

IV. Privacy Issues

Since Internet-based technologies are becoming increasingly pervasive and exhibit a tendency to neglect users' privacy, addressing privacy violations is of utmost importance. In this section, we highlight privacy-related challenges along with state-of-the-art countermeasures.

A. Addressing

As stated above, an 128-bit IPv6 address consists of a network prefix and an interface identifier. While the first is given by the network on which the host resides, the interface identifier is independently generated by the host. Initially, the modified EUI-format containing the MAC address was proposed for generation of the interface identifier [12]. Since using a hardware address results in unique identifiers even across different subnets, it is easy to track a node's movement through the network. A draft now even proposes their deprecation [65].

Numerous address formats have been proposed as an alternative: (1) The *Privacy Extension* generates an MD5 hash at a regular time interval – typically 24 hours – and uses this as the identifier [66]. While this impedes long-term tracking, short-term tracking is still possible as the identifier does not change simultaneously with the prefix. (2) Another alternative frequently proposed is DHCPv6. However, it relies on the static *DHCP Unique Identifier* (DUID). By sniffing DUIDs locally or requesting the respective DHCP servers directly, an attacker is still able to correlate a node with its current address [67].

With *Mobile IPv6*, there is a trade-off between keeping track of all sessions during network switching and the privacy breach allowing to be traceable across different networks. By including the home address and the temporary care-of address in one packet, a potential adversary is able to eavesdrop on the communication channel and infer the device's location. This may be prevented by encryption, e. g. IPsec. However, nodes communicating with the mobile device can still track the latter. To prevent such privacy breaches, the care-of address and the home address must also be changed simultaneously [16].

B. Reconnaissance

The discovery of unknown nodes is typically the first step in an attack or penetration test, but the sheer size of the address range makes brute-forcing impossible. Thus, more sophisticated methods are necessary: (1) In 2007, an analysis of IPv6 addresses in the wild showed frequent address structures for the first time [68]. While servers and routers tend to follow the modified EUI-Format and "low" addresses, clients have a significant portion of addresses generated by the privacy extension. Further analyses are feasible by *address6* [69]. Results of such analyses have resulted in *scan6* of the same toolkit. This tool searches for low-byte, IPv4-based, port-based or modified EUI addresses.

(2) Another source for addresses is DNS, which will be becoming more popular with IPv6 due to the address length. First, it is possible to query known domains. Second, reverse entries can be exploited at BIND or NDS implementations [70]. As the response for an empty non-terminal differs from other error messages, it is possible to infer whether addresses starting with this prefix are known to this server. (3) Beyond DNS, all other sources of addresses are of interest as well, e. g. *Node Information Queries* [71], *Inverse Discovery* [72] or *whois.net* [73].

(4) A modified version of the smurf attack is also capable of reconnaissance. Instead of spoofing the source address, the attacker inserts its own address and receives responses with previously unknown source addresses. However, one has to be aware that a high number of responses may cause a denial of service to oneself [57]. To prevent revealing individual addresses, servers listening to anycast addresses should also use this anycast address as a source address in the response [39].

But inherent features of IPv6 also make reconnaissance easier: (1) The assignment of more than one address to an interface is legitimate, but for reconnaissance it is sufficient to discover one. (2) Addresses expire after a preferred lifetime, but are still used for an existing connection for some time [13]. (3) Clients using the privacy extension further own a stable address that can be assigned randomly or following the modified EUI format [65]. (4) ICMP must not be totally filtered with IPv6. Even further, filtering *echo requests* and *responses* is said to be less important due to the alleged possible risk from scans [74]. An overview on this topic is also given by [73].

C. Covert Channels

Covert channels are communication channels violating system policies. In total, 22 possible covert channels have been found in the IPv6 header and its extensions [75]. The most well known covert channels are the *flow label* with 20 bit [45] and the *traffic class* with 8 bit, as their use is still vaguely defined. While the latter is allowed to be changed en route, the modification of the *flow label* was previously prohibited [76]. This, however, has changed: resetting is allowed in case a covert channel imposes a serious risk [45]. Another covert channel of 64 bit is provided by the interface identifiers. As the privacy extension causes frequently changing random addresses, it is highly unlikely that these secret messages are detected [77].

V. Systematization of Knowledge

Systematization means arranging something so as to present the content more clearly. Section III and Section IV explained security and privacy vulnerabilities as well as countermeasures for IPv6 verbally. This section presents them so that they can be taken at a glance and serve as a checklist for researchers and practitioners alike. With the more in-depth verbal description in the previous sections and this systematic overview, this paper presents the subject in multiple ways, allowing it to be used as a reference guide.

The methodology has to fulfil two goals: (1) a clear arrangement and (2) a brief description of the attacks. In Section V-A, an appropriate approach is presented. Section V-B contains the systematization for vulnerabilities, Section V-C for countermeasures and Section V-D shows the adequacy of countermeasures to vulnerabilities.

A. Methodology

[78] developed an extendible common language for describing computer security incidents. According to this work, "an attack is a series of steps taken by an attacker to achieve an unauthorized result". It consists of a tool for exploitation, a vulnerability describing a system weakness, an event – a directed action intended to change the state of a system – and an unauthorized result. The event consists of an action performed by the attacker on a certain target. We adapted this common language to the purpose of describing IPv6 security and privacy vulnerabilities and the respective countermeasures. The original common language did not offer a description for countermeasures, but we believe describing them as a sequence of steps as well is adequate.

B. Systematization of Vulnerabilities

The vulnerabilities have been systematized by means of six attributes: (1) action, (2) object, (3) target, (4) unauthorized result, (5) origin, and (6) type.

The action describes the activity of the attacker and is further specified by the object and the target. The object describes the entity the action is performed on. The target defines the victim node. If the latter attribute is left free, all types of nodes are likely to be attacked. While object and target are not enumerated, a limited number of values exist for action. The following list defines them in accordance with place holders for object and target in brackets:

- *assign:* set the address for [target] to [object]
- *flood:* emit a high number of [object] to [target]
- *insert:* include [object] into [target]
- *listen:* eavesdrop on the traffic for [object]
- *scan:* iterate through the addresses of [target]
- *send:* emit a packet including [object] to [target]
- *spoof:* emit [object] to [target] pretending to be another node

The unauthorized result describes the aftermath of the malicious action. Further, the origin of a vulnerability and a threat type is defined. The attribute vulnerability indicates whether the vulnerability results from a design, implementation

TABLE II: Classification of Security Vulnerabilities

	ID	Vulnerability	Action	Object	Target	Unauthorized Result	Origin	Type
Security	v01	Fragmentation Header I	send	overlapping fragments		modified header fields	design	modification
	v02	Fragmentation Header II	send	port number in second fragment		middlebox evasion	design	interception
	v03	Fragmentation Header III	flood	fragments		memory shortage	design	interruption
	v04	Fragmentation Header IV	flood	atomic fragments		packet loss	design	interruption
	v05	Routing Header Type 0 I	send	routing header		traffic amplification	design	interruption
	v06	Routing Header Type 0 II	send	routing header		middlebox evasion	design	interception
	v07	Extension Header Options I	send	router alert option		increased workload	design	interruption
	v08	Extension Header Options II	spoof	invalid 10xxxx option	multicast address	multiple responses	design	interruption
	v09	Hop-by-Hop Header	send	hop-by-hop header		increased workload	design	interruption
	v10	New Extension Header	send	unknown extension header		middlebox evasion	design	interception
	v11	New Extension Header	send	unknown extension header		increased workload	design	interruption
	v12	Flow Label I	send	different flow labels		memory shortage	design	interruption
	v13	Flow Label II	send	existing flow label		quality-of-service theft	design	interruption
	v14	Neighbor Advertisement I	spoof	neighbor advertisement		wrongly resolved address	design	interruption
	v15	Neighbor Advertisement II	spoof	neighbor advertisement		traffic redirection	design	modification
	v16	Neighbor Advertisement III	spoof	neighbor advertisement		address assignment prevention	design	interruption
	v17	Router Advertisement I	spoof	router advertisement		new default router	design	modification
	v18	Router Advertisement II	spoof	router advertisement		removed default router	design	modification
	v19	Router Advertisement III	spoof	router advertisement		wrong locally-announced prefix	design	modification
	v20	Router Advertisement IV	flood	router advertisement		multiple address assignment	implementation	interruption
	v21	Router Advertisement V	spoof	router advertisement		prevention of DHCP assignment	design	interruption
	v22	Router Advertisement VI	send	router advertisement		IPv6 activation	implementation	modification
	v23	Redirect I	spoof	redirect		redirected traffic	design	modification
	v24	Redirect II	spoof	redirect		wrong locally-announced node	design	modification
	v25	Echo Request I	spoof	echo request	multicast address	multiple responses	implementation	interruption
	v26	SeND	send	authenticated messages		increased workload	design	interruption
	v27	Tunneling I	send	IPv6 packet as IPv4 payload		middlebox evasion	implementation	interception
	v28	Tunneling II	send	tunnel packet	relay router	cycling packet	implementation	interruption
	v29	Tunneling III	send	tunnel packet		cycling packet	configuration	interruption
	v30	Teredo	send	Teredo bubble	server	cycling packet	design	interruption
	v31	Nesting	insert	packet into packet		packet overhead	configuration	interruption
	v32	Fragmentation Header V	send	packet too big		inclusion of atomic fragments	design	interception
	v33	Neighbor Discovery	scan		subnetwork	memory shortage	implementation	interruption
	v34	Forwarding	send	returning packet		traffic amplification	design	interruption
	v35	Mobile IPv6 I	spoof	binding update	home agent	traffic redirection	design	modification
	v36	Multicast Listener	assign	lowest address	itself	new MDL query router	design	modification

TABLE III: Classification of Privacy Vulnerabilities

	ID	Vulnerability	Action	Object	Target	Unauthorized Result	Origin	Type
Privacy	c01	Fragmentation Header VI	send	overlapping fragments		identification	implementation	interception
	c02	Modified EUI Format	scan	interface identifier	networks	tracking	design	interception
	c03	Echo Request II	send	echo request	invalid multicast address	identification of sniffing nodes	implementation	interception
	c04	Mobile IPv6 II	listen	binding update		tracking	design	interception
	c05	DHCP I	listen	DHCP traffic		tracking	design	interception
	c06	DHCP II	send	DHCP information request	DHCP server	tracking	design	interception
	c07	DNS	send	DNS request	DNS server	reconnaissance	design	interception
	c08	Reverse DNS	send	Reverse DNS query		reconnaissance	implementation	interception
	c09	Echo Request III	send	echo request	multicast address	multiple responses	implementation	interception
	c10	Extension Header Options III	send	packet with invalid option	multicast address	multiple responses	design	interception
	c11	Anycast	send		anycast address	response with unicast address	implementation	interception
	c12	Traffic Class	insert	secret information	traffic class field	leaked information	design	interception
	c13	Flow Label	insert	secret information	flow label field	leaked information	design	interception
	c14	Privacy Extension I	insert	secret information	interface identifier	leaked information	design	interception

or configuration flaw according to the following definitions by [78]:

- *configuration:* "a vulnerability resulting from an error in the configuration of a system"
- *design:* "a vulnerability inherent in the design or specification of hardware or software whereby even a perfect implementation will result in a vulnerability"
- *implementation:* "a vulnerability resulting from an error made in the software or hardware implementation of a satisfactory design"

The threat type is also limited to three values following the definitions by [79]:

- *interception:* "some unauthorized party has gained access to an asset"
- *interruption:* "an asset of the system becomes lost, unavailable, or unusable"
- *modification:* "an unauthorized party not only accesses but tampers with an asset"

The resulting systematization for the above described vulnerabilities is found in Table II and Table III.

C. Systematization of Countermeasures

Countermeasures are described by the two attributes action and object, which have the same purpose as for vulnerabilities. However, the list of actions changes to the following:

- *assign:* set [object]
- *disable:* deactivate [object]
- *encrypt:* encode [object] to be secured against reading and/or tampering
- *filter*[3]*:* remove [object] when passing
- *isolate:* process [object] separately
- *limit:* define maximal value for [object]
- *log:* write message about [object]
- *minimize:* reduce number of [object] as much as possible
- *prohibit:* ban [object]
- *respond:* return with [object]

Object is not enumerated. The countermeasures are further classified into three groups of activity levels: (1) *detective* countermeasures discover a present attack, (2) *preventative* countermeasures are taken before an attack takes place, and (3) *reactive* countermeasures are triggered by the attack. The resulting systematization is found in Table IV.

TABLE IV: Systematization of Countermeasures

ID	Countermeasure	Action	Object
	Detective		
c01	NDP Mon	log	inconsistent NDP msg.
	Preventative		
c02	Use Anycast Address	respond	with anycast as source address
c03	DHCP	assign	addresses statefully
c04	No Forwarding	prohibit	forwarding over same interface
c05	Fragment Isolation	isolate	atomic from other fragments
c06	IPsec	encrypt	packets
c07	IPsec with Manual Keys	encrypt	packets
c08	No IPv6 Support	disable	IPv6
c09	Format Deprecation	prohibit	modified EUI format
c10	Multicast Listener Address	assign	lowest address to router
c11	No Multiple Edge Routers	disable	other edge routers
c12	No Multiple Tunnels	disable	other tunnels
c13	No Multicast Responses	prohibit	answers to multicast addresses
c14	No Overlapping Fragments	prohibit	overlapping fragments
c15	Packet Rate	limit	packet rate
c16	Physical Protection	prohibit	physical access to network
c17	Privacy Extension	assign	temporary random address
c18	RA Throttler	limit	router advertisements
c19	No RAs	disable	router advertisements
c20	No Routing Header Type 0	prohibit	routing header type 0
c21	Router Preference	assign	highest preference
c22	Segmentation	segment	network
c23	SeND	encrypt	NDP messages
c24	Subnet Size	minimize	subnet size
c25	Temporary DUID	assign	temporary DUID
c26	No Tunneling	disable	all tunnels
c27	Uniform Format	limit	number of ext. header formats
	Reactive		
c28	Address Change	assign	new addresses simultaneously
c29	Address Checks	filter	inconsistent addresses
c30	Change Field en route	assign	default value
c31	Echo Requests	filter	echo requests
c32	Hop-by-Hop Options	filter	hop-by-hop extension header
c33	Routing Header	filter	routing headers
c34	Fragmented Packets	filter	packets with port not in 1st frag.
c35	Invalid Options	filter	options of type '10xxxx'
c36	Link Layer Access Control	filter	unauthorized clients
c37	Message Checks	filter	invalid ICMP msg.
c38	NDP Inspection	filter	inconsistent msg.
c39	RA Guard	filter	invalid router advertisements
c40	RA Filtering	filter	router alert options
c41	Router Listing	filter	msg. from other tunnel routers
c42	Tunnel Enc. Limit	limit	number of nested packets
c43	Tunnel Ingress and Exit	filter	at tunnel end points
c44	Unused Addresses	filter	unused addresses

D. Vulnerabilities and Appropriate Countermeasures

Table V shows the adequacy of countermeasures to vulnerabilities. We created a matrix where each row represents a vulnerability and each column a countermeasure. A checkmark indicates that a countermeasure is adequate. There is no distinction between various levels of mitigation, e. g. total mitigation vs. some improvement of status quo.

The introduction of a certain countermeasure may lead to new vulnerabilities. For example, the use of SeND to prevent RA attacks creates a vulnerability to denial-of-service attacks due to increased calculation efforts. Likewise, the use of the privacy extension prohibits tracking, but makes it possible for the interface identifier to be used as a covert channel. Thus, a method may be a vulnerability and a solution to another vulnerability at the same time. Further, there are vulnerabilities that cannot be mitigated easily by means of the mechanisms presented here, e.g. memory shortage due to fragment flooding.

[3]Discarding has been included in filtering as it can also be understood as removing messages.

TABLE V: Evaluation of Countermeasures

	NDP Mon	Answer with Anycast Address	DHCP	No Forwarding	Fragment Isolation	IPsec	IPsec with Manual Key Configuration	IPv6 Support	Format Deprecation	Multicast Listener Address	No Multiple Edge Routers	No Multiple Tunnels	No Multicast Responses	No Overlapping Fragments	Packet Rate	Physical Protection	Privacy Extension	RA Throttler	No RAs	No Routing Header Type 0	Router Preference	Segmentation	SeND	Subnet Size	Temporary DUID	No Tunneling	Uniform Format	Address Change	Checks	Change Field en route	Echo Requests	Hop-by-Hop Options Header	Fragmented Packet Filtering	Invalid Options	Link Layer Access Control	Message Checks	NDP Inspection	RA Guard	RA Filtering	Router Listing	Tunnel Encapsulation Limit Option	Tunnel Ingress and Exit	Unused Addresses
Fragmentation Header I														✓																														
Fragmentation Header II																															✓												
Fragmentation Header III																																											
Fragmentation Header IV					✓																																						
Routing Header Type 0 I																			✓																								
Routing Header Type 0 II																			✓																								
Extension Header Options I																														✓							✓					
Extension Header Options II												✓																		✓		✓											
Hop-by-Hop Header																														✓												
New Extension Header																									✓																		
New Extension Header																									✓																		
Flow Label I																																										
Flow Label II																																										
Neighbor Advertisement I	✓					✓									✓						✓	✓												✓	✓	✓						
Neighbor Advertisement II	✓					✓									✓						✓	✓												✓	✓	✓						
Neighbor Advertisement III	✓					✓									✓						✓	✓												✓	✓	✓						
Router Advertisement I	✓					✓									✓		✓	✓		✓	✓	✓												✓	✓		✓					
Router Advertisement II	✓					✓									✓		✓	✓			✓	✓												✓	✓		✓					
Router Advertisement III	✓					✓									✓		✓	✓			✓	✓												✓	✓		✓					
Router Advertisement IV	✓					✓							✓		✓		✓	✓			✓	✓												✓	✓		✓					
Router Advertisement V	✓					✓									✓		✓	✓			✓	✓												✓	✓		✓					
Router Advertisement VI	✓					✓	✓								✓		✓	✓			✓	✓												✓	✓		✓					
Redirect I	✓														✓						✓	✓												✓	✓							
Redirect II	✓														✓						✓	✓												✓	✓							
Echo Request I												✓			✓														✓													
SeND															✓																				✓								
Tunneling I										✓	✓														✓															✓		
Tunneling II										✓	✓														✓		✓												✓			
Tunneling III																									✓		✓															
Teredo																									✓																	
Nesting											✓														✓														✓			
Fragmentation Header V															✓																											
Neighbor Discovery													✓										✓																			✓
Forwarding			✓																				✓																		
Mobile IPv6 I					✓	✓																																		
Multicast Listener									✓																																
Fragmentation Header VI													✓																												
Modified EUI Format								✓								✓																										
Echo Request II																													✓												
Mobile IPv6 II					✓																					✓															
DHCP I																								✓																	
DHCP II																								✓																	
DNS																																									
Reverse DNS																																									
Echo Request III												✓																	✓												
Extension Header Options III												✓																			✓											
Anycast		✓																																							
Traffic Class																											✓														
Flow Label																											✓														
Privacy Extension I			✓																																						

VI. Future Challenges in Research

Large-scale IPv6 deployment is unquestionably a practitioners' task. However, in this case, practice and research live in mutual symbiosis. The practical experience gained from large-scale deployments typically reveals previously unknown security issues that are not easily solved. As such, they are bounced back to research, where in-depth investigation takes place. In this paper, we described IPv6's status quo with the objective of identifying such back-bouncing topics. While many vulnerabilities have already been considered in practice, the results from our systematization suggest that there is a variety of research challenges to be investigated. In this section, we infer these main challenges regarding IPv6 and propose possible approaches for mitigation.

A. Addressing

Every proposed addressing solution has a serious drawback: (1) The modified EUI-format is easily traceable by benign administrators as well as attackers using out-of-the-box tools like ping. (2) The usage of DHCP does not mitigate this issue because of the unique and stable DUID, and (3) the privacy extension is highly volatile. Therefore, especially administrators fear its negative impact on logging. (4) Manual address assignment is possible for servers and routers, but not for a large amount of clients. These drawbacks highlight the lack of an adequate address assignment structure for the clients' side.

To strike a balance between full randomness and foolproof tracking, requirements for client addressing have to be defined prior to the development of another approach. From this we deduce that the ability to guess a node's address depends on a person's role: (1) Administrators must be able to correlate addresses belonging to its sub-network to physical hosts. (2) Outsiders must not be able to correlate addresses of the same physical node from different networks. This leads to the conclusion that the administrator must have an advantage in terms of knowledge, e. g., through the creation of a pseudo-random addressing scheme seeded by the administrator.

B. Securing the Local Network

Securing ICMPv6 with IPsec has proven to be inadequate due to a bootstrapping problem: IPv6 requires prior setup by means of *router advertisements*, *neighbor solicitations* and *neighbor advertisements*. Securing with IPsec in turn requires a previous key exchange over IP, which is not ready for use at this point.

With this insight, SeND was proposed. But even the toughest solution fails if the acceptance is low and no practical implementation is available. We therefore conclude that although high effort has been put into the development of a general security solution for ICMPv6, there are no advantages over its predecessor IPv4.

As a consequence, protection has to be provided on other layers, e. g., preventing attackers from accessing the local network (physical protection) or link-layer access control. However, physical protection is not feasible in wireless communication and the growth of cloud computing leads to shared local networks among foreign parties. Link-layer access is inappropriate with "bring-your-own-device" policies where the IT department are unable to support the various types of devices. This also applies to decentrally organized organizations like universities.

In such cases, only specific countermeasures such as *router advertisement* guards or throttles remain. The disadvantage is their limited domain and the unknown impacts of combining them. Thus, we strongly encourage researchers to pick up this topic again to develop a more practical general security solution for ICMPv6.

C. Reconnaissance

Even though reconnaissance in IPv6 has been considered impossible, various techniques have proven the opposite. Nevertheless, they have some drawbacks: (1) DNS querying reveals mainly servers that are intended to be found anyway. (2) Messages to multicast addresses invoking responses may result in a denial of service and their deprecation is foreseeable. (3) Eavesdropping, i. e., passive listening to network traffic, does not work for outside attackers as it is unlikely that packets originating within the victim's prefix will run into the attacker in an arbitrary location on the Internet.

Considering this, scanning is still the most promising reconnaissance type due to (1) invoking active responses from the victim, (2) revealing the stable address instead of a temporary one, (3) its local as well as global applicability, (4) its independence from certain protocols and (5) the difficulty of mitigating it due to using the inherent functionality of protocols. What seems to be legacy is brute-force scanning, i. e., iterating through all possible addresses – the method of choice in IPv4. In conclusion, research has to find new address selection algorithms for active probing to replace brute-forcing and manage the large amount of IPv6 addresses in this way. We believe that the exploitation of address structures is promising. Research, therefore, requires data sets of IPv6 addresses. Thus, we strongly encourage the collection of such data sets that make it possible to get more in-depth knowledge on assignment in various environments. Nevertheless, reconnaissance will be also dependent on the developments in addressing.

VII. Generation Next - Generation Best?

Although IPv6 undoubtedly implies significant privacy and security flaws, it must be noted that neither was its ancestor fully secure, yet still contributed to today's interconnected world. Next generation IP will neither be Internet security's patron nor its tortfeasor. Thus, this chapter describes the idea behind protocol application in (1) IPv4 as we know it from today's Internet, (2) IPv6 as primarily intended before the turn of the millennium, and (3) the current state of IPv6. This allows further deliberation of the extent of security and privacy flaws in different phases of IP.

IPv4 was developed as a packet-switching protocol in 1981. At this time, Internet attacks were rare because the network was an academic network connecting universities with a high number of trusted users. This changed with the Internet's commercialization, providing targets with great financial gain and a changing user group. More central solutions came into existence to tackle corporate needs. Since then, a controversy

has existed between the corporate world and academia still aiming at the end-to-end principle.

Initially, IPv6 had been planned to restore the Internet's end-to-end principle, enabling flexibility, decentrality and equality. Measures thereof were the prohibition of fragmentation or other extension headers on intermediate routers, or self-configuration of addresses by SLAAC and the restriction to basic functions in the main protocol header. Additionally, security was valued by the mandatory introduction of IPsec. However, this turned out to be a pitfall presumably caused by limited security knowledge and experience at that time. The decentralized approach was also not fully pervasive as numerous technologies were reused, e.g. DNS. Resolution of domain names even seemed to become a more vital role due to the unwieldy IPv6 addresses.

Like IPv4, the new protocol version experienced an evolution in the past decades based on gained experience as well as a changed environment, e.g. the increased number of mobile nodes. Similar to before, a trend towards centrality becomes apparent. It seems to be driven by corporate administrators who prefer to limit their users in order to achieve manageability, controllability and security. This has led to a reintroduction of various protocols, e.g. DHCP, or the wide acceptance of central middle boxes. The standardization efforts are further an action to anticipate the development of various flavors of implementation like experienced with NAT in IPv4.

Considering all these attacks and the failed security approach with IPsec, IPv6 seems less secure and leads to the final question: Is IPv6 in general more or less secure than IPv4? Our results suggest that this protocol is less secure than it could be if the experience with its predecessor had been taken into account. Further, we conclude that IPv6 is not less secure than IPv4: (1) Fragmentation attacks are known for both versions and (2) securing the local network has always been done on lower layers. (3) SeND vulnerabilities will not play a major role due to its lacking acceptance in practice. (4) Attacks aimed at denial-of-service of routers en-route prevent the goal of router offloading, but IPv6 has at least achieved offloading from the performance-intensive task of fragmentation.

Nevertheless, one major issue remains – transition technology which causes roughly 30 percent of the presented security vulnerabilities. Originally, transition was intended as an interim phase of dual-stack nodes natively supporting both protocol versions. However, this process did not gain momentum for a long time -– also due to distrusting IPv6 security, and now the time has passed for this approach leading to tunneling and translating. In conclusion, a number of security flaws have been introduced by fearing IPv6.

VIII. Conclusion

In this paper, we contextualized security as well as privacy vulnerabilities of IPv6 and evaluated available countermeasures. Then, we systematized the vulnerabilities with respect to the following criteria: *action, object, target, unauthorized results, origin* and *type*. Furthermore, the countermeasures were systematized by *action, object* and *activity level*. The evaluation showed that a countermeasure could be found for the majority of vulnerabilities, which leads to the conclusion that IPv6 is a rather secure protocol. However, some countermeasures create new vulnerabilities. For example, SeND prevents *router advertisement* attacks but increases the risk of denial of service due to increased calculation effort.

Finally, we targeted imperfectly addressed vulnerabilities and identified three major research challenges left with regard to IPv6: (1) addresses providing protection against outside tracking but easy logging for administrators, (2) once more picking up the idea of a general security solution for local network discovery, (3) and the development of an address selection technique that allows reconnaissance through active probing.

Acknowledgment

This research was funded by the Austrian Science Fund (FWF): P 26289-N23 and COMET K1, FFG - Austrian Research Promotion Agency.

References

[1] "IPv4 Address Report." [Online]. Available: http://www.potaroo.net/tools/ipv4/index.html
[2] R. Hinden and S. Deering, "IP Version 6 Addressing Architecture," RFC 4291 (Draft Std), IETF, Feb. 2006, updated by RFCs 5952, 6052.
[3] S. Kent and K. Seo, "Security Architecture for the Internet Protocol," RFC 4301 (Proposed Std), IETF, Dec. 2005, updated by RFC 6040.
[4] S. Kent, "IP Authentication Header," RFC 4302 (Proposed Std), IETF, Dec. 2005.
[5] ——, "IP Encapsulating Security Payload (ESP)," RFC 4303 (Proposed Std), IETF, Dec. 2005.
[6] E. Jankiewicz, J. Loughney, and T. Narten, "IPv6 Node Requirements," RFC 6434 (Informational), IETF, Dec. 2011.
[7] S. Deering and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification," RFC 2460 (Draft Std), IETF, Dec. 1998, updated by RFCs 5095, 5722, 5871, 6437, 6564, 6935, 6946.
[8] A. Conta, S. Deering, and M. Gupta, "Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification," RFC 4443 (Draft Std), IETF, Mar. 2006, updated by RFC 4884.
[9] T. Narten, E. Nordmark, W. Simpson, and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)," RFC 4861 (Draft Std), IETF, Sep. 2007, updated by RFC 5942.
[10] R. Droms, J. Bound, B. Volz, T. Lemon, C. Perkins, and M. Carney, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)," RFC 3315 (Proposed Std), IETF, Jul. 2003, updated by RFCs 4361, 5494, 6221, 6422, 6644.
[11] R. Droms, "Stateless Dynamic Host Configuration Protocol (DHCP) Service for IPv6," RFC 3736 (Proposed Std), IETF, Apr. 2004.
[12] S. Thomson and T. Narten, "IPv6 Stateless Address Autoconfiguration," RFC 2462 (Draft Std), IETF, Dec. 1998, obsoleted by RFC 4862.
[13] S. Thomson, T. Narten, and T. Jinmei, "IPv6 Stateless Address Autoconfiguration," RFC 4862 (Draft Std), IETF, Sep. 2007.
[14] IAB and IESG, "IAB/IESG Recommendations on IPv6 Address Allocations to Sites," RFC 3177 (Informational), IETF, Sep. 2001, obsoleted by RFC 6177.
[15] T. Narten, G. Huston, and L. Roberts, "IPv6 Address Assignment to End Sites," RFC 6177 (Best Current Practice), IETF, Mar. 2011.
[16] C. Perkins, D. Johnson, and J. Arkko, "Mobility Support in IPv6," RFC 6275 (Proposed Std), IETF, Jul. 2011.
[17] E. Nordmark and R. Gilligan, "Basic Transition Mechanisms for IPv6 Hosts and Routers," RFC 4213 (Proposed Std), IETF, Oct. 2005.
[18] J. Haas and S. Hares, "Definitions of Managed Objects for BGP-4," RFC 4273 (Proposed Std), IETF, Jan. 2006.
[19] B. Carpenter and K. Moore, "Connection of IPv6 Domains via IPv4 Clouds," RFC 3056 (Proposed Std), IETF, Feb. 2001.
[20] C. Huitema, "An Anycast Prefix for 6to4 Relay Routers," RFC 3068 (Proposed Std), IETF, Jun. 2001.
[21] R. Despres, "IPv6 Rapid Deployment on IPv4 Infrastructures (6rd)," RFC 5569 (Informational), IETF, Jan. 2010.

[22] W. Townsley and O. Troan, "IPv6 Rapid Deployment on IPv4 Infrastructures (6rd) – Protocol Specification," RFC 5969 (Proposed Std), IETF, Aug. 2010.

[23] B. Carpenter and C. Jung, "Transmission of IPv6 over IPv4 Domains without Explicit Tunnels," RFC 2529 (Proposed Std), IETF, Mar. 1999.

[24] J. Wu, Y. Cui, X. Li, M. Xu, and C. Metz, "4over6 Transit Solution Using IP Encapsulation and MP-BGP Extensions," RFC 5747 (Experimental), IETF, Mar. 2010.

[25] F. Templin, T. Gleeson, and D. Thaler, "Intra-Site Automatic Tunnel Addressing Protocol (ISATAP)," RFC 5214 (Informational), IETF, Mar. 2008.

[26] C. Huitema, "Teredo: Tunneling IPv6 over UDP through Network Address Translations (NATs)," RFC 4380 (Proposed Std), IETF, Feb. 2006, updated by RFCs 5991, 6081.

[27] D. Thaler, "Teredo Extensions," RFC 6081 (Proposed Std), IETF, Jan. 2011.

[28] G. Tsirtsis and P. Srisuresh, "Network Address Translation - Protocol Translation (NAT-PT)," RFC 2766 (Historic), IETF, Feb. 2000, obsoleted by RFC 4966, updated by RFC 3152.

[29] C. Aoun and E. Davies, "Reasons to Move the Network Address Translator - Protocol Translator (NAT-PT) to Historic Status," RFC 4966 (Informational), IETF, Jul. 2007.

[30] F. Baker, X. Li, C. Bao, and K. Yin, "Framework for IPv4/IPv6 Translation," RFC 6144 (Informational), IETF, Apr. 2011.

[31] C. Bao, C. Huitema, M. Bagnulo, M. Boucadair, and X. Li, "IPv6 Addressing of IPv4/IPv6 Translators," RFC 6052 (Proposed Std), IETF, Oct. 2010.

[32] X. Li, C. Bao, and F. Baker, "IP/ICMP Translation Algorithm," RFC 6145 (Proposed Std), IETF, Apr. 2011, updated by RFC 6791.

[33] M. Bagnulo, P. Matthews, and I. van Beijnum, "Stateful NAT64: Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers," RFC 6146 (Proposed Std), IETF, Apr. 2011.

[34] M. Bagnulo, A. Sullivan, P. Matthews, and I. van Beijnum, "DNS64: DNS Extensions for Network Address Translation from IPv6 Clients to IPv4 Servers," RFC 6147 (Proposed Std), IETF, Apr. 2011.

[35] J. Abley, P. Savola, and G. Neville-Neil, "Deprecation of Type 0 Routing Headers in IPv6," RFC 5095 (Proposed Std), IETF, Dec. 2007.

[36] D. Johnson, C. Perkins, and J. Arkko, "Mobility Support in IPv6," RFC 3775 (Proposed Std), IETF, Jun. 2004, obsoleted by RFC 6275.

[37] C. Partridge and A. Jackson, "IPv6 Router Alert Option," RFC 2711 (Proposed Std), IETF, Oct. 1999, updated by RFC 6398.

[38] S. Krishnan, J. Woodyatt, E. Kline, J. Hoagland, and M. Bhatia, "A Uniform Format for IPv6 Extension Headers," RFC 6564 (Proposed Std), IETF, Apr. 2012.

[39] E. Davies, S. Krishnan, and P. Savola, "IPv6 Transition/Co-existence Security Considerations," RFC 4942 (Informational), IETF, Sep. 2007.

[40] G. Ziemba, D. Reed, and P. Traina, "Security Considerations for IP Fragment Filtering," RFC 1858 (Informational), IETF, Oct. 1995, updated by RFC 3128.

[41] I. Miller, "Protection Against a Variant of the Tiny Fragment Attack (RFC 1858)," RFC 3128 (Informational), IETF, Jun. 2001.

[42] A. Atlasis, "Attacking ipv6 implementation using fragmentation," *Black-Hat Europe*, 2012.

[43] S. Krishnan, "Handling of Overlapping IPv6 Fragments," RFC 5722 (Proposed Std), IETF, Dec. 2009, updated by RFC 6946.

[44] F. Gont, "Processing of IPv6 "Atomic" Fragments," RFC 6946 (Proposed Std), IETF, May 2013.

[45] S. Amante, B. Carpenter, S. Jiang, and J. Rajahalme, "IPv6 Flow Label Specification," RFC 6437 (Proposed Std), IETF, Nov. 2011.

[46] N. Moore, "Optimistic Duplicate Address Detection (DAD) for IPv6," RFC 4429 (Proposed Std), IETF, Apr. 2006.

[47] X. Yang, T. Ma, and Y. Shi, "Typical dos/ddos threats under ipv6," in *Computing in the Global Information Technology, ICCGI 2007. International Multi-Conference on*. IEEE, 2007, pp. 55–55.

[48] P. Nikander, J. Kempf, and E. Nordmark, "IPv6 Neighbor Discovery (ND) Trust Models and Threats," RFC 3756 (Informational), IETF, May 2004.

[49] T. Chown and S. Venaas, "Rogue IPv6 Router Advertisement Problem Statement," RFC 6104 (Informational), IETF, Feb. 2011.

[50] E. Levy-Abegnoli, G. V. de Velde, C. Popoviciu, and J. Mohacsi, "IPv6 Router Advertisement Guard," RFC 6105 (Informational), IETF, Feb. 2011.

[51] R. Draves and D. Thaler, "Default Router Preferences and More-Specific Routes," RFC 4191 (Proposed Std), IETF, Nov. 2005.

[52] F. Gont, "Security Assessment of the Internet Protocol Version 4," RFC 6274 (Informational), IETF, Jul. 2011.

[53] F. Gont and W. Liue, "Security Implications of IPv6 Options of Type 10xxxxxx," Mar 2013. [Online]. Available: tools.ietf.org/html/draft-gont-6man-ipv6-smurf-amplifier-03

[54] J. Arkko, J. Kempf, B. Zill, and P. Nikander, "SEcure Neighbor Discovery (SEND)," RFC 3971 (Proposed Std), IETF, Mar. 2005, updated by RFCs 6494, 6495.

[55] T. Aura, "Cryptographically Generated Addresses (CGA)," RFC 3972 (Proposed Std), IETF, Mar. 2005, updated by RFCs 4581, 4982.

[56] H. Rafiee, A. Alsa'deh, and C. Meinel, "Winsend: Windows secure neighbor discovery," in *Proceedings of the 4th international conference on Security of information and networks*. ACM, 2011, pp. 243–246.

[57] M. Heuse, "Thc-ipv6-attack-toolkit." [Online]. Available: http://www.aldeid.com/wiki/THC-IPv6-Attack-Toolkit

[58] G. Nakibly and F. Templin, "Routing Loop Attack Using IPv6 Automatic Tunnels: Problem Statement and Proposed Mitigations," RFC 6324 (Informational), IETF, Aug. 2011.

[59] G. Nakibly and M. Arov, "Routing loop attacks using ipv6 tunnels," in *Proceedings of the 3rd USENIX conference on Offensive technologies*. USENIX Association, 2009, pp. 7–7.

[60] P. Savola and C. Patel, "Security Considerations for 6to4," RFC 3964 (Informational), IETF, Dec. 2004.

[61] A. Conta and S. Deering, "Generic Packet Tunneling in IPv6 Specification," RFC 2473 (Proposed Std), IETF, Dec. 1998.

[62] I. Gashinsky, J. Jaeggli, and W. Kumari, "Operational Neighbor Discovery Problems," RFC 6583 (Informational), IETF, Mar. 2012.

[63] M. Kohno, B. Nitzan, R. Bush, Y. Matsuzaki, L. Colitti, and T. Narten, "Using 127-Bit IPv6 Prefixes on Inter-Router Links," RFC 6164 (Proposed Std), IETF, Apr. 2011, updated by RFC 6547.

[64] J. Arkko, V. Devarapalli, and F. Dupont, "Using IPsec to Protect Mobile IPv6 Signaling Between Mobile Nodes and Home Agents," RFC 3776 (Proposed Std), IETF, Jun. 2004, updated by RFC 4877.

[65] F. Gont, D. Thaler, and W. Liue, "Deprecating EUI-64 Based IPv6 Addresses," Internet Engineering Task Force, Oct. 2013. [Online]. Available: http://tools.ietf.org/html/draft-gont-6man-deprecate-eui64-based-addresses-00

[66] T. Narten, R. Draves, and S. Krishnan, "Privacy Extensions for Stateless Address Autoconfiguration in IPv6," RFC 4941 (Draft Std), IETF, Sep. 2007.

[67] S. Groat, M. Dunlop, R. Marchany, and J. Tront, "What dhcpv6 says about you," in *Internet Security (WorldCIS), 2011 World Congress on*. IEEE, 2011, pp. 146–151.

[68] D. Malone, "Observations of ipv6 addresses," in *Passive and Active Network Measurement*. Springer, 2008, pp. 21–30.

[69] F. Gont, "Si6 networks' ipv6 toolkit." [Online]. Available: http://www.si6networks.com

[70] P. van Dijk, "Finding v6 hosts by efficiently mapping ip6.arpa." [Online]. Available: http://7bits.nl/blog/posts/finding-v6-hosts-by-efficiently-mapping-ip6-arpa

[71] M. Crawford and B. Haberman, "IPv6 Node Information Queries," RFC 4620 (Experimental), IETF, Aug. 2006.

[72] A. Conta, "Extensions to IPv6 Neighbor Discovery for Inverse Discovery Specification," RFC 3122 (Proposed Std), IETF, Jun. 2001.

[73] T. Chown, "IPv6 Implications for Network Scanning," RFC 5157 (Informational), IETF, Mar. 2008.

[74] E. Davies and J. Mohacsi, "Recommendations for Filtering ICMPv6 Messages in Firewalls," RFC 4890 (Informational), IETF, May 2007.

[75] N. B. Lucena, G. Lewandowski, and S. J. Chapin, "Covert Channels in IPv6," in *Proceedings of the 5th International Conference on Privacy Enhancing Technologies*, ser. PET'05. Springer, 2006, pp. 147–166.

[76] J. Rajahalme, A. Conta, B. Carpenter, and S. Deering, "IPv6 Flow Label Specification," RFC 3697 (Proposed Std), IETF, Mar. 2004, obsoleted by RFC 6437.

[77] J. Lindqvist, "Ipv6 stateless address autoconfiguration considered harmful," in *Military Communications Conference, 2006. MILCOM 2006. IEEE*, 2006, pp. 1–5.

[78] J. D. Howard and T. A. Longstaff, "A common language for computer security incidents," *Sandia Report: SAND98-8667, Sandia National Laboratories*, 1998. [Online]. Available: http://www.cert.org/research/taxonomy_988667.pdf

[79] C. P. Pfleeger and S. L. Pfleeger, *Security in computing*. Prentice Hall Professional, 2003.