IPv6: Nowhere to Run, Nowhere to Hide

Stephen Groat*† Matthew Dunlop*† Randy Marchany† Joseph Tront*
*Bradley Department of Electrical and Computer Engineering
†Virginia Tech Information Technology Security Office
Virginia Polytechnic Institute and State University, Blacksburg, VA 24061, USA
Email: {sgroat,dunlop,marchany,jgtront}@vt.edu

Abstract

Due to a large address space, Internet Protocol version 6 (IPv6) uses stateless address autoconfiguration to assign network addresses to hosts. This unmanaged technique creates a static value derived from the Media Access Control (MAC) address of a network interface as the host portion, or interface identifier (IID). Static IID assignment provides third parties (whether malicious or not) with the ability to track a node's physical location, correlate network traffic with a specific user, and collect details about a node's operating system. Using our live production IPv6 network, we demonstrate not only the feasibility of IID monitoring, but also the ease with which an attacker can accomplish it. We then highlight some possible nefarious applications where IPv6 address tracking and analysis could assist the cyber criminal. In order to prevent this privacy breach, we offer solutions that disassociate the IPv6 address from its user.

1. Introduction

Reports of cyber crime continue to increase at alarming rates each year. The reported monetary loss for 2009 was nearly $560 million. This is over double the $264 million reported in 2008 [14]. As researchers develop techniques to combat cyber crime, attackers work to discover new attack vectors. The key to keeping ahead of these cyber criminals is to anticipate and circumvent potential attack vectors before they are exploited. To that end, it is crucial to examine security in the Internet Protocol version 6 (IPv6).

The current Internet Protocol, version 4 (IPv4), is rapidly running out of address space [19]. IPv6 solves the address space issue by providing nearly $8 \cdot 10^{28}$ times more addresses. Unfortunately, the cost, administrative overhead, and potential security issues have delayed IPv6 deployment. Researchers have even created some stop-gap techniques such as Network Address Translation (NAT), to prolong the life of IPv4. Despite these efforts, however, the majority of the Internet community will be forced to make the transition to IPv6 soon [18]. As with any new protocol, there will likely be undiscovered vulnerabilities. Now is the time, before IPv6 is deployed globally, to discover these vulnerabilities.

One vulnerability that exists is a direct result of how IPv6 forms addresses. In order to reduce administrative burden, IPv6 designers implemented a technique for nodes to generate their own addresses. This technique is called stateless address autoconfiguration. Unfortunately, this method exposes a host's Media Access Control (MAC) address globally. Additionally, the portion of the address formed using the MAC remains static regardless of the network the host connects to. A third party could use this static portion of the IPv6 address to track users from virtually anywhere in the world. The static address even facilitates targeted monitoring of network traffic. The reduced administrative burden is not worth the sacrifice in privacy.

This sort of tracking was not possible in IPv4. In IPv4, a node's MAC address is restricted to the local subnet. Additionally, the MAC address is not associated with the IPv4 address. In fact, the Dynamic Host Configuration Protocol (DHCP) usually issues IPv4 addresses to hosts based on address availability. Furthermore, NAT also provides the unintentional benefit of protecting a host's identity by placing it within a private address space, not globally addressable.

To expose the issues and concerns regarding IPv6 stateless address autoconfiguration, we organize the remainder of the paper as follows. We start our discussion in Section 2 by providing background on IPv6 and some of its known vulnerabilities. Section 3 describes related work regarding security in IPv6 addressing. In Section 4, we demonstrate that IID geotemporal tracking and traffic analysis are indeed possible. We then go on in Section 5 to discuss potential privacy ramifications and applications that may result from IID tracking monitoring. Some methods for protecting a user's privacy are provided in Section 6. In Section 7, we discuss future work. We conclude in

1530-1605/11 $26.00 © 2011 IEEE

Section 8.

2. Background

The Internet Protocol version 6 was developed as a solution to the rapidly depleting IPv4 address space. Addresses were expected to deplete so quickly that the White House issued a mandate directing all government agencies to transition their backbones to IPv6 by June 30, 2008 [10]. Technologies such as NAT, however, have staved off the transition. Two full years after the initial 2008 deadline, the Internet community once again finds itself facing the threat of an IP address shortage [19]. Despite its impending fielding, a large portion of the community is still unfamiliar with IPv6 [6]. In an effort to foster a better understanding of IPv6, we provide a brief overview of some of the main features of IPv6. We also provide some background on how addresses are determined in IPv6.

Like any new protocol, flaws are inherient in the design of IPv6. We discuss some of the main flaws. We also introduce an issue resulting from hosts configuring their own addresses. This is the main issue we focus on because it has been largely glossed over. We feel it is important to discuss in detail since it can lead to a compromise of the IPv6 user's privacy.

2.1. Features of IPv6

As previously mentioned, the address space of IPv6 is larger than IPv4. Where IPv4 allocated 32 bits for the address, IPv6 allocates 128 bits. This equates to approximately $5 \cdot 10^{28}$ addresses for every one of the 6.8 billion people [27] in the world. The 4.3 billion addresses provided by IPv4 is not even enough for one address per person. In today's Internet age, it is not uncommon for a person to have multiple devices connected to the Internet.

Larger address space was not the only improvement made in IPv6. The header format was simplified. Unused fields were removed and the header length in IPv6 was fixed to 40 bytes. Another improvement was moving the options out of the header. In IPv6 options are now located in the payload section of the packet. This allows for more options if desired. It also provides space for the defining of new options. The addition of flow labels was also incorporated into IPv6. Flow labels allow traffic to be classified and potentially handled differently by routers. The final major improvement to IPv6 is the incorporation of IPSec [16]. In IPv4, IPSec is not integrated. It was designed after the protocol was fielded, primarily because security was not initially a concern. When IPSec was integrated into the Open Systems Interconnection (OSI) [28] model, it only fit between the network and transport layers. This creates additional overhead and storage requirements as the data has to travel back up the stack for encryption and authentication in tunnel mode. Including IPSec as part of the network layer provides better efficiency and throughput.

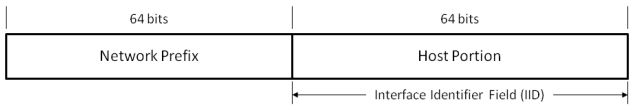

Figure 1. IPv6 128-bit address format

We pointed out the increased address space of IPv6, however, the large address space has the potential to add a significant administrative burden. In an effort to mitigate this burden, IPv6 designers included a method for hosts to configure their own addresses. This technique is referred to as stateless address autoconfiguration.

2.1.1. Automatic addressing in IPv6. A new network address configuration architecture is included in IPv6 to simplify network administration. The two parts that make up this mechanism are a Neighbor Discovery Protocol (NDP) [21] and stateless address autoconfiguration. These two pieces together allow a node to self-determine its IP address. NDP was designed as a replacement for the Address Resolution Protocol (ARP). It facilitates nodes within a particular subnet learning of other nodes on the link using Internet Control Message Protocol version 6 (ICMPv6) messages. Once an NDP message is received, the node uses the network portion of the address to configure the first 64 bits of its IPv6 address. For the last 64 bits, the node automatically configures an address, designated as the IID of the address. The final step combines the 64-bit network address with the 64-bit host address to form a complete 128-bit IPv6 address (See Figure 1).

The Neighbor Discovery Protocol and stateless address autoconfiguration eliminate the need for DHCP adressing services currently implemented on the majority of IPv4 networks. DHCP implements a client-server architecture in which a DHCP server assigns addresses to clients and keeps state of which addresses have been assigned to particular clients. DHCP has also been implemented in IPv6 in the form of DHCPv6. The sparse address space and the ease of address autoconfiguration, however, make DHCPv6 addressing an unnecessary service. The extra expense and network complexity involved in DHCP addressing have been removed with NDP and stateless IPv6 addressing.

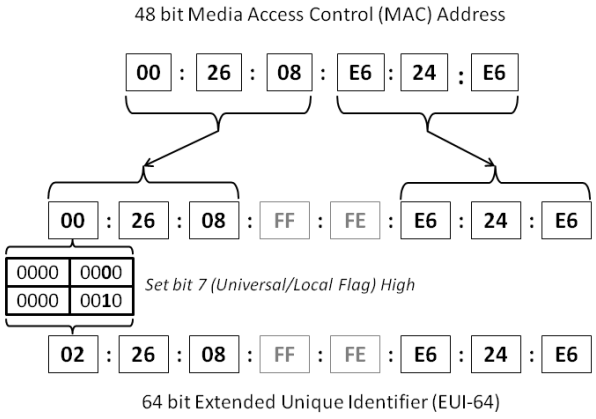

Figure 2. 64-bit Extended Unique Identifier (EUI-64) format

2.1.2. Deterministic IID. As mentioned in Section 2.1.1, the IID is automatically configured by the host. The current accepted definition of the stateless address autoconfiguration on most operating systems provides an IID that is deterministic across networks. The IID makes up the last 64 bits of the IPv6 address and is automatically configured by the host based upon the MAC address of its network interface. This is accomplished by extending the 48-bit MAC address to 64 bits through the EUI-64 format [12]. The EUI-64 format splits the 48-bit MAC address into two 24-bit halves. The 16-bit hex value `0xFFFE` is inserted between the two halves to form a 64-bit address. Also, the universal/local flag, located at bit seven of the 64-bit host portion, is set to 1. This process is illustrated in Figure 2.

2.2. Threats in IPv6

Many of the same threats that plague IPv4, such as man-in-the-middle attacks, sniffing, flooding, and application layer attacks, are still possible in IPv6 [5]. Though IPSec is integrated into the protocol, it is not required. With no authentication or encryption required in the protocol, IPv6 is susceptible to man-in-the-middle attacks and network traffic sniffing. Flooding, whether to deny service or to attempt a buffer overflow attack, is also still possible in IPv6. Finally, all application layer attacks are still applicable since IPv6 simply transports any application layer data protocols in the same way as IPv4.

As with any new protocol, undiscovered threats exist. Examples of these threats include covert channels and exploitations of transition mechanisms. Another area that researchers tend to overlook are vulnerabilities that are not traditionally classified as attacks. One such vulnerability was discussed in Section 2.1.1. Allowing hosts to automatically form addresses relieves some of the administrative burden, but causes additional privacy issues that warrant further study.

2.2.1. Covert channels. Most of the current commercial Intrusion Detection Systems (IDS) and Intrusion Prevention Systems (IPS) cannot effectively recognize or stop IPv6 threats, specifically covert channels. As IPv4 threats are recognized and signatures are created for IDSs and IPSs, many of these threats are evolving into IPv6 threats to avoid detection. By entering a system on IPv4 and then using an IPv6 covert channel to control the system, viruses and trojan horses are mutating to bypass the current commercial IDSs and IPSs. Since most IDSs and IPSs default to allowing traffic originating from within a network, IPv6 covert channels created by attackers often go undetected when blocking inbound IPv6 connections. While individual systems may deploy firewall rules which block all IPv6 traffic, corporate networks cannot take such a draconian approach towards IPv6 without a potential loss in revenue. As IPv6 deployment quickly becomes necessary to support connectivity, the lack of threat signatures allows for covert channels to easy bypass most IDSs and IPSs.

2.2.2. Transition mechanisms. To assure a smooth transition to IPv6, tunnels and other transition mechanisms have been created to allow IPv6 traffic to properly flow over IPv4 networks. Many internal networks may not implement IPv6, yet they require compatibility to assure connectivity. Automatic and manual tunneling protocols have been created to address the need for IPv6 compatibility. 6to4 tunneling creates automatic tunnels at router endpoints using embedded addressing [22]. Teredo is an automatic tunneling technique which encapsulates IPv6 packets in UDP IPv4, defaultly enabled in Windows Vista [13]. ISATAP, another automatic tunneling protocol, creates local IPv6 networks over an IPv4 networks by internally mapping IPv6 addresses to IPv4 nodes [25]. Finally, 6in4, a manually configured tunnel, uses a similar technique as 6to4 tunneling, but with manually configured router endpoints [22]. IPv6 transition mechanisms use a mixture of IPv4 and IPv6 features to create compatibility.

IPv6 transition mechanisms create security risks and are vulnerable to attacks against IPv4, IPv6, and the tunneling protocols. IPv4 tunnels allow IPv6 to bypass firewalls and other security measures on IPv4 networks. Since most networks using tunnels will not have specific IPv6 security measures enabled, tunnels create unmonitored holes in the network's security and can be used for attack. Exploiting the hosts or

tunnel endpoints allow attacks on both protocols. One example is ping flooding an IPv6 tunnel through IPv4. By targeting the tunnel, both protocols are simultaneously stressed at the endpoint, increasing the chance of a successful attack. Also, specific attacks against tunneling protocols, such as changing the autoconfigured tunnel endpoints to malicious hosts acting as men-in-the-middle, are hard to detect and devastating to security. IPv6 transition mechanisms have created significant security risks, bypassing current security measures and creating new vectors for attack.

2.2.3. Static addressing. While operating systems (OS) configure IPv6 addresses differently, no current OS implementations of IPv6 stateless addressing dynamically obscure the IID of all IPv6 addresses on the system. OS X and common Linux distributions, such as CentOS and Ubuntu, follow the EUI-64 format. The MAC address appears virtually unaltered in the IPv6 address. The Windows operating system obscures the host portion of an IPv6 address according to RFC 4941 and sets a temporary address [20, 26]. However, Window operating systems also carry another IPv6 address used for neighbor solicitation. This other IPv6 address contains an IID that is obscured but never changes, regardless of the subnet the node connects to. Not dynamically obscuring a user's host portion of all of the IPv6 addresses associated with a system threatens a user's privacy. The static IID currently implemented in major operating systems can be identified with a particular node, even as the node changes networks.

Many mobile devices, such as Android and iPhone, support IPv6 in WiFi. Their implementations follow the EUI-64 format providing these mobile devices with static IIDs that are easily tracked on their WiFi connections. Since most users frequently carry their mobile devices and leave them on and connected, the ability to track a user is increased dramatically. While the need to address the privacy concerns in Mobile IPv6 has been identified, it does little good until the privacy concerns due to IID tracking are addressed. Since Mobile IPv6 would only be applied to the cellular connections and the majority of these wireless devices also deploy WiFi, users can still be tracked through their wireless devices as they move between different WiFi networks. Therefore, address privacy must be addressed in all connections of a mobile device to assure complete privacy.

3. Related work

A significant amount of work examines the format and structure of IPv6 addresses. Researchers have concluded that using the 64-bit Extended Unique Identifier (EUI-64) format in Mobile IPv6 could compromise a user as subnets move with users following Network Mobility (NEMO). What makes our work unique is that we show how both mobile and stationary nodes connected to the Internet can be geographically tracked and identified through traffic analysis. This is due to the deterministic implementation of IID generation. This capability exists for both the EUI-64 format and the onetime hash used by Windows operating systems.

The realization that using a MAC within the IID of an IPv6 address can potentially reveal information about a user is not in and of itself novel. Narten et al. discussed this problem in RFC 4941 and concluded that a non-changing interface would allow an eavesdropper to correlate unrelated information with a particular node [20]. Haddad even goes so far as to address the fact that mobile nodes using IPv6 stateless address autoconfiguration can reveal their location to an eavesdropper [11]. Our work builds on these ideas by discussing and demonstrating how an interested party can eavesdrop on a user from anywhere on the Internet using basic network tools. Additionally, we identify multiple ways users' stateless autoconfigured addresses can be exploited in their current implementations and offer solutions to protect users' privacy.

Some work addresses issues related to potential privacy problems with regards to Mobile IPv6. Koodli discusses how a mobile node's home or care-of address can be used to reveal that the mobile node has roamed [17]. Castelluccia et al. and Qiu et al. also discuss how mobile nodes can be tracked using their home and care-of addresses [3, 23]. While in principle these concepts relate to privacy concerns with tracking of IPv6 node location, they focus on a completely unrelated vulnerability. Additionally, the vulnerability we address affects both mobile and stationary IPv6 nodes. Mobile IPv6 is still in development and has not been reliably implemented on a production network. Privacy concerns associated with the standard IPv6 must be addressed before Mobile IPv6 can be secured.

4. Testing

In order to prove geotemporal tracking and traffic analysis through IID analysis, testing was performed using IPv6 nodes on a live IPv6 network. Geotemporal tracking and traffic analysis were performed on an Android mobile device on the Virginia Tech wireless network with the cellular connection turned off. Since the Android operating system deploys an EUI-64 IID

Figure 3. Geotemporal plot of a wireless node's movement within the Virginia Tech network

in stateless address autoconfiguration on WiFi IPv6, both geotemporal tracking and traffic analysis were possible.

Testing was conducted on a production IPv6 network. Virginia Tech has a fully functional IPv6 network, providing globally unique addresses through stateless address autoconfiguration to every wireless and wired node on the network. The network contains six core routers which serve distinct geographic areas on campus. Subnets correspond with the core routers and, therefore, with distinct geographic locations on the Blacksburg campus. Packet capture for traffic analysis was performed at the border to assure that all traffic sent from different subnets could be captured and analyzed. Geographic tracking and traffic analysis were performed on the IPv6 network through the use of subnet analysis, network sniffing, and IID analysis.

4.1. IID tracking

Reconnaissance on the campus at Virginia Tech located six unique wireless subnets. A simple script was created to continuously ping a specific IID on these six subnets and to record the date and time when the node successfully responded. Using a wireless node which was set to automatically associate with Virginia Tech's wireless network, the script was run while the node moved around the campus and associated with different access points. The results of this testing can be seen in Figure 3.

Geographically tracking users with static IIDs is possible on any stateless autoconfiguration network. While this example only demonstrates tracking node movement on the Blacksburg campus of Virginia Tech, tracking could easily be expanded to cover other geographical areas that support IPv6. Currently, expansion is not possible due to the lack of production IPv6 networks outside of campus. By predetermining the network portion of the IPv6 addresses within an area of interest (e.g., a metropolitan area), an attacker can remotely scan for a user on any network and accurately determine the user's location. Tracking is also possible without knowledge of the network addresses or the subnets. An attacker could accomplish this by using ping sweep for the node and traceroute to determine location. However, the accuracy of the determined location is degraded and the time necessary to find a user is increased without reconnaissance.

4.2. IID traffic analysis

To perform traffic analysis on IPv6 IIDs, data was sent using an Android OS mobile device from multiple subnets at different times. We used the same wireless subnets that we used to track user locations in Section 4.1. At the network border, a sensor was placed to monitor, sniff, and record all IPv6 traffic traveling over the network. The primary traffic collected was Google search queries. This is due to Google having a AAAA Domain Name System (DNS) record on the network at Virginia Tech which returns an IPv6 address. Other traffic collected included YouTube search queries, Jabber client transmissions, and Gmail data.

The use of Transport Layer Security (TLS) inhibits IID traffic analysis since the data is encrypted for transmission. Some types of traffic, such as webmail traffic, bank traffic, and chat protocols, are often encrypted by default to prevent personally identifiable information (PII) from being intercepted by an attacker. Other traffic, such as search queries, social media posts, and daily browsing habits, have historically been unencrypted due to the extra bandwidth and processing required by TLS. Transmitting this traffic unencrypted allows for attackers to intercept data. Collected PII from this unencrypted data can be used to build a profile of users and determine the users' identities. For example, we were able to monitor the user name and tweets sent by our test node over Twitter. Since many Twitter users tweet about their activity and location, it would be trivial for an attacker to identify a user through their Twitter traffic. As protecting PII becomes more important due to increases in identity theft and related crimes, industry will likely respond by implementing encryption in everyday browsing activities. The introduction of more security focused websites, such as Google search through TLS, will make identifying a user through search traffic analysis increasingly more difficult. Until such a time, the large volume of unencrypted PII makes it relatively easy for attackers to exploit users

Table 1. Top five NIC manufacturers accessing Virginia Tech's network from EUI-64 systems

NIC Manufacturer	EUI-64 Traffic
Apple, Inc.	86.33%
Broadcom Corporation	5.23%
Intel	2.47%
3 Com Corporation	1.58%
Dell	0.42%

through traffic analysis.

IID traffic analysis of users utilizing Windows operating systems is extremely difficult due to the use of privacy extensions [20]. Since the privacy extensions are configured to automatically change the IID of a node at specific time intervals and as a node changes networks, it is impossible to use only the IID of a temporary address to analyze traffic. It is possible, however, to collect network traffic that most likely contains the target node's traffic. This is accomplished by analyzing the Time To Live (TTL) values of ping and traceroute packets sent to a node's permanent address and scanning for similar TTL values in packets sniffed on the network. This technique will, however, also contain many other nodes since TTL values may not vary for large portions of a network.

4.2.1. OUI Analysis. A MAC address is composed of two parts. The first half is called the organizationally unique identifier (OUI) and identifies the vendor that issues the network interface card (NIC). The second half is assigned by the vendor and is designed to act like a serial number to make the MAC unique [7]. Usually, agencies will contract a vendor to produce all of a specific product. This means that, with high probability, an entire product line shares the same OUI.

We conducted some analysis on the OUIs of the captured traffic to determine the types of computers and operating systems communicating on our network using IPv6. Of the 72,377 IPv6 addresses collected in 24 hours, 12,356 were expanded with the EUI-64 expansion format. Since Windows obscures the IPv6 address using privacy extensions, we reason that the other 60,021 addresses are Windows systems. It is worth noting that the 12,356 EUI-64 addresses could contain a small margin of error since it is possible that privacy extensions could produce IIDs that mimic valid EUI-64 expanded addresses. We conclude, therefore, that approximately 83% of the network at Virginia Tech is comprised of Windows systems while the remaining 17% is made up of systems running some other operating system.

For the 17% of systems utilizing the EUI-64 expanded addresses, we analyzed the OUI of each IID and compiled a list of the top five manufacturers as seen in Table 1. The large majority of these systems had wireless NICs registered to Apple, Inc. Since no current mobile Apple operating system deploys IPv6, all of the IPv6 traffic containing Apple OUIs comes from Apple computers. The remaining devices in the OUI analysis are registered to network interface manufactures. These OUIs most likely come from Linux and Unix systems using the default EUI-64 expansion format.

OUI analysis on collected traffic allows attackers to determine the most effective types of attacks to run on a specific network. To effectively use resources to gain entry into a network, attacks should be run against the most common operating systems on a network. For example, the OUI analysis on the network at Virginia Tech shows that the majority of computers run Windows. To effectively launch attacks against a Windows machine in IPv6, attackers must obtain the permanent address of the machine. Therefore, a local device must be connected to the network that listens for the Neighbor Solicitation messages and any other multicast messages which use the permanent addresses of the Windows systems. While tools such as Nmap and Metasploit offer OS fingerprinting, these tools would waste resources scanning the large IPv6 address space and would return invalid, temporary addresses. Analysis of OUIs from captured traffic gives attackers a new tool to effectively collect statistics on system types connecting to a network.

An attacker could also use OUI analysis to locate all of a specific type of asset. This provides an attacker with the locations and numbers of specific types of systems. This may not seem that powerful. Imagine, however, that an attacker is able to identify a vulnerability specific to a particular brand of device. The attacker can target an exploit against those devices specifically. This type of attack may provide an attacker with another vector into critical or sensitive systems.

5. Privacy implications

The static IID created by the EUI-64 format and the Windows operating systems compromises a user's privacy. Creating a static IID from a MAC address allows nodes to be logically and geographically tracked as they travel to different networks. Since the EUI-64 format results in a deterministic IID, users can be tracked on a network by scanning different subnets and searching for the MAC-generated IIDs. Using simple commands such as ping and traceroute, the location of a user can be determined within reasonable

geographic accuracy. Even the Windows obscuration of the IPv6 IID does not protect a user. By locally capturing a user's traffic once, a specific user can be paired with the deterministically obscured IID and tracked with the same technique of searching subnets as used for unobscured host addresses. Since the obscuration occurs independent of the network, a Windows host carries the same obscured IID between networks.

By monitoring the traffic on a network over an extended period of time, a single user's traffic can be identified and analyzed. Armed with this data, a third party (whether malicious or not) can potentially tie a device to its actual user. As the user crosses different subnets, traffic can be collected and correlated by examining the static IID. This vulnerability to tracking does not typically apply when using IPv4. Most medium to large IPv4 networks implement DHCP, which changes user addresses randomly. As a result, DHCP logs are needed to tie traffic sniffed from a network with a particular user. Due to the deterministic IID in IPv6 address autoconfiguration, simple filters could be created to filter out the traffic of a single user on any subnet. This would allow an interested party to identify and monitor a user's on-line activity through traffic analysis. In a dual-stack implementation where a node uses a mixture of IPv4 and IPv6, special ICMPv6 Neighbor Solicitation messages provide an interested party with the IPv4 address linked to an IPv6 address. This correlation allows for traffic collection to extend to IPv4 for a single session.

Tracking a user or monitoring his/her on-line activity is not the only concern. If it was known that location or traffic monitoring was occurring, a malicious host could spoof the IID of an innocent node. The malicious node could then masquerade as the innocent node and create false traffic or locations using the innocent node's IID.

Applications of static IID analysis, both geotemporal tracking and traffic analysis, provide opportunities for cyber crime. Used by a cyberstalker, IID analysis could help to track and monitor a victim. The same broadcast information could also be used by a terrorist to find effective targets.

Cyberstalking using common methods involves an active effort on the part of the stalker. Additionally, there is a level of expertise required; the average cyberstalker will not be able to hack into a victim's machine or even place and run a packet sniffer. Assuming that the cyberstalker possesses the skills to accomplish these tasks, there is the risk of the intrusion being traced back to the cyberstalker. There is also the possibility that the victim has a secured machine, making penetration difficult. A mobile victim makes the cyberstalker's task more difficult. As a victim moves between subnets, the DHCP address provided is not logically connected to the victim. If the cyberstalker does not have a physical presence on the victim's machine, tracking the victim becomes much more difficult.

Stateless address autoconfiguration in IPv6 alleviates many of the challenges and risks for a cyberstalker. Since the IID is static, the cyberstalker always knows half of a victim's IP address. The other half, the subnet portion, is easily discovered by conducting a thorough reconnaissance of the geographic areas frequented by the victim. Armed with the IID and subnets, the cyberstalker can continually ping the likely subnets for the victim's IID. A successful ping reply indicates that the victim is at that specific location. This form of attack will not alert the victim, yet will provide the cyberstalker with the victim's daily movements. The attacker may even be able to establish a movement pattern for the victim, which could be used to plan a physical assault, burglary, or other crime. The proliferation of handheld network capable devices aids a cyberstalker's ability to keep constant tabs on victims.

A potentially disastrous application of IPv6 stateless address autoconfiguration is toward forwarding the goals of terrorists. The best way to cause fear is through surprise and shock value. Terrorists can take advantage of the tracking capabilities provided by static IIDs to plan and organize attacks without attracting attention.

Since users can be easily identified through their IIDs, terrorists can use the IID to target individuals. A terrorist can use traffic analysis as described in Section 4 to pair a target with his/her IPv6 IID. The terrorist can then use cyberstalking techniques to develop a movement pattern for the target. Once a pattern is discerned, the terrorist can choose the most suitable location to trap the target and wait. If the terrorist's intention is assassination, the chosen location can be set up to remotely detonate an explosive device when the victim's IID is detected. If the goal is fear, a similar trap can be sprung remotely. In either case, the terrorist can be miles away and evade capture. This type of attack would be difficult to detect and has the potential to cause widespread panic.

6. Privacy protection

Regardless of the intent behind IID tracking, users are entitled to the expectation that their privacy will be maintained. The argument exists that if a person has nothing to hide, then they should not care if they are monitored [24]. Aside from the malicious possibilities we mentioned in Section 5, most people just

like privacy. When a homeowner constructs a privacy fence in their backyard, are we to assume that it is because they want to participate in some sort or illicit or illegal behavior? Similarly, if we do not plan to discuss illegal activities, should we agree to having all our phone conversations recorded? The answer to both of these hypothetical scenarios is emphatically, "No." Therefore, it is important to prevent IID tracking before IPv6 is globally deployed.

To that end, there are four different approaches to preventing IID tracking in IPv6. The first approach uses cryptographically generated addresses [2] while the second obscuration approach uses what are called privacy extensions [20]. Another way of preventing IID tracking is through the use of the Dynamic Host Configuration Protocol for IPv6 (DHCPv6) [8]. The final method is through the use of the Encapsulating Security Payload (ESP) for Internet Protocol Security (IPSec) [15].

One method of obscuring the IPv6 IID is through the use of cryptographically generated addresses (CGAs). In general, a CGA is formed by hashing the sender's public key along with some other parameters [2]. The original purpose of CGAs was to prevent denial of service attacks against the Secure Neighbor Discovery (SEND) protocol [1]. Since CGAs also dynamically obscure IPv6 autoconfigured addresses, they can be applied as a defense against IPv6 address tracking.

The main disadvantage to using CGAs is the computational cost. Producing an acceptable CGA involves the generation of two hash values, Hash2 followed by Hash1. The complexity of generating Hash2 depends on the strength of a security parameter (Sec). The security parameter can take on any value from 0-7 and indicates the number of leading zeros Hash2 must contain. The number of zeros is determined by multiplying Sec by 16. On average, it takes $O(2^{16 \cdot Sec})$ iterations to generate Hash2. Once an acceptable Hash2 is computed, Hash1 is generated using some of the final Hash2 parameters as well as the subnet prefix. The leftmost 64 bits of Hash1 are used as the IID with the exception of five bits used for other purposes [2]. At this point, duplicate address detection is conducted [1]. If three duplicate addresses are detected, the IID is rejected and the process starts anew. The large number of hash calculations required to generate a CGA could quickly overwhelm a power-constrained device.

Privacy extensions provide another means of obscuring a user's IID. Privacy extensions generate a random IID by hashing the concatenation of a user's EUI-64 IID with a 64-bit "history value" and taking the leftmost 64 bits. The "history value" is initially produced from the leftmost 64 bits of a pseudo-random number. From this point, "history values" are produced using previously calculated IIDs. Using "history values" instead of pseudo-random numbers for each IID calculation limit the number of duplicate address collisions that occur due to only using 64-bits of the hash. If a duplicate address is detected, a new "history value" is formed and the process is repeated [20].

The disadvantages of using privacy extensions are not as severe as those of using CGAs. Assuming no address collisions, only one hash calculation is required of the sender to produce an obscured IID. Privacy extensions also carry parameters to limit the time an obscured IID remains valid. Unfortunately, the default values of these parameters are set too long. It is feasible for an IID using privacy extensions to remain static for as long as one week. During this time period, a malicious node could still successfully profile a target host. Fortunately, RFC 4941 allows users to modify these defaults [20].

The main factor that makes IID obscuration an attractive solution for hiding IPv6 addresses is that there is no need for any management overhead. Obscuration and verification both occur at the respective end hosts with no intervention by a trusted third party required. Although CGAs use a public key, the key is self-generated by the sender [2]. Privacy extensions use a history value that is generated based on a pseudo-random number. This lack of management makes IID obscuration scalable.

The Dynamic Host Configuration Protocol for IPv6 (DHCPv6) [8] is not an IID obscuration technique but rather a means to provide stateful address configuration. In IPv6, DHCPv6 works very similar to DHCP in IPv4. Instead of clients generating their own addresses, a DHCP server leases temporary addresses to clients. The main advantage of this technique is that the addresses issued by the DHCP server are typically not tied to the identity of the clients. In principal, each time a client connects to a network, the DHCP server could issue a new address. Unfortunately, this does not happen in practice. RFC 3315 promotes the issuance of non-temporary addresses to clients. Clients do have the ability to request temporary addresses, which mask their location and activities globally. Locally, however, an attacker can still track clients through a DHCP Unique Identifier (DUID) that is transmitted between the client and server. The scope of this method of tracking is limited to the subnet of the client, server, or any relays [8]. There is also an administrative management burden that accompanies the use of DHCPv6.

Internet Protocol Security (IPSec) also provides a means to protect users from tracking. In IPSec, this is accomplished through the use of encapsulating se-

curity payload (ESP) in tunnel mode. The reason this hides the identity of the target node from being tracked is that the target node's entire packet, including address, gets encrypted. This encrypted portion then becomes part of the payload of a new packet using the address of the tunnel start point [15]. Of course, the tunnel start point cannot be the same as the target host or tracking will again be possible. One major advantage to using IPSec in tunnel mode is that the cryptographic burden of encryption and decryption is offloaded to the tunnel endpoints. This is extremely beneficial for power constrained devices.

There are, however, a number of severe disadvantages to using IPSec as a privacy protection mechanism. The most striking is that IPSec used in this way requires a global key management infrastructure that does not currently exist [4]. Another disadvantage is that IPSec in tunnel mode only protects target nodes from those nodes external to the tunnel. Nodes residing on the same subnet as either tunnel endpoint will still be able to track the target nodes. This may provide a slight obstacle to the majority of malicious nodes, but will provide no obstacle to administrators. Depending on a user's point of view, this could be seen as either positive or negative.

While these prevention techniques seem simple and easy to implement, there are obstacles to implementation. The extra overhead and decreased performance required for frequent hash calculations [9] in embedded devices has caused IID obscuration to be ignored. DHCPv6 will not be implemented any time soon due to the additional network configuration and equipment needed. Also, since the lack of DHCP is touted as a feature of IPv6, few network administrators are choosing to implemented the service. IPSec suffers obstacles to global deployment because of the requirement for a global key management infrastructure. As more embedded devices become Internet capable, a user's identity becomes easer to determine as more attributes of a user are sent over the Internet. Regardless of the obstacles, some method of prevention should be implemented now. Of those mentioned, obscuration through the use of privacy extensions seems to be the best option in terms of benefits versus computational cost.

7. Future work

There are many ways that systems using stateless address autoconfiguration can be exploited, both negatively and positively. In future research, we will explore in detail some of the fields that could be impacted. Aside from cyber stalking and terrorism applications, there are undoubtedly other malicious ways a cyber criminal can exploit a user's or organization's privacy. On the other hand, similar types of information can be gathered by law enforcement officials for forensic analysis. Regardless of the intent, most users will not feel comfortable knowing that their activities can be monitored.

Another future phase of work involving IPv6 privacy involves examining the DCHPv6 protocol and associated weaknesses in privacy [8]. In the configuration steps of the DHCPv6 protocol, DHCPv6 relies on a DHCP Unique Indentifier (DUID). While the DUID is not based off the MAC address or deterministic, the DUID is static and persists over different networks and over time. On the local link or any of the relay links, the DUID can be used to identify a node. Further testing and examination of the protocol is needed to determine if this analysis can also be done globally according to the specification.

8. Conclusion

As the inevitable deployment of IPv6 quickly approaches, this untested protocol opens the door for new cyber crime opportunities. The potential for IPv6 stateless addressing to be used for cyberstalking or even terrorism is proof enough that there are issues that need to be addressed. Combined with the threat of rehashed IPv4 network attacks, the IPv6 Internet may be an unstable, insecure platform for mission critical applications. Before IPv6 is extensively deployed, research must be conducted to ensure privacy and security concerns are protected.

Though simple methods of obscuring IIDs have been identified, no current method decreases system overhead enough to protect mobile systems. To protect users' location and identity, the privacy extensions described by RFC 4941 [20] provide the optimal solution, obscuring the IID with minimal computational overhead. Obscuration through privacy extensions is certainly feasible in unconstrained systems. However, the extra power and computational demand required for IID obscuration may be unacceptable in embedded and mobile device, where geotemporal privacy is particularly vital. Static address components due to stateless addressing must be corrected before IPv6 threatens users' privacy.

9. References

[1] J. Arkko, J. Kempf, B. Zill, and P. Nikander. SEcure Neighbor Discovery (SEND). RFC 3971 (Proposed Standard), Mar. 2005.

[2] T. Aura. Cryptographically Generated Addresses (CGA). RFC 3972 (Proposed Standard), Mar. 2005. Updated by RFCs 4581, 4982.

[3] C. Castelluccia, F. Dupont, and G. Montenegro. A simple privacy extension for mobile IPV6. In *Mobile and Wireless Communication Networks, IFIP TC6 / WG6.8 Conference on Mobile and Wireless Communication Networks (MWCN 2004)*, pages 239–249, Oct. 2004.

[4] A. Choudhary. In-depth analysis of IPv6 security posture. In *The 5th International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom 2009)*, pages 1 – 7, Nov. 2009.

[5] S. Convery and D. Miller. IPv6 and IPv4 threat comparison and best-practice evaluation, 2004.

[6] M. Courtney. What's holding up IPv6? *Computing*, Mar. 2009.

[7] R. Deal. *Cisco Certified Network Associate Study Guide (Exam 640-802)*. The McGraw-Hill Companies, 2008.

[8] R. Droms, J. Bound, B. Volz, T. Lemon, C. Perkins, and M. Carney. Dynamic Host Configuration Protocol for IPv6 (DHCPv6). RFC 3315 (Proposed Standard), July 2003. Updated by RFCs 4361, 5494.

[9] M. Durvy, J. Abeillé, P. Wetterwald, C. O'Flynn, B. Leverett, E. Gnoske, M. Vidales, G. Mulligan, N. Tsiftes, N. Finne, and A. Dunkels. Making sensor networks IPv6 ready. In *SenSys '08: Proceedings of the 6th ACM conference on Embedded network sensor systems*, pages 421–422, New York, NY, USA, 2008.

[10] K. S. Evans. Transition planning for Internet Protocol version 6 (IPv6). Available at: `http://www.whitehouse.gov/omb/memoranda/fy2005/m05-22.pdf`, Aug. 2005. Memorandum M-05-22.

[11] W. Haddad. Privacy for mobile and multi-homed nodes: MoMiPriv problem statement, 2005.

[12] R. Hinden and S. Deering. IP Version 6 Addressing Architecture. RFC 4291 (Draft Standard), Feb. 2006.

[13] C. Huitema. Teredo: Tunneling IPv6 over UDP through Network Address Translations (NATs). RFC 4380 (Proposed Standard), Feb. 2006.

[14] Internet Crime Complaint Center. 2009 Internet crime report. Available at: `http://www.ic3.gov/media/annualreport/2009_IC3Report.pdf`, 2010.

[15] S. Kent and R. Atkinson. IP Encapsulating Security Payload (ESP). RFC 2406 (Proposed Standard), Nov. 1998. Obsoleted by RFCs 4303, 4305.

[16] S. Kent and K. Seo. Security Architecture for the Internet Protocol. RFC 4301 (Proposed Standard), Dec. 2005.

[17] R. Koodli. IP Address Location Privacy and Mobile IPv6: Problem Statement. RFC 4882 (Informational), May 2007.

[18] J. Leyden. Net shakeup looms as IPv4 resources start running low. *The Register*, June 2010.

[19] R. L. Mitchell. The grill: John Curran. *ComputerWorld*, Apr. 2010.

[20] T. Narten, R. Draves, and S. Krishnan. Privacy Extensions for Stateless Address Autoconfiguration in IPv6. RFC 4941 (Draft Standard), Sept. 2007.

[21] T. Narten, E. Nordmark, W. Simpson, and H. Soliman. Neighbor Discovery for IP version 6 (IPv6). RFC 4861 (Draft Standard), Sept. 2007.

[22] E. Nordmark and R. Gilligan. Basic Transition Mechanisms for IPv6 Hosts and Routers. RFC 4213 (Proposed Standard), Oct. 2005.

[23] Y. Qiu, J. Zhou, F. Bao, and R. Deng. Protocol for hiding movement of mobile nodes in Mobile IPv6. In *62nd IEEE Vehicular Technology Conference*, volume 2, pages 812 – 815, Sept. 2005.

[24] D. J. Solove. 'I've got nothing to hide' and other misunderstandings of privacy. *Social Science Research Network Working Paper Series*, July 2007.

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

[26] Introduction to IP version 6. Available at: `http://download.microsoft.com/download/e/9/b/e9bd20d3-cc8d-4162-aa60-3aa3abc2b2e9/ipv6.doc` accessed on 24 May 2010.

[27] U.S. & World population clocks. Available at: `http://www.census.gov/population/www/popclockus.html/` accessed on 4 Mar 2010.

[28] H. Zimmermann. OSI reference model–the ISO model of architecture for open systems interconnection. *Communications, IEEE Transactions on*, 28(4):425 – 432, Apr. 1980.