Privacy and Security in IPv6 Addressing

Stephen L. Groat

Thesis submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Master of Science
in
Computer Engineering

Joseph G. Tront, Chair
Randolph C. Marchany
Scott F. Midkiff

April 21, 2011
Blacksburg, Virginia

Keywords: IPv6, Network Addressing, Privacy, Security
Copyright 2011, Stephen L. Groat

Privacy and Security in IPv6 Addressing

Stephen L. Groat

ABSTRACT

Due to an exponentially larger address space than Internet Protocol version 4 (IPv4), the Internet Protocol version 6 (IPv6) uses new methods to assign network addresses to Internet nodes. StateLess Address Auto Configuration (SLAAC) creates an address using a static value derived from the Media Access Control (MAC) address of a network interface as host portion, or interface identifier (IID). The Dynamic Host Configuration Protocol version 6 (DHCPv6) uses a client-server model to manage network addresses, providing stateful address configuration. While DHCPv6 can be configured to assign randomly distributed addresses, the DHCP Unique Identifier (DUID) was designed to remain static for clients as they move between different DHCPv6 subnets and networks. Both the IID and DUID are static values which are publicly exposed, creating a privacy and security threat for users and nodes.

The static IID and DUID allow attackers to violate unsuspecting IPv6 users' privacy and security with ease. These static identifiers make geographic tracking and network traffic correlation over multiple sessions simple. Also, different classes of computer and network attacks, such as system-specific attacks and Denial-of-Service (DoS) attacks, are easier to successfully employ due to these identifiers. This research identifies and tests the validity of the privacy and security threat of static IIDs and DUIDs. Solutions which mitigate or eliminate the threat posed by static identifiers in IPv6 are identified.

Acknowledgments

I would like to thank the members of my committee, Dr. Joseph Tront, Mr. Randy Marchany, and Dr. Scott Midkiff, for their guidance and support of this research. Their mentorship and assistance allows me to focus my research and succeed in my different academic and professional endeavors. Specifically, Dr. Tront's assistance with publishing my academic research and Mr. Marchany's support in providing space and resources in the IT Security Office and Lab continues to be invaluable.

I would also like to thank Lt. Colonel Matthew Dunlop for his assistance and guidance in the graduate process. Working with LTC Dunlop sparked a passion for graduate work and network security research and has motivated me to continue my studies.

Members of the IT Security Office have also been a valuable resource. Will Urbanski, Phil Kobezack, and Rich Sparrow provide teamwork, insight and experience in addressing many problems. Also, I would like to acknowledge additional former and current members of the IT Security Lab, including Jon Paul Dunning, David Shelly, Brittany Clore, and Jennifer McGuire, for their friendship and support.

Finally, the assistance of Phil Benchoff, Carl Harris, and others in Communication and Network Services at Virginia Tech has been essential. Their knowledge of IPv6 and willingness to teach and to share allowed me to complete my research at an accelerated pace.

Contents

List of Figures

List of Tables

List of Abbreviations

6rd . . . . . .IPv6 Rapid Deployment on IPv4 Infrastructures

ARP . . . . . .Address Resolution Protocol

CALEA . . . . . .Communications Assistance for Law Enforcement Act

CGA . . . . . .Cryptographically Generated Address

DAD . . . . . .Duplicate Address Detection

DHCP . . . . . .Dynamic Host Configuration Protocol

DHCPv6 . . . . . .Dynamic Host Configuration Protocol version 6

DNS . . . . . .Domain Name System

DNSSEC . . . . . .DNS Security Extensions

DoS . . . . . .Denial-of-Service

DUID . . . . . .DHCP Unique Identifier

ESP . . . . . .Encapsulating Security Payload

EUI . . . . . .Extended Unique Identifier

FPGA . . . . . .field programmable gate array

HTTP . . . . . .Hypertext Transfer Protocol

IANA . . . . . .Internet Assigned Numbers Authority

ICMPv6 . . . . . .Internet Control Message Protocol version 6

IDS . . . . . .Intrusion Detection System

IID . . . . . .interface identifier

IP . . . . . .Internet Protocol

IPS . . . . . .Intrusion Prevention System

IPsec . . . . . .Internet Protocol Security

IPv4 . . . . . .Internet Protocol version 4

IPv6 . . . . . .Internet Protocol version 6

ISATAP . . . . . .Intra-Site Automatic Tunnel Addressing Protocol

ISP . . . . . .Internet Service Provider

LAN . . . . . .local area network

LLMNR . . . . . .Link-local Multicast Name Resolution

MAC . . . . . .Media Access Control

mDNS . . . . . .Multicast DNS

MITM . . . . . .man-in-the-middle

MT6D . . . . . .Moving Target IPv6 Defense

MTU . . . . . .maximum transmission unit

NAT . . . . . .Network Address Translation

NDP . . . . . .Neighbor Discovery Protocol

NEMO . . . . . .Network Mobility

NIC . . . . . .Network Interface Controller

NIS . . . . . .Network Information Service

NTP . . . . . .Network Time Protocol

OS . . . . . .operating system

OSI . . . . . .Open Systems Interconnection

OUI . . . . . .Organizational Unique Identifier

P2P . . . . . .peer-to-peer

PII . . . . . .personally identifiable information

PKI . . . . . .public key infrastructure

QoS . . . . . .Quality of Service

SEND . . . . . .SEcure Neighbor Discovery

SIP . . . . . .Session Initiation Protocol

SLAAC . . . . . .StateLess Address Auto Configuration

SNTP . . . . . .Simple Network Time Protocol

SSID . . . . . .service set identifier

SSL . . . . . .secure socket layer

SSN . . . . . .social security number

TLS . . . . . .Transport Layer Security

TTL . . . . . .time to live

UDP . . . . . .Unreliable Datagram Protocol

ULA . . . . . .unique local address

Virginia Tech . . . . . .Virginia Polytechnic Institute and State University

VoIP . . . . . .Voice over IP

WAN . . . . . .wide area network

Chapter 1

Introduction

As the address space in the current Internet Protocol version 4 (IPv4) is depleted, networks will have to transition to the Internet Protocol version 6 (IPv6). IPv6 increases the address size to 128 bits and was designed to support the increasing numbers of users and emerging classes of devices that require globally unique addresses. While Network Address Translation (NAT) has increased the available addresses in IPv4 and staved off the transition to IPv6, the technology has fundamental flaws and is reaching its limit. The Internet Assigned Numbers Authority (IANA) reported in Feburary 2011 that no IPv4 addresses remain. The increased address space of IPv6 fixes the address space issues of IPv4; however, adoption of IPv6 has been slow. IPv6 must be implemented quickly to assure global addressability and connectivity as IPv4 addresses are quickly exhausted.

To make the transition from IPv4 to IPv6 easier, many network administrators are using either StateLess Address Auto Configuration (SLAAC) or Dynamic Host Configuration Protocol version 6 (DHCPv6) to configure addresses on their networks. SLAAC eases the administrative burden of managing the more than $1.8 \cdot 10^{19}$ possible nodes on a single subnet, allowing for nodes to configure their addresses independently using network information broadcast by routers. DHCPv6, similar to Dynamic Host Configuration Protocol (DHCP) in IPv4, is a more familiar method of stateful network address assignment and management that offers more options and control than SLAAC. While currently SLAAC and DHCPv6 are the only automated methods of address configuration available in IPv6, both threaten the privacy and security of users and hosts by exposing a static identifier which allows for tracking and targeting by attackers.

1.1 Motivation

Static identifiers are exposed in both SLAAC and DHCPv6. In SLAAC, the static identifier is exposed as a piece of the global IPv6 address and is available on the wide area network (WAN), compromising users' privacy on any network as they travel. The static identifier exposed in DHCPv6 is only available on the link-local network or local area network (LAN) and, therefore, is more limited in scope. These static identifiers allow attackers to physically track users' locations with publicly broadcast information that is necessary for network connectivity. Also, if attackers are monitoring network traffic, users' network activity can be correlated over multiple sessions, threatening their privacy by intercepting personally identifiable information (PII). Finally, the static identifiers allow network attacks, both system-specific and general Denial-of-Service (DoS) attacks, to be successfully perpetrated against hosts. While static identifiers, such as static addresses connected to Domain Name System (DNS) records, are common for publicly available network hosts, these systems often have less of a need to protect their geographic location and network activity compared to clients and mobile hosts.

The tracking, monitoring and network attacks made possible by static identifiers have both beneficial and malicious uses. These attacks can be used to further crimes such as cyber stalking, identity theft, or terrorism. Also, these techniques could be used by marketing companies to gather valuable data or by legal authorities to help protect citizens by tracking and monitoring criminals. Although the static identifiers can be used to further positive or negative goals, solutions must be implemented to prevent static identifiers in network addresses and to protect privacy and security as IPv6 is deployed. To remove publicly available static identifiers from IPv6, address privacy solutions are recommended which help mitigate the risks.

1.2 Original Contribution

The privacy and security threats posed by static identifiers in IPv6 have never been tested in SLAAC or identified in DHCPv6. While the decrease in privacy of nodes with static identifiers has already been examined by Narten et al. and Haddad, no work has previously verified how a target can be geolocated or have their traffic sniffed on an IPv6 network [11, 23]. The experiment tracking users through their SLAAC IPv6 address, sniffing their traffic, and using statistical analysis of traffic captured for more effective reconnaissance

and targeting is novel. Locating and exploiting the static identifier in DHCPv6 is novel in its entirety. The experiments designed and exploits discovered to track a user through their DHCPv6 leased address are novel. Since these two exploits threaten privacy and security for all known forms of non-static addressing in IPv6, they provide a significant contribution to the development and deployment of IPv6.

1.3 Organization

In Chapter 2, a introduction to IPv6, stateless and stateful addressing, and existing threats in the protocol are given. Chapter 3 discusses the static identifiers in SLAAC and DHCPv6. Related work in the areas of address tracking and network-layer anonymity is discussed in Chapter 4. The design of the study of stateless and stateful network addressing privacy and security and the results and analysis are given in Sections 5 and 6, respectively. To combat the privacy and security risks of static identifier, solutions to the weaknesses in IPv6 addressing are provided in Chapter 7. Chapter 8 addresses future work in the areas of network layer security and dynamic addressing and Chapter 9 concludes.

Chapter 2

Background

IPv6 is a major improvement to the foundation protocol for the Internet. The new version of the protocol provides additional address space, performance features, and security options to networks. The exponentially larger address space, however, also creates an additional administrative burden for network administrators. To accommodate the rapid expansion of the address space, new stateless and stateful address systems were developed to facilitate management.

2.1 Benefits of IPv6

As previously discussed in Section 1.1, the primary motivation behind IPv6 is a larger address space. IPv4 addresses consist of 32 bits, providing approximately 4.2 billion globally unique addresses. This address space is not sufficient to support the emerging class of embedded computing devices beginning to connect to the Internet. Therefore, the IPv6 address was expanded to 128 bits. The expanded address space allows for 2^{128} globally unique addresses, providing more than $7.9 \cdot 10^{28}$ networks with the same address space as the entire IPv4 Internet.

In addition to the larger address space, IPv6 was designed with four other main improvements. First, IPv6 provides a simplified header format fixed at 40 bytes, removing unnecessary or unused fields in the IPv4 header. The smaller header speeds processing in routers and reduces protocol overhead, increasing the maximum transmission unit (MTU). The second improvement moves the options field out of the header and into the payload of the packet, providing a flexible length to the options field only limited by the MTU of the link. Flow labeling, the third design improvement in IPv6, allows for classification of

packets belonging to particular network flows. By including a flow label in the IPv6 header, each router can determine which flow a packet belongs to and prioritize the packets appropriately, increasing Quality of Service (QoS) and throughput on bandwidth-critical and time-sensitive applications. Finally, IPv6 integrates authentication and encryption into the protocol. When IPv4 was designed and defined, security was not addressed. As Internet commerce and secure communication have become critical to maintain users' privacy and security on the Internet, Internet Protocol Security (IPsec) was developed to address the security concerns, adding authentication and encryption to IPv4. Since IPsec was developed after IPv4 was implemented, it became a layer 3.5 protocol in the Open Systems Interconnection (OSI) model, requiring additional overhead and processing compared to application layer encryption at layer 7. Integrating IPsec functions within IPv6 means that the entire operation can be performed at layer 3. This adds efficiency by increasing throughput and eliminating the need to store the packet during processing.

2.2 Stateless Address Auto Configuration

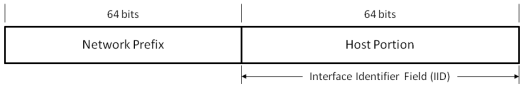

Figure 2.1: IPv6 128-bit address format

The large size of the IPv6 address space requires a new network address configuration architecture to simplify network administration. For this reason, IPv6 contains a Neighbor Discovery Protocol (NDP) [24] and SLAAC to allow for a node to self-determine its IPv6 address. Designed as a replacement for the Address Resolution Protocol (ARP), NDP 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 interface identifier (IID) of the address. The final step combines the 64-bit network address with the 64-bit host address forming a complete 128-bit IPv6 address (See Figure 2.1). SLAAC is currently considered the default IPv6 address protocol.

The NDP message exchange starts with a node reserving its link-local address. Then, the node either solicits the router for a router advertise message or waits for a periodic

router advertise message for network configuration information. This information includes the network portion of the IPv6 address and other optional information, including DNS servers and Network Time Protocol (NTP) servers. Once this is received, the node calculates and assigns itself address. The node then attempts to reserve the address on the network. During either the link-local address or the configured address reservation phase, if the address is already in use, the node goes through Duplicate Address Detection (DAD) and begins the process with a different IID. This process is shown in Figure 2.2.

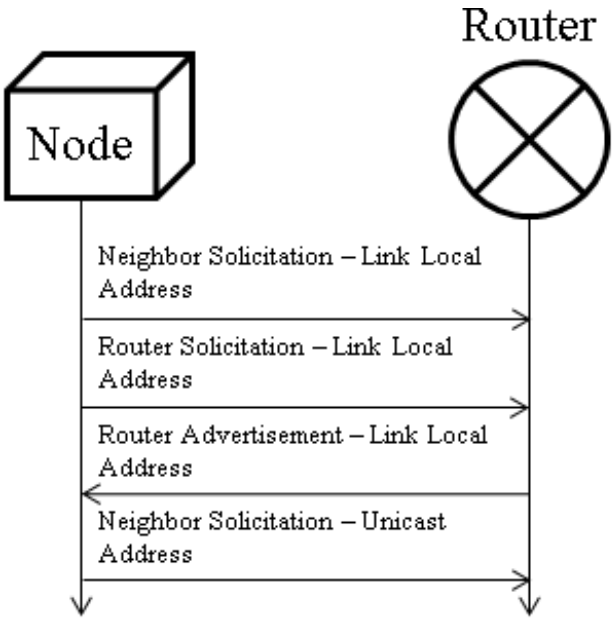

Figure 2.2: IPv6 SLAAC Message Exchange

2.3 Stateful Address Configuration in IPv6

DHCPv6 [9] provides a stateful, managed alternative to SLAAC. While DHCPv6 is not the default addressing mode as DHCP is in IPv6, the management and configuration options offered by the protocol can be useful in large, complex networks. Different realms can be established to configure different addresses for different parts of the networks. Other custom solutions can be also implemented, including servers which rely on public key infrastructure (PKI) for authorization for address leasing.

When a node connects to a network, it sends a multicast SOLICIT message to special addresses reserved for DHCPv6 servers with the node's unique DHCP Unique Identifier

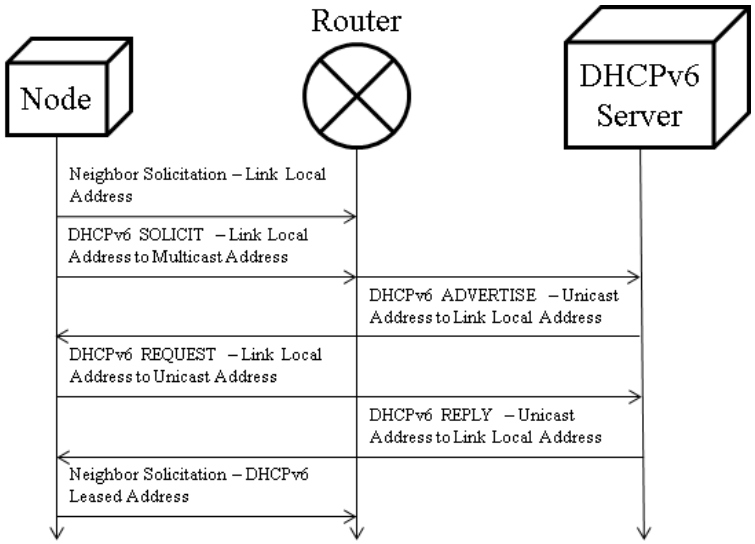

Figure 2.3: IPv6 DHCPv6 Message Exchange

(DUID). The server then responds to the node using the link-local address provided in the initial message. The response is in the form of an ADVERTISE messages containing a leased address and any other configuration parameters necessary for the network. DHCPv6 can also be configured to provide other configuration information to network nodes, such as DNS and NTP servers. To reserve the DHCPv6 leased addresses and notify routers and other clients, servers and clients also use NDP as in SLAAC. This message exchange is depicted in Figure 2.3.

2.4 Threats in IPv6

Many of the threats that plague IPv4, such as man-in-the-middle (MITM) attacks, sniffing, flooding, and application layer attacks, are still possible in IPv6 [5]. Though IPsec is integrated into the protocol, it is not required. Many believe that the implementation of DNS Security Extensions (DNSSEC) will increase security; however, DNSSEC only provides authentication for hosts using DNS and no authorization or access control. With no authentication or encryption required in the protocol, IPv6 is susceptible to MITM 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.

Though IPv6 has inherent security built in through IPsec, the protocol remains vulnerable to the same attacks as IPv4.

2.4.1 Covert Channels

Most of the current commercial Intrusion Detection Systems (IDSs) and Intrusion Prevention Systems (IPSs) 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 only 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 for attacks using the new protocol allows for covert channels to easily bypass most IDSs and IPSs.

2.4.2 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. IPv6 Rapid Deployment on IPv4 Infrastructures (6rd) uses 6to4 tunneling to create automatic tunnels at router endpoints using embedded addressing [6, 25]. Teredo is an automatic tunneling technique which encapsulates IPv6 packets in Unreliable Datagram Protocol (UDP) IPv4, enabled by default in Windows® XP SP2, Vista and 7 [13]. Intra-Site Automatic Tunnel Addressing Protocol (ISATAP), another automatic tunneling protocol, creates local IPv6 networks over an IPv4 network by internally mapping IPv6 addresses to IPv4 nodes [32]. Finally, 6in4, a manually configured tunnel, uses a similar technique as 6to4 tunneling, but instead relies on manually configured router endpoints [25]. 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 due to incorrect system handeling of the traffic and the inability of some system defenses to handle encapsulated traffic. 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 auto configured tunnel endpoint to a malicious host acting as a MITM, 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.

Chapter 3

Problem Statement

Static identifiers allow attackers to locate and target systems. SLAAC in IPv6 produces static IIDs, which allow users to be geotemporally tracked, to have their network traffic monitored, and to be targeted for system-specific and general network attacks. DHCPv6 uses DUIDs, a static value which is passed in DHCPv6 messages and allows an attacker to discover a user's address. This allows for all of the same attacks as in SLAAC except system-specific attacks.

3.1 Static IID

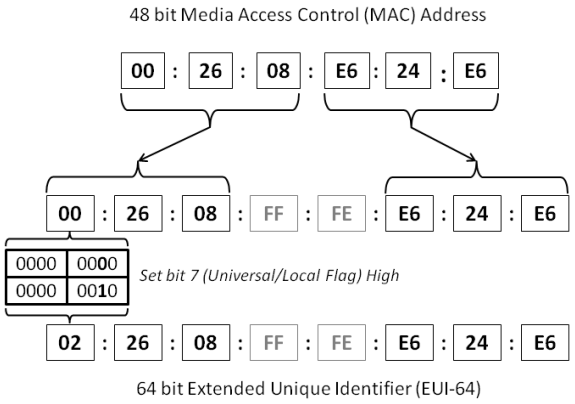

Figure 3.1: 64-bit Extended Unique Identifier (EUI-64) format

As mentioned in Section 2.2, the IID is automatically configured by the host. The current accepted definition of SLAAC on most operating systems (OSs) 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 Media Access Control (MAC) address of its network interface. This is accomplished by extending the 48-bit MAC address to 64 bits through the Extended Unique Identifier (EUI)-64 format [12]. The EUI-64 format splits the 48-bit MAC address into two 24-bit halves. The first 24-bit half, the Organizational Unique Identifier (OUI), is specific to each manufacturer. The second 24-bit half is the Network Interface Controller (NIC) specific part and can be unique to the interface, identify a model, or serve any function designed by the manufacturer. 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 new 64-bit host portion, is set to 1, signaling that the address has been locally configured. Figure 3.1 illustrates this process.

While OSs configure IPv6 addresses differently, no current OS implementations of IPv6 SLAAC 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 [23, 33]. However, Windows operating systems also carry another IPv6 address used for neighbor solicitation. This additional IPv6 address contains an IID that is obscured but never changes, regardless of the subnet the node connects to. The static IID currently implemented in major operating systems can be identified with a particular node, even as the node changes networks. Not dynamically obscuring a user's host portion of all of the IPv6 addresses associated with a system threatens a user's privacy.

Many mobile devices, such as Android and iPhone, support IPv6 in Wi-Fi™. These device implementations follow the EUI-64 format providing these mobile devices with static IIDs that are easily tracked on their Wi-Fi™ connections. Since most users frequently carry their mobile devices and leave them on and connected, it is easy for an attacker to track a user. 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 Wi-Fi™, users can still be tracked through their wireless devices as they move between different Wi-Fi™ networks. Therefore, address privacy must be addressed in all connections of a mobile device to assure complete privacy.

3.2 Static DUID

While the insider threats of a DoS or MITM attacks have already been recognized [9], the privacy implications of using static DUIDs have not been evaluated. The static identifier, created to be publicly broadcast and globally unique, provides a simple vector for an attacker to identify a unique node. The SOLICIT and ADVERTISE messages sent by the DHCPv6 server allow an attacker to identify a node through its static DUID and monitor its traffic for the session through the leased address. If the attacker misses a targeted node's ADVERTISE messages, but already knows the node's DUID, the attacker can send unauthenticated client-initiated INFORMATION-REQUEST messages to the DHCPv6 server requesting information on a leased address.

Since the DHCPv6 DUID is defined as a piece of the protocol, tracking nodes and addresses by DUIDs is not OS dependent [9]. To ensure system interoperability, all implementations of DHCPv6 follow the protocol and include a static DUID. While minor differences may exist in the messages passed by different implementations, the privacy and security deficiency of the DUID lies in the overall design of the protocol.

3.3 Threat to Privacy and Security

The IID and DUID expose a static identifier globally and locally, respectively. A static identifier on a networked device allows it to be geolocated, have traffic captured and correlated over multiple sessions, and to be targeted for attack. As more network devices become mobile, the ability to geographically locate a device can threaten the privacy of the user by exposing their location. If implanted or embedded devices have their addresses compromised, the threat to the user or organization increases as well. With traffic captures, users can often be identified by the sites they visit or any other unencrypted information that is transmitted. Also, systems are more vulnerable to attack with static identifiers, since an attacker has multiple sessions to exploit the vulnerabilities of a system.

While the limited scope of the exposed DUID seems to pose little threat as a static identifier, the ability for DHCPv6 messages to be relayed significantly expands the threat. The threat of static identifiers also exists in IPv4, specifically with MAC addresses and ARP. The threat to privacy and security posed by these vectors has been dismissed since the scope of the threat is limited to the system's physical hardware network. DHCPv6 messages are Internet Protocol (IP) based messages which, once relayed, can travel an infinite distance.

The threat that once existed only on the physical link is now exploitable on a much larger scale.

Chapter 4

Literature Review

While many security systems address computer and network security between the Transport and Application layers of the OSI model, few systems address security at the Network layer. The IP address has been researched as a vulnerable vector which can be exploited to geographically track users and their network activity. Also, the ability to identify a user through their IP address and the need to anonymize IP addresses have been identified. This research differs through its examination of the IPv6 address for new privacy and security vulnerabilities compared to those in the IPv4 address.

4.1 Address Tracking

A significant amount of work examines the format and structure of IPv6 addresses. Researchers have concluded that using the EUI-64 format in Mobile IPv6 could compromise a user as subnets move with users following Network Mobility (NEMO). What makes this research unique is that it shows how both mobile and stationary nodes using IPv6 can be geographically tracked and identified through traffic analysis. This is due to the static implementation of IIDs and DUIDs. This capability exists for both the EUI-64 format and the onetime hash used by Windows operating systems.

4.1.1 SAA

The realization that using a MAC address 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 IID would allow an eaves-

dropper to correlate unrelated information with a particular node [23]. Haddad even goes so far as to address the fact that mobile nodes using IPv6 SLAAC can reveal their location to an eavesdropper [11]. This work builds on these ideas by discussing and demonstrating how an interested party can eavesdrop on an IPv6 user from anywhere on the Internet using basic network tools. Additionally, this work identifies multiple ways users' stateless auto configured addresses can be exploited in their current implementations and offers 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 [19]. Castelluccia et al. and Qiu et al. also discuss how mobile nodes can be tracked using their home and care-of addresses [3,27]. 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.

4.1.2 DHCP

Tracking address assignment in DHCP in IPv4 has been established by patent for a particular method of DHCP address tracking. Tams et al. [31] patented a system in which devices' MAC addresses and address leases are correlated. By querying the server for leased addresses and maintaining a database, the DHCP addressed nodes in an IPv4 network can be tracked. While the nature of this work is similar, the research executes DHCPv6 tracking on a new protocol using a different vector, the static DUID. Also, since their work relies on the MAC addresses of nodes, their system is dependent on DHCP in IPv4 and the ARP, which is not available in DHCPv6 and IPv6.

4.2 Address Anonymity

The need to protect users' IP addresses from exposure in network traffic has already been identified as an important aspect of maintaining privacy. When sharing logs of network traffic, even anonymizing or removing IP addresses from the logs can still expose a user and threaten their privacy. Tor networks were originally invented to hide addresses and

anonmyize traffic through distributed networks, but their dependence on exit nodes has rendered them ineffective. Finally, creating a truly dynamic address for a host has also been identified as a method of protecting privacy through address obscuration.

4.2.1 Logs

In order to advance network security and share information about network attacks with trusted and untrusted partners, many have examined how to successfully share logs that have been anonymized of IP address information. Slagell et al. [30] analyzed different techniques used to identify IP addresses from logs and developed a trust model for sharing logs. While this work focused on removing IP addresses from logs, the trust model allows for IP addresses of hosts being served by DHCP to remain unaltered. The static identifier of the DUID would be exposed in the logs and, therefore, allow sensitive address information to be gleaned from the logs. Koukis et al. [20] used web site signatures and fingerprinting to determine host addresses in anonymized IP logs. While this method is ineffective for tracking dynamic hosts, this work on the DUID in DHCPv6 could facilitate the tracking of dynamic stateful client addresses. Log privacy must be reexamined due to the network level vulnerabilities in IPv6.

4.2.2 Tor Networks

The need to have an anonymous network address to maintain security and privacy has been explored. Originally, Dingledine et al. created Tor, or onion routing, a network in which multiple layers of encryption were used so that individuals could rely on an anonymous network to create security and privacy [7]. Yet, the Tor exit nodes, where traffic leaves the Tor network, remain static in the network and are often blocked to prevent anonymous browsing. Johnson et al. [14] identified the need to anonymize addresses and built a trust model into Tor networks called Nymble. Nymble hides clients' IP address from servers while only allowing trusted nodes to use unblocked exit nodes. Shields et al. [29] created another Tor-based anonymity protocol named Hordes. Hordes focuses on a secure system that does not decrease network performance. While these systems focus on hiding the publicly available addresses once the packet is transmitted to the Tor entrance node, this work analyzes vectors for tracking and attack which exist before the entrance node, for which none of the protocols above provide anonymity or protection.

4.2.3 Dynamic Addresses

While no other academic work has been discovered in increasing the security and privacy effects of addressing. Two patents attempt to capitalize on dynamic addressing to create security. A technique by Sheymov (Sheymov, 2010) is designed with the goal of dynamic obscuration. Sheymov's objective behind dynamic obscuration is to provide intrusion protection from certain classes of network attacks. While Sheymov's method uses dynamic addressing, it relies on an Intrusion Detection System to trigger address changes. Fink et al. (Fink, Brannigan, Evans, Almeida, & Ferguson, 2006) also propose a technique for dynamically obscuring host addresses called Adaptive Self-Synchronized Dynamic Address Translation (ASD). ASD uses symmetric keys established through a handshake process between a trusted sender and receiver enclave. This technique adds additional overhead due to repetition of the handshake process. A dynamic addressing technique must reduce overhead to be feasible for implementation. While these systems achieve security by avoiding static addresses, neither of these systems are effective in IPv6.

Chapter 5

Stateless Addressing and DHCPv6 Exploit Experiment Design

In order to demonstrate geotemporal tracking and traffic analysis through IID analysis, testing was performed using IPv6 nodes on a live IPv6 network. The Virginia Polytechnic Institute and State University (Virginia Tech) IPv6 network uses SLAAC and the NDP to allow nodes to self configure addresses. 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 in SLAAC on the Wi-Fi™ IPv6 adapter, both geotemporal tracking and traffic analysis were possible.

The scenarios described in Section 5.2 were also performed on the Virginia Tech IPv6 network using a Dibbler [22] DHCPv6 server and client running Ubuntu 10.04. By providing a DHCPv6 server on the network, nodes that were set to self-configure IPv6 addresses continued to operate normally. Nodes deploying the Dibbler DHCPv6 client received stateful configured addresses from a Dibbler DHCPv6 server. Different LANs were provided with DHCPv6 access through the use of Dibbler DHCPv6 relays. The DUIDs and link-local addresses of these nodes were recorded and traffic was sniffed at predetermined locations.

5.1 IID Tracking

The Virginia Tech IPv6 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.

5.2 DHCPv6

To exploit the weakness of static DUIDs in DHCPv6, address correlation was performed on the LAN and remotely by sniffing DHCPv6 messages. Once collected, the DUID in each message was analyzed. DUIDs and the associated DHCPv6 leased addresses, available either in the sniffed traffic or by querying the DHCPv6 server, were correlated. An attacker could use this data to pair sniffed LAN traffic to specific users. To correlate DHCPv6 addresses on the LAN, messages were sniffed directly, between the DHCPv6 server and client, and indirectly, through DHCPv6 relays programmed to pass DHCPv6 messages between different segments of the LAN. Remote address correlation was accomplished through compromising DHCPv6 relays, set to forward DHCPv6 messages to a remote third party.

Due to the design and small size of the experimental network used for testing, address tracking and traffic correlation were also possible to accomplish through MAC address correlation. Since the experiments were run on a network where sniffing was accomplished before the packets traveled more than one hop, the MAC address of each node was exposed and could be used to correlate traffic. In a production environment, the MAC address would most likely be changed before the packet was sniffed since it would have multiple hops before reaching the border of the LAN. Therefore, this information was ignored in the experiment. It is worth noting, however, that packet and address correlation can be accomplished through MAC address correlation if the packet can be sniffed before the Ethernet frame is modified.

5.2.1 Local Monitoring

DHCPv6 addresses can be correlated locally by sniffing and spoofing DHCPv6 messages inside the LAN. Since the LAN allows for link-local and multicast messages to pass freely, an attacker can read the messages sent to and from a DHCPv6 relay. An attacker can also spoof the identity of a DHCPv6 client to query a DHCPv6 server for more information, including the addresses leased, for a specific DUID. In large DHCPv6 addressed networks, addresses can be correlated at three locations. The first location is on the same router or switch as the client. An attacker can easily sniff the network traffic and correlate DUID and client

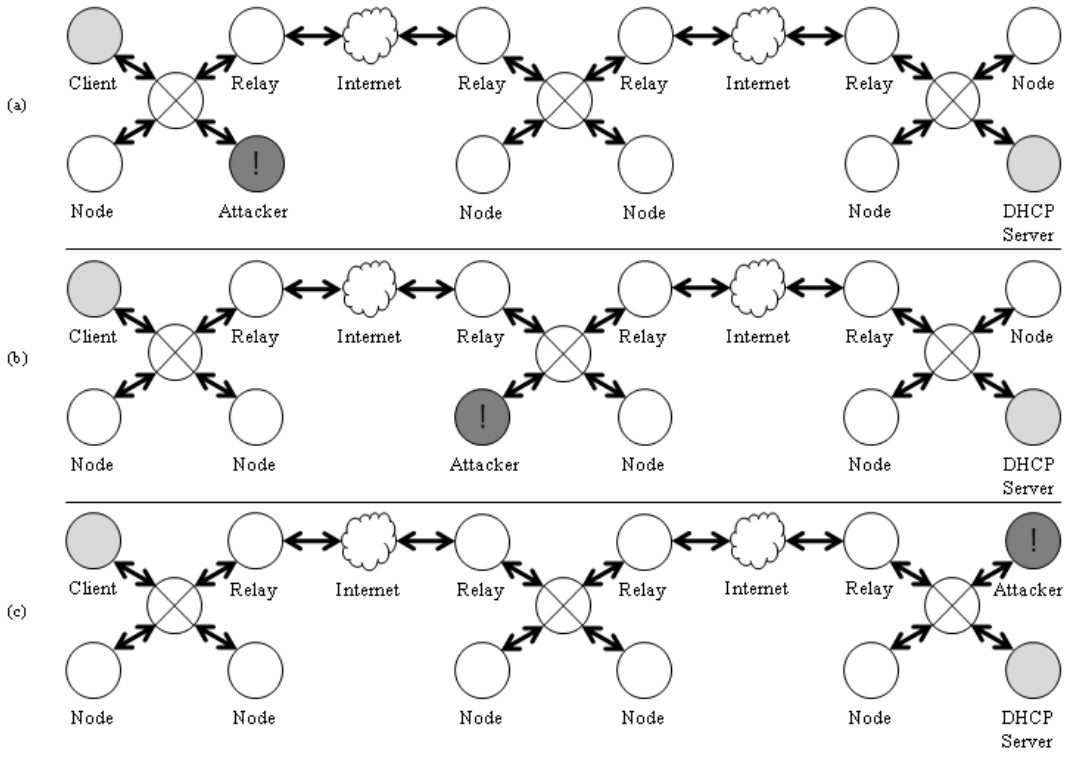

Figure 5.1: Three different scenarios of DHCPv6 message sniffing inside a LAN

address information without querying the DHCPv6 server. The second location is between relays. The attacker is on neither the same switch as the DHCPv6 server or the client. The attacker relies on sniffing messages passed between the relays to collect DUIDs and then queries the server to gain client address information. Finally, when on the same switch as the DHCPv6 server, an attacker can sniff traffic for DUID and address information. He/she can also masquerade as the client and intercept address leases and traffic, denying the user network access.

In Figure 5.1(a), the attacker is on the same router as the targeted client. When the client initially connects to the network, a SOLICIT message is sent to the multicast address of the DHCPv6 server. This messages contains the DUID and link-local address of the client. By sniffing this message, the attacker can capture the client's identity. The server responds to the client's link-local address with the ADVERTISE message, containing the leased address and any other configuration parameters. The attacker captures this response sent to the link-local address and matches it with the link-local address in the SOLICIT message containing the client's DUID. With this information, the attacker is able to compromise the client's

address and identity for the session.

The attacker can also move to a subnet that contains neither the DHCPv6 server nor the client, but does contain a DHCPv6 relay. This scenario is shown in Figure 5.1(b). On a relay subnet, a client's address can still be compromised using an approach similar to the approach in Figure 5.1(a). The SOLICIT multicast message sent to the DHCPv6 server from the client is forwarded by the relay. Since the message is multicast, the attacker can register itself as a receiver of the DHCPv6 multicast addresses with the router through NDP. This NDP exploit allows the attacker to sniff the server's traffic. When the server sends the response using the link-local address, the attacker can either register itself as a relay, receiving the message, or query the DHCPv6 server through an INFORMATION-REQUEST for the address leased to the sniffed DUID. Since the server does not perform address validation on the source of the message, the message will be returned to the attacker with the client's address, again compromising the session.

Figure 5.1(c) shows an attacker sniffing DHCPv6 messages on the same router as the DHCPv6 server. This configuration gives the attacker direct access, since the router can be configured through NDP to send the traffic of the client being attacked directly to the attacker. The attacker can again use the same methods of attack as for the scenario in Figure 5.1(a). When the client initially sends its SOLICIT message to the server, the attacker can sniff the message and analyze the DUID and link-local address. When the server responds with the DHCPv6 configured address to the client, the attacker can signal the router to register the leased addresses to itself. Creating a man-in-the-middle attack, the client is passed packets only after the attacker has intercepted and forwarded them.

5.2.2 Remote Monitoring through DHCPv6 Relays

To monitor the addresses of DHCPv6 nodes remotely or on a WAN, a compromised DHCPv6 relay can sniff DHCPv6 SOLICIT messages and send them to a remote host for analysis as illustrated in Figure 5.2. Since there is no mandatory authentication or authorization of DHCPv6 relays running on a network, a modified DHCPv6 relay could be enabled on a network. Receiving all of the same messages as a legitimate DHCPv6 relay, the modified relay can forward the messages to a remote attacker and allow for remote correlation of leased addresses and any sniffed traffic.

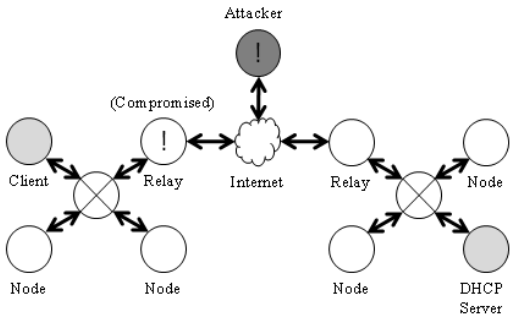

Figure 5.2: A compromised DHCPv6 relay passes DHCPv6 messages from a LAN to an attacker

5.2.3 Dynamic DUID

To create a dynamic DUID, two different methods were selected for evaluation. Both were based off of protection of static IIDs in SLAAC. Cryptographically Generated Addresses (CGAs), part of SEND, use a public key and two separate hash calculations to create their addresses. Privacy extensions simply use one hash calculation and a single pseudo random number (PRN) generation to create an obscured IID. Since privacy extensions have less computational overhead than CGAs, the dynamic DUID implementation was modeled after privacy extensions.

To calculate the dynamic DUID, the randomization technique used in IPv6 privacy extensions was implemented. When the client is first initialized on a system with stable storage, the system generates a random number (RANDOM) and computes the MD5 digest of that number. Then, the newly generated and randomized DUID is archived (DUIDARCH) for future use. For subsequent IID calculations, DUIDARCH is concatenated with a new random number and the MD5 digest is performed again. MD5 was chosen since all IPv6 clients must have an MD5 implementation to use for the mandatory IPsec [18]. All 128 bits of the MD5 digest are used in the DUID. On a system that lacks stable storage or if the client wants to reduce storage overhead, RANDOM can be used for the first and all subsequent IID calculations without the need for storage of DUIDARCH. This implementation adds less entropy to the DUID and is dependent on the quality of the random number generator of the system. If these issues are addressed in the implementation, a system without storage should be able to implement the dynamic DUID without issue. To initiate DUID regeneration, three different triggers or interrupts were used. The first method was a simple time-based system; after a set period of seconds, a new DUID was generated and a new

address was leased from the DHCPv6 server. Following the model of privacy extensions, two values were used: a preferred lifetime for the DUID (TEMP_PREFERRED_LIFETIME) and a maximum time for the DUID (TEMP_VALID_LIFETIME). The system only attempted to renew the DUID at TEMP_PREFERRED_LIFETIME if no active network sockets were open. If sockets were open until TEMP_VALID_LIFETIME, the DUID was forced to regenerate and a new address was leased, often meaning loss of connectivity for the current active network connections. The second trigger system detected changes in the network prefix of the IPv6 router address advertised. The client generated a new DUID and leased a new address whenever any changes in the network prefix were detected. Finally, the third trigger for DUID regeneration was system state; only changes in the state of the host, such as a reboot or standby, caused the DUID to be regenerated. Unlike in privacy extensions, there is no ability to perform any type of duplicate detection for dynamically generated DUIDs. Since only the DHCPv6 server knows the DUIDs that currently have leased addresses, it is impossible for a host to determine if the DUID already exists. By using the entire 128 bits of the DUID space in DHCPv6, there are 2128, or 340 undecillion, possible DUIDs. Therefore, DUID collision is unlikely. A dynamic DUID will not have any effect on other DHCPv6 extensions. Since the DUID has no effect on the DHCPv6 options, additions to the protocol, such as Session Initiation Protocol (SIP) configuration and DNS configuration are not broken by a dynamic DUID [8, 28]. Also, the dynamic DUID does not conflict with any of the proposed authentication overlays, Network Information Service (NIS) or Simple Network Time Protocol (SNTP) configuration options for DHCPv6 [15, 16, 21]. These configuration parameters will have to be reestablished for each connection, causing additional overhead. No further modifications to the DHCPv6 protocol were necessary to maintain functionality of all extensions.

Chapter 6

Results and Analysis

The vulnerabilities discovered in SLAAC and DHCPv6 were thoroughly tested on the Virginia Tech IPv6 network. The geographic tracking and traffic correlation attacks against the static IID were successful. The address tracking attack against DHCPv6 was also successful with the results are limited since little can be shown other than what was already discussed in Section 5.2.

6.1 StateLess Address Auto Configuration

The privacy and security flaws in SLAAC, explained earlier in Section 3.1, were exploited in three different ways. A host was geographically tracked around the Virginia Tech campus using the static IID. Also, traffic was captured at the border and later correlated to using IID analysis. Finally, the captured traffic was also analyzed by the OUIs available in the EUI-64 expanded IIDs to determine system and manufacturer types. The results are discussed below.

6.1.1 IID Tracking

Reconnaissance on the campus at Virginia Tech's six unique wireless subnets was performed under the `2001:468:c80::/48` global IPv6 network prefix. 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 VT-Wireless service set identifier (SSID), the script was run while the node moved around the campus and associated with

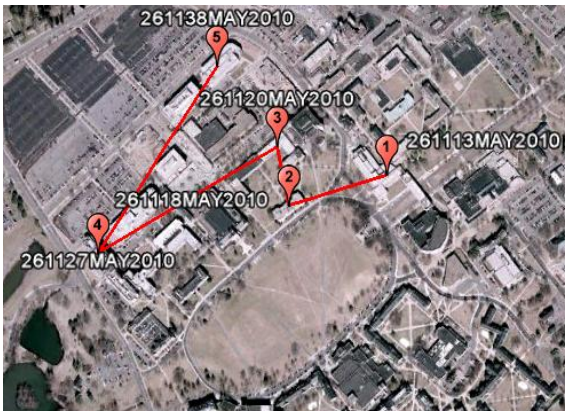

Figure 6.1: Geotemporal plot of a wireless node's movement within the Virginia Tech Network

different access points. The results of this testing can be shown in Figure 6.1.

Geographically tracking users with static IIDs is possible on any SLAAC 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 ping sweeping for the node and tracerouting to determine location. However, the accuracy of the determined location is degraded and the time necessary to find a user is increased without reconnaissance. The use of IPv6 transition tools, such as tunneling, degrades the accuracy of the traceroute. Also, ping sweeps in IPv6 are practically impossible due to the large address space.

6.1.2 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. In the experiment, the same wireless subnets were used to track user locations as in Section 6.1.1. 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 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 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, user name and tweets sent by our test node over Twitter were successfully monitored, captured at the border, and correlated with the test node. 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 through traffic analysis.

IID traffic analysis of users utilizing Windows operating systems is extremely difficult due to the use of privacy extensions [23]. 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.

OUI Analysis

Analysis was conducted 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, the assumption

Table 6.1: Top five NIC OUI Registrars accessing Virginia Tech's network from EUI-64 systems of the 12,356 systems using EUI-64 expansion

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

was made 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. 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, the OUI of each IID was analyzed and a list was compiled of the top five manufacturers as seen in Table 6.1. The large majority of these systems had wireless NICs registered to Apple, Inc. Since no mobile Apple operating system deployed IPv6 at the time of testing, 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. However, if an attacker is able to identify a vulnerability specific to a particular brand of device, the attacker can then target and exploit those devices specifically. This type of attack may provide an attacker with another vector into critical or sensitive systems.

Spoofing

If it is known that an innocent node is being monitored, a malicious host can attempt to slander the innocent host by spoofing the innocent host's IID. By spoofing a host's IID, the attacker can make the victim appear in potentially incriminating locations. Additionally, the attacker can produce slanderous traffic that would be associated with the victim. Of course, these types of attack are only detrimental in environments where hosts are being monitored. For this scenario, four different operating systems were tested: Ubuntu Linux, OS X, Windows Vista SP2, and Windows 7.

Spoofing an IID using Ubuntu Linux and OS X is trivial. Since neither operating system uses any IID obscuration techniques, all the attacker needs to do is modify the MAC address and let SLAAC reconfigure the address with the new IID. In trials, the MAC address was modified using the command `ifconfig [interface] hw ether [hwaddr]`. Once the interface is restarted, it produces a new IID. The same exploit can be performed using a virtual machine with a manually configured MAC address. Using the MAC replacement technique requires the attacker to be on the same network segment as the victim. If the attacker is located elsewhere, he/she can simply spoof the entire IPv6 address.

Spoofing a Windows IID is a more difficult due to default privacy protection enabled in the operating system. Since Windows operating systems use privacy extensions [23], typical communications use a temporary, non-deterministic IID. Spoofing the temporary IID is not useful since it changes often and is not tied to a specific host. Therefore, the permanent IID is required. The only way to get the permanent Windows IID is by monitoring the neighbor solicitation messages, Link-local Multicast Name Resolution (LLMNR) messages, or Multicast DNS (mDNS) messages. This requires the attacker to be on the same network segment as the victim at some point since these messages are sent using the link-local or multicast messages, respectively. Since even the permanent IID is obscured, the MAC replacement technique does not work. This was tested using the same NIC on multiple Windows Vista and 7 machines. A different permanent IID was received each time. On a specific machine, however, the permanent address does not change as a node changes networks or over time. Therefore, IPv6 address spoofing works without issue.

Even with knowledge of the permanent IID, attacks against Windows operating systems are limited. Since Windows hosts communicate using a temporary IID, traffic analysis as discussed in Section 6.1.2 is likely not feasible. This leaves location slander as the only attack available to the malicious node. Since a Windows host will respond to ICMPv6 echo requests using its permanent address, the attacker can make an innocent host appear to be at incriminating locations.

6.2 Dynamic Host Configuration Protocol

The results of testing the different DUID tracking scenarios described in Section 5.2 proved tracking DHCPv6 IPv6 addresses on a DHCPv6 network is possible. A custom network was created on the `fc00::/7` unique local address (ULA) IPv6 network prefix. All SOLICIT and ADVERTISE messages were successfully captured in the trials. INFORMATION-REQUEST messages were also successfully sent and received to determine a node's leased address. Once these addresses were captured, all traffic sniffed was correlated with a specific user, even after a new DHCPv6 address was leased to the client. Since this testing was conducted in a laboratory environment, geographic tracking could not be tested on the network due to it configuration for SLAAC. Yet, since the subnet exposes a user's location, the user could be geographically tracked on a geographically large production DHCPv6 IPv6 network, providing the same mapping of subnets with geographic locations required for tracking in SLAAC.

6.2.1 DUID Formation

DUIDs are formed using three different methods, all of which are static. These methods vary in the amount of system-specific information they expose, but all expose vendor information. Other methods show complete link layer addresses, which exposes unnecessary information about DHCPv6 nodes.

A DUID is commonly formed by combining a link local address with the time, defined as a DUID Based on Link Layer Address plus Time (DUID-LLT). The DUID-LLT is formed with the first two octets set to type 1, the second two octets showing a hardware type defined in RFC 826 [26], the following four octets as the time in seconds since January 1, 2000 modulo 232, and a variable length link layer address. Since the layer address is usually the MAC address, exposing the MAC in the DUID creates the additional privacy and security

threats identified in SLAAC. The amount of the link layer address exposed depends of the OS implementation; common operating systems, such as Windows 7 and Ubuntu, expose only vendor information in this variable length field.

The second method, DUID assigned by Vendor Based on Enterprise Number (DUID-EN), is the most secure method of DUID and exposes the least amount of information. In this method, the first two octets are set to a type of 2, followed by a variable length Enterprise Number and an eight octet unique identifier. The Enterprise Number is a number assigned by IANA and is unique to a vendor. The unique identifier is an identifier determined by the manufacturer. By default, the only information that can be pulled from the DUID is the vendor of the host. Gleaning more information from a DUID-EN would require extension reconnaissance and knowledge of any vulnerability associated with how specific manufacturers implement the unique identifier.

The third and most vulnerable method forms the DUID from the link layer address only, referred to as the DUID-LL. This method uses two octets to show a type code 3, followed by the two octet hardware identifier used in DUID-LLT and a variable length link layer address. With more room to expose the link layer address, most implementation will use the entire MAC address of a system and expose unnecessary system specific information in the DUID. This same information was already shown to be vulnerable in SLAAC and could be used for activities such as system-specific targeting, a method through which attackers target machines based off of known vendor vulnerabilities.

Identity associations (IAs) and identity association IDs (IAIDs) are used by DHCPv6 servers to maintain complex configurations where a single DUID leases multiple IPv6 addresses. IAs and IAIDs are not a threat; since each IA and IAID is only unique to each DUID, multiple IA and IAIDs can exist within a single DHCPv6 without conflict. Therefore, a host cannot be identified through their IA or IAID.

Chapter 7

Solutions to Security and Privacy Flaws in IPv6 Addressing

With problems existing in SLAAC and DHCPv6, default solutions must be implemented which avoid static identifiers in addresses to protect users' privacy and security. The global scope of the static IID in SLAAC requires systems which dynamically obscure the IID. While the DUID is only exposed on the LAN, dynamically obscuring the DUID, combined with strict firewall and router rules, would help to secure DHCPv6. Since SLAAC and DHCPv6 are the only managed addressing options in IPv6, static identifiers must be removed from the protocols to avoid users from having to manually set and change their IP addresses to ensure their privacy.

7.1 Stateless Address Auto Configuration

Regardless of the intent behind IID tracking, users are entitled to the expectation of privacy when accessing the Internet using system defaults. Different schemes have been devised to secure the address in SLAAC and avoid a static IID. As a result of an effort to secure the NDP, SEcure Neighbor Discovery (SEND) includes a system to generate cryptographic addresses which avoid static identifiers. Privacy extensions, already implemented in Windows systems, maintain a secondary address which is non-deterministic and varies between sessions. Finally, IPsec offers address protection through encryption. It is important to prevent IID tracking before IPv6 is globally deployed.

7.1.1 Cryptographically Generated Addresses

One method of obscuring the IPv6 IID is through the use of CGAs, a piece of the SEND protocol. In general, CGAs are 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 SEND protocol [1]. Since CGAs also dynamically obscure IPv6 SLAAC, they can also 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 CGAs could quickly overwhelm a power-constrained device.

7.1.2 Privacy Extensions

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 limits 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 [23].

The disadvantages of using privacy extensions are less 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 [23].

The main factor that makes IID obscuration an attractive solution for hiding IPv6 addresses is the absence of need for any management overhead. Obscuration and verification both occur at the respective end hosts without the necessity for intervention by a trusted third party. 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 for management makes IID obscuration scalable.

7.1.3 IPSec

IPsec also provides a means to protect users from tracking. In IPsec, this is accomplished through the use of Encapsulating Security Payload (ESP) in tunnel mode. This hides the identity of the target node from being tracked by encrypting the target node's entire packet, including its address. This encrypted portion then becomes part of the payload of a new packet using the address of the tunnel start point [17]. 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 especially beneficial for power constrained devices.

There are, however, a number of serious 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.

7.1.4 DHCPv6

DHCPv6 [9] is not an IID obscuration technique but rather a means to provide stateful address configuration. The main advantage of this technique is that the addresses issued by the DHCP server are not globally tied to the identity of the clients. In principle, 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 have the ability to request temporary addresses, which mask their location and activities globally. Locally, however, an attacker can still track clients through the DUID that is transmitted between the client and the server. The scope of this method of tracking is limited to the subnet of the client, server, or any relays [9]. There is also an administrative management burden that accompanies the use of DHCPv6. If the LAN is trusted, DHCPv6 can be configured out of its default settings to avoid static identifiers.

7.2 Stateful Addressing

Due to the multiple computing devices often owned by one individual, DUIDs can provide location and identity information about the user and could be classified as PII. With the proliferation of Internet-connected devices such as smart phones and laptops, the DUIDs associated with these personal devices could be used to identify the device owner. Through the DUID, traffic sniffing and traceroute could provide identity and location information about a user, thus exposing PII. Though the DUID is not a scientifically exclusive identification factor, because multiple devices could compute the same DUID, it provides a relatively accurate marker of identity with personal Internet-connected devices.

7.2.1 Dynamic DUID

A dynamic DUID would address the privacy problems in DHCPv6 without a serious impact on network performance or usability. One of the primary motivations for implementing a DUID was to provide hosts with the same DHCPv6-provided address each session. Due to the privacy risks associated with a static address described in Section 3, maintaining the same address over multiple sessions is an undesirable feature. Obscuring the address often helps protect a user. For those systems that require static addresses, portions of the subnet can be easily configured for static address space.

The dynamic DUID was a successful defense against DUID tracking. When the DUID is set to change on the time-based scheme, it becomes impossible to track the host through the DUID included in DHCPv6 messages. While tracking the host through their MAC address was still possible when available, this type of tracking is not specific to DHCPv6 and is a valid attack for all forms of Ethernet traffic.

When evaluating the overhead of the dynamic DUID, the network, client, and DHCPv6

server overhead must all be accounted for separately. On the network, for each new DUID generated, six ICMPv6 messages are required to lease the address. Out of a 100000 address lease sample of a client and server on the same router, the average time for a DHCPv6 address lease was 1.62 seconds. This overhead is acceptable for clients, which often have long periods of time with little or no network activity. Servers that need to maintain connectivity and availability may need more stable addressing modes. For clients, the overhead can be measured through system calculations. Each client must compute a hash calculation and a pseudo random number generation for each DUID. For clients without resource constraints, the effect of these calculations is negligible. Finally, the system overhead of the DHCPv6 can be significant. The server default stores each DUID until the address timeout period is reached. Since each client is frequently leasing new addresses before their previous address expires, the server often maintains large state tables when clients implement a dynamic DUID. The effect on DHCPv6 server performance was noticeable; after approximately 20000 new DUIDs, the server began to slow in its operation, specifically, leasing addresses.

The most effective trigger for DUID regeneration that balanced privacy and security and system overhead was a combination of DUID regeneration on valid lifetime and network prefix changes. Each interrupt protected against a situation which caused a DUID to remain static. The valid lifetime trigger prevented systems with no movement and long periods of uptime, such as desktops or servers, from having the same DUID. The network prefix trigger protected systems which move frequently without rebooting, such as mobile devices. DUID regeneration on system state changes was effective for most systems, but adjustments to the valid lifetime setting caused similar DUID regeneration and, therefore, similar privacy and security protections.

With dynamically generated DUIDs, there is a remote possibility of an ID collision. When two unique hosts with the same DUID register on the same DHCPv6 server, the server releases the same address to both hosts. The same address then exists twice on the network. Network transmission for this address then fails for one of the hosts and the host must reconnect. Many operating systems have implemented a system in which a RELEASE and REQUEST message are sent to the server to attempt to obtain a new address. If this happens, it is possible that both hosts would receive the new address and the cycle would continue. To prevent address collisions when a DUID collision occurs, hosts implementing a dynamic DUID should always renew their DUID at any address collision on a leased address. While this may cause the host to incur additional overhead for mistaken address collisions, the possibility of two hosts maintaining the same DUID and cycling through addresses is

too great. As previously mentioned, with the large DUID space, the chances of collision are very low; therefore, the overhead of generating a new DUID for each address collision should be minimal.

Due to the system overhead and frequent DUID regeneration and release, mobile systems may want to consider using SLAAC with privacy extensions instead of DHCPv6. The calculations and overhead required for SLAAC and privacy extensions are minimally less than those for DHCPv6 and dynamic DUIDs, but this small difference may lead to a considerable improvement in battery life.

7.2.2 Protocol Security and Router & Firewall Rules

The lack of authentication and authorization in NDP and traditional firewall and router rules allow for hosts to sniff multicast and link-local DHCPv6 messages. NDP, an insecure protocol, is used in conjunction with DHCPv6 to allow systems to advertise addresses to routers and other hosts. The lack of authentication in NDP allows any system, authorized or not, to identify themselves as a recipient of multicast DHCPv6 messages. These systems are then able to sniff DUIDs and link-local address. A secure alternative to NDP has been developed, called SEND protocol [1]. The lack of implementation in routers, however, has prevented the effective deployment and adoption of SEND.

The privacy threat created by DHCPv6 messages can be minimized through proper firewall and router rules. While link-local messages must be passed on the local network, routers and firewalls can be configured to secure access to the reserved multicast DHCPv6 addresses. Proper firewall rules will prevent attackers from sniffing messages sent to DHCPv6 servers containing DUIDs. Multicast DHCPv6 messages are configured, by default, to only stop at the network border, allowing any client inside the LAN to see the traffic. By minimizing the locations where multicast DHCPv6 messages are passed on a network, the privacy and security threat created by DHCPv6 can be minimized.

Chapter 8

Future Work

While the obscuration techniques discussed in Chapter 7 seem simple and easy to implement, the extra overhead required for hash calculations in embedded devices and the additional network configuration and equipment needed for DHCPv6 have caused IID obscuration to be ignored. The additional hardware and decreased performance associated with frequent hash calculations [10] have caused embedded designers to use the EUI-64 format for IID calculation. 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. Also, since the lack of DHCP is advertised as a feature of IPv6, few network administrators are choosing to implement the service. Yet, while operating systems neglect to protect users' privacy through IID obscuration, privacy-minded network administrators will be forced to implement DHCPv6 in order to provide IID obscuration and protect the privacy of their users. Although simple in theory, IID obscuration has yet to be adopted by the IPv6 community.

New features in IPv6 must be analyzed to assure security is implemented at the network layer. Flow labels, a currently unused feature in IPv6, allows for groups of related traffic to be identified by the network layer. When implemented, this feature has the potential to increase security but could also provide a simple label through which attackers can correlate traffic. Also, the large address space in IPv6 allows for a new form of dynamic addressing. Since subnets provide $1.8 \cdot 10^{19}$ addresses, clients within a subnet can have multiple addresses without interrupting connectivity for other clients. By continuous changing and using multiple addresses, a moving target defense is created in which an attacker cannot find a node to target. Flow labels and a large address space must be researched to provide increased security at the network layer.

8.1 Flow Labeling

Flow labeling, an unimplemented feature of IPv6, provides the capability to label network flows in the IPv6 header and allow routers and other Layer 2 network devices to prioritize flow transmission, thereby improving QoS. By selectively transmitting higher priority flows first, bandwidth intensive and real-time applications can ensure a high QoS for users. Other proposed applications of flow labels in IPv6 include implementations which allow for Internet Service Providers (ISPs) to tier QoS for users depending on the service levels. In developing the use of flow labels in IPv6, the security of the technology will be a critical consideration.

Flow labels give network administrators the ability to identify and stop harmful network flows, but also allow attackers to identify sensitive network flows and mine them for data. Current network protection, including IPSs and IDSs, maintain the state of every network connections to detect and block harmful network flows. By labeling the damaging network flows, attacks can be efficiently stopped and decrease overhead. While stopping network attacks is important, the potential harm caused by labeling network flows must also be examined. Just as static identifiers allow for the identification of network traffic, network flow labels would also allow the traffic to be correlated. Depending on the frequency of flow identification regeneration, the flow label could become another static identifier if it is never changed. Serious security concerns must be addressed when implementing network flow labels in IPv6.

8.2 Moving Target Defense in IPv6

The goal of this future research is to protect sensitive communications, which are commonly used by government agencies, from eavesdroppers or social engineers. The current research investigated the privacy implications of SLAAC in IPv6. SLAAC, the default addressing system in IPv6, provides a third party a means to track and monitor targeted users globally using simple tools such as ping and traceroute. Authenticated messages expose the identities of both the sender and receiver to a third party. This research focuses on preventing the issue of IPv6 address tracking as well as creating a "moving target defense." **The Moving Target IPv6 Defense (MT6D) dynamically obscures network and transport layer addresses of packets in IPv6 to achieve anonymity and protect against certain classes of network attacks.** MT6D focuses on providing users with anonymity as well as intrusion protection. It accomplishes this through automatic obscuration of addresses

with no outside involvement. The concept uses the hash of IPv6 IIDs, a shared session key, and a timestamp to obscure and dynamically change the host portion of the sender and receiver addresses. This will prevent an attacker from identifying and monitoring a session between two hosts for more than a few seconds. Packets are encrypted to prevent traffic correlation, which provides significantly improved anonymity. In its preferred implementation, MT6D protects against address tracking, traffic correlation, and certain classes of network attacks. MT6D can be implemented embedded on a host device or as a gateway device, either in software or hardware. Use of MT6D requires negligible configuration and is transparent to applications and hosts. It has numerous applications ranging from hosts desiring to keep their locations private to hosts conducting sensitive communications. Although the primary focus is IPv6, these techniques can also apply to IPv4 provided an available pool of unallocated addresses exists.

MT6D provides a means for hosts to communicate with each other over the public Internet while maintaining complete anonymity from targeting, tracking and traffic correlation. The system is comprised of three main components. The first component dynamically obscures the sender and receiver IIDs and port numbers. The second component dynamically obscures the sender and receiver subnets. The final component obscures message authentication from any third-party host.

Dynamic addressing modifies the network-layer and transport-layer addresses of the sender and receiver nondeterministically. MT6D is capable of dynamically changing these addresses to hide identifiable information about a host, obscuring communicating hosts from any third-party host. A key feature of MT6D is that this obscuration can be made mid-session between two hosts without causing connection reestablishment or breakdown.MT6D can support dynamic subnet obscuration (DSO) using encrypted tunnels between federated MT6D nodes. This is achieved by nondeterministically routing packets through trusted MT6D hosts. Each MT6D host that handles the packet will compute a new DSO header for the packet. By routing packets through other MT6D nodes it will be difficult to trace the original packet from the source subnet to the destination subnet. DSO will incur additional latency due to extra hops and processing imposed on each packet. Subnet obscuration is useful, however, in sparsely populated networks where sniffed traffic could assist a third party in correlating network traffic. No other approach that attempts to hide the sender's identity attempts simultaneously to obscure the subnet.

Another primary objective of MT6D is to allow hosts to authenticate traffic to each other while still maintaining their anonymity to any third party. Authentication privacy is

accomplished by wrapping the original authenticated packet inside a MT6D packet. Since the MT6D packet contents are encrypted by default, a third party will be unable to detect that the packets are authenticated. Further, encryption is done using a secret not tied to an identity. This means that the encrypted packets reveal nothing about either the sender's or receiver's true identity.

MT6D also provides the option of encrypting each original packet before appending it with the MT6D header. By encrypting the original packet, a third party is unable to glean any useful information from the packet. For example, if the original packet is sent using TCP, the header gets encrypted so that a third party cannot attempt to correlate network traffic using the TCP sequence numbers. In addition, the nature of the network traffic is kept private through encryption.

8.2.1 Implementations

As a prototype device, implementing MT6D in software as a gateway device supports any operating system or application by providing encryption at the network layer. Since the protocol is implemented in software, performance measurements and protocol modifications are simple to make. As a gateway device, testing different types of traffic from different devices is possible without significant modifications. A network gateway implementation limits performance, but allows for a variety of measurements and modifications to be easily made on the MT6D protocol.

Implementing MT6D for mobile devices must decrease calculations and mitigate its impact on performance and battery life to prevent geotemporal tracking. While many security devices work for systems with minimal constraints, the limited battery resources of mobile devices require specific modifications to assure performance and system security are maintained. To adapt a moving target defense to mobile devices, the different factors which create movement must be analyzed to determine their impact on battery life. Then, the moving target defense must be adapted to limit the factors which negatively affect mobile devices, especially battery life.

To increase performance for high bandwidth application, MT6D must be implemented in hardware. Packet processing is inefficient in software due to the high system overhead. The hardware implementation of MT6D will face additional challenges, such as importing or transmitting connection profiles. The ability to implement modules hardware devices in a hardware prototyping environment, such as a field programmable gate array (FPGA), allows

for an efficient moving target defense to be implemented in hardware.

8.2.2 Applications

While MT6D has obvious applications in protecting security and privacy for simple Hypertext Transfer Protocol (HTTP) and web traffic, MT6D has the potential to offer a new level of security for BitTorrent and Voice over IP (VoIP) traffic. The distributed nature of BitTorrent allows for systems to use the file hash as a shared key, creating a peer-to-peer (P2P) network in which addresses of the network are constantly changing to avoid tracking [**?**]. Also, since MT6D may incur packet loss when addresses are changing, VoIP protocols which dynamically adjust QoS depending on packet loss would benefit from the additional security while mitigating the effects of lost packets.

BitTorrent is a P2P protocol which could gain anonymity, security, and performance from additional network layer security; also, other features of IPv6 could contribute to performance increases of distributed P2P architectures. By using the file hash as a secret key for address rotations, any user could connect to an anonymous network to download or share the file. By adding the additional layer of encryption at the network layer, the performance could be improved by removing the currently implemented application layer encryption. The unique feature of IPv6 multicast could also allow for users on the same subnet to join multicast groups, offloading processing to routers. Implementing a moving target defense in IPv6 could allow distributed P2P protocols to gain additional anonymity and security and use new features in IPv6 to increase performance.

One of the primary protocols of VoIP, SIP is an insecure protocol which incorporates packet loss into application layer processing and QoS adjustments to assure message transmission. The primary methods of securing SIP include TLS and secure socket layer (SSL), requiring application layer processing to ensure message security. By using a moving target defense, message security is accomplished through network-layer encryption and message transmission becomes difficult to track. Also, current wiretapping laws governed by Communications Assistance for Law Enforcement Act (CALEA) require telecommunications service providers to allow for real-time surveillance of telephone and network traffic. By implementing synchronized dynamic addressing with a VoIP protocol such as SIP, decentralized communications are possible where CALEA and other wiretapping laws may not apply.

Chapter 9

Conclusion

Static identifiers in network computing present a serious problem for privacy and security. Many classes of static identifiers, such as social security numbers (SSNs), are protected by individuals and legally regulated to avoid compromises and potential identity theft. New technologies, however, are publicly exposing static identifiers used to classify users. Two examples of these identifiers are the static IID in SLAAC in IPv6 and the DUID in DHCPv6. Since most people have portable computing devices that they carry with them, these static identifiers can be used to geotemporally track users and correlate their network activities.

To test the validity of geotemporal tracking and traffic analysis with the two discovered IPv6 static identifiers, two different experiments were designed. To test static IIDs, the Virginia Tech campus subnets were geographically mapped. Then, a single host moved through the subnets and used network resources while the attacker pinged the host on each subnet at preset time intervals and captured all of the host's traffic at the border. For the DHCPv6 DUID, three common scenarios using DHCPv6 tested for vulnerabilities which would expose the static identifier to an attacker.

The novel results and analysis gained from these experiments shows the security and privacy threats in IPv6 stateless and stateful addressing protocols. SLAAC allowed users to be accurately geotemporally tracked and have their traffic correlated between multiple, unique network sessions. All of the scenarios created to exploit DHCPv6 were successful in allowing an attacker to gain the static DUID and track users' addresses, allowing for the same geotemporal tracking and network traffic correlation as in SLAAC, but limited to to the LAN.

Potential solutions to the solving the static identifier problem in SLAAC and DHCPv6 include address obfuscation, encryption schemes, and tunneling. SLAAC users can use CGAs

and IPv6 privacy extensions to obscure the static IID. DHCPv6 users can use strong router and firewall rules to protect their static DUID, or they can use the novel dynamic DUID scheme presented in Chapter 7.2.1.

The security and privacy threats of static identifiers lead to the future work in analyzing the security of IPv6 flow labels and the application of a moving target defense in IPv6. By labeling each network flow, it becomes trivial for an attacker to correlate traffic. While flow labels have potential benefits in security and QoS, a secure implementation must be used to avoid compromising security and privacy. Also, the large address space of IPv6 allows a user to hide without being detected, creating the opportunity for implementing a moving target defense. By synchronizing the address changes, hosts can maintain anonymous communication channels without being targeted for network attacks.

Bibliography

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

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

[3] Claude Castelluccia, Francis Dupont, and Gabriel 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, October 2004.

[4] A.R. 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, November 2009.

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

[6] R. Despres. IPv6 Rapid Deployment on IPv4 Infrastructures (6rd). RFC 5569 (Informational), January 2010.

[7] Roger Dingledine, Nick Mathewson, and Paul Syverson. Tor: the second-generation onion router. In *SSYM'04: Proceedings of the 13th conference on USENIX Security Symposium*, pages 21–21, Berkeley, CA, USA, 2004. USENIX Association.

[8] R. Droms. DNS Configuration options for Dynamic Host Configuration Protocol for IPv6 (DHCPv6). RFC 3646 (Proposed Standard), December 2003.

[9] 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.

[10] Mathilde Durvy, Julien Abeillé, Patrick Wetterwald, Colin O'Flynn, Blake Leverett, Eric Gnoske, Michael Vidales, Geoff Mulligan, Nicolas Tsiftes, Niclas Finne, and Adam 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.

[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), February 2006.

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

[14] Peter C. Johnson, Apu Kapadia, Patrick P. Tsang, and Sean W. Smith. Nymble: anonymous IP-address blocking. In *PET'07: Proceedings of the 7th international conference on Privacy enhancing technologies*, pages 113–133, Berlin, Heidelberg, 2007. Springer-Verlag.

[15] V. Kalusivalingam. Network Information Service (NIS) Configuration Options for Dynamic Host Configuration Protocol for IPv6 (DHCPv6). RFC 3898 (Proposed Standard), October 2004.

[16] V. Kalusivalingam. Simple Network Time Protocol (SNTP) Configuration Option for DHCPv6. RFC 4075 (Proposed Standard), May 2005.

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

[18] S. Kent and R. Atkinson. Security Architecture for the Internet Protocol. RFC 2401 (Proposed Standard), November 1998. Obsoleted by RFC 4301, updated by RFC 3168.

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

[20] D. Koukis, S. Antonatos, and K. G. Anagnostakis. On the privacy risks of publishing anonymized IP network traces. *Communications and Multimedia Security*, 4237:22–32, 2006.

[21] L. Morand, A. Yegin, S. Kumar, and S. Madanapalli. DHCP Options for Protocol for Carrying Authentication for Network Access (PANA) Authentication Agents. RFC 5192 (Proposed Standard), May 2008.

[22] Tomasz Mrugalski and Marek Senderski. DHCPv6: Dibbler - a portable DHCPv6. Available at: `http://klub.com.pl/dhcpv6/` accessed Aug 2010.

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

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

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

[26] D. Plummer. Ethernet Address Resolution Protocol: Or Converting Network Protocol Addresses to 48.bit Ethernet Address for Transmission on Ethernet Hardware. RFC 826 (Standard), November 1982. Updated by RFCs 5227, 5494.

[27] Ying Qiu, Jianying Zhou, Feng 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, September 2005.

[28] H. Schulzrinne and B. Volz. Dynamic Host Configuration Protocol (DHCPv6) Options for Session Initiation Protocol (SIP) Servers. RFC 3319 (Proposed Standard), July 2003.

[29] Clay Shields and Brian Neil Levine. A protocol for anonymous communication over the Internet. In *CCS '00: Proceedings of the 7th ACM conference on Computer and communications security*, pages 33–42, New York, NY, USA, 2000. ACM.

[30] Adam J. Slagell and William Yurcik. Sharing computer network logs for security and privacy: A motivation for new methodologies of anonymization, 2004.

[31] Jonathan G G. Tams, Ronald Brown, David J Maxwell, and Mark A Pearce. Tracking dynamic addresses on a network. Patent, March 2005. US 6862286.

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

[33] 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.