Security and Privacy produced by DHCP Unique Identifiers

Joseph Tront[2] Stephen Groat[1,2] Matthew Dunlop[1,2] Randy Marchany[1]
Information Technology Security Office and Lab[1]
Bradley Department of Electrical and Computer Engineering[2]
Virginia Polytechnic Institute and State University (Virginia Tech), Blacksburg, VA 24061, USA
Email: jgtront@vt.edu

Abstract

As protection against the current privacy weaknesses of StateLess Address AutoConfiguration (SLAAC) in the Internet Protocol version 6 (IPv6), network administrators may choose to deploy the new Dynamic Host Configuration Protocol for IPv6 (DHCPv6). Similar to the Dynamic Host Configuration Protocol (DHCP) for the Internet Protocol version 4 (IPv4), DHCPv6 uses a client-server model to manage addresses in networks, providing stateful address assignment. While DHCPv6 can be configured to assign randomly distributed addresses to clients, the DHCP Unique Identifier (DUID) was designed to identify uniquely identify clients to servers and remains static to clients as they move between different subnets and networks. Since the DUID is globally unique and exposed in the clear, attackers can geotemporally track clients by sniffing DHCPv6 messages on the local network or by using unauthenticated protocol-valid queries that request systems' DUIDs or leased addresses. DUIDs can also be formed with system-specific information, further compromising the privacy and security of the host. To combat the threat of the static DUID, a dynamic DUID was implemented and analyzed for its effect on privacy and security as well as its computational overhead. The privacy implications of DHCPv6 must be addressed before large-scale IPv6 deployment.

1. Introduction

As the address space in the current Internet Protocol version 4 (IPv4) [17] is depleted, networks will be forced to transition to the Internet Protocol version 6 (IPv6). IPv6 increases the address size to 128 bits and was designed to support the increased numbers of users and emerging classes of devices that require globally unique addresses. The International Assigned Number Authority (IANA) reported in February 2011 that no more IPv4 address blocks remain [15]. 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 StateLess Address AutoConfiguration (SLAAC) 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. SLAAC allows for nodes to configure the interface identifier (IID), or last 64 bits, of their address independently. While this method eases the burden on network administrators, the static configuration used by many common IPv6 network implementations threatens the privacy of users [6].

The static IID used in SLAAC is globally available and exposes a host to geotemporal tracking and traffic monitoring between different sessions and subnets. Since the IID remains the same as users move between subnets, attackers can ping different subnets for a specific user and identify his or her location through basic reconnaissance or traceroute. Additionally, when attackers successfully sniff network traffic in different locations for a particular user, the static IID makes correlation of traffic over multiple sessions trivial.

The inherent privacy and security flaws in SLAAC will lead network administrators to consider the Dynamic Host Configuration Protocol for IPv6 (DHCPv6) as a stateful addressing alternative. DHCPv6 allows administrators more control over address distribution than SLAAC. It provides many of the same features as the Dynamic Host Configuration Protocol (DHCP) implemented in IPv4, including logging and auditing capabilities. To provide more precise auditing and configuration, DHCPv6 deploys a DHCP Unique Identifier (DUID). Similar to how the Media Access Control (MAC) address is used to identify hosts at the data link layer, the DUID is used to identify unique clients to the server. While DHCPv6 is a stateful protocol which can be configured to change nodes' addresses frequently, the static DUID allows nodes to be tracked through a static value in many common DHCPv6 messages.

The DUID allows for correlation of users' addresses over multiple sessions, creating similar privacy and monitoring concerns as SLAAC. RFC 3315 states that the DUID is static [4] and current operating system (OS) client implementations use a permanent DUID. Since the DUID for a client persists between sessions and networks, users can be geotemporally tracked and have their traffic correlated. Typically, attackers must be on the link-local network to receive the SOLICIT and ADVERTISE messages containing DUIDs. Attackers can also remotely monitor and track clients by planting modified relays at targeted sites to forward multicast DHCPv6 messages for analysis. Once the DUID contained in the DHCPv6 message is obtained, attackers can sniff the network for the DHCPv6 responses or query the DHCPv6 server for hosts' addresses. Since stateful addressing has similar privacy concerns as SLAAC, IPv6 address assignment must be improved before large scale deployment weakens privacy for unsuspecting users.

This paper discusses the issues and concerns of a static DUID in DHCPv6. Section 2 provides background information on IPv6 and DHCP. Work related to the fields of privacy in DHCP and IPv6 addressing is surveyed in Section 3. In Section 4, live network demonstrations are conducted of both tracking DHCPv6 DUIDs and of a privacy and security solution involving dynamic DUIDs. The results of these demonstrations is analyzed in Section 5. The paper concludes in Section 6 and Section 7 discusses future work.

2. Research 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 addresses space, however, also creates an additional administrative burden for network engineers. To accommodate the rapid expansion of the address space, a self-configured address system was developed to facilitate management. SLAAC has been proven to provide a vector for attackers to geographically track and logically monitor unsuspecting users.

Some believe that managed address configuration techniques, mainly DHCPv6, will provide users with privacy in IPv6. Yet, the static DUID value publicly broadcast in DHCPv6 messages provides another vector for attackers to correlate nodes and addresses. Similar in nature to the IID, the DUID can provide attackers with a static value to correlate a particular node's addresses over multiple sessions and networks. While the DUID is limited to the local scope, experienced attackers can use modified DHCPv6 relays and other techniques to increase the scope of their attacks.

2.1. Features of IPv6

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

Larger address space was not the only improvement made in IPv6. The header format was simplified. Unused fields were removed and the header length in IPv6 was fixed to 40 bytes. Another improvement was moving the options out of the header. In IPv6 options are now located in the payload section of the packet. This allows for more options if desired. It also provides space for the defining of new options. The addition of flow labels was also incorporated into IPv6. Flow labels allow traffic to be classified and potentially handled differently by routers. The final major improvement to IPv6 is the incorporation of IPSec [8]. In IPv4, IPSec is not integrated. It was designed after the protocol was fielded, primarily because security was not initially a concern. When IPSec was integrated into the OSI model, it only fit between the network and transport layers. This creates additional overhead and storage requirements as the data has to travel back up the stack for encryption and authentication in tunnel mode. Including IPSec as part of the network layer provides better efficiency and throughput.

2.2. Privacy in IPv6 SLAAC

The rapid growth of the address space in IPv6 requires new subnet management techniques to handle the over $1.8 \cdot 10^{19}$ possible addresses in an IPv6 subnet. The default solution to the problem of subnet management in IPv6 is SLAAC. SLAAC allows an administrator to configure the network and subnet portions of the address, while each device automatically configures the IID, or host portion of the address. The IID is often formed by extending the 48-bit MAC address to 64 bits through the Extended Unique Identifier (EUI)-64 format.

Using a node's MAC address in the IID has serious unintended consequences to a user's privacy [6]. The observation that this addressing scheme could allow an attacker to analyze payload, packet size, and packet timing was first made in RFC 4941 [13]. The issue is not only that the MAC address is used as the IID, but also that the IID remains static. As a result, no matter what network the node accesses, the IID remains the same. Common network tools, such as ping and traceroute, permit tracking a node's geographic location from anywhere in the world.

With IPv6 SLAAC, a user's privacy can also be violated through the monitoring of network traffic. Traffic analysis can deduce an identity by correlating traffic captures from a specific IID. This analysis is possible in IPv4, but only for short periods of time. Addresses in IPv4 are assigned using DHCP. Since DHCP addresses change depending on configuration parameters and availability, prolonged tracking is more difficult. In contrast, a static IID permits correlation of a specific user's data over multiple sessions and subnets. Deterministic IPv6

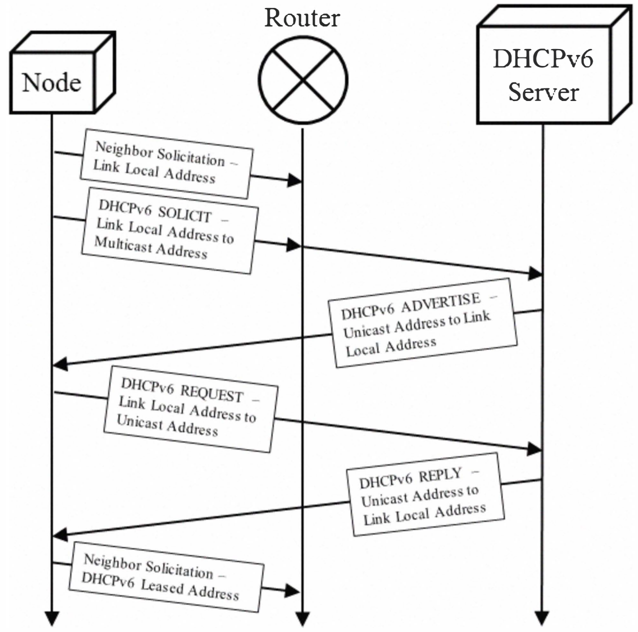
Router
Node
DHCPv6 Server
Neighbor Solicitation – Link Local Address
DHCPv6 SOLICIT – Link Local Address to Multicast Address
DHCPv6 ADVERTISE – Unicast Address to Link Local Address
DHCPv6 REQUEST – Link Local Address to Unicast Address
DHCPv6 REPLY – Unicast Address to Link Local Address
Neighbor Solicitation – DHCPv6 Leased Address

Figure 1. DHCP v6 Message Exchange

addresses with static IIDs globally tie users to each of their packets and make traffic correlation over multiple sessions possible. Once an attacker is able to deduce a user's identity and location, the attacker can then target the user for identity theft or other identity-related crimes. Using static IIDs to monitor traffic for identity theft is one of many potential privacy exploits of deterministic stateless IPv6 addressing.

2.3. Stateful Address Configuration in IPv6

DHCPv6 [4] provides a stateful, managed alternative to SLAAC. While DHCPv6 is not used in IPv6 as the default addressing mode, DHCP is the predominate way of assigning addresses in IPv4. The management and configuration options offered by the protocol can be useful in large, complex networks. Different realms can be established to configure addresses differently for different parts of the networks. Other custom solutions can also be implemented, including servers which rely on public key infrastructure (PKI) for authentication and authorization to lease addresses.

When a node connects to a network, it follows the message exchange sequence illustrated in Figure 1. The client begins by sending a multicast SOLICIT message to special addresses reserved for DHCPv6 servers. One component of the SOLICIT message is the client identifier, which contains the client's unique 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. The ADVERTISE message contains both the client's and server's unique DUIDs. DHCPv6 can also be configured to provide other configuration information to network nodes, such as Domain Name System (DNS) and Network Time Protocol (NTP) servers [3]. To reserve the DHCPv6 leased addresses and notify routers and other clients, servers and clients also use Neighbor Discovery Protocol (NDP) [14] as in SLAAC.

While the insider threats of a Denial-of-Service (DoS) or man-in-the-middle (MITM) attacks have already been recognized in DHCPv6 [4], the privacy implications of using a static DUID 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. As mentioned, the SOLICIT and ADVERTISE messages contain users' DUIDs. This inclusion allows attackers to identify nodes through their static DUIDs and monitor traffic for the session through the leased address. If attackers miss a targeted node's ADVERTISE messages, but already know the node's DUID, they can send unauthenticated client-initiated INFORMATION-REQUEST messages to the DHCPv6 server requesting information on a leased address associated with a particular DUID.

3. Related Work

Address tracking and traffic correlation through DHCP is a relatively unexplored area. During the literature review, no other research involving DHCPv6 privacy implications was discovered. A technique by Tams et al. [19] exists that tracks DHCP users in IPv4. Hosts using both IPv4 and IPv6 simultaneously could be victim to this technique, but only on the IPv4 address. For that reason, this technique is discussed. We also discuss research that investigates the privacy implications of SLAAC to illustrate that both stateful and stateless addressing in IPv6 is flawed.

Tracking address assignment in DHCP in IPv4 has already been established by patent for a particular method of DHCP address tracking. Tams et al. [19] 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 our work is similar, we execute 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 Address Resolution Protocol (ARP), neither of which is available in IPv6.

Since the focus of this research is on the privacy vulnerabilities of DHCPv6, it is also pertinent to address work illustrating more significant vulnerabilities of IPv6 SLAAC. Groat et al. [6] discussed the ability to geographically track a user and monitor and correlate their traffic using the static IID of the autoconfigured address.

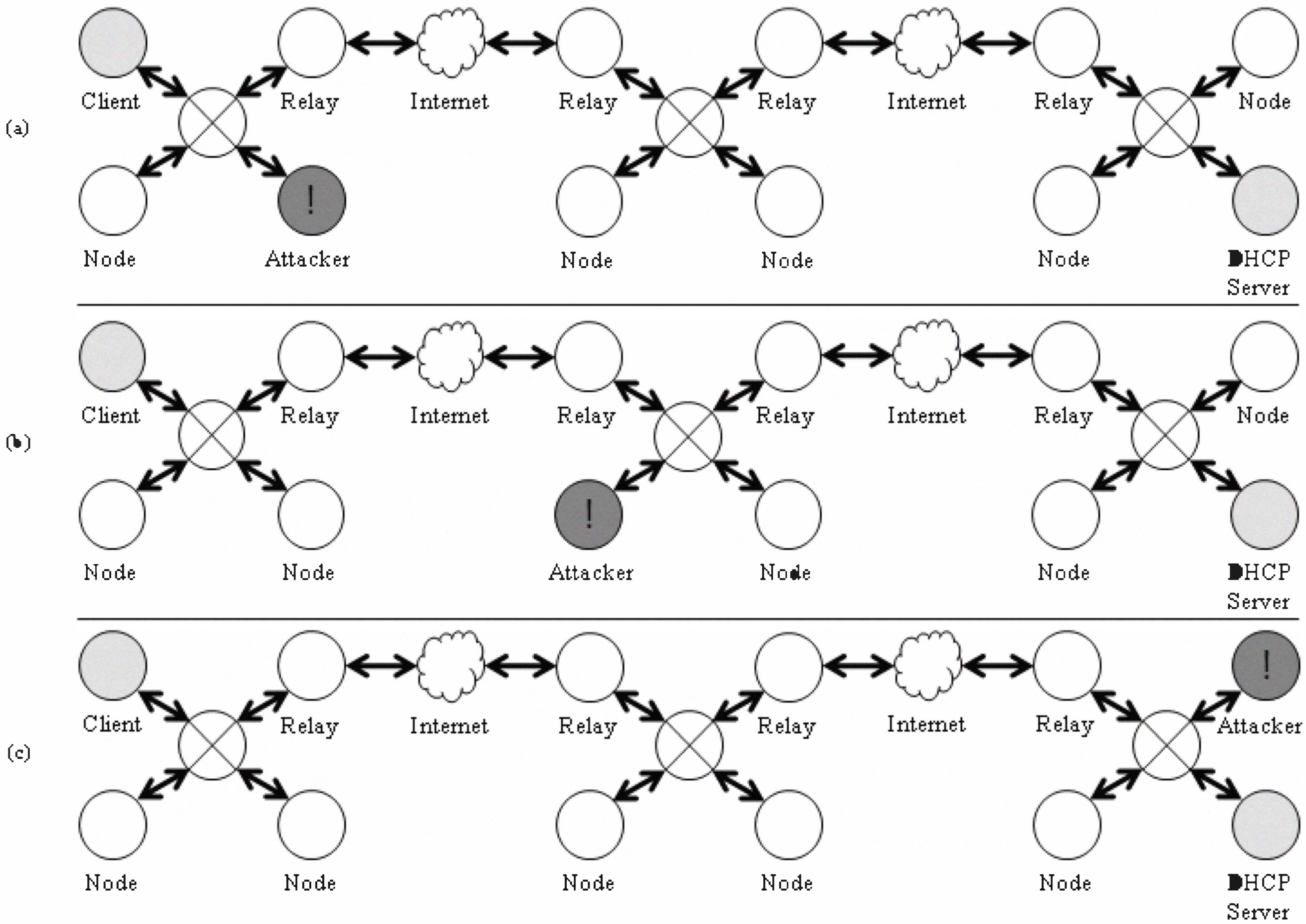

Figure 2. Three different scenarios of DHCPv6 message sniffing inside a LAN

RFC 4941 identified this problem [13] and concluded that a non-changing interface would allow an eavesdropper to correlate unrelated information with a particular node. Haddad addressed the fact that mobile nodes using IPv6 SLAAC can reveal their location to an eavesdropper [7]. Our work builds on tracking stateless autoconfigured addresses by showing the privacy of stateful addresses is also compromised through the static DUID.

4. Study Design

To exploit the weakness of static DUIDs in DHCPv6, address correlation was performed both on the local area network (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. We were able to verify that the 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 the DHCPv6 message 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.

4.1. 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 to 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 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 nor the client. The attacker relies on sniffing messages passed between the relays to sniff DUIDs and then querying 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 or she can also masquerade as the client. Being on the same subnet as the server, it is trivial for the attacker to intercept address leases and traffic and send false responses to the server before the actual client is able to. By doing so, the attacker is capable of denying the user network access.

In Figure 2(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 message 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 2(b). On a relay subnet, a client's address can still be compromised using an approach similar to the approach in Figure 2(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 2(c) shows an attacker sniffing DHCPv6 messages on the same router or switch 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 2(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. The attacker then acts as a man-in-the-middle, passing packets to the client only after he/she has intercepted and forwarded them.

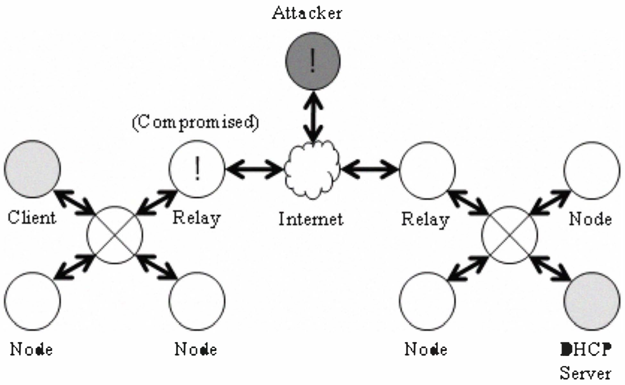

Figure 3. A compromised DHCPv6 relay passes DHCPv6 messages from a LAN to an attacker

4.2. Remote Monitoring through DHCPv6 Relays

To monitor the addresses of DHCPv6 nodes remotely, a compromised DHCPv6 relay can sniff DHCPv6 SOLICIT messages and send them to a remote host for analysis as illustrated in Figure 3. 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.

4.3. 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, a simpler IID obstruction technique, 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 [8]. 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 clients' 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 2^{128}, or 340 undecillion, possible DUIDs. Therefore, DUID collision is unlikely. In the unlikely event of an address collision, two hosts will be issued the same address, possibly creating a race condition.

There is the remote possibility of a DUID collision occurring with dynamically generated DUIDs. When two unique hosts with the same DUID register on the same DHCPv6 server, the server releases the same address to both hosts. At this time, the same address 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 RELEASE and REQUEST messages are sent to the server to attempt to gain 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 occur, 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 frequency with which this occurs is rare enough to accept the risk, especially when compared to the possibility of two hosts maintaining the same DUID and entering a duplicate DUID race condition. As previously mentioned, with the large DUID space, the chances of collision are improbable. Therefore, the overhead of generating a new DUID for each address collision should be minimal.

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 SIP configuration and DNS configuration, are not broken by a dynamic DUID [5,18]. 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 [9,10,12]. These configuration parameters will have to be reestablished for each connection, causing additional overhead. All DHCPv6 extensions, however, will still continue to function. No further modifications to the DHCPv6 protocol were necessary to maintain functionality of all extensions.

5. Analysis of Results

The scenarios described in Section 4 were performed on the Virginia Tech production IPv6 network using a Dibbler [11] DHCPv6 server and client running Ubuntu 10.04. The Virginia Tech IPv6 network uses SLAAC and the NDP to allow nodes to self-configure addresses. By providing a DHCPv6 server on the network, nodes that were set to self-configure IPv6 addresses continued to operate using SLAAC. 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 the prescribed locations. All SOLICIT and ADVERTISE messages were successfully captured in the trials. INFORMATION-REQUEST messages were also successfully sent and received to learn a node's leased address. All of the exploits described in Section 4 were successfully run to discover the address of the target.

The three major vulnerabilities we discovered in the current implementation of DHCPv6 are the formation of the DUID, the static DUID, and the security of IPv6 routers. Different DUID formats reveal unnecessary information about the user. The static DUID is the primary vector for address correlation in DHCPv6, providing a link to a specific computer and, often, a human identity. To prevent the DUID from being exposed, the security of DHCPv6 and NDP must be corrected.

5.1. DUID Formation and Identity Associations

DUIDs are formed using three different methods, all of which are static and permanent to systems. 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 expose unnecessary static, identifiable information about DHCPv6 nodes.

A DUID is commonly formed by combining a link local address with the time. This method is defined as the 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, the following four octets as the time in seconds since January 1, 2000 modulo 2^{32}, and a variable length link layer address. Since the link layer address is usually the MAC address, exposing the MAC in the DUID creates the same additional privacy and security threats identified in SLAAC. The amount of the link layer address exposed depends of the OS implementation; the less exposed, the more secure the implementation. Common operating systems, such as Windows 7 and Ubuntu, expose only vendor information in the Organizational Unit Identifier (OUI) of the MAC address in this variable length field, but the variable length field has the potential to expose the entire address.

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 type 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 each vendor, similar to an OUI. 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 extensive reconnaissance and knowledge of any vulnerability associated with how specific manufacturers implement the unique identifiers.

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 then a variable length link layer address. With more room to expose the link layer address, most implementations will use the entire MAC address of a system, exposing 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, in which attackers target machines based off of known vendor vulnerabilities.

The variable length of the DUID also has an effect on its security. While the DUID has a maximum length of 128 bits, common implementations, such as Microsoft Windows 7, only use 112 bits. The decreased size of the DUID decreases the entropy in the DUID and makes it easier for an attacker to track. The network and computational overhead required to process the extra information is trivial; all DUIDs should be expanded to a mandatory 128 bits.

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

5.2. Static DUID

Due to the multiple computing devices often owned by one individual, DUIDs can provide location and identity information about their user and could be classified as personally identifiable information (PII). Since people carry 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 (since multiple devices could feasibly compute the same DUID) the DUID provides a relatively accurate marker of identity with personal devices.

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 2, 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.

Since a dual-stack implementation of IPv4 and IPv6 would require both versions of DHCP to statefully address nodes, the recommendation has been made to add the DUID configuration parameter to DHCP for IPv4 [2]. Currently, DHCP for IPv4 uses a MAC address to keep state. By adding the DUID to another protocol, sensitive identity information would be further exposed. Modifying or removing the DUID once it has been integrated into two different protocols will be more difficult. Therefore, the security concerns of a static DUID must be addressed now.

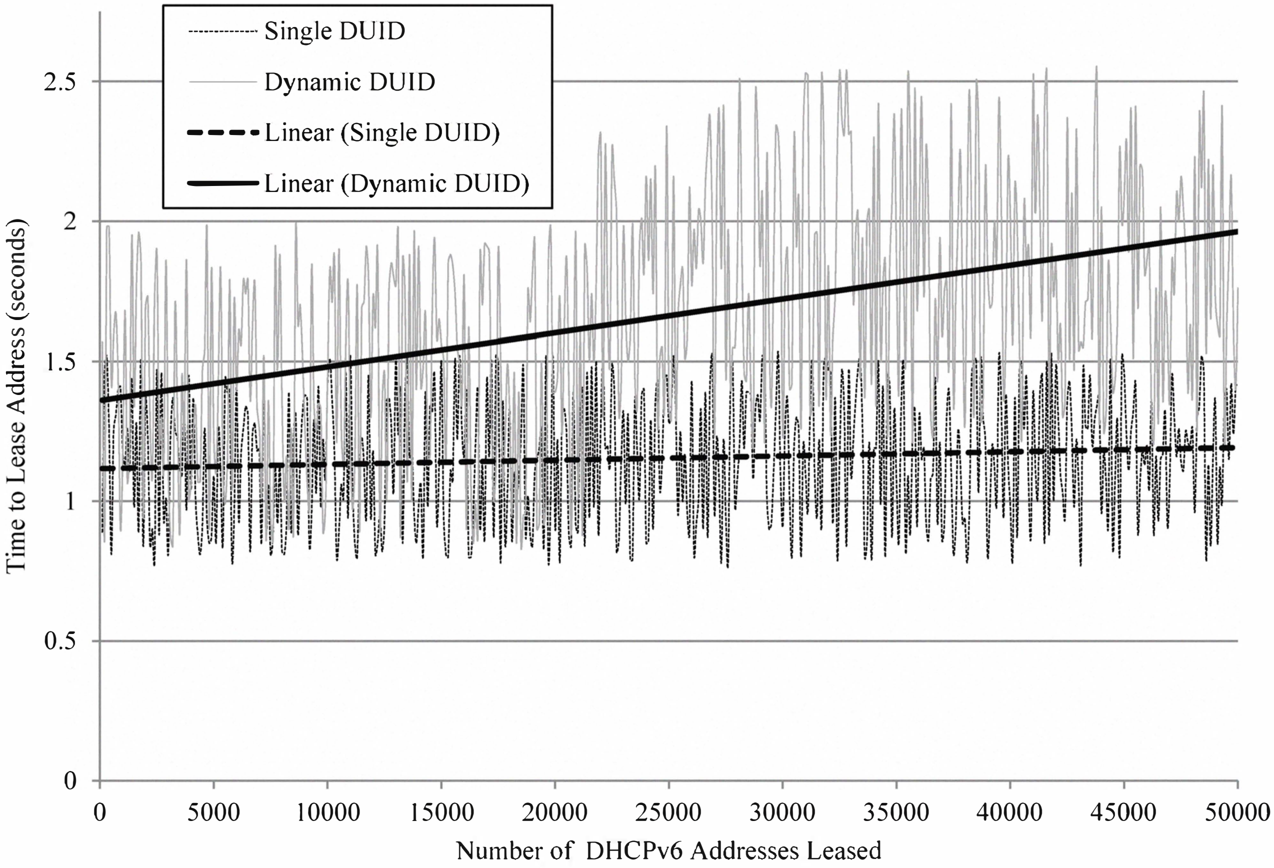

Figure 4. Comparison of lease time for static and dynamic DUIDs

5.3. Protocol Security and Packet Filtering

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 SEcure Neighbor Discovery (SEND) [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 LAN, the privacy and security threat created by DHCPv6 can be minimized.

5.4. Dynamic DUID

The dynamic DUID is 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 hosts 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 100,000 address lease sample of a client and server on the same router, the average time for a DHCPv6 address lease was approximately 1.95 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 generate a pseudo random number and execute a hash calculation for each DUID. For clients without resource constraints, the effect of these calculations is negligible. For resource constrained devices, the system overhead of DHCPv6 can be significant.

By default, the server 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 20,000 new DUIDs, the server began to exhibit a large amount of latency in its operation, specifically, leasing addresses. This decrease in performance can be seen in Figure 4. Since the DHCPv6 server only had to track the state of a single client, the non-changing DUID was trivial for it to maintain. For a single client, lease delays are not apparent when static DUIDs are used. Since the tables of DUIDs and leased addressed remain small, the server does not have to process large files. Yet, the dynamic DUID caused a dramatic increase in lease time after 20,000 DUIDs. The lease delays seen in dynamic DUIDs could be mitigated by changing the type of storage the DHCPv6 servers uses for DUID state tables. If a database or other type of fast lookup is used, lookup performance and lease time could be improved. Since the storage techniques currently used by DHCPv6 are simple file and memory allocations, any improvements to handle the volume of DUIDs generated in a dynamic DUID scheme would lead to large performance gains.

The most effective trigger for DUID regeneration that balanced privacy, 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 period 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, from being geotemporally tracked. DUID regeneration on system state changes was effective for most systems. Adjustments to the valid lifetime regeneration trigger caused similar DUID regeneration and, therefore, similar privacy and security protections.

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 using dynamic DUIDs. This small difference, however, may lead to large differences in battery life.

6. Conclusion

IPv6 is a definite improvement over IPv4, allowing more devices to connect to the Internet using globally unique addresses. However, the privacy implications of static DUIDs should be addressed before the protocol is globally deployed. The ability to track users' DHCPv6 issued addresses combined with the ability to track stateless addresses with static IIDs compromises privacy in all known addressing methods in IPv6. While DUID correlation and analysis is limited in scope, the possibility of relay networks or compromised relays extends an attacker's reach beyond local networks. With both methods of IPv6 addressing compromised, changes must be made to assure users' privacy.

A number of different methods can be used to obscure users' DUIDs from monitoring. While DHCPv6 servers can be configured to increase the randomness of addresses issued, a dynamic DUID configured by the OS appears to be the best privacy option. By changing the DUID at each network connection, all DHCPv6 servers in any configuration will issue randomly available addresses to each client. The lack of client and network support for DHCPv6, however, will likely mean the issue will be ignored until subnets begin to crowd and DHCPv6 is required for efficient operation. Regardless of the solution implemented, some method of DUID obscuration should be deployed as part of operating systems and embedded devices to protect the privacy of users.

7. Future Work

There are many ways that systems using stateful addresses can be exploited, both negatively and positively. In future research, we will explore in detail some fields that could be impacted. Static DUIDs allow for cyber stalkers to gather local information and read targets' traffic. Also, static DUIDs permit terrorists to gather information about targets without alerting authorities. On the other hand, similar types of static identifiers can be captured for legitimate analysis by marketers or gathered by law enforcement officials for forensic analysis. Regardless of intent, users' location and Internet activity should be protected from passive monitoring.

The next phase of the research will focus on designing and implementing a dynamic, nondeterministic method of leasing addresses for DHCPv6 clients. Currently, a client is forced to use a DHCPv6 leased address minimally for of a complete session. With the address being the same for that session, traffic can be correlated within the session. Any unencrypted traffic may expose a user's identity, allowing traffic over multiple sessions to be correlated. By using multiple addresses per session, it becomes statistically infeasible for an attacker to correlate traffic over multiple sessions, even if the traffic is unencrypted. Also, implementing authentication in lease query requests, either through source addresses or other methods, would help to prevent some of these attacks.

8. References

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

[2] T. Chown, S. Venaas, and C. Strauf. Dynamic Host Configuration Protocol (DHCP): IPv4 and IPv6 Dual-Stack Issues. RFC 4477 (Informational), May 2006.

[3] R. Droms. Stateless Dynamic Host Configuration Protocol (DHCP) Service for IPv6. RFC 3736 (Proposed Standard), Apr. 2004.

[4] 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.

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

[6] S. Groat, M. Dunlop, R. Marchany, and J. Tront. The privacy implications of stateless IPv6 addressing. In Proceedings of the Sixth Annual Workshop on Cyber Security and Information Intelligence Research, CSIIRW '10, pages 52:1-52:4, New York, NY, USA, 2010. ACM.

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

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

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

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

[11] T. Mrugalski and M. Senderski. DHCPv6: Dibbler - a portable DHCPv6. Available at: http://klub.com.pl/dhcpv6/, (accessed Aug 2010).

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

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

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

[15] Remaining IPv4 address space. Available at: http://www.iana.org/assignments/ipv4-address-space/ipv4-address-space.xml, Feb. 2011.

[16] 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.

[17] J. Postel. Internet Protocol. RFC 791 (Standard), Sept. 1981. Updated by RFC 1349.

[18] H. Schulzrinne, Dynamic Host Configuration Protocol (DHCPv6) Options for Session Initiation Protocol (SIP) Servers. RFC 3319. Proposed Standard. Jul. 2003.

[19] J. G. G. Tams, R. Brown, D. J. Maxwell, and M. A. Pearce. Tracking dynamic addresses on a network. Patent, Mar. 2005. US 686228

9. Acknowledgements

The authors would like the thank Communications and Network Services at Virginia Tech for their support of this research. Specifically, the knowledge, advice and support of Phil Benchoff and Carl Harris have been invaluable. Their IPv6 expertise allowed this research to be conducted on a production IPv6 network and properly tested and evaluated.