2013 27th International Conference on Advanced Information Networking and Applications Workshops

DNS Update Extension to IPv6 Secure Addressing

Hosnieh Rafiee, Martin von Löwis, Christoph Meinel
Hasso-Plattner-Institut, University of Potsdam
P.O. Box 900460, 14440 Potsdam, Germany
{Hosnieh.Rafiee, Martin.vonLoewis, Christoph.Meinel}@hpi.uni-potsdam.de

***Abstract*— The Domain Name System (DNS) is a primary component of the Internet. It was for this reason that a DNS update mechanism was created and implemented giving hosts the ability to dynamically change DNS entries. But this new mechanism exposed DNS servers to new security vulnerabilities so some security protocols were introduced to address these issues. In Internet Protocol version 4 (IPv4), these mechanisms did resolve most security issues which concerned the authentication between a node and the DNS server while in Internet Protocol version 6 (IPv6) networks this became more difficult. This is due to the fact most protocols introduced to organize the large IPv6 address space do not support DNS authentication or have no option for secure DNS updating. We propose an extension to SEcure Neighbor Discovery (SEND) that will allow for updating DNS records dynamically at the same time that the host sets its IP address and sends its Neighbor Advertisement (NA) message announcing its IP to other nodes on the network. If the SEND-enabled DNS server is in the same local link, then the DNS server updates the DNS Resource Records (RRs) by listening to these NA messages. If not, we propose the use of a Controlling Node (CN) with which to communicate with the DNS server via common, secure DNS Update messages. This CN mediates between local link nodes and the DNS server.**

***Index Terms*—SEND, DNS update, DNS authentication, SEND extension**

I. INTRODUCTION

The DNS (RFC 1034) is a vital element of the Internet. It allows other hosts, on the Internet, to send their queries to DNS servers and to ask about a host's domain name or IP address on the Internet. The DNS creates a meaningful mapping between a hostname and its physical IP address. DNS thus provides a service where domain names can be used in the place of the actual cryptic host IP addresses.

DNS consists of a hierarchical database where data is stored in a particular format in what are called Resource Records (RRs). These RRs are distinguished by their record types -- such as MX, NS, AAAA, A, etc. An administrator can add, delete, or modify RRs residing in a DNS zone file. The DNS service needs a restart in cases where the DNS version in use does not support the type of automatic updating which allows the DNS server to re-read the zone file and then apply these updates. This restart has a negative impact on DNS performance. During a DNS service restart process, the DNS servers (name servers) are unable to process DNS queries, which have been requested by other hosts on the Internet. To address this issue, the Dynamic DNS Update (DDNS) (RFC 3007) was introduced. DDNS solves the problem where other protocols, such as DHCP, which work in conjunction with DNS, need the capability of making real-time updates to RRs.

Even though DNS is a very critical Internet element, it only supports basic security mechanisms. Any new DNS functionality, such as DDNS, open new security issues with DNS as to how to prevent attackers from changing DNS records -- in other words, how to authenticate the host's desire to change RRs on DNS servers. The number of unique IPv6 addresses is 2^{128-32} times greater for IPv6 than that for IPv4. In order to organize this large address space, two different mechanisms have been proposed -- Dynamic Host Configuration Protocol (DHCPv6) (RFC 3315) and Neighbor Discovery Protocol (NDP) (RFC 4861). These two mechanisms are collectively referred to as IPv6 Autoconfiguration. Unfortunately, security in the DNS update process, is still the main issue with these two mechanisms. For example, when using DHCPv6, no options have been added to the DHCPv6 messages to allow for host authentication of the DNS server. The changeable nature of IPv6 addresses causes another problem for these mechanisms. Because of privacy reasons, and in order to prevent attackers from tracking a node in IPv6 networks, the IPv6 addresses are valid only for a short period of time, which is determined by network policy. Moreover, in the NDP addressing mechanism, no control is offered over which nodes can join the IPv6 network. These unmanageable and temporary addresses make it difficult to authenticate a host to a DNS server during the DNS record update process.

To address these issues we propose the use of an extension to SEcure Neighbor Discovery (SEND). We would add the DNS update message as an option to the Neighbor Advertisement (NA) messages. This would give nodes the ability to update their DNS records at the same time that they are setting their IP address and, thusly, announcing it to the other nodes on the network. The remaining sections of this paper are organized as follows:

Section II -- briefly explains the DNS Update, its vulnerabilities and current security mechanisms used in updating DNS. Section III -- introduces IPv6, autoconfiguration mechanisms, threats, and SEND. Section IV -- contains our proposed approach. Section V -- describes our evaluations and analysis of threats. Section VI -- summarizes this paper.

II. DNS UPDATE

The DNS mapping data is stored in RRs consisting of different types, such that, each type gives more information about a domain name or a host's physical IP address. An example for the need to update RRs occurs when a user changes his hosting service -- the RRs related to his domain should be updated in order to point the domain to the correct host. Another example occurs when a new node joins a new

978-0-7695-4952-1/13 $26.00 © 2013 IEEE
DOI 10.1109/WAINA.2013.117

network. The hostname and IP address should be added to the DNS server so that other hosts will be able to find this host by using its hostname in that network. In a local network, many applications work using hostnames instead of IP addresses for their communication with other nodes in that network.

DNS update is the process of adding or updating DNS RRs. In this process one or several RRs can be updated. The old mechanism used to process updates was a manual update process. This manual process negatively impacted the performance of DNS servers because of the need for human intervention. Human intervention also opened the door to increased DNS attacks due to human error. When DNS servers needed a restart after an update, they were unable to process DNS queries.

The Dynamic DNS update (DDNS) was introduced to address these issues. DDNS uses a basic protection mechanism to prevent other nodes from unauthorized updates. This is done by checking whether or not the source IP address is the same as that on the list of authorized updaters. The flaw here is that attackers can spoof this IP address and update DNS RRs so as to redirect all traffic to their desired hosts rather than the intended ones. The attackers can also execute other attacks -- phishing attacks, infection of other computers, Distributed Denial of Service (DDoS) attacks, etc. They can also redirect traffic to the victim's host which will inundate that service with messages and render its service unavailable (DoS).

To address these problems some security mechanisms and protocols were introduced -- Transaction SIGnature (TSIG) and DNS Security Extension (DNSSEC).

A. Transaction SIGnature (TSIG)

TSIG (RFC 2845) is a protocol that can be used to secure a Dynamic Update. It can also be used to assure the slave name server that a zone transfer is from the original master name server and that it has not been spoofed by hackers. It does this by verifying the signature using a cryptographic key shared between itself and the receiver. In this mechanism, the keys are manually exchanged between a host and its DNS server and must be kept in a secure place. The human intervention needed for this mechanism to work makes it difficult to use in IPv6 networks because of the nature of NDP in IPv6. NDP is added to IPv6 to simplify the management of the large IPv6 address space. In this case, all processing is done, automatically, when the node first joins the network.

B. DNS Security Extension (DNSSEC)

The DNSSEC (RFC 4033) was introduced by the Internet Engineering Task Force (IETF) as an extension to the DNS for use in the validation process of DNS query operations. The DNS server digitally signs data so that the other host can be assured that the DNS message is from the intended DNS server and that it has not been spoofed by a hacker.

RFC 5011 was introduced to allow a DNS resolver to update any DNS servers that reside in a trust chain of DNSSEC (Trust Anchors). Like the TSIG mechanism, this mechanism also requires some human intervention and this thus makes it difficult to use in IPv6 networks.

III. IPv6 ADDRESSING MECHANISMS

IPv6 was introduced to resolve the lack of addresses that exists in older versions of this protocol, i.e., IPv4. IPv6 supports 2^{128} unique IP addresses. The address scheme is in hexadecimal format (fe08:1a63:2001:50e9::). This format makes it virtually impossible to memorize IPv6 addresses. Also, administrators do not want to have to manually set these IP addresses for the hosts on their network and thus look for other mechanisms to manage them. To organize this large address space IPv6 autoconfiguration mechanisms were introduced -- Dynamic Host Configuration Protocol v6 (DHCPv6) and Neighbor Discovery Protocol (NDP).

A. Dynamic Host Configuration Protocol v6 (DHCPv6)

DHCPv6 (RFC 3315) allows a DHCP server to automatically assign an IP address to a host from a defined range of IP addresses configured for that network. This protocol works in conjunction with the DNS to dynamically update RRs on behalf of its clients on the DNS server.

When a client needs to update its RRs (AAAA and PTR) on the DNS, it makes this request via the DHCP server. It adds the Fully Qualified Domain Name (FQDN) (RFC 4702) to its update request along with instructions as to how to process this update on the DNS.

This mechanism requires some human intervention to configure and to administer the DHCP servers. For example, if a client wants to add a new option to the message sent to the DHCP server, the server will be unable to handle the new option in that message unless that option is manually defined by the administrator of the DHCP servers.

B. Neighbor Discovery Protocol (NDP)

Neighbor Discovery (ND) (RFC 4861) allows hosts to discover their neighboring routers and hosts and presents a means for obtaining router information from them. It also allows all nodes on the network to check the reachability of other neighboring hosts and routers. ND and SLAAC (RFC 4862), together, constitute NDP. NDP is a key feature of IPv6 suites. It uses five ICMPv6 messages -- Router Advertisement (RA), Router Solicitation (RS), Neighbor Advertisement (NA), Neighbor Solicitation (NS) and Redirect message. A NDP-enabled node can configure its IP address automatically as soon as it plugs into a new network. This newly joined node first generates its Interface Identifier (IID), which is represented by the rightmost 64 bits of the 128 bit IPv6 address. It concatenates the IID with the local link layer prefix that starts with fe08, and sets this on its network adapter. It then sends a RS message to all neighboring routers requesting router information. Routers respond to this message with a RA message containing routing information and subnet prefixes (64 bits). The subnet prefix is the leftmost 64 bits of the 128 bit IPv6 address. This node then sets its global IP address (as a temporary address) with the subnet prefix obtained from the RA message and sends a NS message to all nodes on that network to prevent any possibility of collisions with its IP address (process Duplicate Address Detection (DAD)). If it does not receive any NA messages after a certain period of time, (the standard is about 1 second) from any nodes claiming to have its IP address, then it changes the status of this IP

address to permanent and starts using it. Otherwise it generates a new IID and repeats the process.

1) NDP Threats

NDP works in a manner similar to those of Address Resolution Protocol (ARP), router discovery, and Internet Control Message Protocol (ICMP) used in IPv4 networks. It is thus vulnerable to the same kinds of threats as are those protocols. RFC 3756 gives a list of the possible types of attacks that can be used against NDP. Some of these attacks are:

a) DoS attacks against DAD: An attacker prevents a node from configuring its IP address by responding to the nodes NS message with a new NA message where the attacker claims that IP address. This prevents new nodes from obtaining IP addresses and thus from joining the network.

b) Address stealing: An attacker can easily steal an address and start using it since there is no mechanism in place in NDP to check for the address ownership of a node. . As a result, neither endpoint is certain of the true identity of the entity he is communicating with.

c) Bogus router: An attacker can advertise fake RA messages and redirect all traffic to his desired host.

d) Replay Attack: A malicious node can replay the NDP messages, whenever he wishes, by sniffing these messages and then resending them.

2) SEcure Neighbor Discovery (SEND)

In order to protect NDP enabled nodes against the types of attack stated in section 1, SEND (RFC 3971) was introduced. SEND provides protection by adding four options to the NDP message. These options are timestamp, nonce, Cryptographically Generated Addresses (CGA) (RFC 3972) and signature.

a) timestamp: This option is used to prevent replay attacks. It incorporates the send time for the current message into the message.

b) nonce: This option uses a random number that represents a session ID between two communicating nodes. This ID is variable and depends on the sender's selected random number, but it should never be less than 6 bytes.

c) CGA: This is an important option in the SEND message. It verifies the address ownership of the sender by finding a relationship between the sender's IP address and his public key [1, 2]. It can do this without a dependency on Public-Key Infrastructure (PKI). A SEND enabled node uses one-way hashing algorithms to populate the IID section of its IP address [3]. The following steps are executed by a node to generate the CGA:

1. It first generates a random number which is called a modifier
2. It concatenates this modifier with other parameters such as a zero value prefix (64 bits), a zero value collision count (8 bits) and the RSA public key(variable length)
3. It executes the Secure Hash Algorithm (SHA1) on the output from step 2 and uses the 112 bits from the digest and calls it Hash2
4. It compares the leftmost 16×Sec bits of Hash2 to zero. If the condition is not met, it increments the modifier by one and repeats steps 2 through 4. If the condition is met it proceeds to the next step
5. It concatenates the modifier with the prefix, the collision count, and the public key. It then executes another SHA1 on that output and calls it Hash1. It takes the 64 bits of Hash1 and sets the first 3 left-most bits to the sec value. It also sets bits 7 and 8 (known as u and g bits) and calls the result the Interface ID (IID)
6. It concatenates the subnet prefix and the IID and executes Duplicate Address Detection (DAD) in order to prevent address collision on the network. It sends all CGA parameters (modifier, subnet prefix, collision count, public key) with the messages so that other nodes can verify the address ownership

d) RSA signature: This is the last option in SEND. The data following the ICMPv6 checksum field is signed and the resulting signature is saved in the RSA signature field. The sender digitally signs this data by using its own private key. The receiver then can verify this data by using the public key of the sender. The attacker cannot replay or spoof the content of NDP messages because each message will have a different signature (nonce and timestamp included in the signature). The attacker thus does not have the private key of the sender which he would need in order to accomplish his attacks.

IV. IPv6 AND DNS AUTHENTICATION CHALLENGES

The main reason for the NDP proposal is to ease the management of the large address space in IPv6 networks and reduce the need for human intervention in address configuration. The end result of this eliminates the need to memorize the IPv6 hexadecimal addresses. A node might join an IPv6 network, and have its IP address automatically configured by use of the NDP mechanism, which needs no further intervention by the administrators of those networks.

As stated earlier, this new addressing mechanism, NDP, has an issue with how to authenticate a DNS server during the DNS Update process without, or with minimal, human intervention, while remaining within the goals of this protocol. Moreover, privacy is an important issue in IPv6, when nodes on the network must frequently change their IP address in order to prevent being tracked by attackers. This makes it difficult to authenticate who the update requestor of the DNS RRs is, based solely on the source IP address. Other security mechanisms, such as TSIG or DNSSEC, need manual key exchange or signature generation between the DNS server and a host before a secure authentication can be started. In IPv6 networks, it becomes harder to apply these authentication mechanisms. Although in IPv6 the manual update process is a major concern, in IPv4 it is an acceptable procedure for the following reasons:

- Using Active Directory (AD) to simplify the authentication process

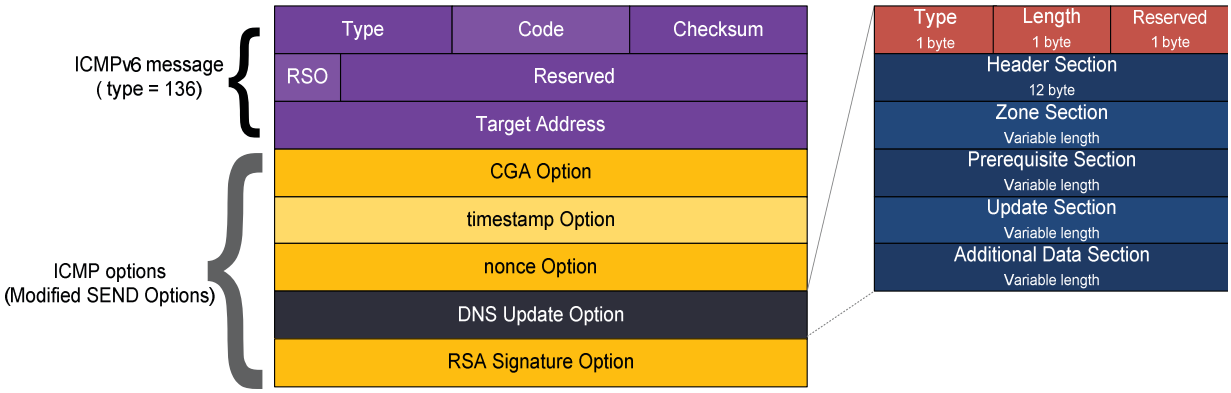

Fig. 1. Modified NA message with DNS Update option

- Advantage: nodes are already authenticated so that they can update their DNS records
- Disadvantages: the administrator manually adds the new node to this network.

- The addressing mechanism in IPv4 is not a completely automatic process -- it is either totally manual or requires network administrator intervention for DHCPv4 server configuration. These administrators can thus exchange the keys required for TSIG or other current DNS update security mechanisms between the DNS server and the DNS update requestor.

V. Proposed Approach

When a new SEND enabled node joins a new network it exchanges NDP messages, which contain SEND options, with a router or other nodes in this network for the purpose of configuring its IP address. These messages are Certificate Path Solicitation (CPS), Certificate Path Advertisement (CPA), RA, RS, NS and NA messages. Among these messages, the NA message may contain important information for the DNS server. This information could be an advertisement for the node's new IP address, the IP address that is required for the addition to DNS RRs, along with the hostname of this new node. A NA message has three flags which are represented by R, S, and O (RFC 4861). The R flag is set to indicate that the sender of this message is a router. The O flag is set to indicate that neighboring nodes should override their cache data about this node (the existing node in that network generated a new IP address). The S flag is a solicitation flag. When set, it indicates that the node wants to respond to the NS message and lay claim to the IP address that was sent by the sender of the message. When a node wants to advertise its IP address, it does not set the S flag. Our proposed approach is to modify the NA advertisement message (S=0) so that it will include the DNS update message. This will simplify the DNS update mechanism on local networks and will utilize a secure authentication mechanism, i.e., the SEND verification process.

Fig. 1 shows the NA message format with modified SEND options. As is illustrated in the figure, the DNS update is a new option for this message. It is also included in the RSA signature so that the DNS server will be sure of the integrity of this data. It also assures the DNS server that this node, with this IP address, is the actual owner of this hostname. The DNS Update option also includes the checksum calculation for NA messages.

The proposed approach can be applied in local link networks where the DNS server and other nodes that want to update their RRs are in the same network. This approach can also be used where the DNS server is outside of the local link network. This can be done by using a Controller Node (CN), which will intercept and verify all NA messages and generate the DNS Update message using the DNS Update Option of successfully verified NA messages. The CN will then forward this DNS Update message to DNS servers to apply the RR records' changes.

A. The format of DNS Update Option

Like all ICMP options, a DNS Update option is comprised of the following fields:

- Type: It has a numeric value -- like nonce or other ICMPv6 options. For the DNS Update we propose to set it to 15. This number is taken as the next sequential number after the last type of the ICMPv6 option. (Section 5.3.2 RFC 3971) (1 byte)
- Length: It is the length of the total data field in the DNS Update option (1 byte)
- Reserved: This byte is reserved for future use. The sender should set it to zero and the receiver should ignore it.
- Header Section: Contains the control information (RFC 2136)
- Zone Section: Identifies the zones that this update should be applied to. (Section 4.1.2 RFC 1035)
- Prerequisite Section: the RR records that must be in the DNS database.
- Update Section: the RR records that must be modified or added.
- Additional Data Section: The data that is not a part of the DNS update, but is necessary to process this update.

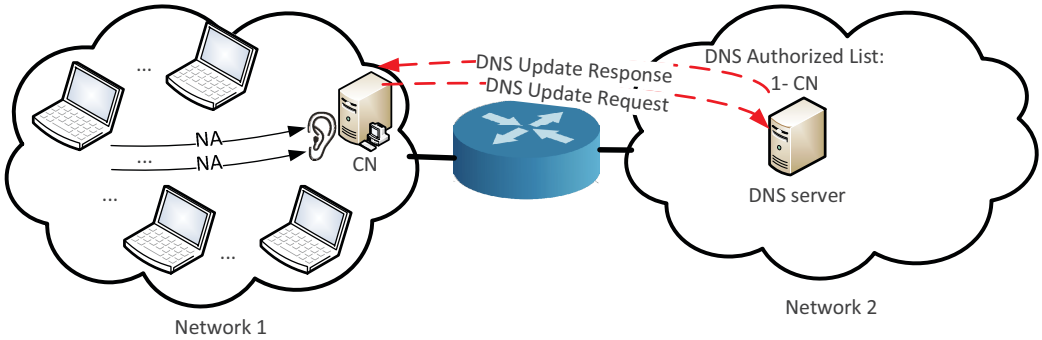

Fig. 2. Modified NA message with DNS server outside the local link

B. Process of the DNS Update Option

There are two different scenarios in play here -- one where all the hosts are in the same local link with the DNS server and one where the DNS server is located outside the bounds of local link.

1) DNS server within the local link

When a NA advertisement message is received by a SEND enabled DNS server, it starts the verification process to ensure that the one who sent the message is really the owner of that IP address. The verification process is made up of the following steps:

a) Executes CGA verification process which is comprised of the following steps:

1. Checking the subnet prefix: The subnet prefix, which is obtained from the source IP address, is checked against the one contained in the CGA parameters. If it is the same, then process step 2. Otherwise, this will be considered an attack and the message will be silently discarded.
2. Comparing Hash1 to Interface ID: Secure Hashing Algorithm (SHA1) is executed on the CGA parameters obtained from the packet. The leftmost 64 bits of this digest (Hash1) are compared to the Interface ID (IID). Any difference in the first leftmost three bits of the IID (sec value) and the u and the g bits (see fig 1.) are ignored. If both are the same, go to step 3. Otherwise, this will be considered an attack and the message will be silently discarded.
3. Checking Hash2 with CGA parameters: the DNS server will set the subnet prefix and the collision count obtained from the CGA parameters to zero and will execute SHA1 on the result. It will then compare the leftmost 112 bits of the digest (Hash2) to zero. If they are the same, the verification process will have been successful. Otherwise, this will be considered an attack and the message will be silently discarded.

b) Executes the SEND verification process: The DNS server obtains the public key from the CGA parameters in order to verify the RSA signature. If the verification process is successful the neighbor cache will be updated. Otherwise, this will be considered an attack and the message will be silently discarded. After successful verification, the public key and IP address of the node should be stored in a file if one does not already exist with matching data. If the data matches, i.e., the node generated a new IP address and already has a RR in the DNS server, then the DNS server should store the old IP address in a temporary place in memory and replace the new IP address with the old one in file.

c) Checking the RR records in the DNS Update option: The IP address of the sender is checked against the one contained in the RR records to ensure that the sender only wants to update his own RR record and not modify other records that he is not authorized to update. If the IP address is not the same, then the DNS server will check the IP address that was stored in temporary storage. If it finds a match, it updates the IP address with that new IP address that was stored in a file in section b. Then it will process this update message. Otherwise it will check the access list of the updater for a match. If it finds a match there, then it will allow this node to update other records as well. This is applicable when a DNS master wants to update several RR records on a DNS slave. If the IP address is not found on the authorized list, the update will not be processed. In other words, DNS servers only accept update requests from the owner of the RR record or from someone who authorized the update of others' RR records. The SEND verification thus provides this DNS server with the authentication needed by the updaters. If the node is a new node and wants to update just his own hostname, after successful IP address verification, his public key and IP address is stored in a file so that if this node changes its IP address and wants to update its own RRs, it can do it

After a successful verification process, the DNS Update option is fetched from the NA message. The SEND process then calls the DNS Update component to process the updated RR records in that message. This component then updates the zone file with the new RR record values.

2) DNS server outside the local link

When the DNS server is outside the local link network, newly added nodes will send a NA message to advertise their new IP address to other nodes on that network. A Controller Node (CN), which is listening to all NA message traffic on the local network, will intercept and verify the new node's NA message. The CN's verification process will be the same as outlined in section B.1, for the DNS server (DNS server within the local link). If it proves valid, a DNS update request

message will be composed and sent by the CN to the DNS server. CN uses the TSIG algorithm in order to authenticate the DNS server in a secure manner and also to establish a secure channel with DNS server. CN is listed as an authorized updater in the DNS server. The DNS server responds to the CN with a DNS update response message (see fig 2). The advantage of this approach is the transparency that is offered to each newly joined host on this network. This node, therefore, is not aware of the existence of the CN in this network. Another advantage is reduction in the administration tasks. The network administrator needs only set the DNS server and CN once to use a secure DNS Update (TSIG). There is no administration involvement needed to process DNS Updates for the other hosts in this network. The hosts in this network need only be configured to support our modified SEND solution.

VI. EVALUATION

A. Practical view of our approach

Because DNS is an application layer protocol, that overlays its data with UDP packets, or sometimes TCP, it allows the servers that support a DNS service to listen to a different port than that in the ICMPv6 (NA message). It is not feasible to change all current DNS applications to enable them to process the ICMPv6 messages (NA messages). Therefore, to benefit from our approach, we propose to modify the current SEND implementation. Our approach will be applied within the local networks, where the nodes can send and receive ICMPv6 messages without being rejected by firewalls. In contrast to NDP, SEND is a feature that is not natively support by Operating System (OS) manufacturers. There are some implementations for Linux, free BSD, and Windows [4] that have been implemented by external groups. These implementations are more open source and, as such, more easily lend themselves to the addition of a component that would allow them to fetch the DNS Update option and apply those updates directly to the DNS zone files. The SEND process was given, at its inception, all the administrative privileges necessary to access network adaptors and to perform all the necessary SEND functions. It is for this reason that no further administrative privileges are needed for SEND to process zone files as it already has the administrative privilege (root) for dropping a packet or accessing a network adaptor card.

Our approach will also decrease the DNS traffic on the local network. In order to securely update DNS RR records, a secure session is established, where a minimum of four messages are exchanged between a host and a DNS server (especially in DNSSEC). Equation 1 shows the percentage of traffic reduction in a case where the DNS server is in the same network as other hosts. In this equation, D_{min} is the minimum number of messages that are needed to be exchanged between a host and the DNS server in order to process a DNS update (IPv6). N is the number of nodes in the IPv6 network. P_N is the probability of a node generating DNS traffic. T is the total amount of DNS (IPv4 + IPv6) traffic.

$$\propto_t \cong \frac{(D_{min} \times N) \times P_N}{T} \times 100 \qquad (1)$$

For example, if the number of nodes on a network is 10, the number of messages exchanged between the DNS server and each host per second is 5, then the probability of each node sending this DNS update message is 0.5 and the total DNS traffic on that network is 100 packets per second. The percentage of traffic reduction per second in this example would be 25%.

It is also possible to decrease the number of messages exchanged between the DNS server and the CN. We propose to wait for some seconds (this waiting time depends on the network policy and the size of the network which can vary from 1 second to 3 minute) before sending the DNS Update to the DNS server. During this time, the CN might receive more than one NA message but needs only generate one DNS Update message on behalf of several nodes.

Another use for our approach occurs when a DNS slave wants to ask for a zone transfer from the DNS master (primary). In order to start communicating, and in order to find the other nodes' IP address, the DNS slave server asks for the IP address of the DNS master. It can thus verify the address ownership of the master server and use the SEND verification process as a tool to authenticate the DNS master. When the nodes use SEND to configure their IP addresses, this resolves the authentication issue between nodes with the DNS server.

B. Experimental Results

To test our approach, our proof of concept, we installed powerDNS on a Virtual Machine (VM) running a Centos 6.3 (Linux kernel 2.6.x) operating system. We used this setup as our DNS server. We used MySQL as the powerDNS database and configured it using the local IP address of 127.0.0.1. This was configured to accept updates from our CN node. The CN node used Centos 6.3 as its Operating System (OS) with TSIG configured as a secure DNS Update. On the two other VMs (pc1.dnstest.com and pc2.dnstest.com) we installed windows 7. We modified the Windows SEND (WinSEND) implementation to include the DNS Update option [4] which we then installed in pc1.dnstest.com. We used pc2.dnstest.com as a router emulator, a program that has all the functionality of a router -- sending RA messages or CPA messages, and verifying the CGA and the signature. We carried out several experiments by forcing pc1.dnstest.com to regenerate its IP address. After each IP generation, and successful Duplicate Address Detection (DAD), pc1.dnstest.com sent an unsolicited NA message. With the DNS server in the same local link as pc1.dnstest.com, we had NDprotector [5] call a small program written in c++ which then processed the DNS options of the NA message and added or modified the host's resource records (AAAA) to/in the MySQL database. Figure 3 depicts the powerDNS query result for pc1.dnstest.com when its IP address, 2001:abc:def:1234:2b23:89f8:e6f3:8c80, has then added to the AAAA records of MySQL database.

In a second scenario, we used the PowerDNS in another local link where we set the IP address for the powerDNS server to 10.0.1.15. In this scenario the CN node listened to the NA messages and, after receiving the NA message from pc1.dnstest.com, and a successful verification process by NDprotector, NDprotector then called our small program to process the DNS Update option. For all tests the security level for the CGA algorithm was done using a RSA key size of 1024 bits and a sec value of 1. The sec value is a number between 0

and 7 that is used to dictate the strength of the CGA algorithm against brute force attacks. Our proof of concept demonstrated how easy it was to use our approach in preventing spoofing attacks.

During the proof of concept, we maintained the host, wishing to add or update his resource records, public keys on the PowerDNS server. In this case, when the verification process succeeds a small program is used to check whether the IP address and the hostname are already present in the MySQL database. If they are, then it checks to see if the public key is the same as that in the clientkeys table along with the associated name (pc1.dnstest.com). If it is, then it updates the IP address associated with AAAA with this new IP address. Otherwise, it considers this an attack and does no further processing. Figure 4 shows the format of the clientkeys table. It is also possible to save this data in a separate file, but since we used MySQL, it was easier to use another table to save this data.

We did notice that when our approach was used the DNS Update process took less time than when the approach was not used. This was probably due to the fact that, with our approach, no separate messages are used for the DNS update.

C. Analyzing Threats

1) *IP spoofing and Message spoofing*: As shown by our experiments, our approach yielded benefits when using CGA and RSA signature SEND options. This process prevented modification to the source IP address and to the content of the DNS Update. This is because the DNS Update options, as well as other options, are included in the signature. Any verification would fail if any of that data was modified.

2) *Configuration attacks:* The use of our approach eliminates the need for a manual update to the DNS configuration file. As a result, the vulnerably due to human error is eliminated.

3) *DDoS attacks:* One type of DDoS attack is called the reflector. The attacker uses the victim's IP address as a source for DNS queries and then floods the DNS server with these queries. The DNS server then inundates the victim with responses to these queries which keeps the victim busy processing these packets. Our approach will prevent this attack, too, because it prevents IP spoofing by the use of the CGA algorithm.

4) *Zone transfer attacks:* Zone transfer is designed to allow the slave DNS server to receive a copy of the primary DNS server's DNS data. The slave DNS server can then be used as a backup to the master server. An attacker can obtain a version of the zone transfer and modify it, and then try to transfer it to the slave DNS server by pretending to be the master server. Since the signature verification process will fail, the slave name server will be unable to authenticate the attacker, thus preventing updates from the attacker's node.

```
[...  @dnstest ~]$ host pc1.dnstest.com 127.0.0.1
Using domain server:
Name: 127.0.0.1
Address: 127.0.0.1#53
Aliases:

pc1.dnstest.com has IPv6 address 2001:abc:def:1234:2b23:89f8:e6f3:8c80
```

Fig. 3. Query Output for pc1.dnstest.com using PowerDNS

```
mysql> show columns from clientkeys;
+-----------+--------------+------+-----+---------+----------------+
| Field     | Type         | Null | Key | Default | Extra          |
+-----------+--------------+------+-----+---------+----------------+
| id        | int(11)      | NO   | PRI | NULL    | auto_increment |
| domain_id | int(11)      | YES  |     | NULL    |                |
| name      | varchar(255) | YES  |     | NULL    |                |
| pubkeys   | blob         | YES  |     | NULL    |                |
+-----------+--------------+------+-----+---------+----------------+
```

Fig. 4. Tables in MySQL database for keeping host's public keys

VII. CONCLUSION

DNS data maintained on DNS servers is kept current by use of an important DNS function called DNS Update. An important mechanism of the DNS Update is DDNS, which gives nodes, on the network, the ability to make automatic updates. Unfortunately this mechanism supports only a basic protection technique which is based on the source IP address. To make this process more secure, especially in IPv6 SLAAC where there is no authentication mechanism, and because the current mechanisms of IPv4 are difficult to use in IPv6 because of the need for human intervention, we propose an extension to the SEND feature which will allow for the update of the DNS RR records at the same time the node configures its IP address. The DNS server will thus benefit from the SEND's verification process as a tool for authenticating the node whose set of DNS RR records it wants to update. This will also result in a decrease in the amount of DNS traffic on the network because it will eliminate the need for the message exchange that is necessary in establishing a secure channel before the DNS update can be processed.

REFERENCES

[1] T. Aura, "Cryptographically Generated Addresses (CGA)," Lecture Notes in Computer Science, Springer, vol. 2851/2003, pp. 29–43, 2003.

[2] G. Montenegro, C. Castelluccia, "Statistically Unique and Cryptographically Verifiable (SUCV) Identifiers and Addresses," ISOC Symposium on Network and Distributed System Security (NDSS 2002), The Internet Society, 2002.
http://www.isoc.org/isoc/conferences/ndss/02/papers/monten.pdf.

[3] P. Nikander, "Denial-of-Service, Address Ownership, and Early Authentication in the IPv6 World," Lecture Notes in Computer Science, Springer, vol. 2467/2002, pp. 22-26, 2002.

[4] H. Rafiee, A. Alsa'deh, and Ch. Meinel, "WinSEND: Windows SEcure Neighbor Discovery", 4th International Conference on Security of Information and Networks (SIN 2011), 14-19 November 2011, Sydney, Australia. 2011.

[5] T. Cheneau, "NDprotector", http://amnesiak.org/NDprotector/.