DOMAIN NAME SECURITY EXTENSIONS

The aim of this paper is to provide information with regards to the current status of Domain Name System (DNS) and its evolution into Domain Name System Security Extensions (DNSSEC).

INTRODUCTION

This paper will introduce the concepts of DNS and DNSSEC to the average reader by providing an outline of both technologies, whilst also discussing their limitations. The DNS is the technology that makes the Internet usable to the every-day user. DNSSEC is an evolution of DNS but provides greater security in authenticating the originality of DNS records.

AN OVERVIEW OF DNS

The Domain Name System (DNS) is a tiered distributed name system for devices or services connected to the Internet or a private network. It will correlate an easy to remember hostname with an Internet Protocol (IP) address of the host E.g. http://www.replyltd.co.uk resolves to the IP address 91.218.224.46. A common analogy is to that of a phone book, translating human names for their equivalent phone number. A device is required to know the IP address of the server it is communicating with, to exchange data else a connection will not be established.

The process of IP resolution through DNS as is seen in the figure above is as follows:

- **Step 1** - a DNS client will make a request to their assigned DNS Server A to obtain the IP address of a URL.
- **Step 2** – DNS Server A would either respond with the correct IP or request the information from another DNS Server, which holds this information in this case Server B.
- **Step 3** - DNS Server B has the IP Address in its database and returns the IP Address to DNS Server A.
- **Step 4** – Server A will then cache the response and forward on the IP Address to the DNS Client.
- **Step 5** - The DNS client is then able to initiate a connection to the webservers www.replyltd.co.uk and continue normally.

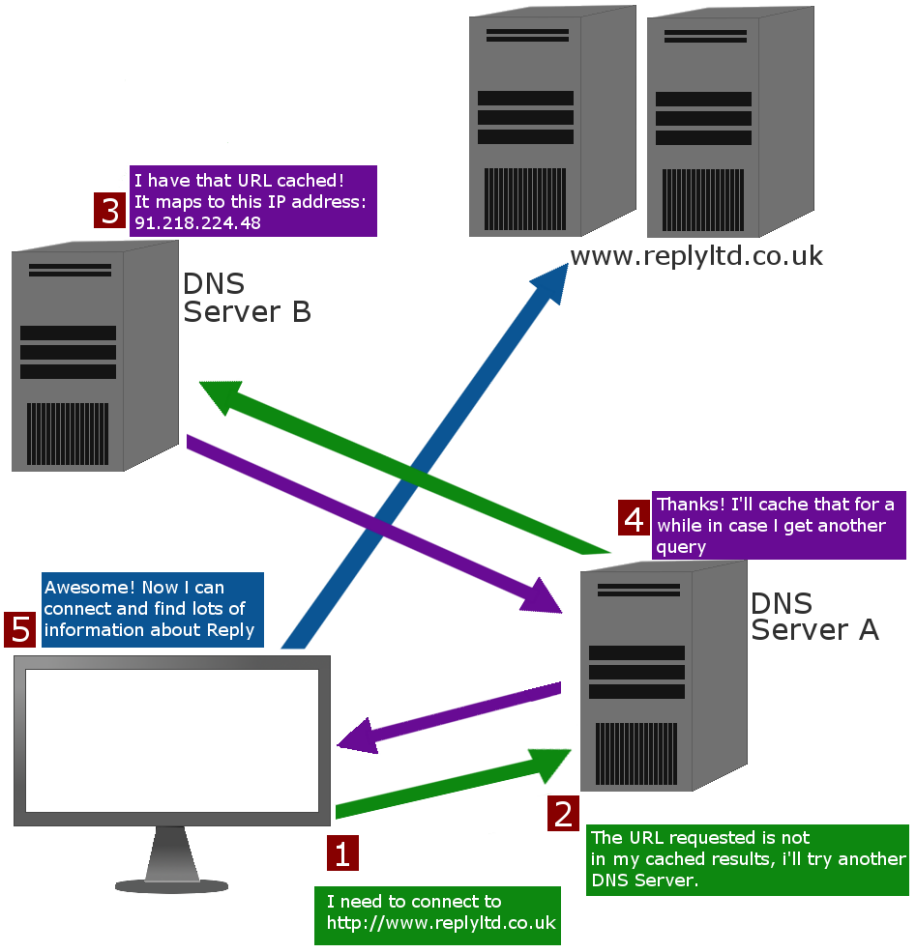

Figure 1 - How DNS Works

DNS functions on port 53 primarily using User Datagram Protocol (UDP) to facilitate the exchange of information between client and server. A DNS query consists of a single UDP request from the client to the server, which returns a single UDP response to the client's request.

DNS Servers also have the ability to communicate through Zone Transfers, this refers to the transfer of DNS records between DNS severs. A Zone is defined as section of the DNS records available on the master DNS server's database. A DNS server will query its master (it's more authoritative) DNS server to identify any changes in the records, only then when it has identified a difference it initiates a Zone Transfer with its master DNS server.

KNOWN DNS SECURITY ISSUES

An attacker can target either the protocol itself or attack a DNS server directly. There are different protocol based attacks that a malicious user could utilise to compromise the protocol. Examples of such will be presented in the following section, including the Client Misdirection and the Cache Poisoning. Server based attacks consist of DoS/DDoS (making communication to a DNS server impossible) and Vulnerability Manipulation.

CLIENT MISDIRECTION

The first step of client misdirection is the process of spoofing information provided to a client by intercepting communications, commonly referred to as a man-in-the-middle attack. The

definition of spoofing, when related to this DNS attack, is answering a request, which is intended for the client with a manipulated DNS record. As a result of this the client will be directed to the supplied website in the DNS response. When spoofing the response, an attacker will spoof the source-address field that is sent back to the client pretending to be the real DNS Server. The attacker will also be required to spoof the DNS ID and port number that are used to identify queries made by the client else the client will ignore the response. Gaining access to the DNS ID and port number is simple when the attacker is within the same Local Area Network (LAN) as the client. A tool that provides the functionality of client misdirection on the same LAN is Cain & Able (Available at http://www.oxid.it/cain.html). However when the attacker is not in the same network his only option is to guess the ID and port. This can only be done by testing all possible values or to randomly generate the values, which is difficult with around a billion combinations but possible with a prolonged attack and a simple script to generate the values (Example Script Available at http://cr.yp.to/djbdns/dns_random.html).

CACHE POISONING

Cache Poisoning is the act of tampering with the information stored in a DNS server's cache, essentially corrupting its database by mapping a domain name to a different and incorrect IP address. DNS servers have been vulnerable in the past but were fixed when Cache Poisoning incidents arose. Nowadays there are only two forms of cache poisoning that occur; first when a DNS server has been fully compromised by a hacker and had its data manipulated (Infosecurity Magazine, 2013). Secondly it can be caused by misconfiguration of DNS records; an example of this occurred in 2010 where popular website traffic was directed to servers in China instead of the correct server. This happened because of a misconfiguration of a DNS server, which fetched records from one of China's Root Zone servers (McMillan, 2010).

DOS AND DDOS

It is possible to take out a DNS server via DoS (Denial of Service) or DDoS (Distributed Denial of Service) and effectively remove a webserver from operating due to users not being able to resolve the IP address of the domain name. This can be achieved by flooding the DNS server with a large volume of traffic making it impossible for legitimate requests to be received. It is also possible to use various DNS servers in order to carry out a DDoS of an intended victim through a process of DNS Reflection DDoS (Piscitello, n.d.). This is where the source address of the DNS packet is spoofed to be the IP address of the victim client and sent to many DNS servers. The DNS servers then respond to the victim with large packets and flood the server so that legitimate communications cannot be made.

VULNERABILITY MANIPULATION

It is possible for a malicious user to find vulnerabilities within the software being used on the DNS server. Utilising these vulnerabilities, the malicious user is able perform various actions such as escalating their privileges to the DNS database and editing records through to causing errors in the software resulting in the software crashing. Any exploited vulnerabilities result to the data, held within the DNS server, being vulnerable to manipulation and should be seen as compromised (SAINT, 2014).

DNSSEC

HISTORY OF DNSSEC

In 1995, DNSSEC became a hot topic within IETF and took four years for RFC-2535 to be published in 1999. BIND (open source software) was the first software to be capable of

implementing RFC-2535. By 2001 it became apparent that implementing RFC-2535 in a large network was infeasible. This was due to a problem in syncing data between parent and child as DNS servers were often out of sync, which isn't a problem for DNS however with DNSSEC when out of sync it could create a self-created denial of service (Wikipedia, 2014). Therefore IETF decided to rewrite the protocol. DNSSEC was then split into three new RFCs 4033, 4034 and 4035 which were implemented first by Sweden; they were then followed by others shortly after. By 2012, 90 Top Level Domains had DNSSEC implemented and this number is still growing (Verisign, n.d.).

AIM OF DNSSEC

DNS is a critical part of the Internet as it allows users to navigate with ease, however security was not considered in the design of the DNS protocol and therefore DNSSEC was created. The purpose of DNSSEC is to securely authenticate records and create a secure network of DNS Servers across the Internet providing users with the correct data.

ROOT ZONES AND TOP LEVEL DOMAINS

The first step of the DNSSEC implementation was to secure the Root Zones and Top Level Domains (TLDs) (IETF, n.d.). This is to ensure that the origin of all DNS records are secure and can then filter down to other zones whilst being trusted to be accurate. TLDs are domains such as '.it' and '.uk'; TLDs are the most sensitive domains, it is imperative that these are secured to prevent large scale manipulation. The diagram below in Figure 2 illustrates the principle of TLDs being part of the Root Zone and second-level domains that are outside of the Root Zone below TLDs.

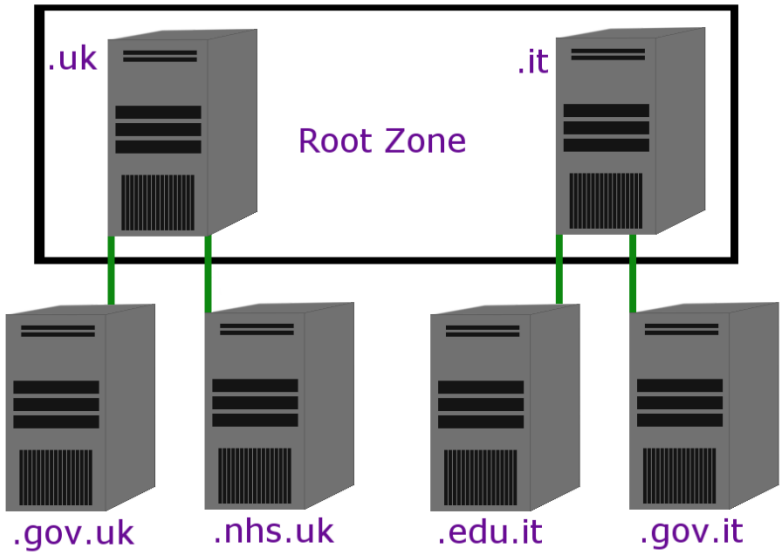

Figure 2 - Top Level Domains

PRINCIPLE OF SIGNING

DNSSEC uses public key cryptography to securely sign data transferred from Zone to Zone. This is a hierarchical process from Parent Zone to Child Zones for example, the Root Zone signs for sub-zones below it. A Parent Zone will publish its public key whilst keeping its private key safe; signing individual DNS records in that Zone to create a digital signature to distribute along with the DNS records. The public key will then be used to decrypt the messages so that the Child may ensure the data's integrity and origin.

Authoritative name servers are used within each Zone that distributes data to recursive

name servers, which are the DNS servers which deal with end user requests. When an end user requests access to a website it will use its own DNS records initially and if unsuccessful contact a recursive name server. The recursive server would request this record from the relevant authoritative name server, whilst at the same time request the Zone key. Using this key the recursive name server will verify the record that is held on the authoritative name server. Using the result of this verification to control what is sent to the end user; either the server will allow the record to be sent to the end user or in turn not provide the end user with the potentially malicious address. By communicating with DNSSEC and knowing that the data you receive is securely signed and holds its integrity, it helps to prevent man-in-the-middle attacks from being able to be launched.

Normally with DNS when a record does not exist within a DNS server, it will send a response with a blank answer. This in turn will cause an issue with signing a record if there is nothing to sign. To mitigate this issue, NSEC was created. NSEC creates a chain of records around the missing record to verify that a record exists. As an example if you have record A and C, but you request record B NSEC will send you NSEC result C which is signable making it possible to find if a record does not exist.

ISSUES WITH DNSSEC

The main issue that arises continuously with organisations implementing DNSSEC is not with the DNSSEC protocol itself but with surrounding technologies. As previously mentioned in this paper DNS uses UDP, however the use changes slightly when implementing DNSSEC. Typical DNS implementations had a max communications of 512 bytes long and when communications exceed this it was required that DNS would revert to TCP and use a fully-established TCP connection to communicate the data. EDNS (Extension Mechanisms for DNS) was developed to allow for a much larger DNS response sizes, most commonly 4096 bytes.

Implications arose from this where Firewalls had been set by default or recommended to limit UDP to 512 bytes therefore preventing DNSSEC from resolving IP addresses. EDNS also is hindered by another issue, Maximum Transmission Unit (MTU) that controls frame sizes. Typically MTU's are approximately 1500 bytes and therefore it is possible for responses larger than the MTU to be fragmented over multiple packets. This leads to another requirement of some firewalls that block UDP fragmentation by default and if this is the case DNSSEC will not be able to resolve IP addresses when it encounters this. Most Firewalls can be configured to prevent these issues by using different solutions to validate the integrity of packets and therefore allow fragments of DNS responses with a larger UDP packet size of 512 bytes. Firewalls can reassemble fragmented packets and then check the validity and deal with them effectively. This leaves the only remaining risk to the DNS server being DDoS attacks, which are apparent regardless of the UDP size or fragmenting (Wilson Lian, 2013).

DNSSEC ADOPTATION

According to studies carried out (Michaelson, n.d.), it was noted that less than 9% of 2,498,497 public DNS resolvers were performing DNSSEC validations in 2013. The top countries performing DNSSEC were Sweden, Slovenia, Luxembourg, Vietnam and Finland. It is clear therefore that other measures are required to take place to ensure that DNSSEC adoption is increased. Some believe that it is the role of Governments to mandate that all domain owners initiate DNSSEC functionality, whereas others perceive that it is the role of large domain name server providers to offer DNSSEC by default (Thia, 2011).

CONCLUSIONS

As highlighted in this paper, DNSSEC provides much greater levels of security than DNS. Nevertheless; due to the technical constraints that DNSSEC places onto a network approximately only 9% of DNS resolvers were performing DNSSEC validations in 2013. There are many security issues that are apparent in the DNS protocol, however many of these are common in every software such as DDoS and Vulnerability Manipulation. The vulnerabilities which are only apparent within the DNS protocol have been solved with the implementation of DNSSEC. It is clear that when designing new implementations it should be imperative that architects ensure that both DNS and DNSSEC are properly encompassed, only then can the DNS protocol start to be phased out.

BIBLIOGRAPHY

IETF, n.d. *Protocol Modifications for the DNS Security Extensions.* [Online]
Available at: http://www.ietf.org/rfc/rfc4035.txt
[Accessed 20 May 2014].
Infosecurity Magazine, 2013. *Syrian Electronic Army Steps Up a Gear – Re-Directs Major Websites to its Domain.* [Online]
Available at: http://www.infosecurity-magazine.com/view/34209/syrian-electronic-army-steps-up-a-gear-redirects-major-websites-to-its-domain
[Accessed 20 May 2014].
McMillan, R., 2010. *China's Great Firewall spreads overseas.* [Online]
Available at:
http://www.computerworld.com/s/article/9174132/China_s_Great_Firewall_spreads_overseas
[Accessed 20 May 2014].
Michaelson, G. H. a. G., n.d. *Meausring DNSSEC Use.* [Online]
Available at: http://www.iepg.org/2013-07-ietf87/2013-07-28-dnssec.pdf
[Accessed 20 May 2014].
Piscitello, D., n.d. *Anatomy of a DNS DDoS Amplification Attack.* [Online]
Available at: http://www.watchguard.com/infocenter/editorial/41649.asp
[Accessed 20 May 2014].
SAINT, 2014. *DNS Vulnerabilties.* [Online]
Available at: http://www.saintcorporation.com/cgi-bin/doc.pl?document=vulnerability/DNS_vulnerabilities
[Accessed 20 May 2014].
Thia, T., 2011. *Govt mandate aids DNSSEC uptake.* [Online]
Available at: http://www.zdnet.com/govt-mandate-aids-dnssec-uptake-2062301179/
[Accessed 20 May 2014].
Verisign, n.d. *Domain Name System Security Extension (DNSSEC).* [Online]
Available at: https://www.verisigninc.com/en_US/innovation/dnssec/index.xhtml
[Accessed 20 May 2014].
Wikipedia, 2014. *Domain Name System Security Extensions.* [Online]
Available at: http://en.wikipedia.org/wiki/Domain_Name_System_Security_Extensions
[Accessed 25 06 2014].

Wilson Lian, E. R. H. S. a. S. S., 2013. *Measuring the Practical Impact of DNSSEC Deployment.* [Online]
Available at: https://www.usenix.org/conference/usenixsecurity13/technical-sessions/paper/lian
[Accessed 20 May 2014].