DNS SECURITY EXTENSION (DNSSEC)

I assure on my honour that I have actively participated in these solutions, and that the solutions were developed independently from other groups.

811217-5073	Frans Eliasson	frel8817@student.uu.se
	Lu Hai Loc	Lolu7877@student.uu.se

1. Introduction:

DNS stands for domain name system whose main job is to translate domain names like www.uu.se to IP addresses. DNS system is structured in a hierarchical order where there're DNS roots and sub DNS systems. When one user enters www.uu.se into the browser, his computer will send out query to ask its DNS what the corresponding IP addresses is. And the DNS will search from its local cache or if it cannot find, it will make recurring query to higher DNS servers until the IP is found. Finally it will send back its response. Therefore, the DNS plays a very critical role in the Internet structure. However, DNS packets were not designed for any encryption or verification of authentic sources.

On the Internet, this association between IP and top-domain is ultimately decided by one of the 13 root DNSs controlled by ICANN in USA. These servers keep child DNSs updated, that keep other child DNSs updated etc.

2. DNS Attacks:

This also means that there are a lot of attempts to forge the DNS responses from DNS servers. Imagine what happens if instead of going to the authentic servers to receive information, you are redirected to an attacker's servers with full of malicious codes, viruses when you might unknowingly give out sensitive information.

There are many types of DNS attacks:

- *Monkey in the middle*: eavesdropping on requests and spoofing responses to beat the real response from the DNS servers. This can easily be done because the DNS query and response are contained in a single unsigned and unencrypted UDP packets.
- *Cache poisoning*: The attacker captures the responses and injects false data which could be like replacing the IP address by the IP address of the server that the attacker controls or simply saying "no such domain" exists. Due to the fact that the victim server will cache this false information for a while, the effect of the attack will affect all the users who make this DNS request.
  The attacker can easily provoke the victim to make a DNS request by simply embedding a 1x1 pixel web bug graphic in a peace of HTML mail. If the mail program of the victim tries to follow this link, he or she accidentally makes a DNS query for a name chosen by the attacker.
- *DoS*: DNS is vulnerable to DoS and even used as DoS amplifier because the DNS response packets usually significantly longer than the DNS query packets.

3. Solution – DNSSEC:

With those vulnerabilities, there comes DNSSEC.

The basic idea is by attaching digital signatures to the DNS responses, the DNS clients now can verify the authenticity, data integrity of the DNS responses, or even when the responses are "no such domain" exists, these responses can also be authenticated.

The DNS servers will have one public key and one private key. The private key is kept secret and used to sign the signature attached to each response. The public key will normally attached to the response for the DNS clients to verify.
Even if the attacker has captured and modified the DNS responses, the DNS client should be able to detect that the responses has been altered and not come from the authoritative sources.

How does DNSSEC work?

In order to achieve this, DNSSEC introduces 4 more main components:

- DNSKEY: This is the pair private and public key. The private key is guarded by the zone admin. The public key is published for verification.

- RRSIG: This is the signature itself.

- NSEC: This covers the case of "there's no such name" responses. Because the domain names in one zone are ordered in a canonical form, NSEC is inserted to cover the gaps.

- DS (The Delegation Signer): This is introduced to counter the case that if the attacker can intercept all your traffic, he is there from the beginning and signs response using his own private key and attaches his own public key to the response. Therefore, how can you know you are communicating with the correct server and how can you verify that the public key is really from the correct server?
  This can be fixed thanks to the chain of trust property. The public key of the DNS server will be signed by is parent DNS server, whose signature is signed by the grand parent DNS server, and so on. Up till on root DNS server that we choose to trust. And the brilliant idea is to the advantages DNS hierarchical delegation structure itself without introducing any other third party.

So, the whole idea is when requested, the DNSSEC add more the above additional information to the response so that the DNS client verify the authentication of the response. If the attacker captures the signed response and try to insert false information, the DNS client will be able to detect the altered response. When the DNS clients receive

the response, it will be able to verify the attached signature using the public key. Before that, it can also verify the authenticity of that public key through the chain of trust.

4. Difficulty of deployment DNSSEC:

- The most significant problem: The "no such domains" NSEC responses can be used by attackers to create the entire domains map of the zone, which can be commercially sensitive information. According to Albitz and Liu in "DNS and BIND" (4th edition), there's a big difference between letting random folks call your company's switchboard and ask for John Q. Cubicle's phone number [versus] sending them a copy of your corporate phone directory." (Zone walking problem)

- Performance problems:
  - The average size of the DNS response message increase, due to the attachment of digital signature. This creates overhead cost if exceeding the size of the UDP message.
  - The number of DNS transactions increase due to the requirement for performing additional chain of trust queries.
  - The resolution process is slowed down due to the additional time validating signed data
  - Heavy load on the server side for signing every response.
  - Small queries creating large responses, potential for DoS.

- Technical problems:
  - DNSSEC deployment on such a big scale as the Internet is very challenging. users typically only deploy a technology if they receive an immediate benefit, but if a minimal level of deployment is required before any users receive a benefit greater than their costs, it risks remaining undeployed.

- Political problem with DNSSEC deployed at root DNS servers: Because most of the root DNS servers are in US, other countries are concerned about U.S. control over the Internet, and may reject any centralized keying for this reason.

Sources:
http://www.circleid.com/posts/dnssec_deployment_and_dns_security_extensions/
http://en.wikipedia.org/wiki/DNSSEC
http://smakd.potaroo.net/ietf/idref/rfc3833/index.html