Tallinn University of Technology

Faculty of Information Technology

Institute of Computer Science

ITC70LT

Ennio Calderoni (132110ITCMM)

DNS SECURITY: ANALYSIS OF ALTERNATIVES AND AN ANDROID DNSSEC-AWARE BROWSER

Master's Thesis (30 ECTS)

Supervisor: Truls Tuxen Ringkjob

Tallinn 2015

Author’s declaration of originality

I hereby certify that I am the sole author of this thesis. All the used materials, references to the literature and the work of others have been referred to. This thesis has not been presented for examination anywhere else.

Author: Ennio Calderoni

Signature:………………

Date:……………………

Abstract

In today’s world, many services have moved to the Internet. We visit fewer physical shops, banks, and municipal offices; if we want to know what is happening around the world, we do not buy newspapers, but read the news online. Most companies have moved a significant part of their services online. Some countries rely strongly on the Internet; for example, in Estonia, it is possible to sign documents and to vote online using one’s ID card, and there are available a lot of other services like e-residence.[1]

To use all these services, users must trust an insecure protocol, The Domain Name System (DNS). This thesis discuss in detail the alternatives to the DNS protocol and how to simplify and make more effective its usage from smartphones, because according to the article “Smartphone now most popular way to browse internet – Ofcom report” [2] from theguardian.com, people access Internet services more from smartphones than PCs. The contribution this thesis gives to the security in the DNS is a Browser for Android that is able to validate the DNSSEC chain of a domain name, block rogue DNSSEC domain names and to block most commune DNS hijacking attacks.

Keywords

Android Security, DNS Hijacking, DNS, DNSCurve, DNSSEC, Namecoin, Network security

Table of abbreviations and terms

APP	Application
CA	Certificate Authority
DANE	DNS based Authentication of Named Entities
DDoS	Distributed Denial Of Service
DNS	Domain Name System
DNSKEY	DNS Public Key
DNSSEC	Domain Name System Security Extensions
DS	Delegation Signer
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
IAB	Internet Architecture Board
ICANN	Internet Corporation for Assigned Names and Numbers
ID	Identification
IETF	Internet Engineering Task Force
IP	Internet Protocol
ISP	Internet Service Provider
KSK	Key Signing Key
MITM	Man In The Middle
MTU	Maximum Transmission Unit
NSEC	Next Secure
PC	Personal Computer
RFC	Request For Comments
RRSIG	Resource Record Signature
SIP	Session Initiation Protocol
SMTP	Simple Mail Transfer Protocol
SSL	Secure Sockets Layer
TA	Trust Anchor
TCP	Transmission Control Protocol
TLD	Top Level Domain
TLS	Transport Layer Security
UDP	User Datagram Protocol
XMPP	Extensible Messaging and Presence Protocol

ZSK	Zone Signing key

Table of Contents

List of Figures and Tables

1. Introduction

1.1.Background

In today’s world, IT plays an extremely significant role in enterprise, and countries’ infrastructures. Companies are offering significant parts of their services online, through online banking, electronic commerce, and cloud computing. Several countries now rely on the Internet, such as Estonia, which offers e-voting and e-residence facilities [1].

This new way of life with many online services does make everyday life easier. In some cases, these services do help save money and time, and buy goods that are not available in other ways. But does it have any disadvantages? Services online can be attacked by hackers, activists, foreign governments, competitors, and, in extreme cases, terrorists or cyber-wars could paralyze an entire nation [1][5].

For years, people have understood the importance of online security and there are organizations like IETF, IAB, and ICANN that are making efforts to make the Internet a more secure place [2]. In spite of the fact that today, the Internet is more reliable than before, one of the most important and used protocols – the Domain Name System (DNS) – remains completely insecure and vulnerable to some of hackers’ most powerful attacks.

1.2.Problem Description

The DNS is a crucial protocol in the Internet; almost every operation relays on it, and therefore, securing DNS is necessary because so many services relay on this protocol.

This thesis will analyze the more promising protocols existing today which can help secure the DNS and deeply analyze their effectiveness in the real world. Furthermore, some Applications that use these protocols will be analyzed in order to discover how they can be used to protect other protocols.

2. Domain Name System (DNS)

In order to access a specific website, it is necessary to give a unique identifier for that page. This Identifier is called the Internet Protocol (IP) address, which is a numerical value assigned to every device inside a computer network. There actually exist two versions of this protocol: version four (IPv4)[6], that consists of four decimals, each between 0 and 255, separated by a dot Figure 1, and version six (IPv6)[8], which uses eight hexadecimal numbers of four numbers, each separated by a colon Figure 2. [6][8]

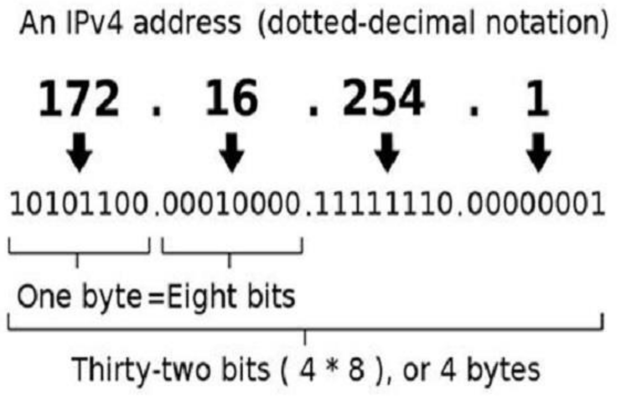

An IPv4 address (dotted-decimal notation)
172 . 16 . 254 . 1
10101100.00010000.11111110.00000001
One byte=Eight bits
Thirty-two bits (4 * 8), or 4 bytes

Figure 1: IPv4 Addresses [8]

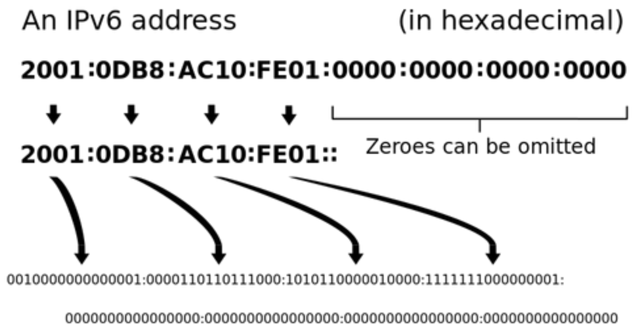


Figure 2: Ipv6 Addresses [9]

People tend to remember a text than IP addresses (and this is even more true in the case of IPv6); if we consider www.example.com and 93.184.216.34, the URL will be easier to remember then an IP address. [15]

The principle is the same as we tend to do with telephone numbers, associating names to telephone numbers in a rubric to know which number to dial to call one person; similarly, we need a database that associates IP numbers with host names and web pages. Initially, this process was done through the maintenance of a hosts.txt file, distributed to all machines on the Internet, but this was possible only in the beginning, because when users started to grow, this method could not be applied anymore, and to serve this purpose, the Domain Name Service (DNS) in the year 1983. [11][15]

2.1.How Does DNS Works?

The DNS plays an extremely important role, both in the Internet and in private networks. It does everything needed for the translation of IPs to make them more readable for humans (host names) and vice-versa; but how does it work?

It has a hierarchical structure that looks like an upside-down tree; in fact, the root of this tree is at the top and it is represented by a dot, while the leaves are at the bottom Figure3. Every zone is authoritative only for its data. Domain names are written in Fully Qualified Domain Name (FQDN) standard, which means that each domain name consists of more labels separated by dots. Every level of the tree has its name and delegates the authority to sub-domains; so the zero node, called root node (directly managed by ICANN) delegates its authority to the Top Level Domain (TLD) and TLD does the same with the Second Level Domain (SLD), and so on. When we write a fully qualified domain name (FQDN) like www.example.com, starting from the right to the left, there is the top-level domain name, the second-level domain name and the third-level domain name, and in addition, we could also have sub-domains. Figure 3 [11][15]

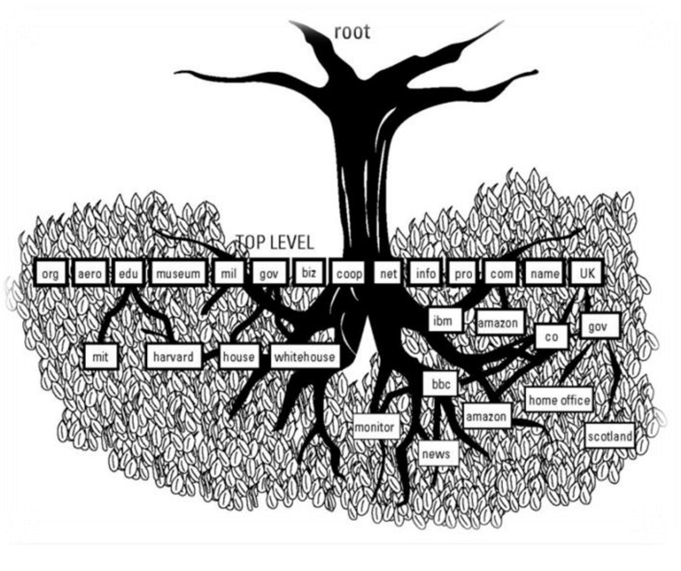


Figure 3:DNS structure [10]

The Domain Name System is a distributed database system, which means that the information between DNS servers is shared by name servers; as a result, a single server does not contain the complete database. Domains have authoritative DNS servers to publish information about themselves and for any server at a lower level. [11][15]

There are three kinds of servers: Primary, Secondary and Caching. Primary Servers are considered to be authoritative for the zone where they are located; this means that the domain’s configuration files reside inside it.[15] Secondary Servers are basically backups and load distributors for the Primary Servers, thus also acting as authoritative servers. Despite the fact that some think that secondary servers should not be authoritative, there is no way to know if the answer came from Primary or Secondary servers[13]. In case of Caching servers, which operate only with name servers, every query has to pass through the entire tree, and this is not efficient because there would be a huge traffic for DNS queries across the Internet. To address this this problem, DNS supports DNS cache servers for storing DNS query results for a certain period of time. Internet service Providers (ISP) usually provide caching servers for their costumers and many private and home networking routers also have DNS caches. DNS servers also implement negative caching, and so store the failed query so as to not repeat it. Negative caching stores the result of the queries in [15]:

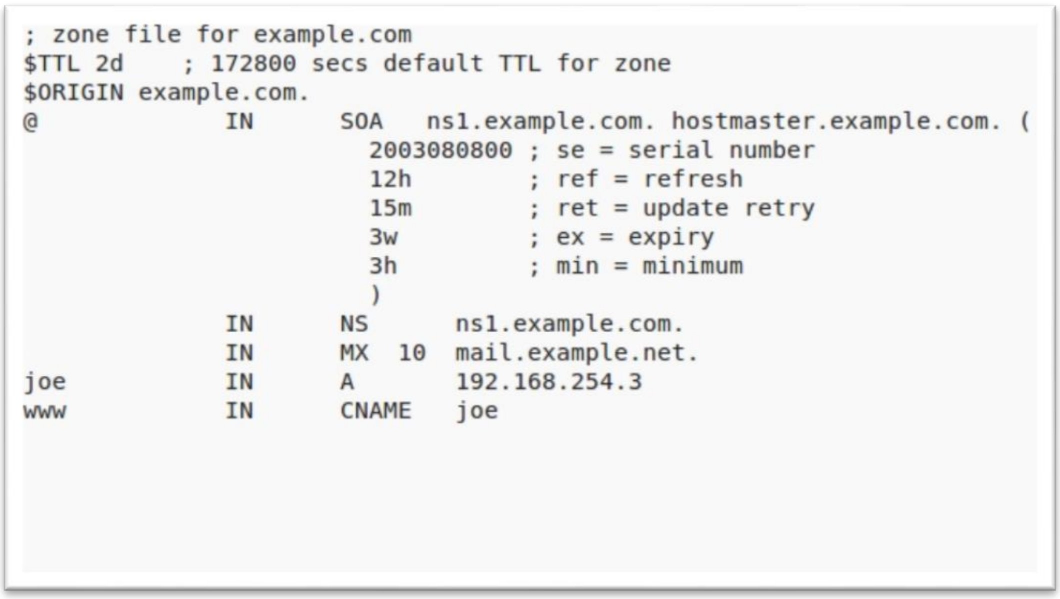
- No host or domain matches the name queried.
- The type of data requested does not exist for this host.
- The server is not responding.
- The server is unreachable because of network problems.

An authoritative name server gives answers on the basis of what has been configured directly on it, and this could be a master server or a slave one. It is a master server when it stores master copies of all its zone records; on the other hand, a slave server uses automatic updating of the DNS protocol to maintain an exact copy of the master records. Authoritative servers put a flag to indicate that it is the authoritative answer; it adds the AA (Authoritative Answer) bit in responding [15].

The responsibility of the registration of domain names has been delegated from ICANN and ccTLDs to registrars that provide services to the public and are responsible to provide registries with information of the registered domain name and the IP address of the authoritative name servers for that domain. If a domain name is registered through a registrar, its installation at a top level domain necessitates a primary server, and, in addition, at least a secondary server to make the domain work even if one of the name server stops working.[15]

The Client of DNS is referred as the DNS resolver and is the one that initializes and sequences the translation of a domain name into an IP address; this can be done in two ways: through a recursive query or a non-recursive query.[14]

It is recursive when a DNS server completely answers the query, even querying other name servers if it does not know the answer. It is non-recursive when that answer contains only records where the server is authoritative or it gives a partial answer. The DNS information is stored in the domain servers in zone files. Given below is an example of zone file and its explanation (Image 4) [14].



```
; zone file for example.com
$TTL 2d    ; 172800 secs default TTL for zone
$ORIGIN example.com.
@              IN      SOA   ns1.example.com. hostmaster.example.com. (
                         2003080800 ; se = serial number
                         12h        ; ref = refresh
                         15m        ; ret = update retry
                         3w         ; ex = expiry
                         3h         ; min = minimum
                         )
               IN      NS      ns1.example.com.
               IN      MX  10  mail.example.net.
joe            IN      A       192.168.254.3
www            IN      CNAME   joe
```

Figure 4: zone file for example.com [14]

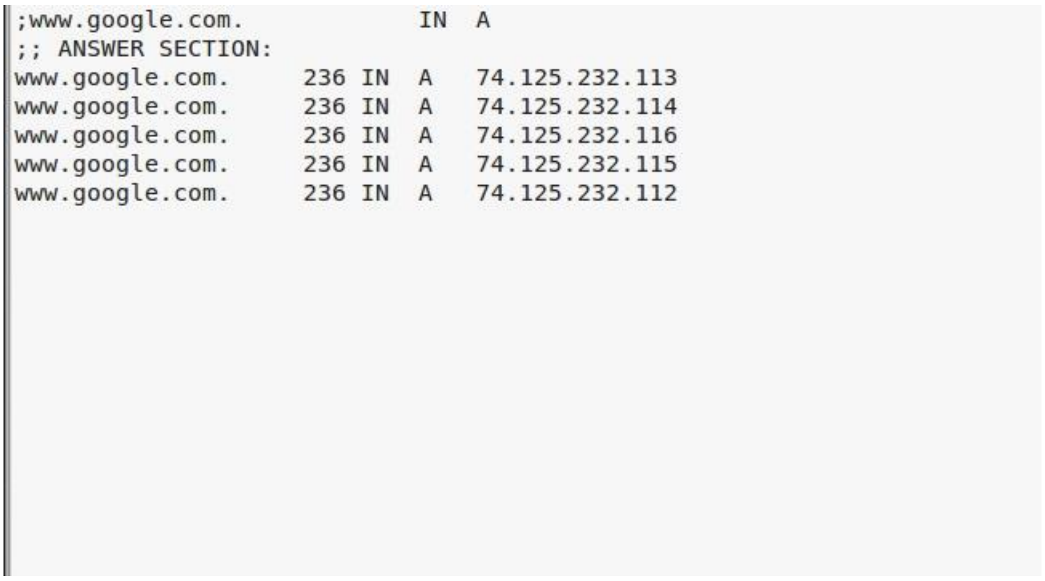
In the following table, (Table1) the syntax used in the example is explained.

Type	**Description**
A	The IP address of the host.
NS	Host name or server name.
CNAME	Host canonical name, that is an alias for the host domain name.

PTR	Host’s domain name, host identified by its IP address.
HINFO	Some information regarding the host.
MX	Mail exchanger.
AXFR	Request for zone transfer.
ANY	Request for all records.

Table 1: Explanation of zone files [14] [15]

Name servers sometime receive multiple records from a query and most name servers return the answer in a random way and this makes it possible to assign a single hostname to several IP addresses. As this allows more machines to work for the same domain, this is useful for very busy domains like www.google.com. For example, with the Linux dig command on, www.google.com it is possible to see different IPs for this domain (Figure 6). [15]



```
;www.google.com.            IN  A
;; ANSWER SECTION:
www.google.com.     236 IN  A   74.125.232.113
www.google.com.     236 IN  A   74.125.232.114
www.google.com.     236 IN  A   74.125.232.116
www.google.com.     236 IN  A   74.125.232.115
www.google.com.     236 IN  A   74.125.232.112
```

Figure 5: dig www.google.com [14]

2.2.Threats in DNS

Unfortunately, when the DNS model was deployed, security was not considered, and as a result, it contains a lot of very dangerous vulnerabilities, and considering that life today relies on Internet, this fact could lead to a number of destructive consequences.[14]

This chapter will describe the well-known Threats in the DNS protocol.

2.2.1. Packet Interception

Packet Interception is the simplest threat against DNS; an attacker could intercept the traffic of a victim and therefore, the DNS queries. There is a lot of free software that makes this attack easy even by non-technical people. What is important to understand is that this kind of attack does not concern just the privacy of users (reading URLs typed), but could also be used for bigger attacks like DNS. [12]

2.2.2. DNS hijacking

It consists of giving fake DNS resolutions in a way that the victim, believing that he is entering the domain, enters a rogue one [26].

This practice is usually done for malicious purpose like phishing, pharming, collecting statistics, or for an illegal means of advertising. Pharming is an attack used to redirect a website's traffic to redirect it to another fake website, mostly used to generate advertising revenue. Phishing is an attack where the goal is to redirect a user to a malicious website that looks like the original website to steal the user's credentials [26].

This attack can be done in a lot of ways and at every level, and the closer the computer attacked is to the root of the DNS tree, more the hosts that will be affected by this attack. One such scenario is when a user downloads a malware (for example, the Estonian DNSChanger Trojan) that changes the DNS configuration of the user's computer or router (if it has some vulnerabilities). Another scenario is when an attacker, after sniffing traffic, does ID Guessing and Query Prediction, thus being able to give fake answers. Even ISPs sometimes use DNS hijacking to introduce advertisements or collect statistics [26].

The level of danger of this attack depends on how the attacker wants to use it [26].

2.2.3. ID Guessing and Query prediction

It is a technique that consists of generating packets that match the casual parameters of the transport protocol. This is possible because DNS uses the UDP/IP protocol and the ID is only 16 bits. As a consequence, the range to brute force is quite small and could the possibility of the attacker guessing it rises; for these reasons, this range is not big enough to guarantee protection and, in addition, the client UDP port and ID can be predicted, because there could be a fixed value to pass the restrictive rules of firewalls. [16]

2.2.4. Cache poisoning

This attack consists of putting fake information into DNS caches so as to modify the association between IP and name server. This leads to the redirection a domain name to a malicious IP. The damage from this attack can be more severe if the malicious answer is cached by a web cache, because it will affect multiple users and will continue to do so until when the cache is purged. [17]

2.2.5. DDoS

“Distributed denial-of-service (DDoS) attack is an attempt to make a machine or network resource unavailable to its intended users.”[18] This attack is used broadly on any service and can be done in a lot of ways. [18]

If an attacker does a DDoS attack on a DNS server, there will be no answer to legitimate requests, so that people will be not able to use the services they need.
A DDoS attack against a DNS server could be even more destructive than against other services, because stopping a DNS server would make unavailable all the services that rely on it. [18]

It can be done at every level of the DNS tree and is more is more destructive the nearer it is to the root zone. Sometimes, even the root servers are attacked - an attack that potentially could put down the entire Internet. [18]

2.2.6. Amplification Attack

DNS Amplification is a Distributed Denial of Service that uses DNS answers to amplify the power of an attack. With this attack, it is possible to cause a DDoS without having a large botnet [19]. Basically, it works in 2 steps:

First, the Attacker spoofs the IP address of the victim and finds an internet domain with many DNS records. After this, the attacker does a DNS query to get all records from the internet domain and puts it like a source of the spoofed IP address of the victim. [19]

2.2.7. **DNS Zone transfer Information Leakage**

DNS Zone transfer is a process wherein a DNS master server passes a copy zone to a slave DNS server. Also, here, the DNS misses some security procedures; in fact, you just have to pretend to be a slave server to ask for the copy if the master server is not well configured. [12]

2.2.8. Cache Poisoning of Mail Handling Domains

Mail servers also use the DNS to communicate with each other, and so, there exists the possibility (like with web pages) of communicating with a rogue one [20].

Jonathan Spring gives an example of how this could happen: “Unless the whole message is cryptographically protected, the intermediate server can read and modify the message, or append malicious content. This attack would defeat opportunistic TLS encryption between MXs that is meant to ensure the confidentiality of messages in transit. And there are some small changes the intermediary could make to the mail that would make the attack worthwhile, such as changing a bank account number for a home purchase deposit.” [20]

2.2.9. DNS Exfiltration

DNS requests are usually allowed and so, even a well configured firewall could let malicious DNS requests from an internal DNS pass and let data getting the data outside a company’s firewall without being detected. For example, an attacker could embed a SQL query in the DNS requests and sniff the answer [23].

What is really dangerous about this technique is that it is difficult to stop, because every company has to allow DNS traffic, as it is necessary for the surfing the Internet, and this allows hackers to successfully exfiltrate important corporate data, even if organizations do not consider DNS a threat for data exfiltration [23].

In summary, this is a powerful technique for both lack of knowledge as well as vulnerability.

2.2.10. DNS Tunneling

The DNS Tunneling (or Encapsulation) attack encodes the data of other protocols in DNS queries and responses. [24] As stated above, DNS is rarely monitored and usually not stopped by firewalls. DNS Tunneling could be used in several ways; one of the most popular is to have free Internet from systems that allow you to authenticate the router but to access the internet only after a subscription and payment; another one includes transfer and stealing of data from an infected computer, or even to have complete remote access to it [24].

3. DNS Attacks Statistics

Attacks against and through DNS are dramatically increasing: in an article in securityweek.com, it is stated that more than three-quarters of companies in the UK and USA have been attacked using the DNS protocol [27]. In another article from net-security.org, it is written that DNS attacks have increased by 170% in the last years, and as you can see from the image, the DNS is the second protocol was most used as the attack vector. This means that DNS protection is to be considered necessary even if a lot of senior managers do not do it [24].

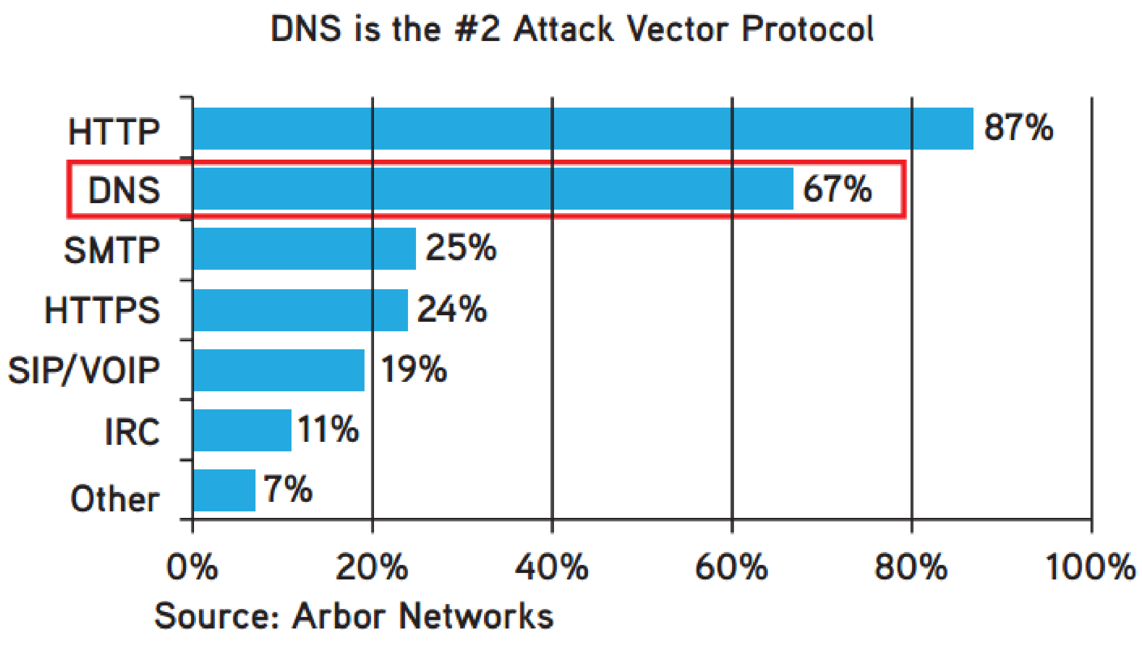


Figure 6: Most common attack vector protocols

Chapter 2.2 has explained how it is possible to misuse DNS for malicious intentions and in chapter 4 will explain the protocols which are thought to bring security to the DNS. In this chapter, statistics are presented along with real facts on attacks mostly used in real life against companies, organizations and users, and why they could be really dangerous. To do this, surveys and articles found on the Internet will be used. The scope of this chapter is to understand whether the protection of the DNS is really a necessity, and if these kinds of attacks are just theoretical or are implemented. According to the results, the following chapters will determine the kind of solution that is best to mitigate those attacks.

3.1.Attacks Against Companies

It is possible to see from the research study done by Vanson Bourne and commissioned by Cloudmark, Inc, that the most common DNS attacks that on companies are [25]:

- DDoS (74%)
- DNS exfiltration (46%)
- DNS tunneling (45 %)
- DNS hijacking (33%)

3.1.1. DDoS

As seen in the webinar organized by IMPERVA/Incapsula [28] (a company that offers security tools for web apps protections), a DDoS attack can be very destructive. In the webinar, they explained the effects that this attack could have for a company through the experience of who have suffered this attack. Testimonials claimed that 1 hour of DDoS costs about $40,000 and that generally these attacks last less than 1 day (about 13 hours), but they can still cause a huge loss, often more than $500,000 for one attack. These losses are shared between the IT team, security/risk management, customer service, and customer sales [28].

Incapsula conducted a survey on DDoS attacks in August 2014, and this highlighted that, in addition of the costs mentioned above, they also have other side effects that cause a loss. The result of the survey claims that 52% of those surveyed had to replace hardware or software, 50% had a virus or a malware activated, and 43% experienced a loss of consumer trust. [28]

3.1.2. DNS Exfiltration

As mentioned in chapter 2.2, this technique makes it easier for hackers to steal data. Of course, this can be very dangerous for companies; for example, if a company produces software and after years of hard work, a hacker steals everything and sells it to competitors, this could lead to a huge loss of money; even worse could be if a hacker steals confidential data of costumers or employees of a company. In both cases, the loss will not stop at data, but we have also to consider side effects like the loss of credibility; costumers will trust the company less and of course, competitors will take advantage of this. Even if people are not aware, these things happen; for example, the Hacking Team hacked and dumped 500GB of data over the Internet [34]. This kind of attack could signal the end of a company or put it in serious difficulties [30].

3.1.3. DNS Tunneling

The damage that a DNS tunnel can create is the same as that of DNS exfiltration, but, in addition, there will be a big loss for companies that offer Wi-Fi Internet connections for a cost, because through this technique, it is possible to have free Internet. In an article in SC magazine, it is written that thanks to this technique, cybercriminals bypass Wi-Fi payments and roaming fees and this causes a revenue loss of millions of dollars to telecommunication providers. [35]

3.1.4. DNS hijacking

This method is arguably the most dangerous and marks the insecurity of the DNS protocol. It could be very dangerous both for costumers as well as for companies, because DNS Hijacking could be used for phishing and therefore, to steal passwords, acting as the man in the middle, sniff conversations, or even to take control of machines inside private networks, and this could be the start for an attacker to do the attacks described above and in general own your private network [26]

Arguably, this technique can be devastating for company that works in cloud, because redirecting the webpage where users access the cloud services (nowadays, most companies are on the cloud) could make available to the attacker all the information he needs to have to access the infrastructure. This, if done fast, could even bypass the second step authentication.

This attack if done against normal users could lead to identity theft, bank accounts theft, and phishing. In addition, it is really difficult to detect by non-technical people.

3.2. Against Routers

According to the Internet security research organization's Team Cymru, some hackers managed to compromise 300,000 home and small-office wireless routers by modifying DNS settings and putting their rogue DNS servers [29].

The routers have been compromised using various techniques, but the most commune attack vector has been the exploitation of vulnerability on the ZynOS firmware used on many routers. This vulnerability allows an attacker to download the configuration file without authentication (it is enough to write the path); the file is encrypted, but it is easy to decrypt, using websites and tools that do it in seconds [29].

According to Ara Labs, a malware infecting routers hijacking the DNS system is able to intercept requests to the Google analytic.com domain, allowing the malware to inject ads in the site people are browsing [32]. In addition, there are malwares that change the DNS settings in users' computers (for example, the Estonian DNS changer) [33]. These attacks show that controlling the DNS settings of a victim means that the attacker controls and decides the website the victim is accessing, so that he could easily be able to do phishing, steal credentials, credit cards numbers, etc. This kind of attack could literally ruin the life of the victim, and the very possibility of these attacks could make costumers scared of conducting operations over Internet, leading to heavy losses for companies that rely on these services.

3.3. Considerations

In the attacks described in sections 3.1 and 3.2, it is clear that these threats are real, and the DNS vulnerabilities are continuously exploited. What the reader has to become aware of is that attacks described in this chapter are a small portion of the ones possible and described in chapter 2 and this does not mean that others are not used or that they are not common, because these attacks are difficult to detect, and if the victim is a normal user, he would never notice that he is under attack. Even an expert could be deceived.

These attacks show how difficult it is to protect the DNS and how the link between the users' node and the DNS resolver and the single node itself are the weakest points of the chain, and as the chain is strong as its weakest point, changing the DNS settings or answers in the "last mile" potentially make useless all kinds of protections that could be present on other levels.

There exists a solution for mitigation of most of the attacks described in this chapter, and they are still widely used for a lack of knowledge in the DNS protocol.

4. How To Secure DNS

In the previous chapters, we have seen that DNS is not secure and that it allows numerous attack vectors, that could make huge damages in many aspects; some of them could attack individuals, but others, as we have seen, are largely used against companies.

In the last decade, there have been other protocols deployed that have claimed to be the solutions for the problems in DNS. This chapter will describe these new protocols, highlighting what they solve, what they do not solve, and what they might get worse. This is done with the aim of identifying the best protocol to increase Internet security and one suitable to real life.

The aim of this chapter is to identify the best protocol, and see how and if they could interact for better security.

4.1.DNSSEC

“DNSSEC is a set of DNS extensions that authenticate the origin of zone data and verify its integrity by using public key cryptography”[41] This means that a set of security extension to the DNS have been rewritten in a way that it will continue its proper functionality but it is finally possible to authenticate one zone's owner. DNSSEC guarantees the authentication of the origin of DNS data, its integrity, and an authenticated denial of existence, and nothing else; this means that it does not offer an entire protection of the DNS packet from beginning to the end, as DNSSEC by design is not able to protect every vulnerability of DNS, but is really focused on allowing a client to authenticate the contents of a DNS response.[40]

4.1.1. New resources record

DNSSEC introduces new resource records to put in addition the ones that DNS had, and those are needed to increase the security of the connections between nodes [37] :

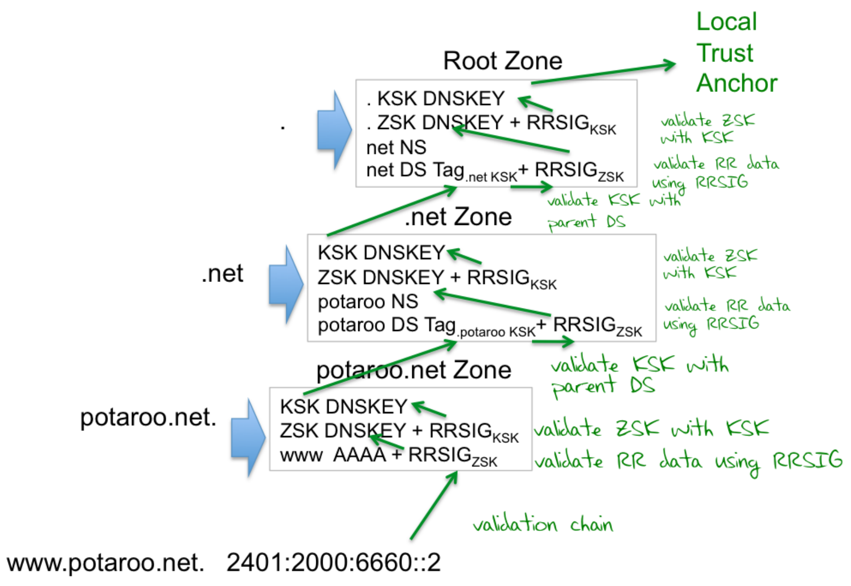
- DNSKEY: (DNS Public Key), Needed for verifying a RRSIG.
- RRSIG: (Resource Record Signature), Signature over RRset made using private key.
- NSEC: (Next Secure), Indicates which name is the next in the zone.

- DS: (Delegation Signer), Pointer for building chains of authentication.

4.1.2. How DNSSEC Works

Similar to the DNS, DNSSEC also works through a chain of trust, but the domain that is above provides validation of information for the one that is under it. The data has a digital signature for demonstrating its authenticity, and this is done by running a secure hash on the data and after the results of the hash are encrypted by the zone's private key. Each digital signature is stored in a RRSIG record in the signed zone file and to verify the signature, it is possible to decrypt it with a public key stored in a DNSKEY-record. Every zone has two keys, one public and one private, and for this, it has a zone-signing key pair and a key-signing key pair. The private zone-signing key signs each Rrset, while the public verifies the signature. To secure delegations, it uses DS-records, which are a hash of the child’s self-signed key-signing key DNSKEY records. The DNSSEC specifications require that if a zone has multiple keys, each of them is tried until the data is validated. This allows the key to be changed without stopping the service. [37][40][12][48]

In Figure 7, it is possible to see how the validation of a DNSSEC Domain Name happens:



Local Trust Anchor
Root Zone
.
. KSK DNSKEY
. ZSK DNSKEY + RRSIG$_{KSK}$
net NS
net DS Tag$_{.net\ KSK}$ + RRSIG$_{ZSK}$
validate ZSK with KSK
validate RR data using RRSIG
validate KSK with parent DS
.net Zone
.net
KSK DNSKEY
ZSK DNSKEY + RRSIG$_{KSK}$
potaroo NS
potaroo DS Tag$_{.potaroo\ KSK}$ + RRSIG$_{ZSK}$
validate ZSK with KSK
validate RR data using RRSIG
potaroo.net Zone
validate KSK with parent DS
potaroo.net.
KSK DNSKEY
ZSK DNSKEY + RRSIG$_{KSK}$
www AAAA + RRSIG$_{ZSK}$
validate ZSK with KSK
validate RR data using RRSIG
validation chain
www.potaroo.net. 2401:2000:6660::2

Figure 7: DNSSEC Domain Name Validation [48]

4.1.3. Authenticated Denial of Existence

In the rfc3833 [49] the IETF highlights the problems of authenticating the Denial Existence of a domain, because an attacker could create an appearance of denial of service answering to the victim. IETF says that is not clear how an attacker could use this vulnerability, but it exists, and so, is worth protecting.

The Denial of Existence is a way to inform a resolver that a domain does not exist or that it does not contain the data required. Let's see how this works in the DNS and later, in the DNSSEC.[49]

4.1.4. DNS (NXDOMAIN)

If someone asks for an existing domain, the DNS server will answer with the status: variable set on NOERROR; meanwhile, if it does not exist, the answer will be status: NXDOMAIN. If the domain exists, but does not contain the data required, the status is set as NOERROR, and NXDOMAIN otherwise. The DNS packet is not signed, and so the status NXDOMAIN cannot be trusted, because the packet could be forged. [13]

4.1.5. NSEC

To sign the negative answer, the negative answer has to be precomputed, and it is impossible to precompute every possible negative answer, and it is also not possible to precompute only one answer for all answers, because this would be vulnerable to the replay attack.[13]

NSEC is the solution to authenticate the nonexistence of a domain. It orders DNS names according to their most significant labels, as described in the rfc4034 [47], and if the requested domain should follow between two contiguous existing domains, it will answer with the two domains near the one requested, indirectly proving the non-existence of the domain requested. [47]

4.1.5.1. NSEC 3

The third version of NSEC is done to try to mitigate two problems of NSEC [50]:

- It allows zone walking
- It could lead to a huge increase of the size of zones.

To solve the first problem in NSEC3, every domain name is hashed (as well as the owner name), and the NSEC3 chain is sorted by the hash and the hashing can be repeated several times, taking as input the previous hash.[50]

To solve the second problem, NSEC3 introduces the possibility to sign only some domains and not necessarily all the zones. This functionality is named Opt-Out. [50]

What is important to know is that NSEC3 is not a substitute NSEC, which is the default option.[50]

4.1.6. Application Uses of DNSSEC

What is really valuable to the DNSSEC protocol is the fact it can be used to link cryptographic keys to domain names and so, permits applications to use them in other security protocols in addition to the DNS. As it is written in the chapter … the SMTP protocol is prone to hijacking and the possibility to read and even modify emails could be really juicy for an attacker and really disastrous for the victim. Today, the only protocol that provides protection to these kinds of attack and many others is DNSSEC, with its application extensions that are going to add protection to other existing protocols like SSH, SSL/TLS, SMTP and others. [51]

Some existing extensions protocols are [51]:

- SSHFP (Secure Shell Host Key Fingerprint): to secure the SSH protocol, it allows to validate the SSH host using the DNSSEC protocol and is supported in the OpenSSH and even enabled by default in FreeBSD 10
- IPSECKEY: it is a method for storing IPsec keying material in DNS
- DKIM TXT: the Domain Keys Identified Mail protects from spamming and phishing attempts, validating a domain name identity associated with an email through DNSSEC.
- DANE: the DNS-based Authentication of Named Entities protocol solves the problem of securing a communication between two parties over the network

4.1.7. DANE

In this chapter, the DNS based Authentication of Named Entities protocol will be described, after explaining what is wrong with actual security mechanisms.

TLS/SSL

When a website is visited, it is important to ensure the integrity of its data. Today, this is achieved by the use of SSL (Secure Socket Layer)/TLS (Transport Layer Security) protocols used in the HTTPS protocol. These protocols encrypt the information shared in a way that this cannot be changed or read even if captured with a man in the middle attack. The client connects to a website with the protocol HTTPS (Hypertext Transfer Protocol over Secure Socket); the server must present a x.509 certificate that will be evaluated by the browser, checking if it is for the right Domain Name, if it has the right IP address and if it was signed by a trusted Certificate Authority. It is important to understand that this certificate protects only the data exchanged between a client and a server, but does not certify the exchange between the client and the right server. [39]

- **Vulnerabilities inside the protocol**

In the past year, the SSL protocol suffered various serious vulnerabilities that put in danger millions of customers and companies; hackers have been and are able to disclose information like usernames and passwords. These vulnerabilities are named HearthBleed and Poodle.

    - HeartBleed

It is a serious vulnerability in the OpenSSl crypto library that affects the Heartbeat Extension implementation. Heartbeat is necessary for connections not being interrupted if there is no data transfer, and to enable the path MTU discovery. The problem in this implementation is that the length of the buffer used to create responses to the Heartbeat’s answer was decided from the host and not checked by the server, leading to the disclosure of sensitive information like passwords, private keys and session IDs. [62]

- Poodle

The Poodle attack exploit the need of servers to be compatible with as many hosts as possible, and many of them do not support new secure protocols like TLS, so, if a host is not able to connect through TLS, the server contains the possibility of connecting with SSL3, which is a weak protocol and the packets encrypted with this method are easier to decrypt. An attacker is able to force the connection between a client and a server through SSL 3 and so, is able to do a man in the middle Attack.[52]

- **The Certificate Authority Problem**

Vulnerabilities such HeartBleed and Poodle are scary and it is necessary to anticipate the possibility of these, for layered security system is necessary to follow the security bulletins and to patch them as soon as possible. But there exists something even more dangerous. For the SSL/TLS negotiation to work, it is necessary that a third party certificate is issued by a Certificate Authority (CA), and here lies the trouble.[40][51]

There is an abundance of CAs (more than 1300) that browsers recognize by default as legit, and these CAs are referenced as Trust Anchors (TAs). Every TA can issue a certificate for any website and it will be recognized by the browser as legit. This means that if an attacker is able to take control of any of the TAs, he will be able to generate a certificate for any website he wants. This means that all the systems are secure as the weakest TA, and it is not possible to associate a website to a specific certificate or CA. [40][51]

TAs are well secured and have high security standards, but 100% secure systems do not exist, as proved by the case of Comodo. [38]

On 15th March, 2011, the Comodo TA suffered an internal security breach and attackers were able to steal these certificates with the following domain names [38] :

- addons.mozilla.org
- login.live.com
- mail.google.com
- www.google.com
- Login.yahoo.com
- Login.skype.com

- Global trustee

With these certificates, an attacker could conduct a man in the middle attack, and would not be detected, because, as written above, a browser is not able to understand if the website should use a certificate issued from a precise label or another; in addition, there are some TAs that do not check if you really own the website that you are signing with the certificate, and all the CA accepted by a browser are considered valid in the same way. The RFC 6698 document of IETF states: "Recent experiences with compromises of CAs or their trusted partners have led to very serious security problems, such as the governments of multiple countries attempting to wiretap and/or subvert major TLS-protected web sites trusted by millions of users."[39] This means if a country owns a CA, it could easily spy on its own citizens and citizens of other countries.[38]

- **DNS-Based Authentication of Named Entities (DANE)**

DNS-based Authentication of Named Entities (DANE) is a protocol that allows certificates used for TLS/SSL to be linked to DNSSEC. This means that it can solve the problems described above because linking a website to a certificate means that only one certificate (or a range of certificates, depending upon the configuration) will be considered valid for that website. DANE introduced new DNS resource records named TLSA and it has to be DNSSEC signed or it will not be valid.[39]

DANE protocol can work with TLS certificates issued by Certificate Authorities (CAs), but also it could be used to add layers of trust to self-signed certificates, thanks to DNSSEC.[39]
DANE gives four options to use a certificate in the TLSA record [66]:

- CA specification – Basically, DANE gives the possibility to say that the only certificates that have to be accepted from the domain you own have to be issued from a specific public CA, so that all other CAs, even if trusted by browsers, have to be untrusted.[66]
- Specific TLS certificate – With this option, it is possible to specify the exact TLS certificate that has to be trusted in the domain. The TLSA record specifies the exact TLS certificate that should be used for the domain (this has to be signed by a valid public CA). [66]

- Trust anchor assertion – This allows to specify the trust anchor that has to be used for validating the TLS certificates for the domain. This means that there are ways to secure using a private certificate authority.[66]
- Domain-issued certificate – The TLS record specifies the exact TLS self-signed certificate that should be used for the domain.[66]

There is a debate whether self-signed TLS is enough to secure, or if it is even worse than the actual TLS method. CAs claims that the best and most secure solution is to use public CAs with DANE. While the best means can only be determined through the number of cases, the fact remains that DANE can solve the big problem of trusting the HTTPS connection used to encrypt network traffic. Another thing of note about DANE is that it offers a way to protect SMTP, XMPP and SIP connections that the normal TLS it is not able to do. This means that DANE is able to solve the attack described in chapter 2.2.6 Cache Poisoning of Mail Handling Domains, but also protect VoIP calls and chats.[39][40][66]

- **DANE Problems**

Today, DANE it is not supported by any of the well-known browsers, because the validation with this system is slow and the goal of the modern browsers is to visualize the page faster and faster. Some plugins exist for Firefox and Chrome, but they do not check everything, such as URLs of images and JavaScript, and in addition, downloading a plugin that validated such an important thing could lead to other attack vectors, because it is possible to download malicious plugins that, instead of checking the DNSSEC path, helps hackers.[39][40]

Conclusively, there are few websites that use the DANE validation.

4.2. Problems in Developing DNSSEC

With IETF standards that seem functional and deployable, and we have instruments to work with the DNSSEC, why are we still using DNS?

Unfortunately, the answer is that DNSSEC also has a series of problems difficult to solve.
Here is a brief list of these problems:

- Network path and availability: DNS messages are limited to 512 bytes, but DNSSEC is using a protocol that supports UDP packets bigger than this size; however, not all the implementations and nodes of the tree support this. [41]
- Every link of a sequence in the network has its own MTU, and if in the network path, there will be an MTU smaller than the packages that DNSSEC requires, this will slow down the connection. [41]
- Some firewalls could fail to deliver large DNS messages. [41]
- KSK roll-over issues: This is the most common error that happens with DNSSEC.[41]

  Considering that a sub-zone has rolled its KSK, but failed to upload it new KSK or has done it incorrectly, mitigating the problem can be done by two alternatives:
  - The first one is to let the subzone operator push up a correct KSK or the older one, but this means that the zone will be invalid for a while. [41]
  - The second solution is to pull the subzone's DS RR pointing to a missing KSK from the parent, but this will make the zone insecure. [41]
- Timing issues: All signatures in a sub-zone could expire or may be published before they are valid, or the system may have a wrong clock, and this will make the zone not valid. [41]
- Signed zones served on DNSSEC-unaware systems: If an admin signs the zone but does not configure the server to be DNSSEC-aware, or there are some problems and the server cannot correctly serve a DNSSEC-signed zone, the zone will answer with a normal DNS response. This will create a problem if there is a DS RR in .gov, because here, a lack of RRSIGs would be a valid failure. [39][40]
- If a DNSSEC-aware recursive name server queries an unsigned zone, the unsigned answer that comes back is accepted as valid. But problems occur when signatures expire or when parent and child zones do not agree on the child’s current DNSKEY record. [36]
- Lack of knowledge and Training: There are few people that know how to develop DNSSEC, and even then, not all are aware if the configuration is good enough; DNSSEC could be even more dangerous than DNS, because, as explained before with TLSA, it is possible to generate certificates, so if someone is able to decrypt or steal both KSK and ZSK, he is basically able to rule the

domain and subdomain he has stolen, as he could change IPs, generate certificates, etc. [39][40]

- In case an attacker is able to steal or decrypt KSK and ZSK, he will be able to authenticate a fake website, and in case you are using DANE, he could do even worse; the disadvantage of this situation is that a key rollover in DNSSEC is slow, and so you have choose whether to allow the crafted keys until the rollover will be complete (therefore allowing the attacker to do his game) or to allow only the new one (making your website unavailable for a lot of costumers for a considerable amount of time). [39][40]
- DNSSEC it is not kind with errors; if everything it is not done perfectly, the service will be down, and it will be really difficult to understand the problem, because DNSSEC does not give any clue, and in addiction, this can happen in any part of the DNSSEC three and so, depending on where it will happen (root, top levels, etc.), a large portion of the Internet will be not available. It slightly extends the risk of down time. [46]
- In the event of DNSSEC error e-mails, voice-over IP and your website will be down, so it is mandatory that an alternative method of communication is devised.[41]
- Another negative side effect it is that will be easier to censor websites. [46]

4.2.1. What it solves

The DNSSEC protocol has been deployed to assure the integrity of the packets transmitted between the server and the client; with this protocol, we can be quite sure that the website we are visiting is the one written in the URL. But in which of the attacks described in the chapter 2.2 does this stop? [37][40]

Considering the statistics in chapter 3 of the most used attacks, we can see it solves DNS hijacking, and from a costumer's point of view, it is probably one of the scariest attacks and could bring a huge amount of problems and stealing of data. It also solves ID Guessing and Query Prediction (guessing the ID and Query will be not enough for a valid DNSSEC answer), and Cache Poisoning of Mail Handling Domains.[39][42]

It solves cache poisoning from the server side; for the clients, browsers still need to validate the DNSSEC websites. [37][40]

4.2.2. What it does not solve

Unfortunately, DNSSEC does not solve every problem in the Domain Name Service protocol. It is important to understand that DNSSEC does not encrypt communications; it just guarantees that the server that answered it is legitimate. Consequently, it is still necessary to use all other security protocol like https for a reliable Internet connection, and this means that it does not solve the packet interception attack. [37][40]

Here, it is described what a hacker could do with the DNSSEC:

- **AMPLIFICATION ATTACK:** DNSSEC does not solve the problem of the Amplification attack described above; actually, it does even worse, because the size of response increases in DNSSEC due to the use of signatures.[6] DNSSEC makes possible a new way of conducting an amplification attack, so that a hacker with a 1 Mbit/s connection would now be able to generate a UDP flood of more than 100 Mbit/s and this could be enough to do some damages; this could be even worse if the attack is done in a coordinated and distributed approach.[45]

- **DDOS ATTACK:** It is possible to create a denial of service against DNSSEC servers, in addition this attack could be even more destructive, because a DDOS attack on any of the servers involved to the process of authentication will lead to a denial of service of all the domains that in the DNSSEC three are under the one attacked.[13]

- **DNS EXFILTRATION:** The situation does not improve with this attack either, because, first of all, the DNSSEC protocol allows you to choose if you want to do a DNS request or DNSSEC, and second, even if it is not easy to find a software for DNSSEC exfiltration, DNSSEC will make it worse, because DNS has a limit of 512 bytes for a request, while DNSSEC has none.[44]

- **DNS TUNNELING:** As mentioned above, the DNSSEC protocol does not deny an attacker from using the DNS protocol, so arguably the situation remains the same.

Anyway, even if the protocol itself does not solve these attacks, these vulnerabilities can be patched with the implementation of the best practices.

4.3.DNSCurve

Like DNSSEC, DNSCurve is an extension of the DNS protocol, and while it is not a new protocol, it adds some functionality to the old one, and is backward compatible. DNSCurve has different goals than DNSSEC: it does not certify that the answer given by the DNS server is the right one, but that the DNS server that gave the answer is the right one, because it is an integration of link-level cryptography (to encrypt information at the data link level between two points) within DNS, and therefore, design decisions have been made to make this integration possible to encrypt and authenticate DNS packets between the client's public key and the server's public key. To do this, it uses elliptic curve security that is faster and more secure than RSA cryptography. [45]

4.3.1. How DNSCurve works

DNSCurve introduces its own protocol format (Streamlined Format) to insert regular DNS packets to efficiently use the bandwidth, and it works in a different way according to if it is a query or a response. A query consists of a client public key, client nonce and cryptographic box, and the response consists of client nonce, server nonce and cryptographic box. This means that this protocol puts legacy DNS messages into the crypto box and each packet contains a unique nonce, making every packet unique so that replay attacks are not possible, and there does not remain the problem of the expiration of signatures, system time is not encrypted, and it is more difficult for an attacker to watch the queries. But this is not so effective because it is still possible to watch the answered IP address or server name in TLS handshake. The DNSCurve uses Curve25519 instead of RSA, and it is online cryptography instead of pre-generated as in DNSSEC. It does not introduce new DNS resources of records and for this, it is far less complicated than DNSSEC. The keys are associated with name servers not with zones. The concept behind DNSCurve is simple: it offers a secure link between a resolver and an authoritative server encrypting the packets. Basically, the client transmits the public key in the query and the server encodes the public key as a server name in the parent zone, giving back its IP. [45]

4.3.2. Problems in DNSCurve

As mentioned above, the DNSCurve offers protection only from two nodes, but the DNS protocol is hierarchal; a Domain Name Server does not have all the answers, and so, it needs to ask others' DNS, and if they are not DNSCurve signed, we can not be sure that their answer is correct, and therefore, they are a necessity. Additionally, all parents of the DNSCurve, including the root zone, are signed, and this is where problems arise. [45]

DNSCurve needs the private keys online on name servers and, for security reasons, this is not feasible for top-level domains and roots; additionally, it does not support multi-hop caching, and this could impact performances. For these reasons, ICANN and the USA government claimed that the support for DNSCurve will never be deployed in the root zones. [45]

Another problem that should be taken into consideration is that the encryption of DNS would make traffic monitoring more complicated.

4.3.3. DNSCrypt

DNSCrypt is the first and most famous implementation of the DNSCurve protocol; it encrypts the DNS traffic of users and OpenDNS. OpenDNS relisted the code of DNSCrypt on github to assure that the code is not malicious and that it does exactly what it claims to do. [53]

4.3.4. What DNSCurve solves

As written earlier, the DNSCurve protocol encrypts part of the conversation between a resolver and a host (could be a resolver as well). Doing this makes it possible to verify that the answer that has been received has issued from a trusted domain name resolver and this could potentially stop the man in the middle attacks and eavesdropping against users. This is especially important nowadays because the world's Internet connectivity is going mobile, connecting on different public Wi-Fi, and so, subject to DNS attacks by man in the middle. Regarding the most used attacks, if every host uses the DNSCurve protocol, it will solve attacks against the routers. [45]

4.3.5. What DNSCurve does not solves

Even the DNSCurve can not automatically protect against the most common attacks against companies; in addition, it can not assure full protection even against man in the middle attacks, because this protocol adds a protection link to link, but the DNS protocol is hierarchal. A Domain Name Server does not have all the answers for all the domains, so it needs to ask other domains, and if the domain that gives the answer is not signed with DNSCurve, the answer could be potentially malicious. [47]

4.4.Namecoin

Namecoin is a crypto-currency and a fork of Bitcoin. The Namecoin protocol is different from the DNS protocol and it is not backward compatible. First of all, it is a decentralized namespace, and this does not have a single point of failure like the root zone in DNS-like protocols. The decentralization of the protocol has many consequences and one of them is the primary scope that the new protocol has, to block the censorship. Recent examples of this censorship include Turkey, where Twitter and YouTube were blocked by redirecting their DNS records. China is also renowned for conducting DNS cache poisoning as a common practice to censor domains for their citizens. Another big example is when the US Department of Justice shut down the wikileaks.org domain. [46]

4.4.1. Zooko's Triangle

Namecoin is the first naming system that clams to have all the desirable properties that a name system should have, and these are [63]:

- **Secure:** there is only one possible match and there is no possibility to give fake answers on domain names.
- **Decentralized:** there is no a central authority.
- **Human-meaningful:** the names should be short enough to allow people to memorize it.

For the points described above, the DNS is able to satisfy only the human-meaningful property; the DNSSEC satisfies Secure and human-meaningful and DNSCrypt satisfies only human-meaningful property, because, as explained before, it is not possible to use it in all the hierarchal chains of the DNS.[46][63]

4.4.2. How Namecoin works

The Namecoin naming system is based on Bitcoin and, like the latter, it uses a distributed blockchain as a proof-of-work to instate the consensus of the ownership of a domain. The blockchain is a transaction database containing all the transactions executed so for with Namecoin, through which transaction clients are able transfer Namecoins from one transaction to another with a signed message. This message is broadcasted to the entire node’s network to be verified from the other nodes using the public key from the sender’s address. [46][63]

The transactions are collected inside blocks and once it is created, it is appended to the blockchain and consequently grows into part of the Namecoin history, meaning that it is not possible to change or remove it, confirming the transaction. Every block contains a cryptographic hash, giving a block reference to the previous block. [46][63]

The proof of work is given by the creation of a block and appending it to the blockchain. This process is called mining and it is an expensive process because it needs a big calculus power to solve a unique and difficult math problem. The solution of this problem is the proof of work. [46][63]

In addition, users can record and transfer names and attach data to keys, and because they are stored in blockchains, they are also decentralized and because they can be arbitrarily chosen, they are human-meaningful. Every node can check the validity of the operations on keys and this means that they are also secure, satisfying all the properties of the Zooko’s triangle. [46][63]

4.4.3. How to Resolve a domain

As mentioned above, Namecoin is not backward compatible with DNS, and so, resolving a domain name is different. The domain stored in the Namecoin Blockchain has as top level domain .bit, which, although a virtual top level domain, is not under the ICANN; this means that it is not assigned in the DNS root zone. For these reasons, DNS servers are not able to solve queries for these kinds of domains. [46][63]

A secure way to resolve the domains is to use a local DNS that will be synchronized with the Namecoin service running in the background to have an updated full copy of the blockchain. Consequently, it is necessary to change the DNS settings of the operative system to use the local DNS server. [46][63]

Another possible solution could be to use a publicly available Namecoin DNS suffix gateway, but this solution is not secure because there is a possibility of middle man attack and censorship. [46][63]

A third solution is to use a full proxy service that allows to access .bit domain, but this solution is also highly insecure because the owner of the proxy server could monitor, do act as the man in the middle, hijack domains, or even modify the traffic of the .bit websites. [46][63]

4.4.4. Problems of Namecoin

In the author's opinion, the worst problem that Namecoin has is domain squatting. The cost of registering domains is really cheap, and so, people can register a huge amount of .bit domains anonymously and never use them. In addition, the service, being anonymous, does not check who is buying the domain; as a result, many people are buying trademarked or copyrighted names. For example, if checking for information about domains in Namecoin Block Explorer, we can see that Google, Alexa, Ebay and LinkedIn belong to the same email, and it does not seem to be the right email for these domains. The problem here is that there is not a central authority and ample information about the domains. Their anonymity means that there is no authority that can solve a dispute resolution, so these domains could be sold at a really high price or used for malicious purposes. [46][63]

Another big problem is that there is no mobile support for it, for, in order to solve Namecoin names in a secure way, a full Namecoin node must be run, along with a local copy of the whole blockchain; this is not really suitable for mobile devices and their actual hardware, and it will be required to root the device to change the DNS settings. Another problem for mobile devices is that the locked-down APIs of iOS and Android make it difficult to have domain resolution for all domains. For example, to do this on Android, the phone should be rooted and this creates other security problems on the device. [46][63]

In addition, Namecoin could have some code quality problems at its core. For example, in October 2003, a big bug that was allowing anyone to steal Domain Names was discovered; also, the version Q.3.73 was not mandatory for the integrity of the key value. [46][63]

Furthermore, Namecoin shares all the theoretical problems that also Bitcoin also contains. If a node has enough computing power and has control of more than 50% of Namecoin's network, it would permit the owner of the node to exclude or modify the blockchain, and by doing this, he could have full control of the validation of the transaction. [46][63]

Even efficiency does not seem to be the strong point of Namecoin; if the entire world uses Namecoin, the block chain file would increase sensibly and this could bring about a number of problems. [46][63]

4.5. Considerations: Which protocol to choose?

In this chapter, the three major alternatives to DNS have been presented, all of which have some peculiarity that makes that protocol better than the other two alternatives in certain situation.

The Namecoin crypto currency is the only one satisfying Zooko's triangle properties [46][63], as it is able to offer a domain name server that is definitely decentralized and human-meaningful, and also secure, but this last point is only in theory because it has showed a lot of problems in this point.

Namecoin, in spite of being a really promising and fascinating alternative, can not be considered a substitution of the DNS because, until everyone can buy any domain there, is no purpose of being secure while communicating with the real Domain Name we want to communicate with, if the owner of that Domain is not the one we think. In addition, the world is going mobile and a technology that is not supported by mobile devices cannot be the right choice, but if Namecoin is able to solve the security problems related to the selling of Domain Names, it would be a great parallel alternative to DNS to fight censorship and a good use of it could be made in the TOR network, where domain names are hashes that are not human-meaningful.

The DNSCurve, on the other hand, being not supported by IETF, cannot be used as a standalone protocol, because it is not able to offer security thought all the three DNS, but it could be used in conjunction with DNSSEC and could offer considerable protection against the “last mile attacks”. Users could use the DNSCurve to connect to a DNS server, adding a layer of security, but it still is not enough if the DNS server we are connected to contains wrong information against which DNSCurve does not give protection.[46]

DNSSEC, like the other two discussed above, does not protect against all the most common attacks against companies and it could actually make things even worse, but these kind of attacks could be stopped or at least mitigated through the best practices. This means that the only possible protection against these threats is to let administrate the DNS only to professionals and this is more true with DNSSEC, because, as explained in 4.1, a misconfiguration in DNSSEC could lead to even more serious damages than an attack. But this should not frighten us too much; in every area, there are jobs that, if they are done by people that are not qualified, would lead to disaster; why, then, should it be different in as important a thing as DNS, few experts exist in DNSSEC, and it is unlikely to be taught in universities and not easy to find updated information about. In spite of all the problems DNSSEC has, it is the only protocol that could bring about real security in the DNS, and the only one that works on any node from the beginning to the end of the hierarchal tree, and this is arguably more important than all the problems it could have. So, according to what has been analyzed so far, it can be concluded that DNSSEC is the right direction to take.

5. An Android DNSSEC Browser

The weak point of the DNSSEC protocol is the connection between a host and the DNSSEC resolver [13]; if the user does not use a DNSSEC aware browser or a DNSCurve link to the DNS server, all the protection that DNSSEC brings disappears. While both solutions can be used from a Windows PC, from a MAC and from Linux (here, there is not official support for DNSCrypt but it is still possible), there are no secure solutions for mobile devices yet.

DNSCrypt/DNSCurve on Android needs the device to be rooted and it is a dangerous and complex method, in addition it invalids the guarantee of the device [53]. Similarly for iOS, the device needs to be jailbroken [54] with the same consequences as rooted Android devices.

Until the beginning of this thesis there existed no DNSSEC browser and the extensions that were working for Firefox and Chrome on Desktop were not working on mobiles.

According to the article "Smartphone now most popular way to browse internet" [2] by theguardian.com users access the Internet more through smartphones than from PCs, so this could mean that people start to access important and sensible services from smartphones, therefore calling for a need to bring the security that is possible on PCs in smartphones.

To achieve this, an Android DNSSEC-Aware Browser has been designed, which will be able to identify if a domain name is signed with DNSSEC and if the key is valid. The purpose of the app is to guarantee to Android users that they are browsing the right web page or alert them if it has been hijacked, or if the Domain is a rogue one. This app assures the security of DNSSEC even in the link from the host to the DNSSEC resolver. In addition unrooted Android devices, it is not possible to change the DNS configuration, so that they use the ISP DNS, but not all ISPs offer DNSSEC verification. For Android clients of ISPs that do not validate DNSSEC domains, using this app is the only way to have DNSSEC verification.

This browser also offers a layer of protection against DNS hijacking if the attack happens in the local network even if the Domain is not signed with DNSSEC, such as when using public Wi-Fi, when this function is missed from Android's browsers.

5.1.Making the App

To build this DNSSEC-Aware Browser, it is necessary to identify the libraries that could help verify DNSSEC signed domains. These have been identified on github the dnssecjava library [55] of Ingo Bauersachs, and based on the dnsjava [56] library. These libraries are the bases of mostly well-known java projects that need the validation of DNSSEC domains.

Android has a class called WebView that is able to handle http, https connections and this is the right one to create a view to be used as a browser. The problem here is how to make the validation on every link clicked inside the browser. In the class WebViewClient there is a method called shouldOverrideUrlLoading, which returns a Boolean according to if the link has to be executed inside the WebView (false) or from the default Browser (true); the last one is the default for this method. Creating a new class that extends the WebViewClient it is possible to override this method to decide what the links to be executed outside the WebView are, and what the links to execute inside it are.

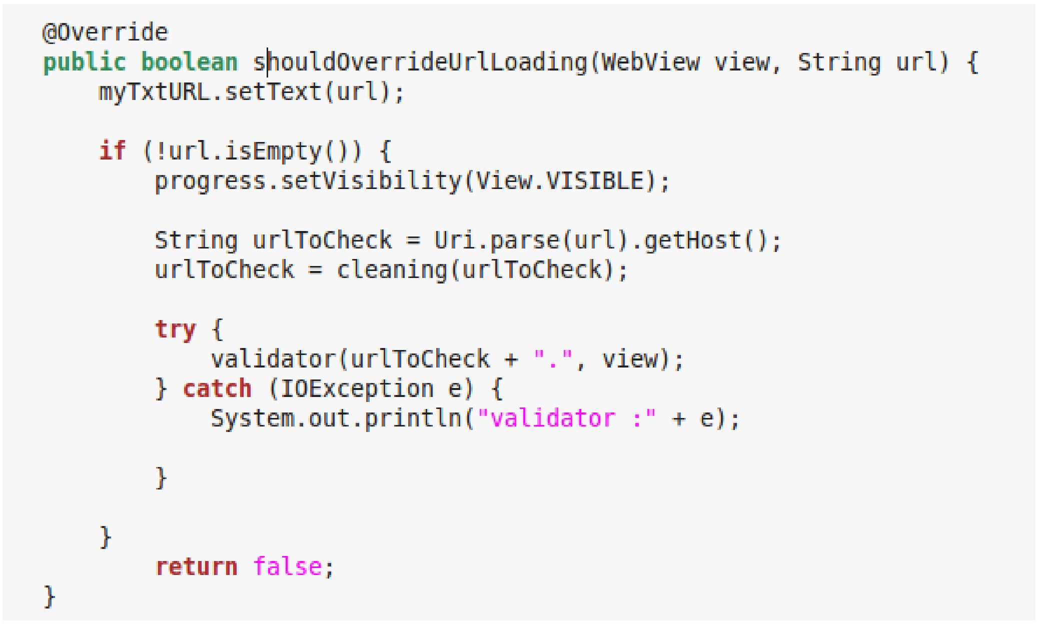
Having this method invoked every time a link is pressed could be used for the purpose of the app. The shouldOverrideUrlLoading method has been modified, returning always false in a way that everything will be executed inside the webview and has been put inside the code to check its validity. First of all, it extrapolates the domain name from the URL that the user wishes to visit. Afterwards, it calls a method, also inside the same class (validator), passing to it the domain to validate Figure 8 .

The validator Figure 9 will validate the domain and according to the results it will show beside the URL a yellow, a red (blocking also the Domain), or a green.

The yellow means that the Domain does not have any DNSSEC key, the red means that the key that the domain has it is not the right one, so it is probably a rogue one or there is a DNSSEC Hijacking attack. The Green means that the domain is correctly validated through DNSSEC. To make the DNSSEC validation as mentioned in the chapter 4.2, a root zone key for the trust anchor is needed and this can be found in the IANA website [57]. It is also important to choose a DNS resolver that handles DNSSEC requests; unrooted Android phones use the DNS resolver that their ISPs offer and not all of them can handle DNSSEC requests. It was decided to use the Google's DNS because, according to the article “Why and How to Use Google’s Public DNS” [58] from blog.dnsimple.com, Google’s DNS has taken measures to protect and prevent common attacks and the resolvers are continually monitored. [58]

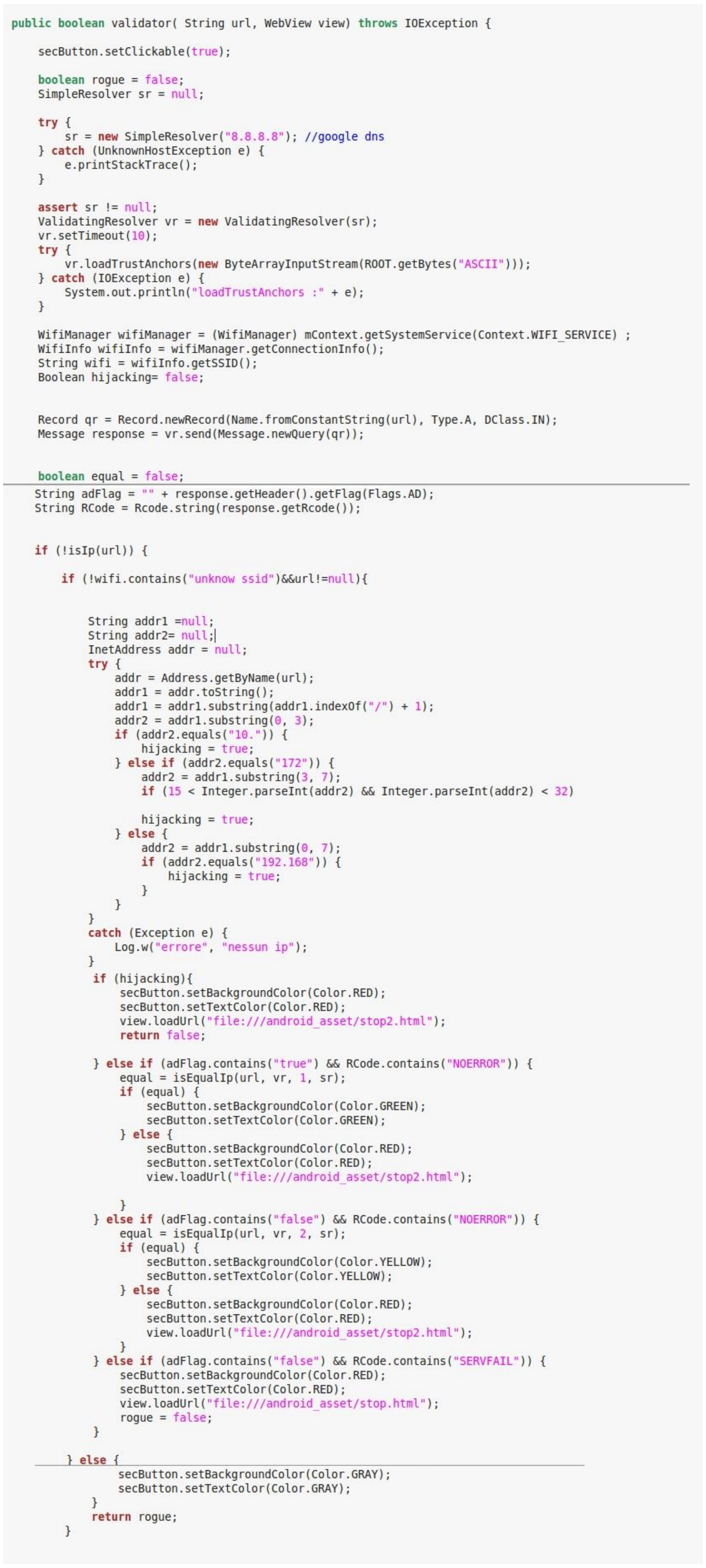
In unrooted devices, it is not possible to change the DNS settings, so that even if the validation can be done from a chosen DNS, the Webview will use the system DNS, for this reason, the app will check if the IPs found in the system DNS are equal to the ones founds in Google’s DNS. This also allows to detect if there is a hijacking attack. This method used to Detect DNS hijacking for DNSSEC Domains cannot be used for the DNS domain because big domains give different answers according to the location of the DNS resolver but they have their own Name Servers and their IP does not change according to DNS resolver. When an attacker spoofs a domain server, he will spoof also the IP of the Name Server of that Domain.

If an attacker using an arp spoofing claims to be the gateway, the methods described above sometime does not work because the two DNS resolver will receive the same answer, for this reason if the IP is a private one the domain will be blocked (it will not stop if a hacker hosts the rogue website on the Internet, but this attack is more difficult and dangerous for the attacker). This means this browser should be used to Browse the Internet and not for intranets. All the techniques described to detect a DNS-Hijacking attack is inside the function isEqualIp Figure 10



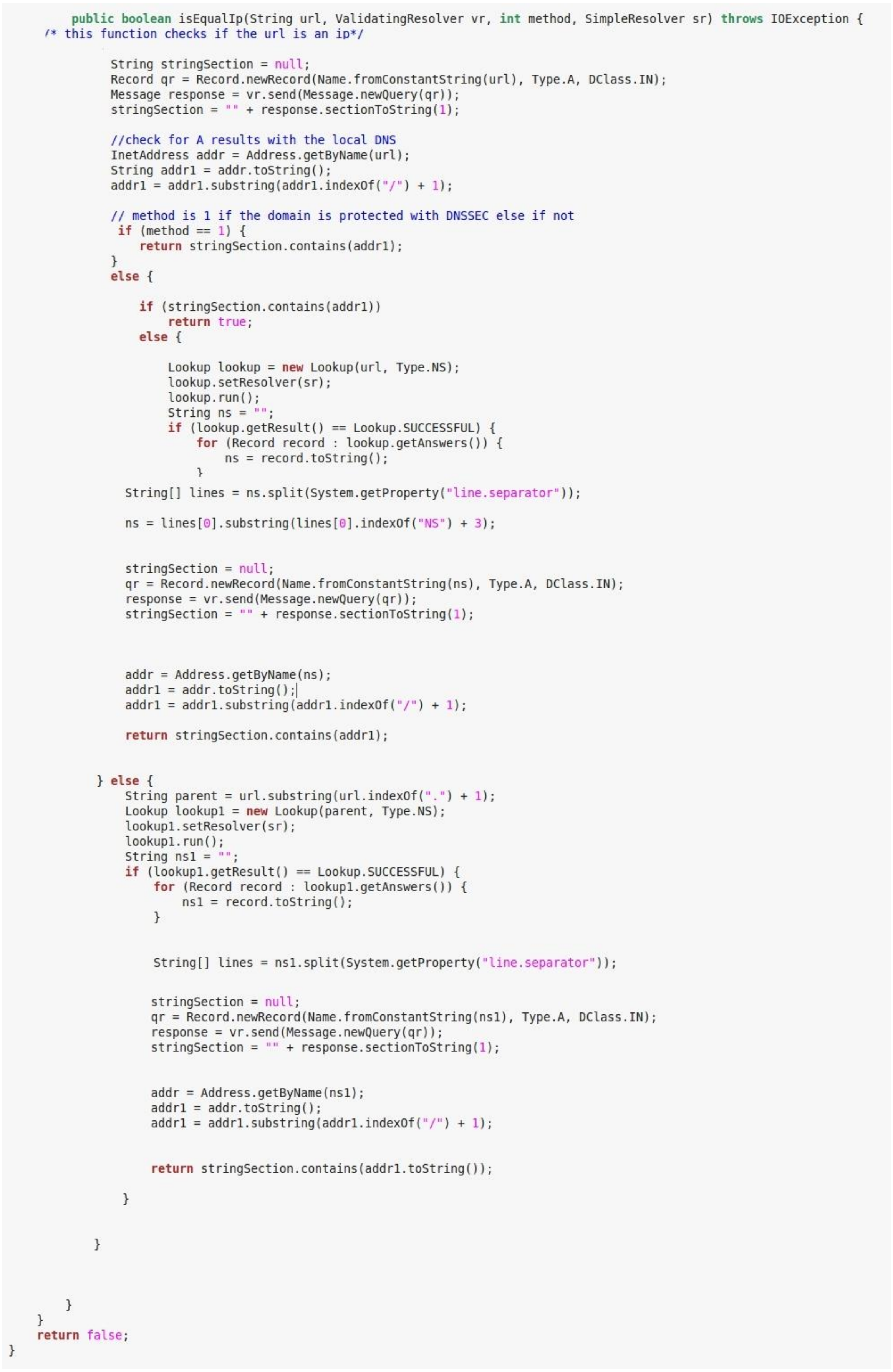
```
@Override
public boolean shouldOverrideUrlLoading(WebView view, String url) {
    myTxtURL.setText(url);

    if (!url.isEmpty()) {
        progress.setVisibility(View.VISIBLE);

        String urlToCheck = Uri.parse(url).getHost();
        urlToCheck = cleaning(urlToCheck);

        try {
            validator(urlToCheck + ".", view);
        } catch (IOException e) {
            System.out.println("validator :" + e);

        }

    }
        return false;
}
```

Figure 8: function shouldOverrideUrlLoading



```java
public boolean validator( String url, WebView view) throws IOException {

    secButton.setClickable(true);

    boolean rogue = false;
    SimpleResolver sr = null;

    try {
        sr = new SimpleResolver("8.8.8.8"); //google dns
    } catch (UnknownHostException e) {
        e.printStackTrace();
    }

    assert sr != null;
    ValidatingResolver vr = new ValidatingResolver(sr);
    vr.setTimeout(10);
    try {
        vr.loadTrustAnchors(new ByteArrayInputStream(ROOT.getBytes("ASCII")));
    } catch (IOException e) {
        System.out.println("loadTrustAnchors :" + e);
    }

    WifiManager wifiManager = (WifiManager) mContext.getSystemService(Context.WIFI_SERVICE) ;
    WifiInfo wifiInfo = wifiManager.getConnectionInfo();
    String wifi = wifiInfo.getSSID();
    Boolean hijacking= false;


    Record qr = Record.newRecord(Name.fromConstantString(url), Type.A, DClass.IN);
    Message response = vr.send(Message.newQuery(qr));


    boolean equal = false;
    String adFlag = "" + response.getHeader().getFlag(Flags.AD);
    String RCode = Rcode.string(response.getRcode());


    if (!isIp(url)) {

        if (!wifi.contains("unknow ssid")&&url!=null){


            String addr1 =null;
            String addr2= null;
            InetAddress addr = null;
            try {
                addr = Address.getByName(url);
                addr1 = addr.toString();
                addr1 = addr1.substring(addr1.indexOf("/") + 1);
                addr2 = addr1.substring(0, 3);
                if (addr2.equals("10.")) {
                    hijacking = true;
                } else if (addr2.equals("172")) {
                    addr2 = addr1.substring(3, 7);
                    if (15 < Integer.parseInt(addr2) && Integer.parseInt(addr2) < 32)

                    hijacking = true;
                } else {
                    addr2 = addr1.substring(0, 7);
                    if (addr2.equals("192.168")) {
                        hijacking = true;
                    }
                }
            }
            catch (Exception e) {
                Log.w("errore", "nessun ip");
            }
             if (hijacking){
                 secButton.setBackgroundColor(Color.RED);
                 secButton.setTextColor(Color.RED);
                 view.loadUrl("file:///android_asset/stop2.html");
                 return false;

             } else if (adFlag.contains("true") && RCode.contains("NOERROR")) {
                 equal = isEqualIp(url, vr, 1, sr);
                 if (equal) {
                     secButton.setBackgroundColor(Color.GREEN);
                     secButton.setTextColor(Color.GREEN);
                 } else {
                     secButton.setBackgroundColor(Color.RED);
                     secButton.setTextColor(Color.RED);
                     view.loadUrl("file:///android_asset/stop2.html");

                 }
             } else if (adFlag.contains("false") && RCode.contains("NOERROR")) {
                 equal = isEqualIp(url, vr, 2, sr);
                 if (equal) {
                     secButton.setBackgroundColor(Color.YELLOW);
                     secButton.setTextColor(Color.YELLOW);
                 } else {
                     secButton.setBackgroundColor(Color.RED);
                     secButton.setTextColor(Color.RED);
                     view.loadUrl("file:///android_asset/stop2.html");
                 }
             } else if (adFlag.contains("false") && RCode.contains("SERVFAIL")) {
                 secButton.setBackgroundColor(Color.RED);
                 secButton.setTextColor(Color.RED);
                 view.loadUrl("file:///android_asset/stop.html");
                 rogue = false;
             }

        } else {
                secButton.setBackgroundColor(Color.GRAY);
                secButton.setTextColor(Color.GRAY);
            }
            return rogue;
        }
```

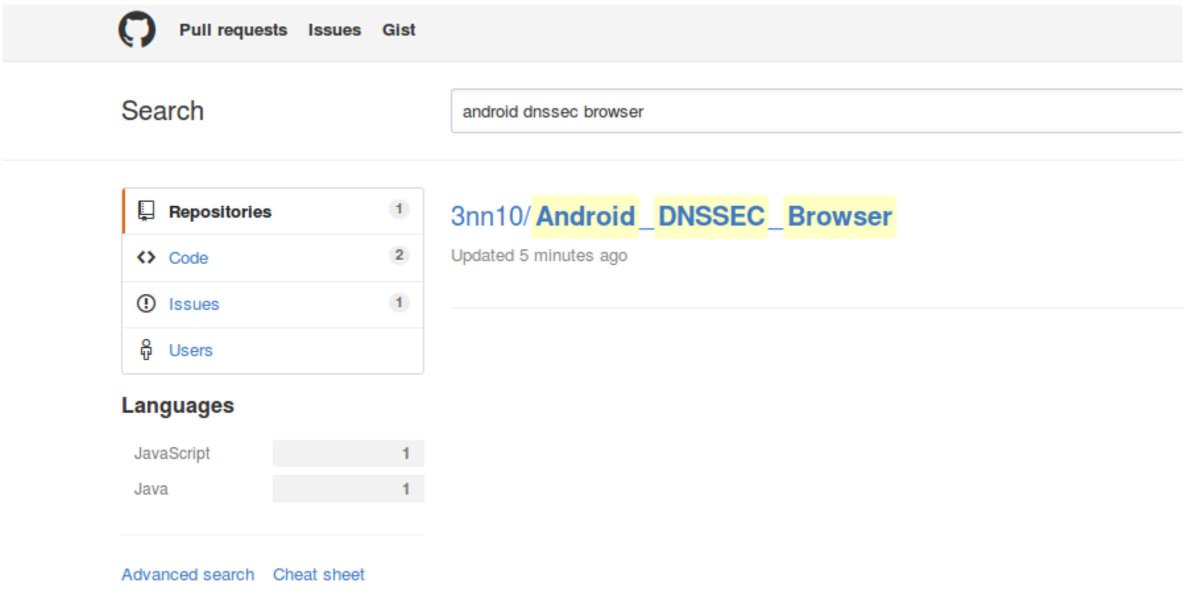
Figure 9: Function validator



```
         public boolean isEqualIp(String url, ValidatingResolver vr, int method, SimpleResolver sr) throws IOException {
     /* this function checks if the url is an ip*/

              String stringSection = null;
              Record qr = Record.newRecord(Name.fromConstantString(url), Type.A, DClass.IN);
              Message response = vr.send(Message.newQuery(qr));
              stringSection = "" + response.sectionToString(1);

              //check for A results with the local DNS
              InetAddress addr = Address.getByName(url);
              String addr1 = addr.toString();
              addr1 = addr1.substring(addr1.indexOf("/") + 1);

              // method is 1 if the domain is protected with DNSSEC else if not
               if (method == 1) {
                  return stringSection.contains(addr1);
              }
              else {

                  if (stringSection.contains(addr1))
                      return true;
                  else {

                      Lookup lookup = new Lookup(url, Type.NS);
                      lookup.setResolver(sr);
                      lookup.run();
                      String ns = "";
                      if (lookup.getResult() == Lookup.SUCCESSFUL) {
                          for (Record record : lookup.getAnswers()) {
                              ns = record.toString();
                          }
                String[] lines = ns.split(System.getProperty("line.separator"));

                ns = lines[0].substring(lines[0].indexOf("NS") + 3);


                stringSection = null;
                qr = Record.newRecord(Name.fromConstantString(ns), Type.A, DClass.IN);
                response = vr.send(Message.newQuery(qr));
                stringSection = "" + response.sectionToString(1);



                addr = Address.getByName(ns);
                addr1 = addr.toString();
                addr1 = addr1.substring(addr1.indexOf("/") + 1);

                return stringSection.contains(addr1);


            } else {
                String parent = url.substring(url.indexOf(".") + 1);
                Lookup lookup1 = new Lookup(parent, Type.NS);
                lookup1.setResolver(sr);
                lookup1.run();
                String ns1 = "";
                if (lookup1.getResult() == Lookup.SUCCESSFUL) {
                    for (Record record : lookup1.getAnswers()) {
                        ns1 = record.toString();
                    }


                    String[] lines = ns1.split(System.getProperty("line.separator"));


                    stringSection = null;
                    qr = Record.newRecord(Name.fromConstantString(ns1), Type.A, DClass.IN);
                    response = vr.send(Message.newQuery(qr));
                    stringSection = "" + response.sectionToString(1);


                    addr = Address.getByName(ns1);
                    addr1 = addr.toString();
                    addr1 = addr1.substring(addr1.indexOf("/") + 1);


                    return stringSection.contains(addr1.toString());

                }


            }



        }
    }
    return false;
}
```

Figure 10: Function isEqualIp

The codes have been uploaded on github Figure 11



Pull requests Issues Gist

Search

android dnssec browser

3nn10/Android_DNSSEC_Browser

Updated 5 minutes ago

Repositories	1
Code	2
Issues	1
Users	

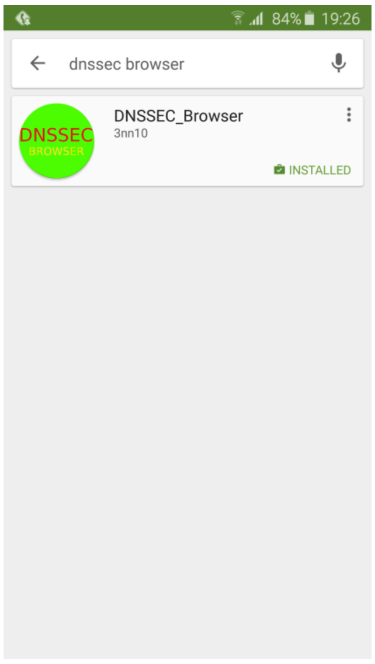
Languages

JavaScript	1
Java	1

Advanced search Cheat sheet

Figure 11: https:// github.com/3nn10/Android_DNSSEC_Browser

And the app can be downloaded for free from the Google Playstore. Figure 12



84% 19:26

dnssec browser

DNSSEC_Browser
3nn10
INSTALLED

Figure 12 Google Playstore

5.2. App Usage

The usage of the app is really simple: the user puts the URL that he wants to visit and presses the search button, and the app will say if the URL is secure or not according to its DNSSEC key. The button at the left of the URL will be yellow, green or red according to the cases; if the user presses it, a popup would state information about the domain. The process also works if the user presses on links or pages opened from search engines.

Now some examples will be reported:

- In case the Domain name is not signed by DNSSEC: Figure 13 Figure 14

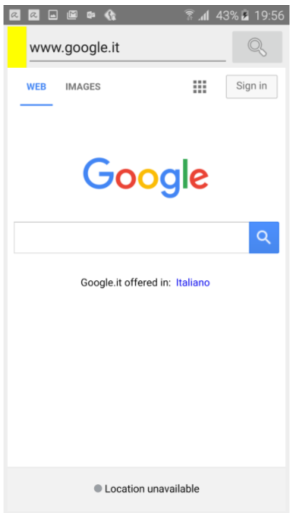
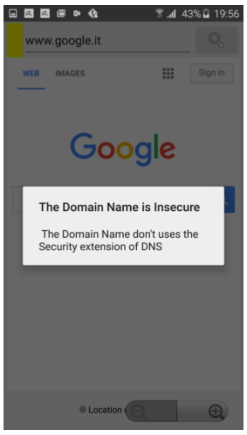


Figure 13: Domain Name is not signed with DNSSEC

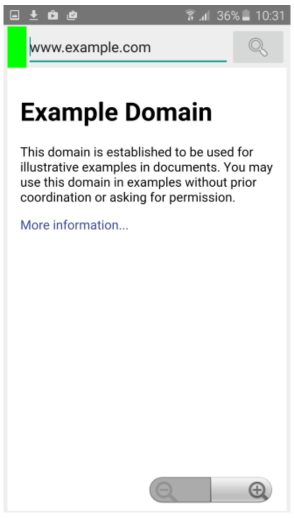


The Domain Name is Insecure

The Domain Name don't uses the Security extension of DNS

Figure 14: Domain Name is not signed with DNSSEC

- In Case the Domain Name is correctly Signed by DNSSEC: Figure 15 Figure 16

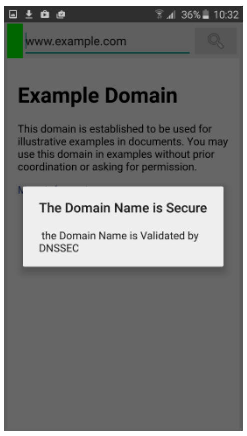


Example Domain

This domain is established to be used for illustrative examples in documents. You may use this domain in examples without prior coordination or asking for permission.

More information...

Figure 15:Domain Correctly Signed



Example Domain

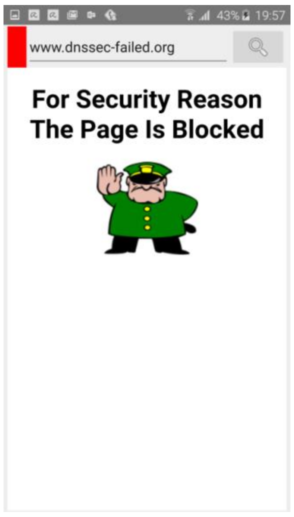
This domain is established to be used for illustrative examples in documents. You may use this domain in examples without prior coordination or asking for permission.

The Domain Name is Secure

the Domain Name is Validated by DNSSEC

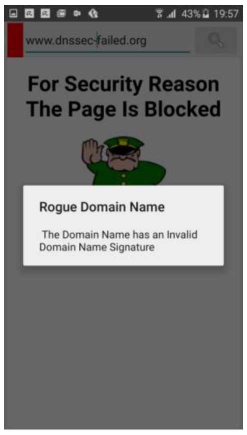
Figure 16: Domain Correctly Signed

- In case it is signed by DNSSEC, but the key is wrong: Figure 17 Figure 18


www.dnssec-failed.org

For Security Reason
The Page Is Blocked

Figure 17: Rogue Domain


www.dnssec-failed.org

For Security Reason
The Page Is Blocked

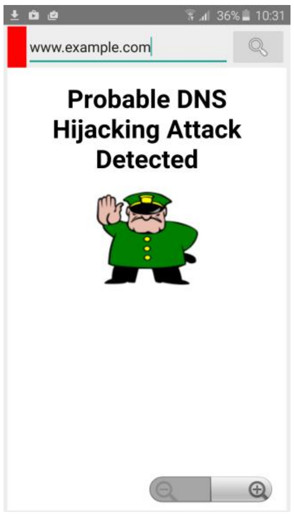
Rogue Domain Name

The Domain Name has an Invalid Domain Name Signature

Figure 18: Rogue Domain

In spite of the add-ons for Firefox or Chrome, it has been decided to completely block the page and not only to alert the user of the risks, because it is what DNSSEC-aware resolvers do.

- Case DNS-hijacking Figure 19



Probable DNS Hijacking Attack Detected

Figure 19:DNS-hijacking

6. Conclusion

DNSSEC, despite not being fully implemented at present, seems a sensible option for the future. It is clear that there is an international effort to use DNSSEC to secure the DNS protocol. There has been a sudden rise in the signing of TLDs from 2013; today, about 80% of TLDs are signed, and this means that 2LDs domains could also start signing their domains in the coming years [59].

Since January 2014, the Estonian TLD (.ee) has begun to be signed by DNSSEC, and it is validated by some Estonian Internet Service Providers like Elion, EMT, Infonet and STV. It has been less than a year since Estonia started DNSSEC signing, and so, only a few domains have been signed, but since this country has always been innovative in computer science and in addition, cares a lot about IT Security since the cyber war that happened in 2007, it is probable that in the following years, all sensible Estonian domains will be signed with the DNSSEC protocol. This is yet to happen for extremely important Estonian domains like Valimised [61], which provides the Internet voting service and wk.ee, which shows the voting results. [60]

Sweden has been one of the first countries to validate its TLD (.se) [65] and as a consequence, it has five banks signed with DNSSEC (alandsbanken.se, handelsbanken.se, lansforsakgringar.se, marginalen.se, sparbankenvm.se). [64]

6.1.Reasons to use Today the DNSSEC Browser

Using this app is the only possibility thus far to securely browse DNSSEC domains from Android devices. Even if there are few domains today signed with DNSSEC, some, such as Valimised, are extremely important.

In addition, the app is able to identify a DNS hijacking attack inside a LAN, and also if the domain is not secured by DNSSEC, but is not able to identify if the answer that the ISP gives is the right one; this can only be done if the Domain uses the DNSSEC protocol.

6.2.Summary

The first part of this research work presents all the threats in the DNS protocol, giving examples of real cases and statistics. Identifying the threats has established that they are really dangerous and could do huge damage in a society based on Internet services.

In the second part, three valid alternatives to the DNS protocol have been analyzed, identifying what they solve, their weaknesses, and their side effects. DNSSEC, in spite of having a lot of side effects, is the only protocol that is really doing what it claims.

In the third part, a contribution to the DNSSEC world is described, which involves making a browser that validates DNSSEC domains through Android devices. The scope of this browser is to stop DNS-hijacks in local networks, but also to stop DNS-hijacks if the Android device is connected to the Internet through a non-DNSSEC validated Internet Service Provider. An android user without a DNSSEC browser cannot know if the DNS query has been answered with a valid DNSSEC key or not. Another contribution that this app could have is to make users aware of the security that DNSSEC brings in a way that more users would ask for better security, forcing the websites that contain sensible information to sign their domains.

6.3.Future work

This kind of Browser does not exist yet in iOS and Windows phones, and therefore, the code of this app should be translated in Objective-C and C# to work with the other two operating systems.

A functionality that the browser should have if the web pages will start to adopt the DANE protocol is the DANE verification.

The app could be integrated with new functionalities and updated.

When most of the well-known websites will use DNSSEC, verification will have to be considered to solve the problem for big domains that their DNS give answers according to the location of the host. It could also be considered to add a high security option by not accepting all normal DNS answers.

7. Bibliography

[1] Components -E-Estonia
url: https://e-estonia.com/components/

[2] Smartphone now most popular way to browse internet – Ofcom report
url: http://www.theguardian.com/technology/2015/aug/06/smartphones-most-popular-way-to-browse-internet-ofcom

[3] Issues in DNS Security
url: http://www.windowsecurity.com/articles-tutorials/misc_network_security/DNS-Security-Part-1.html

[4] What is Cyberwarfare?-Definition from WhatIs.com
url: http://searchsecurity.techtarget.com/definition/cyberwarfare

[5] Cyber War Example- US Cyber War
url: https://sites.google.com/site/uscyberwar/cyber-war-example

[6] RFC791 Internet Protocol
url: https://tools.ietf.org/html/rfc791

[7] RFC2460 Internet Protocol, Version 6 (IPv6) Specifications
url: https://tools.ietf.org/html/rfc2460

[8] File:Ipv4_address.svg -Wikimedia Commons
url: https://commons.wikimedia.org/wiki/File:Ipv4_address.svg

[9] File:Ipv6_address.svg -Wikimedia Commons
url: https://commons.wikimedia.org/wiki/File:Ipv6_address.svg

[10] Tree-Structure.jpg
url: http://chihealer.com/wp-content/uploads/2015/04/Tree-Stucture.jpg

[11] Wale Soyinka “Linux Administration A Beginner's Guide”, chapter 16

[12] DNS AND DNSSECPROBLEMS AND SOLUTIONS PRESENTATION
url:
http://toshendra.com/downloads/DNS%20&%20DNSSEC%20Problems%20&%20Solutions%20presentation.pdf

[13] Michael W Lucas: “DNSSEC Mastery – Securing the Domain Name System with BIND (IT Mastery Book 2)“

[14] DNSSEC course
url: http://www.dnsseccourse.nl/en/player.html

[15] DNS for Rocket Scientists
url: http://www.zytrax.com/books/dns/
[16] Threat Analysis of the Domain Name System (DNS)
https://tools.ietf.org/html/rfc3833
[17] Chache Poisoning-Owasp
url: https://www.owasp.org/index.php/Cache_Poisoning
[18] Distributed Denial of Service (DDos) attack
url: http://blog.ddos-guard.ir/distributed-denial-service-ddos-attack/
[19] DNSSEC and Amplification Attacks
url: https://technet.microsoft.com/en-us/security/hh972393.aspx?f=255&MSPPError=-2147217396
[20] Probable Cache Poisoning of Mail Handling Domains
url: https://www.cert.org/blogs/certcc/post.cfm?EntryID=206
[21] DNS Hacking (Beginner to Advanced)-InfoSec Institute
url: http://resources.infosecinstitute.com/dns-hacking/
[22] Encrypt Your Gmail, Hotmail, And Other Webmail: Here’s How
url: http://www.makeuseof.com/tag/encrypt-your-gmail-hotmail-and-other-webmail-heres-how/
[23] Exfiltrating Data From MS SQL Server Via DNS – pentestmonkey
url: http://pentestmonkey.net/blog/mssql-dns
[24] DNS Tunneling: Is It a Security Threat?
url:
http://www.circleid.com/posts/20131030_dns_tunneling_is_it_a_security_threat/
[25] Cloudmark Press Relase: New Research Reveals More than Three Quarters of Organizations Have Suffered a DNS Attack
url: https://www.cloudmark.com/en/press/new-research-reveals-more-than-three-quarters-of-organizations-have-suffered-a-dns-attack
[26] DNS Hijacking: What is it and how it works
url: http://www.gohacking.com/dns-hijacking/
[27] Nearly 50 Percent of Organizations Hit With DNS Attack in Last 12 Months: Survey
url: http://www.securityweek.com/nearly-50-percent-organizations-hit-dns-attack-last-12-months-survey

[28] DdoS Response Playbook
url: http://lp.incapsula.com/rs/incapsulainc/images/DDoSResponse.pdf

[29] A Team Cymru EIS Report: Growing Exploitation of Small Office Routers Creating Serious Risks
url: https://www.team-

[30] CDN, Website Security, DDoS protection, Load balancing | Incapsula
url: https://www.incapsula.com/

[31] Cyber Attackers Increasingly Sneaking Corporate Data Out Through DNS url: http://www.eweek.com/security/cyber-attackers-increasingly-sneaking-corporate-data-out-through-dns.html

[32] Ad-Fraud Malware Hijacks Router DNS – Injects Ads Via Google Analytics
url:http://sentrant.com/2015/03/25/ad-fraud-malware-hijacks-router-dns-injects-ads-via-google-analytics/

[33] DNSChanger
url: http://krebsonsecurity.com/tag/dnschanger/

[34] Hacking Team' Gets Hacked! 500GB of Data Dumped Over the Internet
url: http://thehackernews.com/2015/07/Italian-hacking-team-software.html

[35] DNS attacks putting organizations at risk, survey finds
url: http://www.scmagazine.com/ddos-attacks-mask-crime/article/389815/

[36] Simple DNS Plus
url: http://www.simpledns.com/help/v52/index.html?rec_dnskey.htm

[37] DNSSEC Course
url: http://www.dnsseccourse.nl/en/player.html

[38] Comodo Report Incident - Comodo detected and thwarted an intrusion on 26 MAR 2011
url: https://www.comodo.com/Comodo-Fraud-Incident-2011-03-23.html

[39] RFC 6698 - The DNS Based Authentication of Named Entities (DANE) Transport Layer Security (TLS) Protocol: TLSA
url: https://tools.ietf.org/html/rfc6698

[40] Tutorial on DANE and DNSSEC
url: https://www.youtube.com/watch?v=BhvU19RJrPY

[41] “ UNIX and Linux System Administration Handbook” Authors: Evi Nemeth, Garth Snyder, Trent R. Hein

[42] “DNS AND DNSSECPROBLEMS AND SOLUTIONS” Author: Toshendra Sharma Wegilant Net Solutions Pvt. Ltd.
url:
http://toshendra.com/downloads/DNS%20&%20DNSSEC%20Problems%20&%20Solutions%20presentation.pdf

[43] Iternet Assigned Numbers Authority
url: http://www.iana.org/

[44] Malware Next Move:DNS
url: http://www.isssource.com/malwares-next-move-dns/

[45] SHAPING DNS SECURITY WITH CURVES
url: http://curvedns.on2it.net/thesis/shaping_dns_security_with_curves.pdf

[46] An overview of secure name resolution [29c3]
url:
https://www.youtube.com/watch?v=eOGezLjlzFU&feature=youtu.be&t=52m26s

[47] rfc4034
url:https://tools.ietf.org/html/rfc4034#section-6.1

[48] DNSSEC a Review
url:http://www.potaroo.net/ispcol/2010-06/dnssec.html

[49] rfc3833
url: https://tools.ietf.org/html/rfc3833

[50] rfc7129
url: https://tools.ietf.org/html/rfc7129

[51] DANE & Application Uses of DNSSEC
url: http://meetings.internet2.edu/media/medialibrary/2014/11/06/20141029-huque-dnssec-

[52] This POODLE Bites: Exploiting The SSL 3.0 Fallback
url :https://www.openssl.org/~bodo/ssl-poodle.pdf

[53] DNSCrypt for Android
https://dnscrypt.org/#dnscrypt-android

[54] DNSCrypt for iOS
url: https://dnscrypt.org/#dnscrypt-ios

[55] dnssecjava
url: https://github.com/ibauersachs/dnssecjava

[56] dnsjava

url: http://www.dnsjava.org/

[57] DNSSEC Trust Anchor Publication for the Root Zone

url: http://data.iana.org/root-anchors/draft-icann-dnssec-trust-anchor.html

[58] Why and How to Use Google's Public DNS
Url: http://blog.dnsimple.com/2015/03/why-and-how-to-use-googles-public-dns/

[59] DNSSEC Deployment Report
url: http://rick.eng.br/dnssecstat/

[60] DNSSEC IN ESTONIA

url:http://www.internet.ee/dnssec-en

[61] Elektrooniline hääletamine

url: https://www.valimised.ee/eng/

[62] Hearthbleed

url: http://heartbleed.com/

[63] Namecoin

url: http://www.net.in.tum.de/fileadmin/TUM/NET/NET-2014-08-1/NET-2014-08-1_14.pdf

[64] Banks with DNSSEC

url: http://dnssecandipv6.se/bankdns/

[65] DNSSEC – The Path to a Secure Domain

url: https://www.iis.se/english/domains/tech/dnssec/

[66] The DANE Protocol – DNS-Based Authentication of Named Entities

url: http://www.internetsociety.org/deploy360/resources/dane/

All Listed references were last checked and working on December 29th , 2015