

Attack on WiFi-based Location Services and SSL Using Proxy Servers

by

Jun Liang Feng

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Applied Science
in
Computer Engineering

Waterloo, Ontario, Canada, 2013

© Jun Liang Feng 2013

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Wireless LANs, or WiFi, are very common in any household or business today. It allows access to their home or business network and the Internet without using wires. Their wireless nature allows mobility and convenience for the user and that opens up a lot of new possibilities in mobile devices such as smartphones and tablets. One application that makes use of wireless LANs is positioning, which can be used in areas where Global Positioning Systems may have trouble functioning or not at all. However, a drawback of using wireless communication is that it is susceptible to eavesdropping and jamming. Once the wireless signal is jammed, an attacker can set up fake access points on different channels or frequencies to impersonate a legitimate access point. In this thesis, this attack is performed specifically to trick WiFi-based location services that are becoming very popular nowadays, especially for mobile devices. It will be shown that the attack works on Skyhook, Google, Apple and Microsoft location services, four of the major location service providers, and works on dual-band hardware. Some countermeasures to such an attack are also presented.

The web is an important part of many people's lives nowadays. People expect that their privacy and confidentiality is preserved when they use the web. Previously, web traffic uses HTTP which meant traffic is all unencrypted so most information is sent in plaintext, which an attacker can intercept and read. This is clearly a security problem so many websites now default to using a more secure protocol, namely HTTPS which uses HTTP with SSL, and will redirect you to the more secure version if you attempt to connect using the non-SSL version of the website. SSL works by exchanging keys between the client and server and the actual data is protected using the key and the cipher suite that is negotiated between the two. However, if a network uses a proxy server, it works slightly different. The SSL connection is broken up into two separate ones and that creates the potential for man-in-the-middle attacks that allow an attacker to intercept the data being transmitted. This thesis analyzes several scenarios in which an adversary can conduct such a man-in-the-middle attack, and potential detection and mitigation methods.

Acknowledgements

I would like to thank my supervisor Professor Guang Gong for her mentorship. She is one of the smartest and kindest person I know and has helped me countless times throughout my Master's career. I would also like to thank everyone in the Communication Security (Comsec) group at the University of Waterloo led by Professor Guang Gong for their friendship, and especially to former member and friend Khizer Kaleem for his help and expertise while I was working on my research, and Kalikinkar Mandal for his help. In addition, I would like to thank my girlfriend Ping He for her love and support while I was completing my Master's degree. Last but not least, I would like to thank my parents for their unconditional love and taking care of me all these years.

Dedication

This thesis is dedicated to my girlfriend and my parents.

Table of Contents

List of Tables	x
List of Figures	xi
1 Introduction	1
1.1 Attack on WiFi-based Location Services Using Dual-band Hardware	2
1.2 Attack on SSL Using Proxy Servers	3
1.3 Outline and Contributions	3
2 Preliminaries	5
2.1 Background	6
2.1.1 Network Infrastructure	6
2.1.2 Optional Network Infrastructure Components	10
2.1.3 SSL	13
2.1.4 IEEE 802.11	15
2.1.5 WiFi-based Location Services	19
2.2 Related Works on WiFi-based Location Services	20
2.2.1 Attack on WLAN-based Positioning Systems on a Single Band	20
2.2.2 Attack on WLAN-based Positioning Systems Without Jamming	20
2.2.3 Portable GPS Spoofer	21
2.2.4 Securing Localization with Hidden and Mobile Base Stations	21

2.2.5	Using Localization to Enhance Security and Services in WiFi Networks	22
2.2.6	Denial of Service Attacks on 802.11 Networks	23
2.2.7	Mitigating Attacks Against Signal Strength-based Localization Algorithms	23
2.3	Related Works on SSL and Proxy Servers	24
2.3.1	Practical Attacks Against SSL	24
2.3.2	A Practical Man-in-the-middle Attack Against SSL	24
2.3.3	Issues with the Certificate Authority Model	25
2.3.4	Implementation Issue in SSL-enabled Applications Opens Applications to Attack	26
2.3.5	Man-in-the-middle Attack on WiFi Clients	26
2.3.6	SSL Session-aware Authentication	27
2.3.7	Lucky Thirteen - A Timing Attack Against SSL	27
2.3.8	Bias in RC4 Used in SSL	28
2.3.9	Attack Against Tor Network Using HTTP-based Application	29
3	Attack on WiFi-based Location Services Using Dual-band Hardware	30
3.1	Vulnerability Analysis of Location Services Providers	31
3.2	Adversary Model and Equipment	36
3.3	Implementation of Attack on WiFi-based Location Services	38
3.3.1	Attack procedure	38
3.3.2	Results of the Attack	39
3.4	Impact of Attack on Location Services	42
3.4.1	Photo Geotagging	42
3.4.2	Social Network Geotagging	43
3.4.3	Location-based Weather	43
3.4.4	Locating Friends	43
3.4.5	Device Recovery Services	43

3.4.6	Location-based Wallet Services	44
3.4.7	Location-based Reminders	44
3.4.8	Search	44
3.4.9	Bluetooth and Bluetooth-based Location Service	45
3.5	Impact of Attack on WiFi Infrastructure	45
3.6	Countermeasures to Attack	48
3.6.1	Use multiple bands	48
3.6.2	Use multiple location service technologies	49
3.6.3	Fingerprinting	50
3.6.4	Summary of countermeasures	50
3.7	Summary of Attack on WiFi-based Location Services	51
4	Attack on SSL Using Proxy Servers	53
4.1	Internet Proxy Servers	55
4.1.1	Attack Implementation	56
4.1.2	Attack Procedure	56
4.1.3	Notes on Attack	56
4.2	Corporate Proxy Servers	57
4.2.1	Attack Implementation	57
4.2.2	Attack Procedure	58
4.2.3	Notes on Attack	58
4.3	Wireless Hotspots	62
4.3.1	Attack Implementation	62
4.3.2	Attack Procedure	63
4.3.3	Notes on Attack	63
4.4	Captured Data	66
4.5	Detection and Mitigation of Attack on SSL Using Proxy Servers	72
4.5.1	Detection	72
4.5.2	Mitigation	74
4.6	Summary of Attack on SSL Using Proxy Servers	74

5	Conclusions and Future Work	76
5.1	Contributions	76
5.2	Future Work	78
5.2.1	Extending Attack on WiFi-based Location Services	78
5.2.2	Extending Attack on SSL Using Proxy Servers	79
	References	80

List of Tables

3.1 Summary of countermeasures	51
--	----

List of Figures

2.1	Network infrastructure	7
2.2	Proxy server in network infrastructure	11
2.3	SSL Protocol	14
2.4	All channels in the 2.4 GHz band [22]	15
2.5	Non-overlapping channels in the 2.4 GHz band [22]	16
2.6	WiFi channels for the 5 GHz band [33]	16
2.7	Location determination process	19
3.1	Skyhook location service query	33
3.2	Google location service query	34
3.3	Apple location service query	35
3.4	Hardware	37
3.5	Location services attack procedure	39
3.6	Access point impersonation attack on Google location service	40
3.7	Access point impersonation attack on Skyhook location service	40
3.8	Access point impersonation attack on Apple location service	41
3.9	Access point impersonation attack on Microsoft location service	41
3.10	Scan for nearby access points	49
4.1	Using SSL with proxy server	54
4.2	Setting proxy settings in Group Policy	59

4.3	Disabling changing proxy settings in Group Policy	60
4.4	Trusting proxy server certificate for entire domain	61
4.5	Proxy configuration file	63
4.6	DHCP option to distribute proxy configuration file	64
4.7	Firewall rules to only allow traffic to/from proxy	64
4.8	Captive portal configuration	64
4.9	Facebook login	67
4.10	Twitter login	68
4.11	LinkedIn login	69
4.12	Google login	70
4.13	Live login	71
4.14	iCloud login	72
4.15	CIBC login	73

Chapter 1

Introduction

Hardware and software have evolved a lot, even in the past few years. The cost have come down a lot too, making them more accessible to more people. Many people now own several devices, including laptops, smartphones and tablets and they are carrying them wherever they go. Meanwhile, companies are looking to take advantage of this to collect information on users. Users' privacy clearly becomes an issue. In addition, this type of information may be stored on a user's device and/or sent to the company's servers for processing. The nature of the data collected gives any person with malicious intent a big motivation to get that data. An adversary can steal the device physically or intercept the communications. For some attackers, simply denying the use of the device would cause enough damage to be worthwhile.

A service that has become very popular now that mobile devices are very common is location service. As the name suggests, the service allows a user's device to get its location. This opens up many new applications that were not possible before. Now, people can search for places to go near them, set reminders if they enter or leave a specific area, or track the location of their device if it is lost, and these only scratch the surface of what developers have come up with already. However, the growing popularity of this service means attackers are gaining interest as well. This thesis talks about an attack on WiFi-based location services using dual-band hardware and Section 1.1 gives a brief introduction on the work.

A security protocol that is very commonly used nowadays is SSL. SSL stands for Secure Sockets Layer and was first introduced in the 1990s. It provides secure end-to-end connections between a client and server that wish to communicate with each other. SSL is used per-application and many applications have adopted SSL for secure connections. Proxy

servers are an intermediary between a client and the server, which means it sits between the client and server. They are more often used in large networks to save bandwidth costs and increase security. Combining SSL and proxy servers means combining two contradicting elements. SSL provides end-to-end connections while proxy servers break them up. This opens up opportunities for an attacker to take advantage of to gain access to users' communications. The second work of this thesis talks about how an attacker can use this fact to attack a network and Section 1.2 gives a brief introduction to the work.

1.1 Attack on WiFi-based Location Services Using Dual-band Hardware

There are many devices that have wireless capabilities on the market today. Arguably the most common wireless technology in use is wireless LAN, more commonly known as WiFi. WiFi is the wireless version of traditional wired Ethernet LANs and it allows people to do more or less the same things, except without wires. It allows people to access network resources without being chained to a network cable. With traditional wired networks, cables need to be run to every place that needs access to the network. With wireless networks, people can take their devices anywhere where there is a signal and be able to use network resources. This has resulted in the creation of new types of devices, namely the smartphones and tablets that are very common today. New types of devices also bring new types of applications and services, and one of the more popular services that have been created as a result of wireless networking is location services.

Traditional location services were accomplished using GPS technology. GPS relies on the use of satellites that orbit the Earth. One of the many limitations with such system is that it does not work indoors and alternative methods had to be created. One of them was WiFi-based location services. The system relies on the information provided by broadcasting WiFi access points. Since access points are not likely to move very often, they are used as anchors for which the location is calculated.

However, as described by [69], one location service provider that uses this type of system is subject to attack. An attacker can jam existing legitimate access points and set up fake access points to broadcast attacker-chosen information. This results in inaccurate access point information sent by the client and the location service provider in turn will provide inaccurate location information.

The work in this thesis extends the work in [69] by attacking four major location service providers. The implementation is an alternative that is easier to perform and costs less

and thus is accessible to more people. An analysis on the impact on location services and WiFi infrastructure in general is also performed.

1.2 Attack on SSL Using Proxy Servers

Secure Sockets Layer (SSL) is one of the most popular security protocols today. It was invented by Netscape in the 1990s and has gone through numerous revisions since its inception and is now called Transport Layer Security (TLS). SSL was created as a result of the need for a client to verify a server's identity. Client's identity is usually verified using a username and password so it is not a requirement of SSL. When a client wants to communicate with a server via SSL, it would initiate an SSL connection to the server and start the handshake process. The handshake involves authenticating the server, establishing keys and choosing the cipher suite. Once the handshake is complete, the server and client has agreed on a key and cipher suite to be used for protecting the communication and they can now communicate in a secure manner. This connection is end to end since a client and the destination server perform the handshake.

Proxy servers serve as an intermediary between a client and a server. All communications between the client and server pass through the server. The proxy server breaks up the connection between a client and a server into two separate ones: one between the client and the proxy server and one between the proxy server and the destination server. Proxy servers are not what one typically finds on a network but more and more people are using it nowadays for personal reasons like getting around restrictions imposed by a company network.

The purpose of SSL and proxy servers are in contradiction to each other. SSL provides an end-to-end connection while proxy servers break up end-to-end connections. Combined together, an attacker can use a proxy server to perform a man-in-the-middle attack on SSL. In this thesis, an analysis on some scenarios where an attacker can perform such an attack is analyzed and some detection and mitigation methods are given.

1.3 Outline and Contributions

The remainder of the thesis is divided into two parts, one concerning the attack on WiFi-based location services and one concerning the attack on SSL using proxy servers. The outline of the thesis and contributions are as follows:

- **Chapter 2** gives some background on some of the technologies that are relevant to this thesis. A literature survey of some of the related works to this thesis is also performed.
- **Chapter 3** talks about the attack on WiFi-based location services. The chapter starts off with a brief introduction and related works, then moves on to an analysis of four of the major location service providers on the market today. The adversary model (attack environment) and equipment used in the experiment are described next, followed by the implementation details and the results of the attack. To the author's knowledge, this is the first time four location service providers are attacked together and results gathered in a single document. The attack is easier to set up and more powerful due to the fact that people with less technical knowledge can easily get the equipment at a fairly low cost and perform the same attack and the cost and effort of the attack decreases dramatically for those with just a little more technical knowledge. An analysis of the impact of the attack on location services and WiFi infrastructure is performed and several countermeasures are proposed. Finally, a summary of the chapter and conclusion are given.
- **Chapter 4** talks about the attack on SSL using proxy servers. The chapter starts off with a brief description of the problem and related works. Then some potential scenarios for launching an attack are analyzed, and the implementation details of an attack is given for some of those attack scenarios. It shows that proxy servers are a very powerful tool in intercepting and extracting the plaintext from SSL traffic and can be used for malicious purposes. Some of the attack scenarios require user interaction but others do not, which makes the attack a very big threat to the privacy and confidentiality of user's data, especially when users believe that the lock icon in many browsers lead them to believe that their data is safe from adversaries. Next, the chapter talks about some ways to detect and mitigate this type of attack. Finally, a summary of the chapter and conclusion are given.
- **Chapter 5** provides a summary and conclusion of the thesis and suggests possible future research direction.

Chapter 2

Preliminaries

The Internet is an interconnection of many networks together that exist around the world. With the invention of the World Wide Web, people were able to access any information made available from anywhere in the world at any time. It experienced heavy growth in the 1990s as more and more people gain access to the Internet and in the modern world, many of us rely on the Internet in our daily lives. The lowering cost of gaining access to the Internet definitely helped and the ever-increasing speed at which networks are capable of running at is enabling many new applications that were not possible before.

On the local networking part, the most popular and common one is Ethernet. Ethernet is a local networking technology that was eventually made into a standard that is still being used today. It competed with a number of other local networking technologies but eventually took over as the dominant technology due to its lower cost and making into a standard. At its standardization, Ethernet was able to transmit at a speed of 10 Mbps and has evolved in the following decades to 100 Mbps, 1000 Mbps, 10 Gbps and above today. In addition to its speed, Ethernet evolved with respect to its MAC layer protocol, physical medium, and others. One can find Ethernet in use in most networks today.

In the 1990s, the IEEE group developed the 802.11 standard that is also known as WiFi, which is a wireless equivalent of Ethernet. Like Ethernet, the standard has evolved since its inception. Speed has increased dramatically, security has been improved, and more ways of reducing the impact of interference were incorporated. Due to its wireless nature, it enabled the creation of new types of devices or enhancement of existing devices, including laptops, smartphones and tablets, and in turn, also enabled brand new applications that were not possible, or at the very least convenient, with wired technologies.

With the invention of all these new technologies, the complexity of networks has in-

creased and many new types of attacks have been created. Networks require a lot of background services that allows it to behave like they do today. These include technologies like DNS that provides name translation and DHCP that allocates IP addresses. Moving from wired to wireless makes it easy for attackers to sniff traffic and opens up new ways to attack. This thesis shows two of such attacks. In this chapter, some background services seen in many computer networks along with some industry standards are introduced that are relevant to the work done for this thesis.

The chapter is organized as follows: Section 2.1 gives some background information; Section 2.2 gives some related works on WiFi-based location services; and Section 2.3 gives some related works on SSL and proxy servers

2.1 Background

This section will give some background information on the attack on WiFi-based location services as well as on SSL using proxy servers. The background information involves the technology used in existing environments where the attacks take place.

The section is organized as follows: Section 2.1.1 gives background information on network infrastructure components; Section 2.1.2 gives background information on the network infrastructure components that are optional; Section 2.1.3 talks about how SSL works; Section 2.1.4 talks about the part of the IEEE 802.11 standard that are relevant to the work done in this thesis; Section 2.1.5 gives background information on WiFi-based location service.

2.1.1 Network Infrastructure

Devices and functions that make up the network infrastructure provide support to user devices. A user device needs a minimum set of information and supporting devices before it is able to communicate with other devices on the same network, and that minimum is increased when it needs to talk with devices on the Internet. This section introduces the components of a network that are required for access to the Internet and to perform elementary tasks, like browsing the web. The procedure for a client connecting to a network and accessing a website on the Internet is given in Procedure 1 and is illustrated in Figure 2.1.

Procedure 1 A client connecting to a network and accessing a resource

- 1: Request IP address from DHCP server
 - 2: DHCP server returns available IP address
 - 3: Client sets IP address
 - 4: Request www.google.com's IP address from DNS server
 - 5: DNS server returns www.google.com's IP address
 - 6: Client makes a request to www.google.com's IP address for data
 - 7: Router forwards client's request to the destination via the Internet
 - 8: Server returns data to the router
 - 9: Router forwards data to the client
-

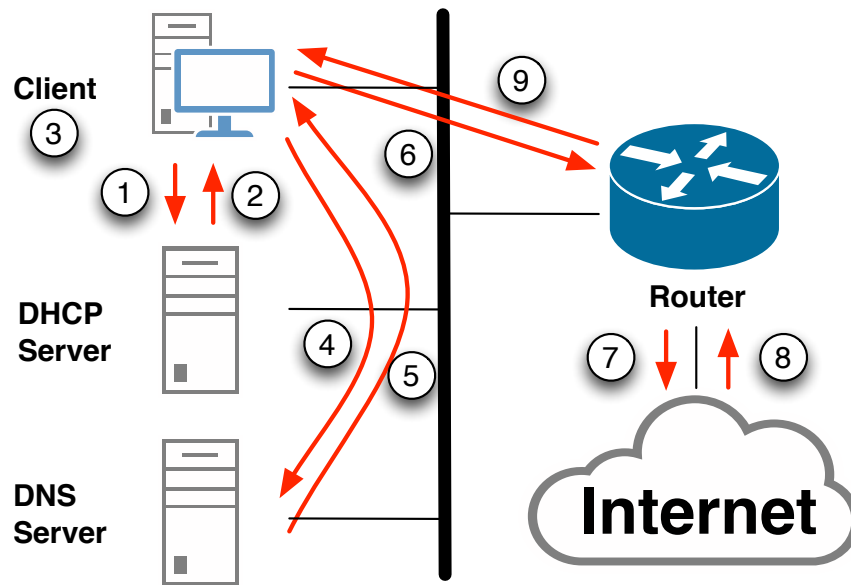


Figure 2.1: Network infrastructure

Client

The client is the user device that needs to access network resources. The client can be a laptop at a company or possibly a tablet at home. In order for a client to communicate with others, it needs to have an IP address. An IP address is basically an identity for the client [25]. The client uses the IP address to communicate with others on a network. When a client needs to communicate, it would build what is called a packet [49], which is the format of the data that travels on the Internet. A packet requires a minimum of three types of information: source IP address, destination IP address and payload. The source IP address is needed so that the destination knows where to send back information. The destination IP address is needed so that the packet travels to the intended recipient. The payload is the data that the client wants to send.

There are two ways that a client can get an IP address: static or dynamic [25]. A static IP address is an IP address that is specifically assigned to and configured on the client, so the client uses that IP address automatically by reading its configuration. A dynamic IP address is an IP address that is given to the client when it connects to the network. The DHCP server is the supporting device that provides the function of managing IP addresses on a network.

DHCP

DHCP is known as Dynamic Host Configuration Protocol [35]. It is a protocol designed to manage IP addresses on a network, and a DHCP server is a device that performs that function. The DHCP server has pools of available addresses on a network or several networks. It keeps a database of the clients that have acquired addresses and prevents them from being assigned again to new clients.

When a client initially connects to the network, it broadcasts a DHCP Discover message to the entire network. The DHCP server picks up the message, unicasts a DHCP Offer message to the client with an available IP address that the client can use. The client, after receiving the offer, unicasts a DHCP Request message to the server to request the IP address, and the server will respond with a DHCP Acknowledgement message to acknowledge the assignment of the IP address to the client [35].

In addition to the IP address, there are other parameters that the DHCP server can provide to the client. These are called DHCP options, and they are usually for special features on different devices. The common parameters that most clients will receive are the subnet mask, which specifies the addressing configuration on the network, and the default gateway, which is a device that connects to other networks, usually a router.

DNS

DNS is known as Domain Name System [43]. On a network and on the Internet, IP addresses define a specific device, but IP addresses can be hard to remember. Names such as apple.com are therefore used as an easy way to refer to devices on the Internet and serve as an alias to the IP address. The job of a DNS server is to have a table of names with their matching addresses and translate the names into their respective addresses when the client requests it.

The DNS protocol is fairly simple. A client sends a request to the DNS server to translate a name to an IP address. The server looks it up then returns the address to the client [43]. While the actual process is much more complicated than that due to the way the entire DNS server structure is organized on the Internet, this high level description of the protocol serves the purposes of this thesis.

Router

A router is a device that directs traffic to its correct destination [47]. The Internet consists of a lot of networks that are interconnected by routers, which is what makes the Internet work. When a client device wants to access something on the Internet, it sets its destination IP address in the packet to the appropriate address and tries to send it. Client devices are generally not on the same network as the device it wants to reach on the Internet, so it will look for its default gateway, which should have been configured either statically or dynamically with DHCP. The default gateway is in almost all cases the router. The client sends the router the packet. The router will look at the destination and forwards the packet to the destination. On its way to the destination, the packet will go through a number of routers that connect various networks together, and they all have the same decision to make, that is, to look at the destination IP address, and decide where to send the packet with the lowest cost. This cost is configurable by the various service providers through which the packet travels and can depend on a number of metrics, such as bandwidth, number of hops or the type of routing protocol used.

Server

A server is a device that is connected to the Internet that serves requests sent by clients around the world. The Internet consists of many servers that provide different services to clients, like hosting web sites, sending and receiving email, or chatting with others on the Internet. The DNS and DHCP services are run on a server and serves client requests.

2.1.2 Optional Network Infrastructure Components

In this section, the components of the network infrastructure that are optional for access to the Internet are introduced. These are not essential to the operation of the Internet but adds additional features to the fundamental network infrastructure. They were also used for the work performed for this thesis.

Firewall

A firewall is a device that filters traffic coming in and out of a network [48]. Companies likely have firewalls to prevent unauthorized traffic from going in to their network. Firewalls have rules that tells the device what traffic to allow and what to deny. These rules are based on various parameters, such as source and destination IP address, source and destination port number, connection state and application layer protocol.

Proxy Server

A proxy server is a device that acts as an intermediary between a client and a server [36]. It splits and manages data transmissions between a client and a server. When a client wants to connect to a server, the client establishes a connection with the proxy instead and makes a request for the server's resources to the proxy. The entire process of a client accessing a resource through a proxy server is given in Procedure 2 and illustrated in Figure 2.2. It is assumed here that a client has already been assigned an IP address and given the IP address of `www.google.com` as per Procedure 1.

Procedure 2 Accessing a network resource via proxy server

- 1: Client makes a request to proxy server for `www.google.com`'s data
 - 2: Proxy server makes request to `www.google.com` for data
 - 3: Router forwards proxy server's request to the destination via the Internet
 - 4: Server returns data to the router
 - 5: Router forwards data to the proxy server
 - 6: Proxy server forwards data to client
-

There are a number of reasons why people would want to use a proxy for accessing resources on the Internet [36]. The most common use of proxy servers are for HTTP connections, which is used for browsing websites. Historically, companies use them as

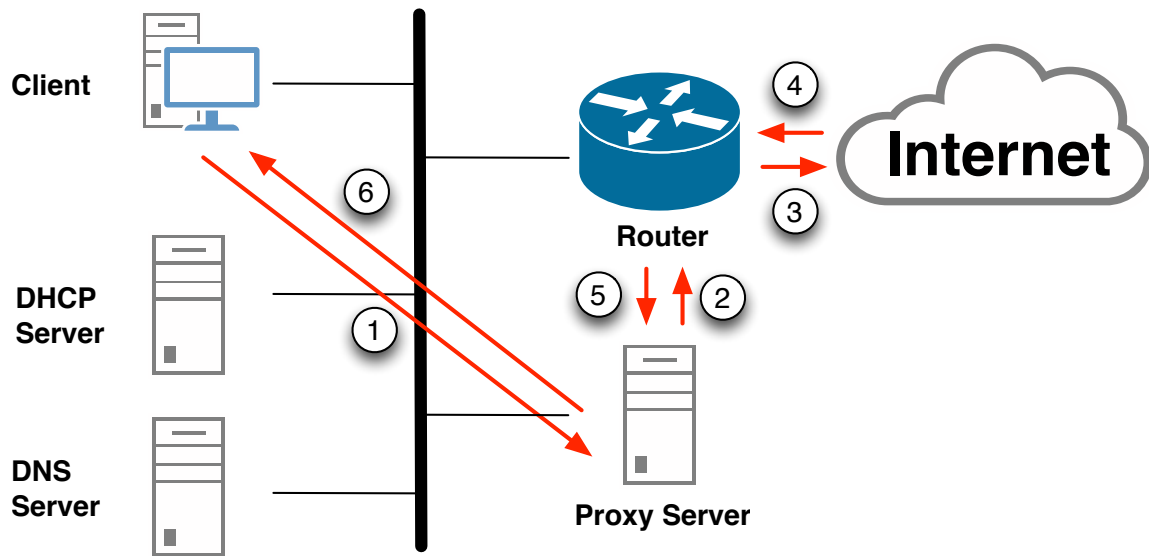


Figure 2.2: Proxy server in network infrastructure

part of their network but many individuals are using them for personal reasons nowadays. Companies use them for a number of reasons, including:

- Reducing amount of end-to-end connections for security reasons
- Caching frequently used content to reduce bandwidth usage
- Log employee Internet usage
- Filtering inappropriate content
- Filtering malicious content

Individuals use them for a number of reasons, including:

- Hiding one's identity
- Access restricted content

Some individuals may choose to use proxy servers for the same reasons as companies.

Active Directory Server

Active Directory is mainly used in a corporate environment so a regular user would not have this at home. Active Directory server is a directory service that manages various types of objects such as users, computers and printers [55]. It provides authentication and authorization for these objects to make sure the right people or computers can use network resources. Active Directory also contains policies that can be pushed or enforced on the users or computers, such as computer configurations or password security requirements.

In a corporate environment, a computer is most likely joined to a domain before it is handed to users. Before a computer can be joined to the domain, it needs to have DNS client settings configured and that can be obtained using DHCP. DNS lookups need to be made to find the main Active Directory server, called the domain controller. During the process of joining the domain, an administrator needs to authenticate the join request, and once authenticated, a computer object is created on the Active Directory server, and the client device pulls all of the policies for the computer. The computer needs to be restarted and the user can now log onto the the domain. The first time a domain user logs in, a new profile will be created on the client device, and all the policies of this user will be downloaded and applied to the profile. After this, the operating system checks the Active Directory server for policy changes periodically, which default to every twenty minutes, and applies them automatically. If an administrator chooses, it can force an update on the client device by issuing a command on the client device, and this may be required when the administrator wants to apply a newly created policy immediately. The software used to define these policies is called Group Policy [67].

Certificate Authority

Certificate authorities are used for security purposes. With billions of devices being used around the world and many servers on the Internet providing services, it becomes hard for people to know who to trust. Certificate authorities are built for that purpose. These are devices that clients trust to verify the authenticity of other devices, especially servers, on the Internet. Certificate authorities are part of public key infrastructures that use public key cryptography to create and sign certificates to bind a public key to its owner [32], that is, to make sure that no attacker can claim to be another entity but with its own public key. When an entity wants to prove to its users that it is an authentic party and not the attacker, the entity would make a Certificate Signing Request to one of the many certificate authorities available on the Internet with its personal or company information, along with the public key. The certificate authority signs the information and creates a certificate

and returns it to the entity. The entity can now present this certificate to any client that connects to it to prove that it is the owner of the device the client is connecting to. Root certificates are loaded onto client devices that certifies the certificate authority's public key. When the client gets a server's certificate to verify, it would check the list of root certificates it has trusted on its device and use the appropriate public key from that certificate authority to verify the certificate. If the signature on the certificate is the same as the one calculated by the client, then the server belongs to who it claims to be and the client trusts this server to provide services to the client. One main assumption of certificate authorities is that the certificate authority itself is trustworthy since the certificate authority is being used to verify the authenticity of other devices. If the certificate authority is compromised in any way, then the entire public key infrastructure is compromised and the attacker is free to create valid certificates that the client will accept as authentic.

2.1.3 SSL

SSL is a protocol for protecting communications between any client and server pair [32]. The protocol runs above the transport layer on the OSI model. SSL is initialized in the session layer, layer 5, and works in the presentation layer, layer 6, by encrypting data from the application layer. When a client wants to communicate with a server, it would send a Client Hello message to the server. The server responds by sending a Server Hello message with its certificate for authentication, cipher suites available and its contribution to the shared key. The client then responds with a message with the cipher suite it wants to use and its contribution to the shared key. The contributions to the shared key are operated on using public key cryptography to get the actual shared key and once that is calculated and the connection is established, symmetric key cryptography is used to encrypt all data exchanged between the client and the server. The result is an end-to-end connection between the client and the server that is encrypted. The protocol is illustrated in Figure 2.3.

SSL relies on public key infrastructure to work, that is, certificate authorities must be set up and certificates distributed in order for SSL to work. The main concern of SSL was server authentication when it was initially designed because a client needs to be sure it is talking to the right server. With the SSL handshake, the client needs to verify the server's certificate using any of the trusted root certificate authorities, which may be preloaded with the operating system or browser or added after the fact. Many certificate authorities exist on the Internet and their root certificates are trusted by software.

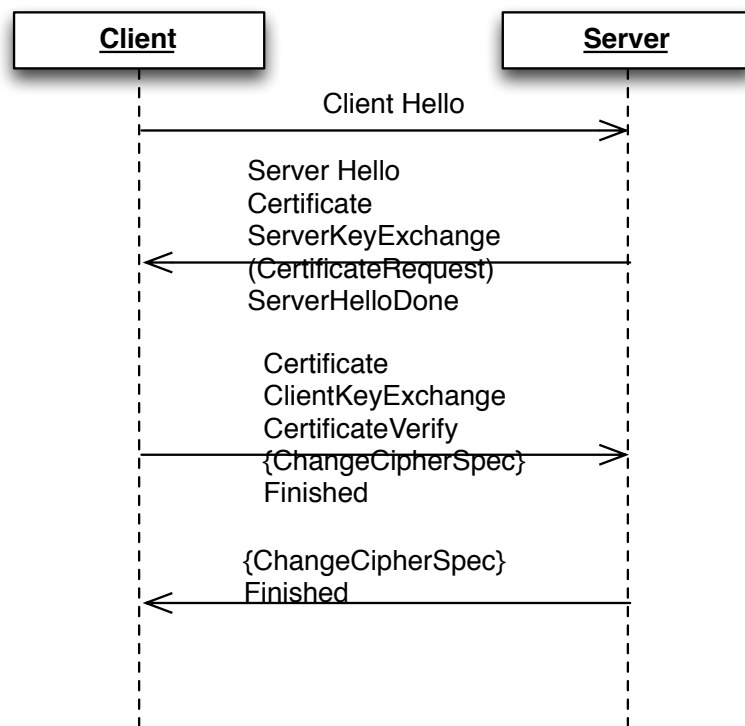


Figure 2.3: SSL Protocol

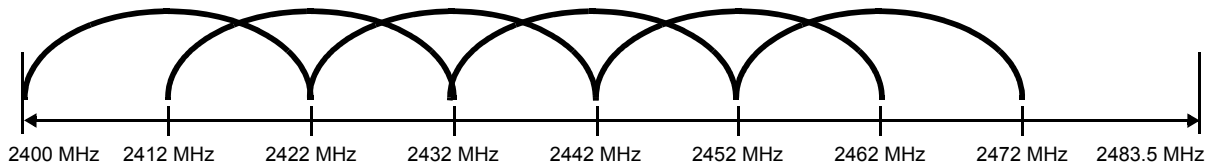


Figure 2.4: All channels in the 2.4 GHz band [22]

2.1.4 IEEE 802.11

The IEEE 802.11 standard was first ratified in 1997. It defines a set of Physical Layer and Link Layer protocols for wireless local area networks, or WLAN. It is now found on many devices on the market today, including laptops, smartphones and tablets. This section introduces parts of the standard that are relevant to the work done for this thesis.

Spectrum

The IEEE defined two frequency bands for WiFi use, with possibly more coming in future revisions. The two frequency bands are 2.4 GHz and 5 GHz. For the 2.4 GHz band, channel one operates at a centre frequency of 2.412 GHz, and channel eleven operates at a centre frequency of 2.462 GHz [22]. Each channel occupies a bandwidth of 20 MHz, so the whole 2.4 GHz band usable in North America has a total bandwidth of 70 MHz. The close centre frequencies and a 20 MHz bandwidth means the maximum number of channels that can be used without interference is three, namely channels 1, 6 and 11 [22]. All channels in the 2.4 GHz band and their spacing are shown in Figure 2.4 and the non-overlapping channels are shown in Figure 2.5. There is an additional problem with the 2.4 GHz band. The band is shared with Bluetooth, some cordless phones and microwave ovens so interference from these devices plus having only three non-overlapping channels is a big concern. For the 5 GHz band, there are many more channels available, and their separation is spaced out so that they are non-overlapping so there are a lot more channels in the 5 GHz band that can be used without interference. The non-overlapping channels in the 5 GHz band are given in Figure 2.6. Notice that the 5 GHz band has twenty-three channels that are non-overlapping, which is way more than the three in the 2.4 GHz band.

In a lot of new devices on the market today, WiFi cards are dual-band, meaning they can operate on 2.4 GHz frequencies or 5 GHz frequencies so users can connect to devices using either frequency bands without worrying about compatibility.

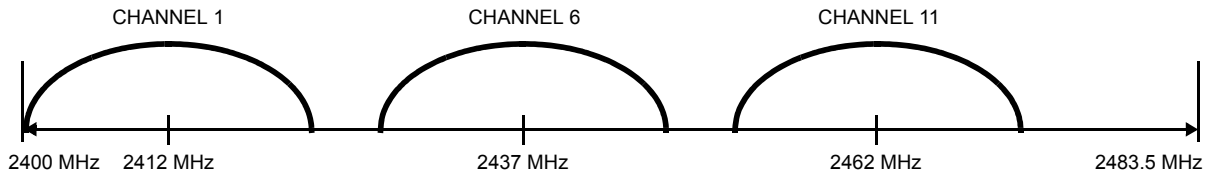
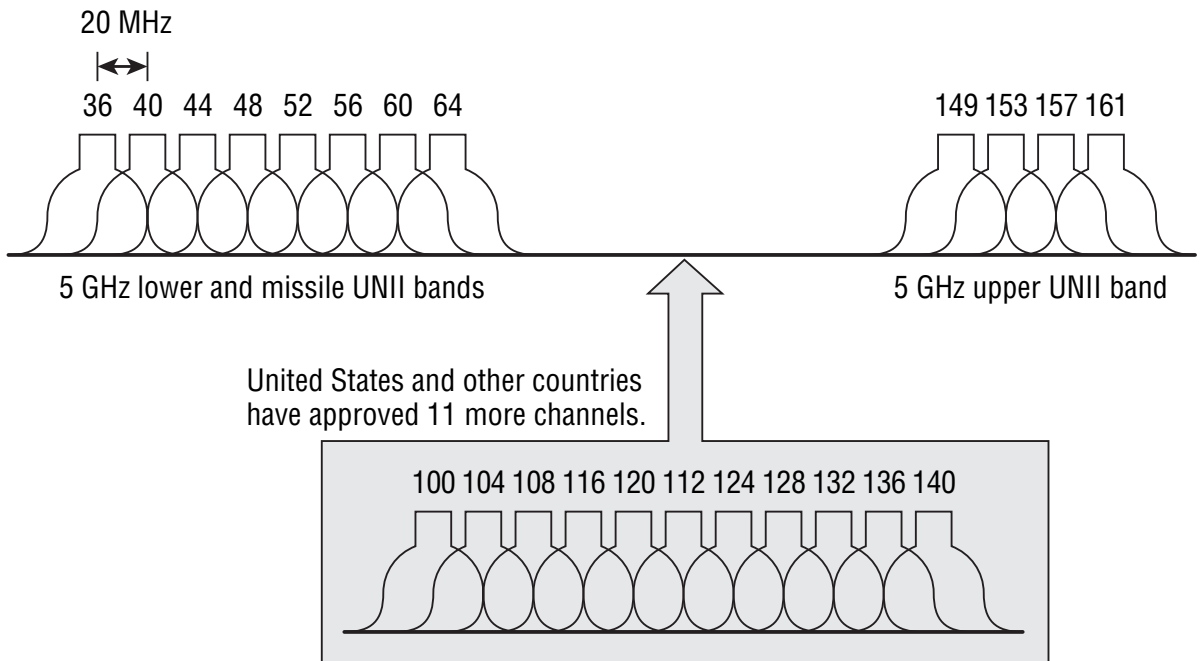


Figure 2.5: Non-overlapping channels in the 2.4 GHz band [22]



Twenty-three non-overlapping channels possibly available at 5 GHz.

Figure 2.6: WiFi channels for the 5 GHz band [33]

Topologies

Wireless LAN infrastructure generally involves client devices, a WiFi access point, and a router to connect to the Internet [33]. A client device associates with the access point and all communications from the client device passes through the access point. Access points broadcast a beacon message periodically to advertise the name of the network, which is the Service Set Identifier (SSID), and its fixed address, which is the MAC address of the WiFi radio interface. Clients would scan the airwaves to see available access points and can optionally send probe requests. The user would choose the network based on the name to connect to, enter authentication credentials and authenticate itself with the access point. After authentication is complete, the user can access the network. In subsequent connections, the device would connect to that access point with the same SSID but not necessarily MAC address without user interaction as it believes it is connecting to the same access point.

Authentication

The authentication schemes in 802.11 has evolved since the standard was first published. For the first iteration of the standard, the working group came up with Wired Equivalent Privacy, also known as WEP. As the name suggests, it was designed to provide the same level of security for wireless networks as wired networks. WEP consists of 10 or 26 hex digits acting as a shared key, meaning the client and access point need to use the same key in order for the client to be authenticated. WEP used the RC4 algorithm for encryption and decryption and CRC-32 checksum for message integrity. The shared key is concatenated with a 24-bit initialization vector, or IV for short, to form the encryption key. There are two authentication methods for WEP. One is called Open Authentication, which basically involves a client requesting access to a wireless network and the access point granting the request and in effect provides no authentication. The other is called Shared Key authentication, where the shared key is used as part of a challenge-response handshake protocol to authenticate the client. A lot of research has been done on WEP and has found many flaws, including very short IV which reduces the key search space, key can be obtained from capturing one set of challenge-response messages and CRC is linear which allows forgery attacks and is no longer recommended for use. Software that is freely available on the Internet can now crack WEP within minutes on real-world devices using a modern computer.

Wi-Fi Protected Access, or WPA, improves on the security of WEP. It was created to fix WEP's security problems before the 802.11i standard was finalized and became what

is now known as WPA2. WPA could be implemented as part of a software update on many existing WEP-enabled devices so the cost of upgrading most devices is just a simple software update instead of requiring the purchase of brand new hardware. Temporal Key Integrity Protocol, or TKIP, was used for WPA. TKIP improves on WEP's security by implementing a key mixing function for increasing the complexity for an attack, a sequence counter to mitigate replay attacks and a 64-bit Message Integrity Check that can help prevent forged packets. For compatibility with older hardware that supports WEP, TKIP uses RC4 as its cipher. Attacks on WPA have been published by several researchers but requires more time and effort than cracking WEP.

Wi-Fi Protected Access 2, or WPA2, is the latest security specification for the IEEE 802.11 standard, known as 802.11i. It is the replacement of WPA which was created as a temporary improvement over WEP that is compatible with older hardware. WPA2 uses AES as its cipher and introduces a four-way handshake and Group Key Handshake protocols. The four-way handshake provides mutual authentication and key derivation from the shared key. The Group Key Handshake is used to update the Group Temporal Key, which is used to decrypt multicast and broadcast traffic. WPA2 introduces Counter Cipher Mode with Block Chaining Message Authentication Code Protocol (CCMP) as the encryption, authentication and integrity protocol. It uses the Counter Mode of AES for encryption and CBC-MAC for authentication and integrity. CCMP uses a 128-bit key and 128-bit block size. One known attack exists for WPA2, and is actually a flaw in a feature implemented on certain access points, not the protocol itself. The Wi-Fi Protected Setup (WPS) allows easy setup of wireless devices to connect to an access point, but researchers found an attack that allows an attacker to recover the WPS PIN followed by the WPA2 password. This attack does not work if an access point has the WPS feature turned off.

An alternate authentication scheme that is used mainly for enterprise networks is the IEEE 802.1X standard. For WPA, the authentication method using a shared key is called WPA-PSK where PSK stands for pre-shared key. For enterprise networks, using a pre-shared key may not be very secure, so WPA-802.1X is used, which uses a RADIUS server for authentication. 802.1X was originally used for port-based authentication for wired networks but has been adopted for wireless use. The protocol uses an Extensible Authentication Protocol (EAP) for authentication. EAP is an authentication framework. It defines a message format for the transport and usage of keying material. Protocols encapsulate EAP messages. Five EAP protocols are adopted as part of the 802.1X standard with many more that exist.

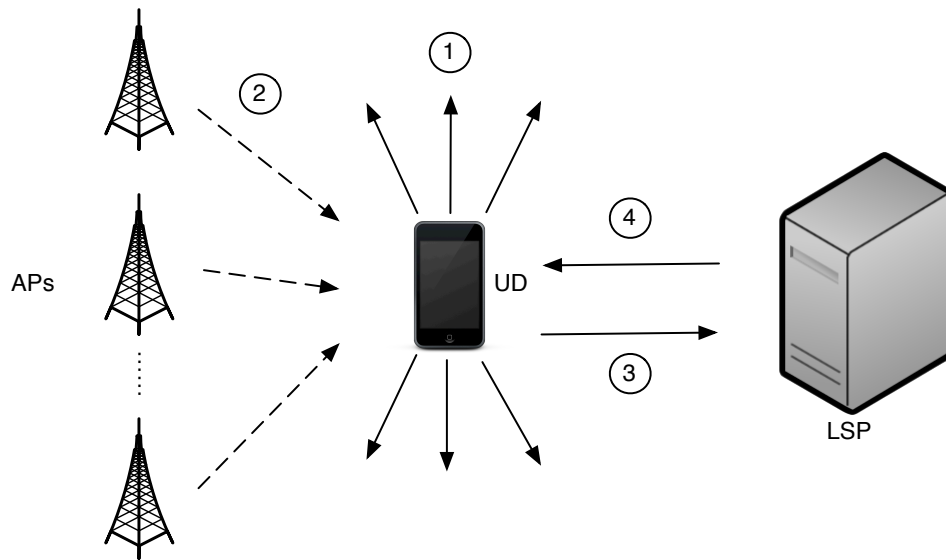


Figure 2.7: Location determination process

2.1.5 WiFi-based Location Services

WiFi-based location services uses WiFi information to provide location information to devices. Information on WiFi access points, including their SSID and MAC address, is collected by the WiFi-based location services companies, usually using vehicles that are equipped with devices that can collect that information. Many of these vehicles drive around in towns and cities around the world so that it can collect as much WiFi access point information as possible and build a database with that information. The process of determining a device's location is illustrated in Figure 2.7. In the diagram, access points are denoted as APs, user device is denoted as UD, and the location service is denoted as LSP. In step 1, the client broadcasts probe requests to ask for surrounding access point information. In step 2, access points that receive this request reply with its information, including its SSID and MAC address. In step 3, the client collects all the responses, along with signal strength information, and send them together to a WiFi-based location service over the Internet. The servers at the service provider will process that information by looking up the access point information in its database, make calculations on the position, taking into account the signal strengths, then in step 5, return a position with longitude and latitude information that corresponds to what it believes to be the device's location.

2.2 Related Works on WiFi-based Location Services

This section gives a literature survey of some related works to WiFi and location services.

2.2.1 Attack on WLAN-based Positioning Systems on a Single Band

In [69], the authors performed an attack on WiFi-based location services using simulated fake access points. The authors used a software defined radio (USRP) to send a uniform noise signal to jam some WiFi bands. In the environment that the authors worked in, the legitimate access points occupies channels 6, 10 and 11, which are all channels in the 2.4 GHz band, and they were all jammed with a uniform noise signal from the USRP. For impersonating access points, the authors wrote a script using a program called Scapy running on a Linux computer. The script listens for client probe requests that asks for the available WiFi access points in the area, then sends responses corresponding to access points that are in a far away location. This information is freely available from WiGLE, also known as the Wireless Geographic Logging Engine [28], which is a service that tracks WiFi access points and their locations. The fake access point uses channel 2. The authors performed the attack on a first generation iPhone that uses Skyhook location services, and laptops using the Loki browser plugin, and was able to get the location service to return a false location that is based on the fake access points. In addition, the authors also conducted the experiment with GSM-based positioning, which showed that when WiFi positioning fails, the iPhone uses GSM-based positioning to calculate its location, but GSM-based positioning provides a much less precise location, which renders the applications that use location services useless.

2.2.2 Attack on WLAN-based Positioning Systems Without Jamming

In [37], the attacker used a slightly different method to attack WLAN-based location services. Without jamming the existing access points, the attacker created many fake access points to trick the location service client. So, it does not matter what channels the legitimate access points are operating in because no jamming is needed. In this setup, the attacker had to create many fake access points. The attacker estimates that in order to perform this attack, he would have to create double the amount of legitimate access points in the area in one case. In the two examples that the attacker gave, in a less WiFi-dense

area with 6 legitimate access points, he had to create around 20 fake access points, and in a denser area with 25 legitimate access points, he had to create around 65 fake access points to make it work. The attacker also used access point information from WiGLE [28]. The hardware used is actually very cheap, with just a Raspberry Pi and a USB WiFi card, which cost around \$100 in total with power supply and SD card for storage. The software used are Aircrack-NG and MDK3, which are software dedicated to cracking and manipulating WiFi. The attacker saves the access point information he got from WiGLE into a text file and launches the software to advertise that information to its surroundings.

2.2.3 Portable GPS Spoofer

In [45], the authors constructed a portable civilian GPS spoofer. The authors note that there was concern with potential vulnerability of civilian GPS receivers if their signals are spoofed as early as the early 2000s. Some examination of spoofing was done and countermeasures were proposed. Amplitude discrimination and time-of-arrival discrimination involve changes to software but is only effective against the simplest attacks. Consistency of navigation inertial measurement unit cross-check, polarization discrimination, and angle-of-arrival discrimination require additional hardware. A portable receiver-spoofers would defeat most of these countermeasures, which is what the authors concentrated on. The portable GPS spoofer consists of a receiver, which, as the name suggests, receives the legitimate GPS signals and the spoofer, which sends the fake GPS signals. The spoofer takes inputs from the receiver module, giving the start times of the k th C/A code period, estimate of beat carrier phase, estimate of the Doppler frequency shift, estimate of the signal amplitudes and the receiver-spoofers current 3D position and velocity. The spoofer tries to align the code phase then increase its amplitude, and at which point the target should be under the control of the spoofer.

2.2.4 Securing Localization with Hidden and Mobile Base Stations

In [31], the authors created a secure localization algorithm using hidden and mobile base stations. The idea behind this is that the position of these base stations are not known to the attacker and thus an attacker would have a hard time to attack them since they don't know where they are. The positions of these hidden base stations act as a secret key to the system. To prevent attackers from learning the positions of these hidden base stations, these devices are made passive, meaning they only listen to wireless signals and not

transmit. The authors distinguish between two types of localization methods: node-centric, where a client computes its location after observing signals from public base stations and infrastructure-centric, where the infrastructure devices like base stations or backend servers computes the location of the client based on their communication. In the infrastructure-centric protocol, the base stations first send a challenge to the client. The client broadcasts the response to all base stations. Upon receiving the response, the hidden base stations calculate the position of the client using time difference of arrival. If the computed position and the measured time difference is inconsistent, then the position is rejected. The authors note that the system's precision is a factor in the attack's effectiveness. If the system is not precise, then more tolerance is accepted and the attacker has more room to fake a location. In the node-centric protocol, the client gets its location from a non-secure positioning system. The base stations sends a challenge to the client and the client responds with its calculated position and an ultrasound message. The hidden base stations then computes the time between when it received the ultrasound signal and the radio signal and computes the distance to the client. The measured distance is compared to the client's reported position and is rejected if they are inconsistent.

2.2.5 Using Localization to Enhance Security and Services in WiFi Networks

In [29], the authors propose enhancements to security on WiFi networks using location information of the user, specifically, a location based firewall system. The authors define two challenges in providing location based security. One is providing updates on user location information in a timely manner in order to restrict network access or not. The other is controlling the firewall to allow or deny network access when users are mobile. The wireless infrastructure environment in which the authors worked on their research includes an enterprise location appliance that stores location information of devices that have connected to the network and allows third-party applications access to that information through an API. This is in contrast to many applications of location services where the client, not the server or central appliance, independently makes a location request to a service provider. After getting the location information of devices currently on the network, the application makes a request to security devices to allow or deny traffic depending on what is required. The authors define four approaches to restricting access to the network and they are: 1) prevent clients from connecting to the wireless network by using MAC filters and de-authentication requests; 2) use role-based firewall and captive portal to restrict access based on user, user type and/or MAC address; 3) dynamically configure firewall; and 4) send traffic from clients to null route.

2.2.6 Denial of Service Attacks on 802.11 Networks

In [40], the authors identified several methods to deny wireless service, specifically the access point. Since in many wireless infrastructure, the access point is the single device that clients connect to for wireless service, it becomes a single point of failure, which is a reason why the attackers chose to attack access points. By analyzing how the MAC layer of 802.11 works, the authors figured out ways to attack the MAC layer to deny service. This is in contrast to jamming which is on the physical layer and does not care about what goes on in the MAC layer or any layer above the physical layer. The authors identified three different attacks to deny service, but they are all flooding requests. The idea is to flood the access point with specific types of packets to overwhelm them and thus not able to respond to traffic from legitimate users. The first attack is the probe request flood. In this attack, the attacker floods the access points with many probe requests each coming from a different MAC address that is faked by the attacker. Since access points must respond to probe requests with a probe response, access points will be busy responding to these requests instead of serving legitimate users, thus denying the service. The second attack is the authentication request flood. The attacker floods the access points with many authentication requests, and the access points will be overwhelmed in trying to authenticate these fake requests, thus denying the service to legitimate users. The last attack is the association request flood. The attacker floods the access points with many association requests. Association requests are not sent by legitimate devices if they are not authenticated. In the author's experiments, many access points actually respond to these association requests that are clearly not from authenticated devices and thus exhausts its resources and denies access to legitimate users.

2.2.7 Mitigating Attacks Against Signal Strength-based Localization Algorithms

In [53], the authors talk about attacks against signal strength-based localization algorithms and how to mitigate it. They used a toy example to show the effect of attenuation on the signal strength readings but the authors say that an attacker can potentially put an absorbing or reflecting material around a wireless device to changes its signal strength reading. In signal strength based localization algorithms, the client performing the algorithm would collect received signal strength (RSS) from static wireless devices, usually access points, and perform calculations on it or send it to a server, the location service provider, for calculation. The location is then returned to the client. When the signal strength is modified, the location of the client may be shifted and thus not entirely accurate. In the paper, the

authors assume that a similar attack is carried out on all access points, which means the signal strength detected at the client is reduced approximately the same amount for all of the signals it received. The authors proposed a new metric for calculating the location, that is, a ratio-based signal strength metric or RSM. The new metric takes advantage of their assumption that the signal at the client is affected similarly for the signals received by the client from all access points. While an attack on all the access points modifies the signal strength at the client, the relative distance between two access points are unaffected. In their experiments, the authors show that the RSM localization method showed less degradation in general than RSS localization method.

2.3 Related Works on SSL and Proxy Servers

This section gives some related works to SSL, man-in-the-middle attacks and proxy servers.

2.3.1 Practical Attacks Against SSL

In [27], the authors talked about some of the issues with using the web and how legitimate websites using SSL is not enough to prevent attacks, especially phishing attacks. The authors say that man-in-the-middle attacks are possible on websites where an attacker can present a website that looks identical to a legitimate site, for example, an online banking site. Instead of sending the login information to the bank, the site actually forwards that information to the attacker, thereby compromising the user's login credentials and anything that can be accessed using those credentials. Modern browsers generally have several indicators when using SSL/TLS protected sites, these include: the full URL of the site with domain name and path, the protocol used (HTTP or HTTPS), and a padlock icon for SSL-enabled sites, but from studies referred to in their paper, users still fail to recognize a fake website. One server-side countermeasure designed to prevent this, that is, site-authentication images, was not enough to prevent users from entering their login credentials to the fake website according to one of the studies. Other countermeasures proposed are special bookmarks/links used by the website for each user, password managers to fill in login information and authenticate websites.

2.3.2 A Practical Man-in-the-middle Attack Against SSL

The authors of [54] provided an implementation of a man-in-the-middle attack on SSL. The attacker places itself in the middle between the client and the server using ARP poisoning,

which makes it look like the attacker's device is the router. The program then looks for HTTP traffic on a network, hijacks that traffic, and replaces any HTTPS links with HTTP ones, that is, replace what is supposed to be links to encrypted websites with unencrypted ones. These replacements are recorded by the attacker. An unsuspecting user that visits any of those links will first make a request to the attacker-controlled server using HTTP, unencrypted. The attacker then makes an HTTPS request to the actual server that the user wanted to access to get the legitimate page. The attacker receives the webpage over HTTPS from the legitimate server, then forwards it to the user using HTTP. When a user enters any information, including any login credentials, that information is sent over HTTP to the attacker and the attacker is able to read all of that information in plaintext, thereby compromising the user's privacy and confidentiality. The attack works due to the problem that users usually type websites into the browser without using "https://", and some websites choose to present an unencrypted homepage with links to encrypted content.

2.3.3 Issues with the Certificate Authority Model

In [60], the authors talk about the issues of the Internet's certificate authority model. Certificate authorities allow people to verify the authenticity of another party in a public key infrastructure. They first create a self-signed root certificate for the certificate authority that has a public and private key pair. The private key is used to issue digital certificates to entities and anyone that wants to communicate with them on the Internet would use the certificate authority's public key from the root certificate to verify the digital certificate. If the digital certificate is verified, then the entity is verified and is assumed to be who it claims to be. One big issue with the current certificate authority model is that there are thousands of certificate authorities around the world. Software vendors have to integrate many if not all of the root certificates of these certificate authorities into their software or else the system does not work. It is impossible to figure out who is actually trustworthy or not, and these certificate authorities can issue certificates for any entity (e.g. website). In addition, if any one of these systems are compromised, then the attacker can potentially sign its own certificates and use them to get access to private information. Another issue is that many certificate authorities delegate verification of ownership to third parties, called registration authorities, and even issuing certificates to third parties, called subordinate certificate authorities. Since the certificate authorities delegate these tasks to them, users just have to trust them, even though they do not need to be audited or identity revealed.

2.3.4 Implementation Issue in SSL-enabled Applications Opens Applications to Attack

In [34], the authors talk about an implementation issue in SSL-enabled applications that are on modern systems that allow them to be compromised using a man-in-the-middle attack. Normally, the SSL handshake involves the client initiating the connection, then the client and server exchanges some parameters, the server provides its certificate for authentication, and they both agree on the keys and cipher suite to be used in the communication. The authors in the paper found a problem with the way some applications verify the server's certificate and that opens up the application to attack. In the attack, the authors get the legitimate server's certificate, and creates a certificate signed by its own certificate authority with all the information from the legitimate server's certificate to make it look like a legitimate certificate. They carried out the attack on a number of applications on the Android platform and found that most of the applications tested are vulnerable to the attack. The authors also wrote a program that performs the correct certificate validation procedure and acts as a proxy for the applications to strengthen security. They find that do this did not affect the performance of the applications significantly and it should protect the applications from the attack.

2.3.5 Man-in-the-middle Attack on WiFi Clients

In [42], the authors showed a man-in-the-middle attack on WiFi. The issues the authors identified that makes the man-in-the-middle attack possible are that there is no mutual authentication with WiFi and that anyone can listen in on wireless signals. This is in contrast to wired networks where the link is protected by physical means and thus is much harder to eavesdrop. The authors used two laptops for their attack, one being the man-in-the-middle and one being the client. The man-in-the-middle has two network interfaces used. One connects to a legitimate network where it has access to the Internet and thus can route traffic to the Internet. The other interface acts as a rogue access point and can serve clients. The two interfaces are bridged so that clients can get access the Internet via the fake access point. An attacker can force clients into connecting to his own rogue access point by sending deauthentication packets to clients. Since the traffic from clients pass through the rogue access point, the attacker can analyze and modify the traffic at his own will. In the paper, the authors wrote a script to look for HTTP download links and replace them with links to their own file that contains malicious content. An unsuspecting wanting to download a file will be tricked into downloading potential malware, which will do harm to the user's computer. The authors note that the attack is just one approach

and can be extended to do even more harm. One countermeasure the authors came up with is to use VPN, which establish a secure connection at the network layer and thus an attacker cannot see or modify the information contained in upper layers.

2.3.6 SSL Session-aware Authentication

In [59], the authors created a method for session-aware authentication for SSL. The authors note that man-in-the-middle attacks are a real threat to SSL-enabled applications and they address this with a new session-aware authentication scheme for SSL. This means user authentication depends on the current SSL session, or binding user authentication to a specific SSL session. By binding user authentication with the current SSL session, even if a man-in-the-middle attack is performed, the attacker cannot use the captured credentials on another session since the credentials are valid for only one session. In order to achieve this, the authors used an authentication token with the ability to digitally sign. The procedure for SSL session-aware authentication is as follows. The user has an ID_U and a secret PIN_U that is shared with a server. The user also has a token that can be used to digitally sign messages, which is identified by its serial number SN_T and has a secret key K_T . When a user wants to connect to the server, it first initiates a connection with the server, then the server will authenticate itself then request the client to also authenticate itself. The client authenticates itself by returning a CertificateVerify message with the cryptographic hash of all previously exchanged messages in the session, which is generated by the token using a public and private key pair common to all tokens. In addition, another value is generated using the secret key K_T and combined with PIN_U to form a User Authentication Code (UAC) that is only valid for the current session. The user sends ID_U , SN_T and UAC to the server for authentication. The server can authenticate using these values since it is in possession of all the secret parameters. Once the server authenticates the client, the server and client can begin secure communication for this session. If they want to communicate for subsequent sessions, they either have to perform the authentication procedure again or they may potentially cache their credentials (but that may defeat the purpose of the protocol).

2.3.7 Lucky Thirteen - A Timing Attack Against SSL

In [39], the authors detailed a timing attack on SSL. The authors note that in most SSL sessions, the encryption algorithm used is AES in CBC mode and HMAC-SHA1 is used as the authentication algorithm to create a Message Authentication Code or MAC. In the

SSL protocol, a MAC is first created on the data and header. The MAC is then appended to the payload. Since AES is a block cipher that encrypts 16 bytes at a time, data plus its MAC must be padded if they are not a multiple of 16 bytes when combined together. When a server receives such a packet, it would decrypt the data, identify the padding and remove it from the packet. The rest of the data contains a 20-byte MAC and the actual payload. The server calculates the MAC of the payload and verifies with the MAC that was sent with the packet. If it does not match, then the server will respond with an error message and terminate the session. For an attacker, he can intercept these communications, but of course cannot decrypt the data since he does not have the key. However, he can take advantage of the fact that an error message is sent and modify packets to get information. An attacker can change two bytes of a packet then truncating it, they can make the server believe that the message was padded. The server will attempt to decrypt the packet but it is obviously not valid since an attacker has modified the packet. The time it takes for the server to send an error message leaks information about the plaintext. The timing difference comes from the fact that HMAC-SHA1 takes some units of time to check the first 55 bytes of data plus an extra time unit for each additional 64-byte block. This allows the attacker to recover two bytes of the original packet. Repeating this, the attacker can recover many more bytes. This attack was done in a very controlled environment and thus not entirely practical due to timing requirements but nevertheless shows that SSL contain some vulnerabilities.

2.3.8 Bias in RC4 Used in SSL

In [26], the authors detail two attacks on the RC4 algorithm. The first attack rely on single-byte biases in the initial 256 bytes of RC4 keystreams. The attack is done over many sessions and the authors said that the number of sessions needed to reliably recover 220 bytes of SSL-encrypted plaintext is around 2^{30} but some bytes can be recovered reliably with 2^{24} sessions. The second attack detailed in the paper can be carried out in a single session and exploits some double-byte biases in RC4 keystreams. Reliably recovering a set of 16 consecutive plaintext bytes requires around $10^2 \cdot 2^{30}$ encryptions but with only $6^2 \cdot 2^{30}$ these bytes can be recovered with 50% reliability.

2.3.9 Attack Against Tor Network Using HTTP-based Application

In [71], the authors showed an attack on the Tor network, which is a service that provides anonymity to its users and is something close to a proxy server. The authors insert malicious Tor routers onto the Tor network, then looks for web traffic. Once that traffic is detected, a custom webpage or modified version of the legitimate webpage from the destination server is returned to the user. The client then generates some web traffic that can be analyzed by an attacker and the attacker can now perform traffic analysis on that traffic to figure out the client and server that are communicating. The authors note that using encrypted web traffic, that is, HTTPS, mitigates this attack since the attacker cannot modify the webpage to get the client to send special traffic that can be analyzed by the attacker.

Chapter 3

Attack on WiFi-based Location Services Using Dual-band Hardware

Mobile devices are now a big part of our life thanks to wireless LANs, also known as WiFi. WiFi is a technology that allows devices to communicate wirelessly [33], and the most common use is accessing the Internet. The use of wireless technology allows us to access the Internet wherever we are, whether it's on a laptop at a coffee shop, a tablet at home or a smartphone at a mall. This opens up many new types of applications that were unimaginable using wired network connections.

However, bringing wireless technology into all of our devices creates big security problems. Due to the fact that data is being transmitted over the air, anyone can snoop in on the transmissions if it is not encrypted. Plus, all of the problems related to wired technology also applies in the wireless world, so there is a lot of research on securing wireless communications, like in [51, 64, 70, 72].

On the physical layer, jamming attacks can be carried out which involves an attacker sending noise signals that make it hard for legitimate users to decode their own signals. For wired networks, an attacker would need access to the physical network in order to carry out the attack, but since data is transmitted over the air in wireless networks, no such restriction exist. An attacker can send the attack signal freely and anyone in range of the signal would be affected. This can have many implications on wireless communication and services that rely on them. In general, jammers jam the control signals which do not use any anti-jamming protection mechanisms in practice [51, 58, 64, 70, 72]. The research on anti-jamming mechanisms have recently looked at using cryptographic related methodologies and approaches, see [65, 68, 73, 74], just to list a few. In the WiFi case, one such service

that is affected that is very popular today is location services.

Location services, as the name suggests, allows a device to get its location. Traditionally, when one thinks about a device that can get its location is one that uses the Global Positioning System, or GPS [38]. GPS devices are mainly used in cars to provide directions from one place to another. It can achieve this because satellites are launched into space, and these satellites together can see the whole world. Signals are sent from GPS devices on the ground to satellites that are in range and replies are received and calculations are performed to determine the exact position in the world. Due to the use of satellites, GPS has a number of limitations, and one of them is that GPS does not work indoors. This limitation motivated many companies to find a mapping solution that works indoors. WiFi-based location service was one of the solutions. Plus, the nature of WiFi means that one should be able to get a more accurate measurement of a device's position. Combined, WiFi-based location service enables brand new applications that were not possible with GPS-based location service. However, this also means more services are compromised if the underlying WiFi infrastructure is compromised. An implementation of such an attack was performed and implemented for this thesis. This attack was conducted on four major location service providers: Skyhook, Google, Apple and Microsoft. GPS can also be compromised in a similar way as demonstrated in [45]. An analysis of the requirements of carrying out a successful GPS spoofing attack was conducted in [68].

The chapter is organized as follows: Section 3.1 gives a vulnerability analysis of four major location service providers: Skyhook, Google, Apple and Microsoft; Section 3.2 shows the environment in which the attack was conducted in and the equipment used in the attack; Section 3.3 describes the implementation of the attack and the results from the attack, Section 3.4 analyzes the impact of attack on location services and various location-enabled applications; Section 3.5 talks about the impact of the attack on WiFi infrastructure; Section 3.6 explains some of the potential countermeasures; and lastly Section 3.7 provides a summary of the chapter.

3.1 Vulnerability Analysis of Location Services Providers

In our experiments, we focused on four location services providers: Skyhook, Google, Apple and Microsoft. Skyhook was founded in 2003 and is one of the earliest companies to offer WiFi-based location services. Skyhook location services was used on Apple iPhone, iPod Touch and iPad until iPhone OS software release 3.2. It is also used in numerous other smartphones. Skyhook also has a browser plugin that allows any computer that can use the plugin to determine its location using WiFi location services. Google's location services

is used on Android operating system-based devices, and in Firefox and Chrome browsers. Apple's location services is more recent, and is used by all of Apple's current generation devices. Their location service is built into their Mac OS X operating system that runs on computers, and iOS operating system that runs on their mobile devices such as the iPhone, iPad and iPod Touch. Microsoft's location services is built into their operating systems, Windows and Windows Phone.

To see what kind of information that needs to be sent to determine a location, a proxy was used to capture the traffic between the client and the location services server. Since all of the services use SSL, it was not possible to use just Wireshark to sniff the packets as that would not reveal the encrypted contents. So, a tool called sslmeat [16] was used. sslmeat is an SSL proxy. It works like an HTTP proxy in that it sets up two connections to establish communications: one between the client and the proxy and one between the proxy and the destination server. However, in addition, the client needs to trust the proxy by trusting the proxy's server certificate in order for the SSL proxy to work. Then, when a client needs to access the Internet, traffic is first passed from the client to the proxy, encrypted using the proxy server's public key. The proxy server decrypts the data, encrypts it again using the destination server's public key and forwards it to that server. The fact that the proxy server can decrypt the client's data allows analysis of the traffic protected by SSL.

The captured information shows the format of the server queries and what exactly is being sent to these servers. For Skyhook, the query is an HTTP POST request in XML format. The MAC address and signal strength of surrounding access points are sent to Skyhook's servers and a sample query is shown in Figure 3.1. For Google, the query is an HTTP POST request, with the MAC address and signal strength of each access point sent to the server, and a sample query is shown in Figure 3.2. For Apple, the query is an HTTP POST request, and it includes all the MAC addresses in the surrounding area, and a sample query is given in Figure 3.3. For Microsoft, we were not able to determine the format and information sent to their servers but [] says that it is a SOAP request that only sends the MAC address.

The information captured here shows that the only uniquely identifiable information collected is the MAC address. The signal strength is just a number that one can arbitrarily choose from the range of acceptable signal strength values. So, only the MAC address needs to be spoofed to impersonate an access point.


```
POST /wps2/location HTTP/1.1
Accept: */*
Content-Type: text/xml
User-Agent: WPS API (3.4.2.20)
Accept-Language: en-us
Accept-Encoding: gzip, deflate
Host: api.skyhookwireless.com
Content-Length: 1822
DNT: 1
Connection: Keep-Alive
Cache-Control: no-cache
```

```
<LocationRQ xmlns='http://skyhookwireless.com/wps/2005' version='2.10'
street-address-lookup='full'><authentication
version='2.0'><simple><username>loki.com</
username><realm>js.loki.com</realm></simple></authentication><access-
point><mac>20AA4B96A40B</mac><signal-strength>-75</signal-
strength><age>30233</age></access-point><access-
point><mac>D86CE92A9365</mac><signal-strength>-56</signal-
strength><age>30296</age></access-point><access-
point><mac>00222D559979</mac><signal-strength>-43</signal-
strength><age>30342</age></access-point><access-
point><mac>3CEA4FA11609</mac><signal-strength>-62</signal-
strength><age>30342</age></access-point><access-
point><mac>B88D12660DCF</mac><signal-strength>-24</signal-
strength><age>30374</age></access-point><access-
point><mac>D86CE9230635</mac><signal-strength>-59</signal-
strength><age>30420</age></access-point><access-
point><mac>BC1401E5F3E9</mac><signal-strength>-73</signal-
strength><age>30436</age></access-point><access-
point><mac>D86CE9290B11</mac><signal-strength>-63</signal-
strength><age>30483</age></access-point><access-
point><mac>C0C1C0264C23</mac><signal-strength>-60</signal-
strength><age>30639</age></access-point><access-
point><mac>386077BAAC91</mac><signal-strength>-56</signal-
strength><age>30654</age></access-point><access-
point><mac>586D8F06431B</mac><signal-strength>-44</signal-
strength><age>30654</age></access-point><access-
point><mac>C0C1C0264C22</mac><signal-strength>-65</signal-
strength><age>30654</age></access-point><access-
point><mac>00222D55997A</mac><signal-strength>-44</signal-
strength><age>31356</age></access-point><access-
point><mac>4C72B9063C0B</mac><signal-strength>-75</signal-
strength><age>31403</age></access-point><access-
point><mac>74D02B3AE37C</mac><signal-strength>-62</signal-
strength><age>31559</age></access-point></LocationRQ>
```

Figure 3.1: Skyhook location service query

```
POST /geolocation/v1/geolocate?key=AIzaSyD-s-
mXL4mBzF7KMRkhTCIbG2RKnRGXzJc HTTP/1.1
Host: www.googleapis.com
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.9; rv:25.0)
Gecko/20100101 Firefox/25.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
DNT: 1
Content-Type: application/json; charset=UTF-8
Content-Length: 526
Connection: keep-alive
Pragma: no-cache
Cache-Control: no-cache

{"wifiAccessPoints":
 [{"macAddress":"66-2a-2f-53-7c-99","signalStrength":-57},
 {"macAddress":"58-6d-8f-06-43-1b","signalStrength":-60},
 {"macAddress":"d8-6c-e9-2a-93-65","signalStrength":-61},
 {"macAddress":"d8-6c-e9-23-06-35","signalStrength":-65},
 {"macAddress":"38-60-77-ba-ac-91","signalStrength":-68},
 {"macAddress":"c0-c1-c0-26-4c-22","signalStrength":-69},
 {"macAddress":"c0-c1-c0-26-4c-23","signalStrength":-74},
 {"macAddress":"20-aa-4b-96-a4-0b","signalStrength":-77},
 {"macAddress":"4c-72-b9-06-3c-0b","signalStrength":-82}]}
```

Figure 3.2: Google location service query

```
POST /clls/wloc HTTP/1.1
Host: gs-loc.apple.com
Content-Type: application/x-www-form-urlencoded
Content-Length: 242
Proxy-Connection: keep-alive
Accept: */*
User-Agent: locationd/1613.5.1 CFNetwork/672.0.8 Darwin/14.0.0
Accept-Language: en-us
Accept-Encoding: gzip, deflate
Connection: keep-alive

en_UScom.apple.locationd
7.0.3.11B5113/4
58:6d:8f:6:43:1b
ea:40:f2:6c:6f:b5
38:60:77:61:c3:63
74:d0:2b:3a:e3:7c
20:aa:4b:96:a4:9
c0:c1:c0:26:4c:23
c0:c1:c0:26:4c:22
3c:ea:4f:a1:16:9
66:2a:2f:53:7c:99 d
```

Figure 3.3: Apple location service query

3.2 Adversary Model and Equipment

In the experiment, the environment is a residential area consisting of houses separated only by a few metres or two backyard lengths. When a scan is done on the airwaves to see available WiFi access points, around ten access points can be seen that uses channels ranging from channel 1 to channel 11, which covers all of the 2.4 GHz channels that are available in North America at full power [22]. This is slightly different than in [69], where the legitimate access points that are available in the authors' vicinity only occupied channels 6, 10 and 11, which is just over half of the 2.4 GHz band. There was space for an access point on the 2.4 GHz band on channels 1 and 2, and the authors in that paper set up their fake access point on channel 2. In our setup, since the whole 2.4 GHz band is used, dual-band devices needed to be used in the experiment so that both the 2.4 GHz band and the 5 GHz band can be used as part of the attack. We assume that a GPS signal is not available since the attack will be performed in an indoor environment.

The hardware used in the experiment are listed below and shown in Figure 3.4.

- 2x USRP N210 software defined radio with RFX2400 daughterboard from Ettus Research
- 1x Dell PowerEdge 2950 server
- 1x PC Engines ALIX.2D13 embedded system
 - 1x Wistron NeWeb Corp. DNMA-92 802.11 a/b/g/n Mini-PCI dual-band WiFi card
 - 2x Dual-band (2.4 GHz and 5 GHz) antenna, 5 dBi nominal gain with SMA connector
- 1x Apple MacBook Pro 15-inch, Mid-2010 (model number MC371LL/A)
- 1x Apple iPad 2 (model number MC775C/A)

The software used in the experiment are as follows:

- Dell PowerEdge 2950:
 - VMware ESXi 5.1 Update 1 build 1065491
 - 2x Ubuntu 13.04 operating system in virtual machines



Figure 3.4: Hardware

- GNU Radio 3.6.4 operates the USRP's
- ALIX.2D13 embedded system:
 - OpenWRT 12.09 (Attitude Adjustment) operating system
 - hostapd to create fake access points
- MacBook Pro 15-inch, Mid 2010:
 - Mac OS X 10.8.4 operating system, build 12E55
 - * Mozilla Firefox 24.0 to test Google location service
 - * Mozilla Firefox 24.0 with Loki plugin to test Skyhook location service
 - Windows 8 operating system
 - * Maps application to test Microsoft location service
- Apple iPad 2
 - iOS 7 operating system, build 11A465
 - Apple Maps application to test Apple location service

3.3 Implementation of Attack on WiFi-based Location Services

In this section, the implementation details of the attack on WiFi-based location services are given along with the results of the attack.

3.3.1 Attack procedure

The WiFi access points in the test environment cover the whole 2.4 GHz band. In order to perform the attack, we would need dual-band capable devices since we needed to jam all the frequencies on which legitimate access points can be seen, which means the whole 2.4 GHz band. So, instead of setting up fake access points on channel 2 like in [69], the experiment takes advantage of dual-band devices that can operate in the 2.4 GHz band and 5 GHz. The hardware had been picked for this purpose. Note that one could have also used a 5 GHz-only WiFi card in a computer as the location services client, but doing so would not demonstrate the effect of jamming as 5 GHz-only WiFi cards cannot receive 2.4 GHz signals.

To carry out the attack, we need to do two things: (1) jam all legitimate access points so that the location services clients cannot determine its location and (2) set up fake access points so that the location services clients will use incorrect information to determine its location and thus get the wrong location. Procedure 3 gives the steps for the attack.

Procedure 3 WiFi-based location services attack

- 1: Use GNU Radio and USRP to send a White Gaussian Noise signal centred at 2.422 GHz to jam about half of the 2.4 GHz band
 - 2: Use GNU Radio and USRP to send a White Gaussian Noise signal centred at 2.452 GHz to jam the rest of the 2.4 GHz band
 - 3: Use WiGLE.net's database [28] to get the MAC address of several access points that are near each other anywhere in the world
 - 4: Set up Internet connection and routing on the embedded system running OpenWRT
 - 5: Set up impersonated access points using these MAC addresses on a 5 GHz channel
 - 6: Connect each location services client to any of the impersonated access points to get an Internet connection so that they can make queries to their respective location services server
 - 7: Have each location services client determine its location
-

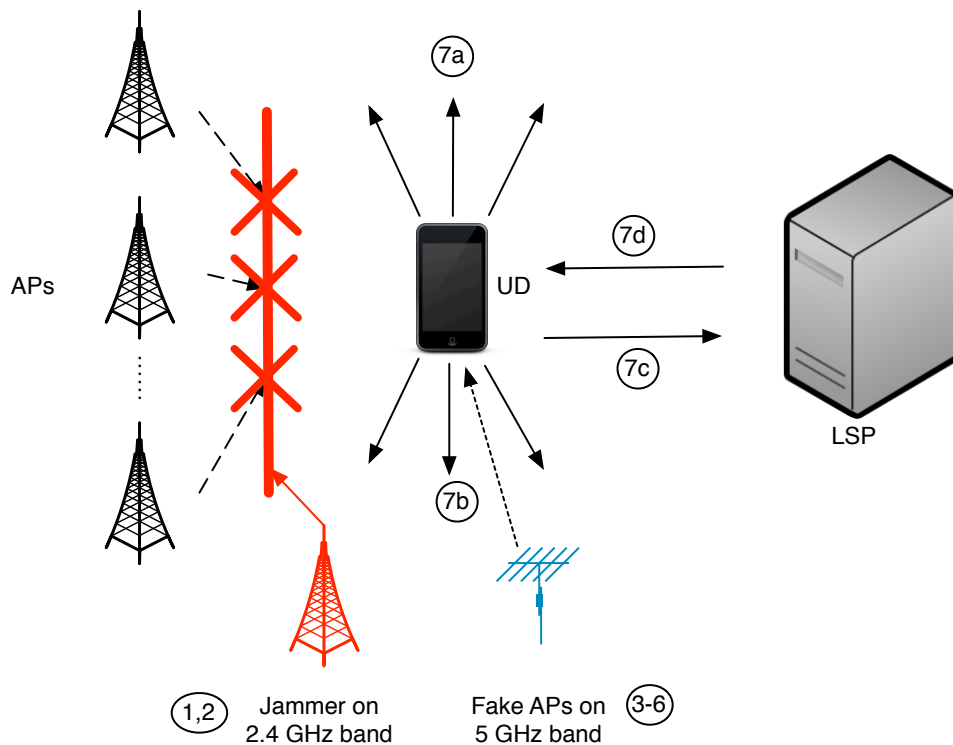


Figure 3.5: Location services attack procedure

A diagram of this attack is given in Figure 3.5. Again, the access points are denoted as APs, user device is denoted as UD, and location service provider is denoted as LSP.

3.3.2 Results of the Attack

The attack was performed on the four WiFi-based location service providers mentioned in Section 3.1: Skyhook, Google, Apple and Microsoft. For all of the WiFi-based location service providers, the first step of the attack was successful, that is, jamming all legitimate access points resulted in the device not being able to get a location, thus denying the service. For the second step of the attack, the attack was also successful on all four location service providers as they all displayed the location of the spoofed access points, which is a location in Bellevue, Washington, U.S. in this case and the attack was conducted in Toronto, Ontario, Canada. The result of this step is shown in Figure 3.6 for Google, Figure 3.7 for Skyhook, Figure 3.8 for Apple, and Figure 3.9 for Microsoft.

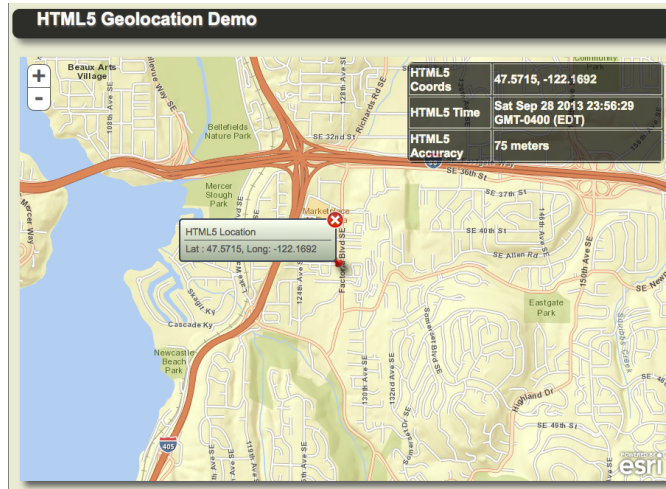


Figure 3.6: Access point impersonation attack on Google location service

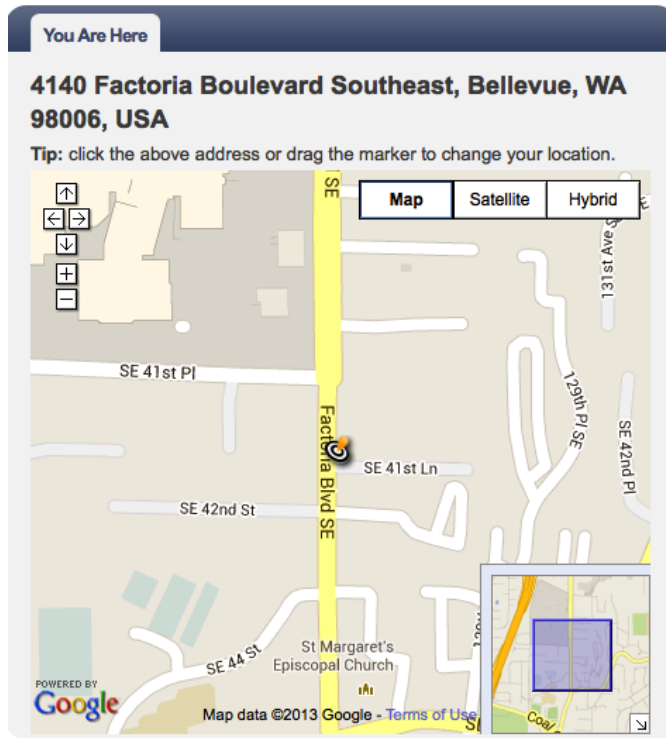


Figure 3.7: Access point impersonation attack on Skyhook location service

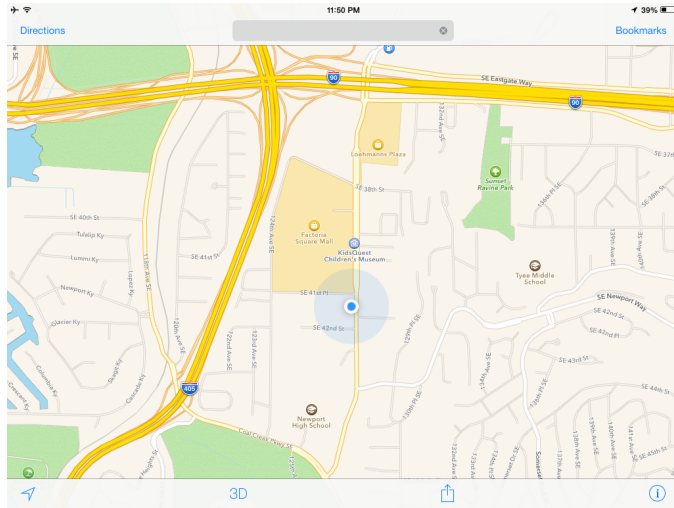


Figure 3.8: Access point impersonation attack on Apple location service

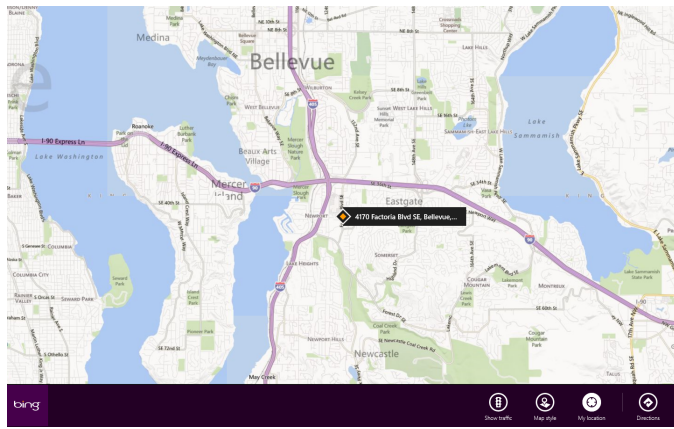


Figure 3.9: Access point impersonation attack on Microsoft location service

To summarize, in Section 3.1, an analysis was first conducted on some of the location service providers. This analysis showed that the unique identifiers used in determining location is the MAC address of access points. While the signal strength is also sent, that is more dependent on the radio hardware and the distance the client is from the access point. The client determines the signal strength when it scans for surrounding access points so it does not need to be spoofed. Next, in Section 3.2, we talked about the environment that the experiment was conducted in, which is a residential area consisting of many houses. A scan at one home in the residential area showed many broadcasting access points spread out over the 2.4 GHz. No 5 GHz band access points were available. In addition, the hardware and software used in the experiment were described, which consisted of virtual machines running Ubuntu Linux to control the USRP software defined radios, an embedded system for creating fake access points and several client devices to test each location service provider. Lastly, in Section 3.3, we outlined the procedure of the attack and showed the results of the attack, which was successful for all four major location service providers tested.

3.4 Impact of Attack on Location Services

With the gaining popularity of location services, there are a lot of applications that are affected by such an attack. This section provides an analysis on some of those applications and an analysis of the cost and how it compares to a previous attack on WiFi-based location service in [69]. Depending on the application, the effect of such an attack would range from being a minor inconvenience for the user to disabling the feature entirely.

3.4.1 Photo Geotagging

Photos contain a lot of metadata that can be used by photographers to perform processing. These kinds of information include the date and time, ISO, aperture size and lens. Photos can be tagged with GPS coordinates that show exactly where the photo was taken. This information can be useful to the user if they forgot where the picture was taken or possibly want to make a video based on the location information like in Apple's iPhoto that generates a video based on that information [1]. If such information contained in the photo is missing or wrong, then it would not be detrimental to the user, but would just cause some minor inconvenience.

3.4.2 Social Network Geotagging

When people make a post on a social network such as Facebook and Twitter, they have the option of providing a location with the post. Location services information would possibly add some value to the posts but it is not critical to the operation of the social networks so having a wrong location displayed would only be a minor annoyance. In extreme cases, a person can see the wrong location of a post made by a significant other and start questioning where that person is, but that is not a technology issue. Other than posts, services like Foursquare allow people to "check-in" to places they are at or have been (for example, a restaurant) based on location information [4]. Having wrong location information would cause minor inconvenience to the user as the user would need to input that information manually.

3.4.3 Location-based Weather

Weather applications can now provide weather information based on the location of the user, such as the application from the Weather Network [18]. This is a very convenient feature that allows a user to check local weather no matter where he/she is as long as the device has access to the Internet. Without location information, the user is forced to enter that information manually, which is a minor inconvenience.

3.4.4 Locating Friends

There are a number of services on the market that allows a user to share his/her location to those he/she gives permission, such as echoecho's location sharing service [3]. The whole service relies on location information so having no location information, or worse, having wrong location information, would make the service worse than useless.

3.4.5 Device Recovery Services

With the increasing use of mobile devices, the amount of thefts have gone up in many cities [57] [24] [63] since people carry them wherever they are. Also, mobile devices are small, so people can easily lose them, or forget them, for example, at a coffee shop. A number of device recovery services have been created to solve that problem, including Orbicule's Undercover service [10]. These services generally works as follows: an application is first installed on the device that a user wants to protect; then, when a device is lost or

stolen, the user would report that to the service provider; the application that is installed on the device frequently checks the service provider to see if the device is reported lost or stolen and begins transmitting information such as screenshots and location to the service provider; then the service provider attempts to recover the device for the user, for example, by contacting law enforcement in that area. It is clear that location information is key to recovering the device, and that information is used to call law enforcement so missing location information would make recovery harder, but wrong location information can cause law enforcement to go to the wrong place, which can impact innocent people.

3.4.6 Location-based Wallet Services

Passbook is an application on Apple’s iOS 6-based devices that allows users to put tickets, membership cards, coupons and more in one place [6]. This application can be location enabled so that, for example, when a person is in a store at which he/she has a membership card, the card will pop up on the lockscreen of the device to provide easy access to the card. Another example is people going to the movies with tickets in Passbook, and Passbook will automatically show a ticket available on the lockscreen to provide easy access to the ticket. If location information is missing or wrong, then the user still has the option of unlocking the device, opening Passbook, then select the membership card, or ticket or coupon that he/she wants, so it is just a minor inconvenience.

3.4.7 Location-based Reminders

Smartphones nowadays have reminders that can be based on location, like Apple’s Reminders application on iOS 6 devices [5]. This works by setting up a “geofence” around the area in which the user wants a reminder, and he/she can set the smartphone to remind them of something when they leave or enter this area. This can be used in conjunction with a time-based reminder. If location information is missing or wrong, then the reminder will not work, and a user relying on such reminder will miss the reminder, which can have big consequences depending on the reminder.

3.4.8 Search

One feature that has been added to search engines is the ability to find places nearby like Google [8]. This can be very convenient to a user who wants to find, for example, the closest

store that is open twenty-four hours. If location information is missing or wrong, then the user would need to input location information manually, which is a minor inconvenience.

3.4.9 Bluetooth and Bluetooth-based Location Service

Bluetooth operates in the same 2.4 GHz band as WiFi, so Bluetooth suffers the same issue as WiFi with respect to jamming. Also, Bluetooth only uses the 2.4 GHz band, so it is worse than WiFi in that it only uses one band instead of two in the WiFi case, which means it is easier to attack Bluetooth than WiFi. The impact of jamming Bluetooth is a little different than jamming WiFi. In WiFi, jamming would mainly affect the traffic to and from the access point and client devices since that is the main use of WiFi. Ad-hoc traffic will also be affected. In Bluetooth, different types of traffic is affected. Bluetooth is mainly used in short range applications [23] so devices like mice, keyboards and headsets would be affected when a jamming attack is performed on Bluetooth, and the author experienced such an issue while performing the experiment when the author's computer was not able to register any movements from the Bluetooth mouse that was paired with the computer.

One area of development in recent years is Bluetooth Low Energy or Bluetooth Smart [23]. This is a specification in Bluetooth version 4.0 that allows devices to communicate with very low energy, thus decreasing energy use and increasing battery life on mobile devices. There are many applications of this new specification, and a lot of companies have already come up with many devices to take advantage of the low power consumption, and many Bluetooth profiles have been created for the various types of applications. Companies have already started taking advantage of the low-power Bluetooth specification and have come up with products such as fitness trackers [17] and smart watches [15] that are connected to one's smartphone. Location services based on Bluetooth have also been proposed like in [30, 44, 61]. The combination of Bluetooth Low Energy and location services can potentially provide a very low-power indoor positioning system. However, as mentioned above, Bluetooth is also susceptible to jamming, and since WiFi and Bluetooth share a lot of the 2.4 GHz spectrum, jamming of WiFi also affects Bluetooth, allowing an attacker to disable both location service technologies.

3.5 Impact of Attack on WiFi Infrastructure

The attack carried out in this experiment not only affected location services but also WiFi infrastructure. With jamming, existing access points are blocked and clients are

denied access. With access point impersonation, unsuspecting users may be tricked into connecting to an access point controlled by the attacker and all the traffic going through that access point will be accessible to the attacker.

The experiment mainly focused on WiFi-based location services. However, there is one big difference between the implementation of the attack in this experiment than in [69] that makes the implementation more powerful. In [69], the authors wrote a custom script using a program called Scapy to impersonate the access points. Scapy is a tool that allows users to easily create various packets or frames, including 802.11 frames. With the proper WiFi card and modified drivers for the Linux platform, the user can use packet injection to send these custom frames. The authors in [69] wrote a script that listens to requests from clients asking for surrounding access point information. When it receives such a request, the script would create a fake response packet and sends it to the client. This script is limited in that not many services other than location services scans for surrounding access points so the use of this script in other attacks is limited. In the implementation done for this thesis, the fake access points are access points that are like any other access point with its own SSID and MAC address. Clients can connect to these access points, get an IP address and can communicate with others on the network and use the Internet. In [69], when clients actually try to connect to these access points, it would fail because the script was not written to accept clients, communicate with others on the network, or route traffic to the Internet, making that attack only suitable for location services. In the implementation done for this thesis, the fake access points can take over existing, legitimate access points and users would not notice. In the implementations in various operating systems [7, 11], the WiFi network manager remembers the SSID of various access points that the user connects to and will attempt to reconnect when it sees that SSID. This can be a big issue since a lot of users go to coffee shops or restaurants, for example, that have free WiFi. An unsuspecting user may connect to the fake access points and not realize that it is doing so, and all of the traffic can be collected by an adversary and the privacy and confidentiality of the user would be compromised. This attack was shown in [42].

Another important consequence of using actual access points for the attack is that it is much easier to set up fake access points than to write a script to send out broadcast beacon packets. A person with malicious intentions can easily set up fake access points on a Linux device with the right hardware, or get a router and put one of the several supported custom firmwares that are freely available on the Internet [9, 19, 21]. Since router setups are fairly user friendly with a web interface, it does not take much technical knowledge to set up a fake access point. However, in the attack in [69], the authors used a Scapy script, which requires knowledge of the language used to write Scapy programs and some knowledge of the parameters of a broadcast beacon packet. In order to use Scapy, the attacker needs

to get a compatible WiFi card, plus install a special driver or patch an existing driver to support packet injection, which takes even more effort and technical knowledge. So, the attack is much easier to implement, and users can carry out the attack with much less technical knowledge.

The cost of the experiment consist of two things: the cost for jamming, and the cost for access point impersonation. For jamming, we used USRPs and the total cost for these USRPs with daughterboards and antennas is thousands of dollars. They were used since they were readily available in the lab. However, USRPs are software defined radios. Their purpose is not specifically to jam WiFi but rather a platform for people to build and test radio enabled applications among possible other uses. There are specialized hardware out there on the market (possibly illegal) that is designed specifically for jamming WiFi and other wireless devices that operate in the 2.4 GHz for around a few hundred dollars or less [20], which is inexpensive, and require little technical knowledge to operate. For the people who have more technical knowledge, there are video tutorials (e.g. [56]) and open source designs for WiFi jammers (e.g. [41]) that are freely available with a simple search on the Internet and the attacker would just have to buy the parts and put them together. So, an attacker can perform jamming without much technical knowledge and at a relatively low cost. These devices can be very small, which means they can be hidden, making them hard to find.

For access point impersonation, there are a number of ways to achieve this. We used an embedded system with a WiFi card in the experiment because this hardware was readily available to us. Potential attackers can use the hardware that they already have, or buy new. Hardware like off-the-shelf routers or WiFi cards for desktops can all work. The requirement for these WiFi hardware is that they support customizable MAC address, and be able to act as an access point. To broadcast multiple access points, one can have multiple WiFi hardware or optionally a WiFi card or router that supports multiple wireless networks, which would definitely be more cost effective than having multiple hardware. The WiFi card in the embedded system we used for the experiment supports multiple wireless networks and was specifically picked for this reason.

Together, to perform an attack, an attacker would only need to purchase several hundred dollars worth of equipment. For those who have more technical knowledge, they may already have the hardware to perform the attack or they can follow the tutorials and build their own, which brings down the cost dramatically. So, a jamming and access point impersonation attack can be easily achieved.

3.6 Countermeasures to Attack

In this section, we present a number of countermeasures. They are using multiple frequency band ranges, using multiple location technologies and performing fingerprinting. These countermeasures are meant to increase the cost of an attack for the attacker rather than completely defend against the attack since it is not possible as of yet to mitigate jamming an entire band.

3.6.1 Use multiple bands

In the experiment, the environment consisted of access points spread across the entire 2.4 GHz band with no access points on the 5 GHz band, so to jam these access points, the author needed to jam the entire 2.4 GHz band. Two USRPs each with RFX2400 daughterboards were used to accomplish this, and the fake access points needed to be on a 5 GHz channel. However, many client devices are dual-band capable nowadays, meaning they can operate on the 2.4 GHz band or the 5 GHz band. One can take advantage of this feature to increase the cost of the attack by requiring access point information from both the 2.4 GHz band and 5 GHz band. The attacker would then need to jam legitimate access points on the 5 GHz band too if the attacker is to carry out the attack successfully on a client that is dual-band capable. The cost is incurred on the additional hardware that is needed to send the jamming signal, which means additional USRPs with a different daughterboard, or another jamming device. When a client is trying to use WiFi-based location services, the location service software should not display any messages if access points from both 2.4 GHz and 5 GHz bands are detected, and should display messages indicating less accuracy if only access points from one band are detected. Also, since 2.4 GHz bands are more common, location services software should look out for empty 2.4 GHz bands but non-empty 5 GHz band because it is a good indication of an attack like the one carried out in this experiment. This may change over time as more and more access points start using the 5 GHz band. This method is not foolproof as the attacker can choose to attack both bands, but the cost of the attack increases and effectiveness of the attack decreases if both bands are used.

The cost of implementing such change is low. It requires no modifications to hardware since a lot of devices are now dual-band capable. The change that needs to be made on the client side is in software. The algorithm for the entire locating process needs to be modified so that additional information is gathered when scanning for nearby access points. In addition to the MAC address and signal strength, the client should also collect band

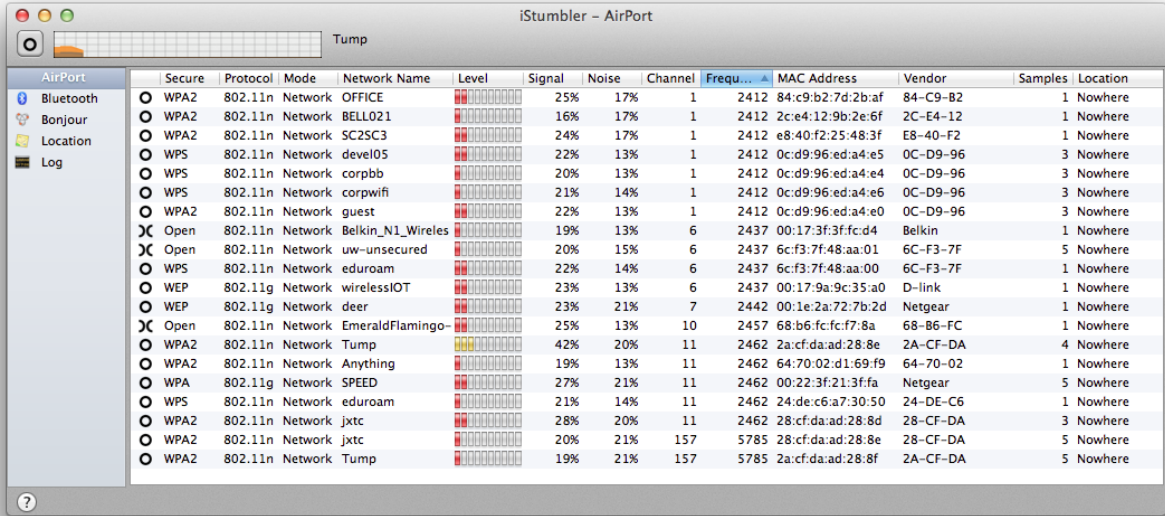


Figure 3.10: Scan for nearby access points

information, that is, whether the broadcasting access point is using the 2.4 GHz band or the 5 GHz band. If there is no mixture of 2.4 GHz and 5 GHz band access points, then the probability of an attack is higher than if access points exist on both bands and the client can display a message to the user indicating such a possibility, or at least point out that the location displayed may not be entirely accurate. This should be a trivial change in the algorithm as the scanning process already goes through all possible frequencies in all bands that the device is capable of, so for dual-band capable devices, access points broadcasting in both 2.4 GHz and 5 GHz bands can be seen as shown in Figure 3.10.

3.6.2 Use multiple location service technologies

There are three common types of location service technologies: GPS, WiFi and cellular. GPS is assumed to be unavailable since the experiment was carried out in an indoor environment so only WiFi and cellular service are available. Using both WiFi and cellular locating methods together will increase the cost of the attack, since an attacker now needs to jam all WiFi frequencies that legitimate access points use, and all frequencies that legitimate cellular networks in the surrounding area use. This means an attacker would need all

jamming equipment used in this experiment, plus jamming equipment for jamming the 5 GHz WiFi band, and cellular bands. Since there are more cellular bands than WiFi, the cost of the attack increases dramatically. Note that cellular-based location service method will likely be used as a verification of WiFi-based location service results since cellular-based location service is much less precise. This method is also not foolproof since the attacker can choose to jam all of the frequencies, and can potentially set up fake cellular base stations, which is available in open source [13], but the cost of the attack increases dramatically. Information from GPS can also be used if available to increase the cost of attack even more.

3.6.3 Fingerprinting

In [69], the authors proposed using unique access point characteristics such as traffic and signal fingerprints. The fingerprints would be stored in the service provider's location database as another parameter that the client needs to send to get accurate location information. If the fingerprint does not match, then no location data is returned, or the most likely location is returned but with a warning about the accuracy of the location.

We propose an additional characteristic to fingerprint, that is, the noise level. Such fingerprinting would be similar to that proposed for traffic and signal fingerprints in [69], where the location service provider would perform fingerprinting and store that information in the database. When clients make a location request, the noise level measurement similar to that described in [62] would be sent along with the MAC address, signal strength and the traffic and signal fingerprints. Fingerprinting of noise level can involve measuring the normal noise level, and interference that may occur in the environment from time to time. For example, in residential areas, since microwaves and some cordless phones operate in the 2.4 GHz band, this can be part of the fingerprinting process to create a more unique fingerprint.

3.6.4 Summary of countermeasures

Table 3.1 shows a summary of the countermeasures based on whether the countermeasure requires client or server side changes, cost for attacker and cost to implement.

One important thing to note is that a lot of devices on the market today are dual-band capable, and mobile devices, especially smartphones, already have multiple technologies built into the phone (WiFi, cellular and GPS) so most of the cost is the time, effort and

Table 3.1: Summary of countermeasures

	Multiple bands	Multiple technologies	Fingerprinting
Server side	Optional	Optional	Required
Client side	Required	Required	Required
Cost for attacker	Proportional to number of bands	Proportional to number of technologies and bands	Proportional to amount of area to attack
Cost to implement	Multi-band hardware + software update	Multi-technology hardware + software update	Collection of info + software update

money spent on updating the software algorithms to take advantage of the hardware and information gathered, making hardware costs negligible for a lot of devices.

3.7 Summary of Attack on WiFi-based Location Services

In this chapter, it was shown that Skyhook, Google, Apple and Microsoft location services can be denied and manipulated to provide the wrong location information, and an alternative implementation of the WiFi-based location service attack that is done on dual-band hardware was presented. The thesis also looked at many services that rely on location services and how the manipulation of location services can impact these services. Some services use location information just for added information while some rely on them to work. As a result of the attack on WiFi-based location services, WiFi infrastructure was also attacked, as legitimate access points are disabled and attacker-controlled access points are used. Users might unknowingly connect to these access points and any traffic passed between the access point and the user's device can be captured and/or manipulated by an adversary for malicious purposes, which makes the attack more powerful than in [69].

A number of countermeasures to the attack have also been proposed. These are using multiple bands, multiple location service technologies and fingerprinting. Many WiFi cards nowadays operate in both 2.4 GHz band and 5 GHz band so changes only need to be made in software to take advantage of it. Having more than one frequency range increases the jamming cost for the attacker. Smartphones now have WiFi, cellular and GPS technology built-in so each one can be used as verification of the others, which increases the cost

dramatically. Some devices can be expanded with new hardware to take advantage of multiple technologies. Lastly, fingerprinting also increases the cost dramatically for the attacker since the attacker needs to collect or fake fingerprint information, which is non-trivial.

The method of the attack used in the experiment can theoretically be applied to non-WiFi-based location services. As noticed during the experiment, the Bluetooth wireless mouse that was used to operate the computer was having issues registering movements on the computer, which shows that Bluetooth was also affected by jamming. The method can be used to gather Bluetooth location service data, create fake Bluetooth access points or beacons and then trick devices. In addition, spoofing attacks on GPS has already been accomplished like in [45], and it should not be very hard to apply the same method of attack to cellular-based location service, which uses cell tower triangulation to get an approximate location. So, with the cost and complexity of jamming plus impersonation attacks decreasing, it is imperative that more research is performed to provide secure solutions.

Chapter 4

Attack on SSL Using Proxy Servers

Proxy servers are optional to a network's infrastructure. They usually add some feature on top of what is already available. As detailed in Chapter 1, proxy servers can be used to save bandwidth, prevent some attacks, filter traffic and more. They are more commonly seen in network infrastructures that have a lot of users, like big companies and university campuses. Proxy servers accomplish these tasks by splitting up the connection between the client and server into two separate ones: one between the client and the proxy server and one between the proxy server and the destination server. The proxy server acts as a relay between the client and the server, and all data reaches the proxy server before going to the client or server.

SSL provides end-to-end protection of data. The client and server negotiates the key and cipher suite to be used for encryption and decryption, message integrity check and authentication. This is done on demand. When a client wants to connect to a server, it would initiate the SSL handshake with the server to negotiate the parameters for this specific session. These parameters may be cached for future sessions if desired.

There is a big problem when combining SSL with proxy servers. SSL provides end-to-end protection of data while proxy servers do not allow end-to-end connections and splits them up instead. Their goals are in direct contradiction of each other and that contradiction can be used to attack SSL. Putting SSL and proxy servers together can be part of a recipe for a man-in-the-middle attack. While proxy servers can be used for legitimate purposes, and are common in organizations, attackers can abuse this for malicious purposes. The process for using SSL with proxy servers is illustrated in figure 4.1. Notice in Figure 4.1 that the proxy server needs to decrypt and re-encrypt data as traffic passes through it whether it is from the client or server, which means the proxy has access to the plaintext

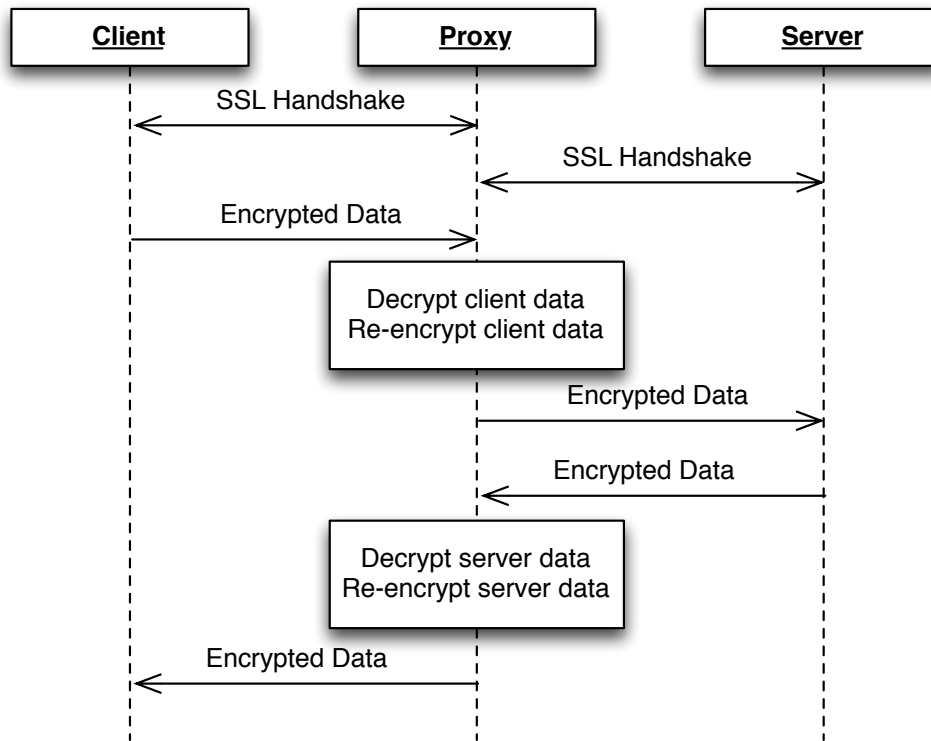


Figure 4.1: Using SSL with proxy server

of all communication that takes place between the client and server if SSL traffic is being proxied. If the proxy server is controlled by an attacker, then the attacker can use the captured plaintext for malicious purposes.

However, in order for the attack to work, there is a major requirement. Since SSL relies on certificate authorities in order to provide authentication, integrity and encryption, it needs to trust the proxy server before it will accept the connection. On modern browsers, if the proxy server's certificate is not trusted, the browser will display an error message warning the user that the certificate cannot be verified. In this chapter, an analysis is conducted on the various scenarios where an adversary can get users to trust (possibly unknowingly) the proxy server's certificate and accept the proxy configuration.

This chapter is organized as follows: Section 4.1 provides an analysis on Internet proxy servers; Section 4.2 gives an analysis and implementation on the attack in a corporate environment; Section 4.3 talks about an implementation and analysis of the attack on wireless hotspots; Section 4.4 shows some of the data that the author was able to capture

using this attack; Section 4.5 provides some detection and mitigation methods; and Section 4.6 gives a summary of the chapter.

4.1 Internet Proxy Servers

Internet proxy services are becoming more popular nowadays. They are more often used for the personal reasons listed in the introduction, that is, to hide one's identity and to access restricted content. An individual would want to hide his/her identity if he/she does not want a website to track the user, especially browsing habits and potentially personal information, or maybe the user does not want to be identified, or maybe some other reason. An individual would want to access restricted content if, for example, a website blocks devices other than the country that the website operates in from viewing the videos on the site due to copyright restrictions, or maybe a user is in a country where the government blocks certain websites and wants to get around it. These two reasons are becoming more serious with more and more websites wanting to track its users to obtain personally identifiable information so that it can target the webpage to each individual user, and with more and more websites preventing access based on location.

When a user wants to hide its identity or access restricted content, the user would sign up for a proxy service. Many of these services are free with possible restrictions surrounding its use but in most cases it serves the user's purpose. The service can provide many proxy servers, each at a different location to get around location blocking restrictions. A user would connect to the proxy server in the desired location and access the websites and content that are available only to that location or region.

Vendor proxy servers are a little more dangerous. Most devices sold on the market come with a lot of certificates that are trusted already. These are mostly Internet certificate authorities, but also include the device maker's certificate. Since the device maker is in control of the entire device package, from hardware to software and anything loaded onto the device, including proxy settings and trusted certificates, a vendor can take advantage of this and become the man-in-the-middle for all communications to and from its users' devices. This can be done without the user's interaction or knowledge since everything is preloaded onto the device and it would greatly compromise a user's confidentiality and privacy. While there is not a known case of this happening, it is definitely a possibility and great threat if this is actually the case.

Since this attack uses a proxy service provider on the Internet, traffic can only be captured by the proxy service provider so the attack is not implemented. The following sections show

a potential implementation, procedure and notes on such an attack if a user uses Internet proxy servers.

4.1.1 Attack Implementation

The setup for this attack is very simple. The setup for this attack include the following:

- Windows 7 acting as proxy client
- Proxy software running on the client

4.1.2 Attack Procedure

The idea of this attack is that the user signs up and willingly uses the proxy service, so any configuration comes directly from the proxy service provider. The user accepts the software that the proxy service provides to them. For the vendor proxy attack, users theoretically accept all the risks when they buy the device since no setup is needed as everything is done at manufacturing and assembly time. The steps to accomplish the Internet proxy server or vendor proxy server attack are given in Procedure 4.

Procedure 4 Internet proxy server attack

- 1: Sign up for proxy service
 - 2: Set up OS or browser to use Internet proxy or install service provider software
 - 3: Browse the web
-

4.1.3 Notes on Attack

In this attack, the user at the very least knows about the benefits of using proxy servers and willingly signs up to use the service. The user may not be aware of the costs and technical details of using the service and may not even care about those details. The problem with this is that by using the proxy service, the user chooses to accept all configuration changes required for the service to work, including proxy settings and trusted certificates. The user is basically entrusting the proxy service provider with all of the data that the user forwards to their proxy servers as a result of using the service, which can potentially include SSL traffic that can be decrypted by the proxy server.

So, the issues with entrusting the proxy service are twofold. First, the proxy service must be secure, meaning an attacker cannot easily attack the proxy service and gain any information on the users. Second, the proxy service provider must be trustworthy. Since the service provider has the ability to gain access to users' information and traffic, it better protect the users' privacy very well. This means not peaking at the decrypted SSL traffic for usernames and passwords, although having the data in plaintext is never a good thing. The service provider should also not sell that information to advertisers or hand over information to the government without proper justification and documentation. The user should be aware of these costs and make an informed decision on whether to use the proxy service and when to use it.

4.2 Corporate Proxy Servers

As mentioned previously, a lot of companies deploy proxy servers on their networks. Companies generally use proxy servers for security purposes, reducing bandwidth usage, and filtering malicious and inappropriate content as outlined above but they can also use them to spy on the employees if they want.

Since companies are in control of all devices that they give out to its employees, they can have the devices use the proxy server, and trust the proxy server or the certificate of the proxy server. Once the employees start to use the devices, all the traffic will be sent to the proxy server, where all the data is decrypted and re-encrypted and sent to the destination server. An employee who has access to the proxy server or a company executive that is curious will be able to examine all of the decrypted data to spy on its victims.

The following sections show the implementation, procedure and some notes on the attack.

4.2.1 Attack Implementation

Since this attack scenario is supposed to resemble a company network, a simplified setup of a corporate environment was used to show the ease with which a company can force you to use their proxy server so that they can read all the traffic that passes through it, including SSL traffic.

For the simplified corporate environment, the setup consists of the following:

- Windows Server 2003 acting as the domain controller and DNS Server

- Ubuntu 11.10 with a piece of software called sslmeat [16] acting as the proxy server
- Windows 7 acting as the clients of the Active Directory server

4.2.2 Attack Procedure

The premise for the corporate proxy attack scenario is that the company's IT administrators can push whatever configuration that they want onto the user's devices, so the procedure mainly consists of forcing these configurations onto the client device. The steps to carry out the attack are given in Procedure 5.

Procedure 5 Corporate proxy server attack

- 1: Push proxy configuration to client devices using Group Policy, which is shown in Figure 4.2
 - 2: Disable modification of proxy configuration using Group Policy, which is shown in Figure 4.3
 - 3: Trust the proxy server's root certificate on the domain controller, which is shown in Figure 4.4
 - 4: Browse the web
-

4.2.3 Notes on Attack

There are a few of things to note about the corporate proxy attack scenario.

For one, the proxy configuration and root certificate are pushed to the client devices because of IT policy, or potentially forced by company executives. No interaction is needed from the end users. In fact, non-tech-savvy users likely have no idea that their computers are configured this way, or even that these settings exist so they unknowingly send potentially sensitive information to the company when they are, for example, surfing the web during their break or lunch.

Secondly, IT administrators have the option to disable modification of proxy settings. This means end users are stuck with the proxy settings, even when they are home and does not want traffic to be monitored. The same can be said for the certificate that was forced onto the computer. If users are not aware of proxy settings or what they do, then they might forward their traffic to the company even if they are at home. This allows the company to also spy on what employees do at home with respect to what websites they

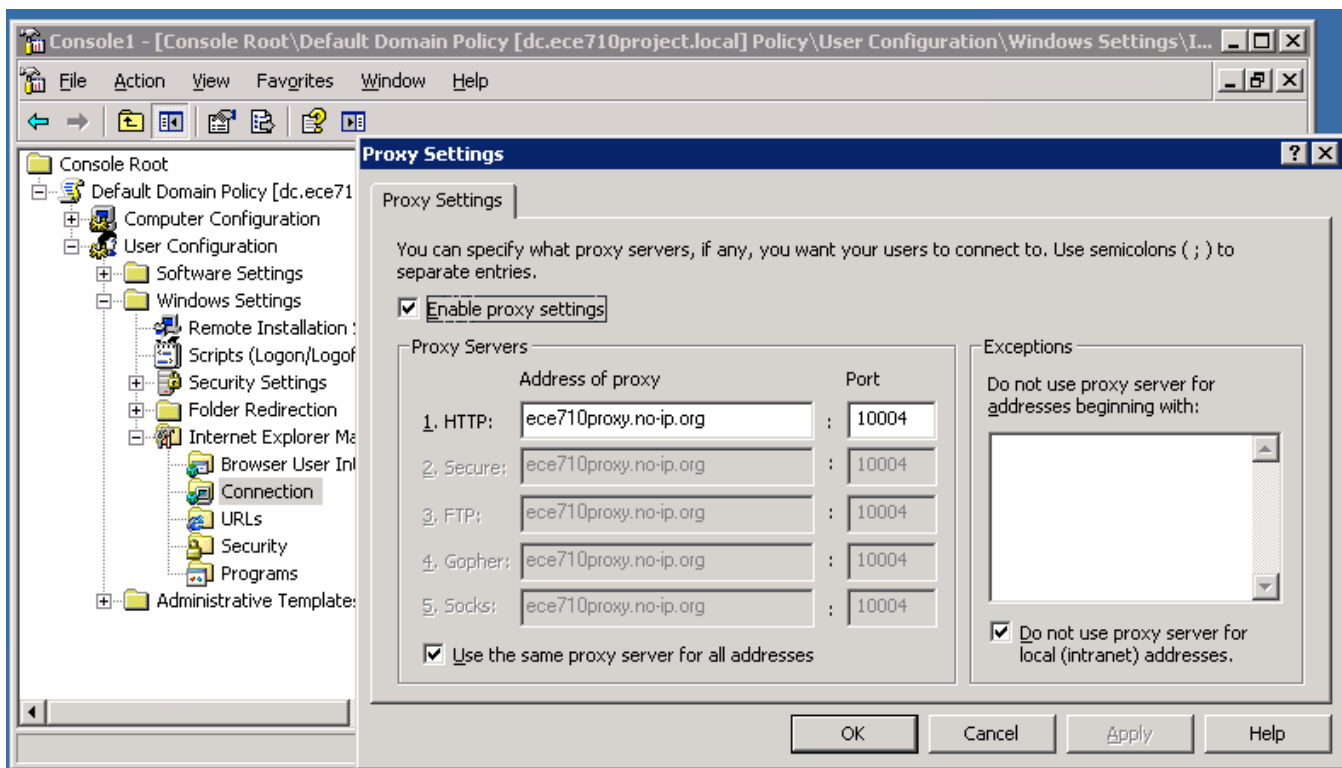


Figure 4.2: Setting proxy settings in Group Policy

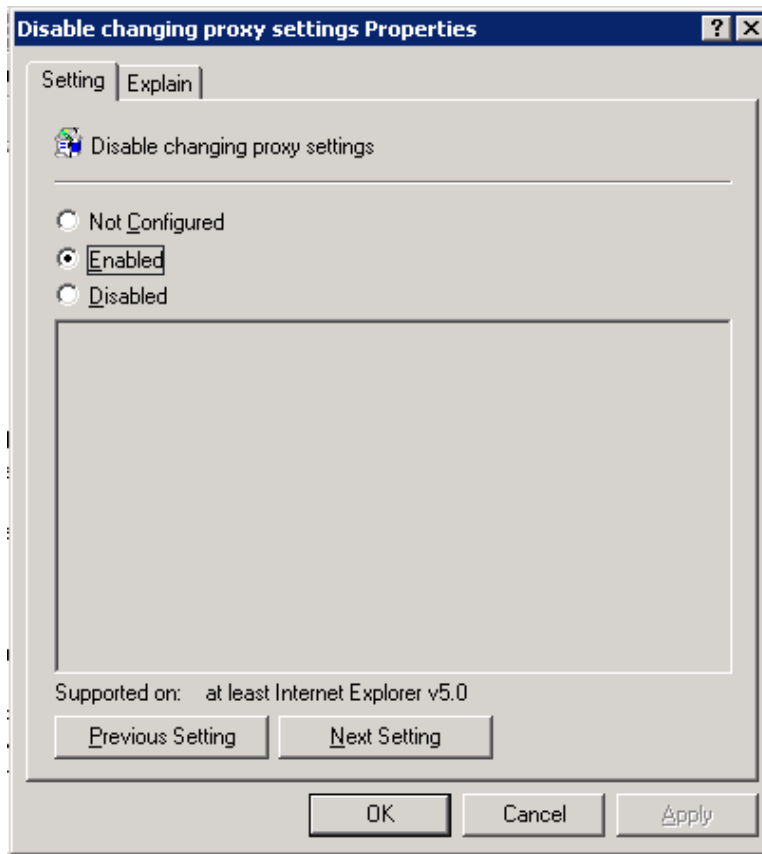


Figure 4.3: Disabling changing proxy settings in Group Policy

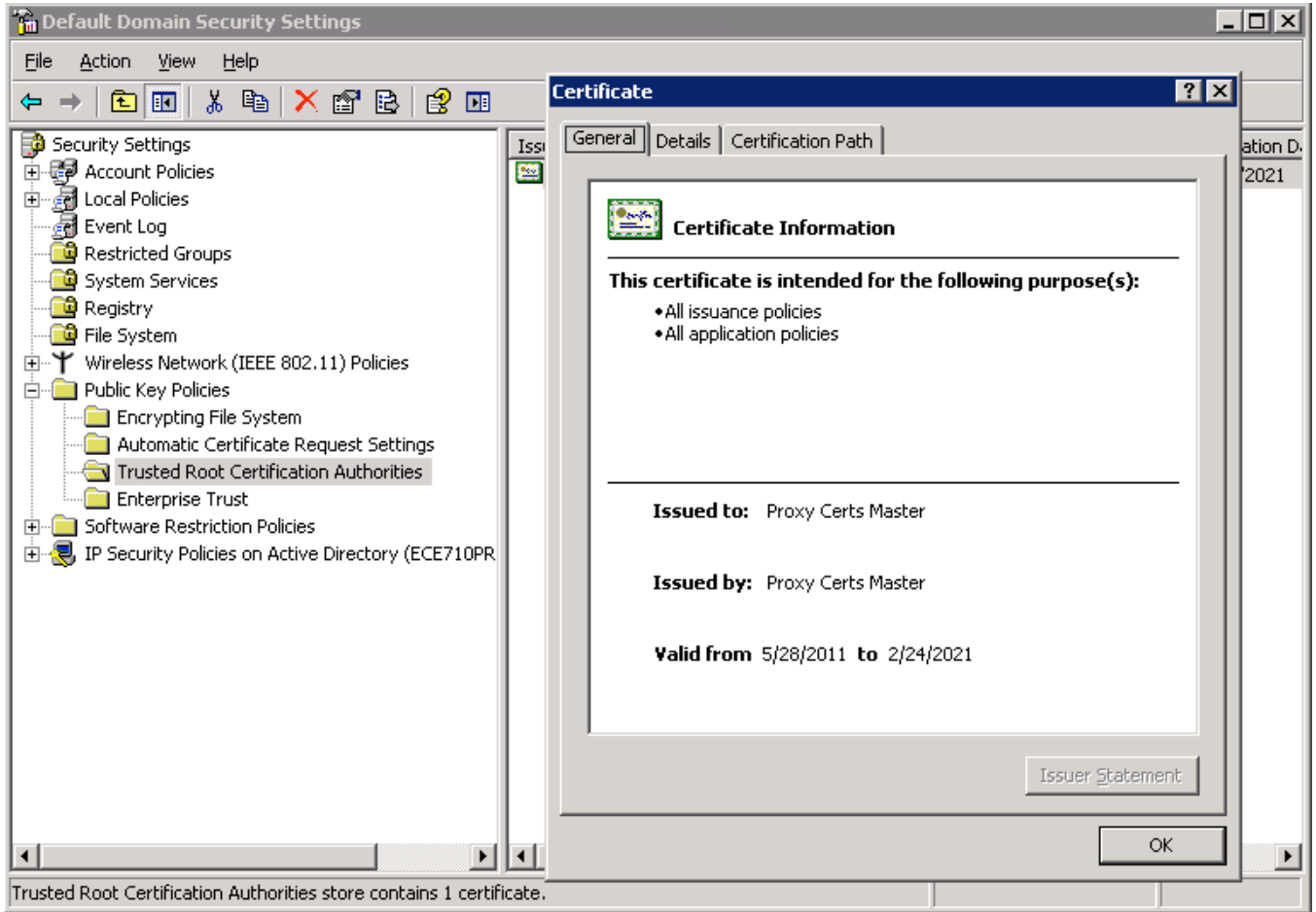


Figure 4.4: Trusting proxy server certificate for entire domain

visit, the usernames and passwords used on the websites and personal information after logging in.

So, users should be careful of using company devices for personal use. Company devices are loaded with company software, policies and settings, and one cannot be sure whether anything a person does outside of work can be performed securely without the company knowing even the tiniest detail. The safest choice is to not use company devices for personal use.

4.3 Wireless Hotspots

Wireless hotspots are generally deployed to attract shoppers by providing free Internet access in stores and restaurants, and sometimes guest Internet access in company offices. They are fairly cheap to deploy so stores are willing to incur the cost and gain the benefit of attracting shoppers. However, stores that operate these hotspots do not care about security since having free Internet access is already very appealing to users so many of the hotspots end up with no lower layer security which means an attacker can easily manipulate access points to carry out an attack. In addition to manipulating existing hotspots, an attacker can set up its own hotspots and masquerade them as legitimate hotspots set up by the stores to lure unsuspecting users.

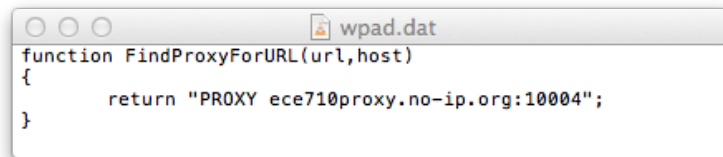
The following sections show the implementation, procedure and notes on the attack.

4.3.1 Attack Implementation

The idea of the attack is to create a fake hotspot that looks just like a legitimate hotspot so that users will not suspect that they are being attacked. The setup mainly consists of what one would use in a legitimate store that provides free wireless Internet access to its customers.

The setup for this environment consists of:

- Ubuntu 11.10 with a piece of software called sslmeat acting as the proxy server
- ALIX embedded system running PfSense operating system acting as the wireless access point, router and DHCP server
- Windows 7 acting as a hotspot user



```
function FindProxyForURL(url,host)
{
    return "PROXY ece710proxy.no-ip.org:10004";
}
```

Figure 4.5: Proxy configuration file

4.3.2 Attack Procedure

Legitimate hotspots likely do not use a proxy server as the providers want to keep the cost down as much as possible. Only an attacker, or a store that wants to spy on its customers would want to put in a proxy server. The main issue in a deploying proxy server is to set up a client device to use the proxy server, and for the client device to accept the root certificate of the proxy server, which is only needed to proxy SSL traffic. So, the procedure involves getting the configurations to the client device with as little interaction on the user's part as possible. The steps to carry out the attack is given in Procedure 6.

Procedure 6 Wireless hotspot with proxy server attack

- 1: Save proxy configuration to a file as shown in Figure 4.5 and put it on a web server
 - 2: Distribute the location of the proxy configuration file using DHCP (or DNS) as shown in Figure 4.6
 - 3: Block all incoming/outgoing traffic on firewall unless it is from/to the proxy server as shown in Figure 4.7
 - 4: Set up captive portal with instructions to trust the proxy server's root certificate as shown in Figure 4.8
 - 5: Connect to network and browse the web
-

Note that the first two steps of the procedure are part of what is called the Web Proxy Autodiscovery Protocol or WPAD and is a way for an operating system or browser to automatically detect and configure proxy settings [50].

4.3.3 Notes on Attack

The Web Proxy Autodiscovery Protocol is a protocol designed to help make it easier to distribute proxy settings. If any changes need to be made, the administrator would just make the appropriate changes to the central configuration file, and the next time the client

Additional BOOTP/DHCP Options

Enter the DHCP option number and the value for each item you would like to include in the DHCP lease information. For a list of available options please visit this URL

Number	Type	Value
252	String	"http://192.168.101.1/wpad.dat"
252	String	"http://192.168.101.1/wpad.da"
252	String	"http://192.168.101.1/proxy.pac"

Figure 4.6: DHCP option to distribute proxy configuration file

Firewall: Rules

S L ?

Floating WAN LAN GUEST

ID	Proto	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description
<input type="checkbox"/>	*	69.165.155.39	*	GUEST net	*	*	none		
<input type="checkbox"/>	*	GUEST net	*	69.165.155.39	*	*	none		

Figure 4.7: Firewall rules to only allow traffic to/from proxy

Services: Captive portal

Captive portal Pass-through MAC Allowed IP addresses Allowed Hostnames Vouchers File Manager

Enable captive portal

Interfaces

WAN
LAN
GUEST

Select the interface(s) to enable for captive portal.

Figure 4.8: Captive portal configuration

tries to connect, it will get the new settings and configure accordingly. For this to work, the operating system or browser must be instructed to automatically detect proxy settings. Old versions of operating systems such as Windows 7 and Windows XP turn this on by default. An attacker can leverage this to push proxy settings to client devices without them realizing and gain access to user data. Such an attack had already been used on a widespread basis [52].

However, pushing proxy settings is only enough to read HTTP traffic. SSL traffic cannot be read since the traffic is encrypted, which is why the procedure involves instructions to trust the proxy server's root certificate. This requires a little bit of social engineering. The captive portal webpage must look like something from a legitimate hotspot, and the process of adding the root certificate to the list of trusted certificates must be very simple or else the user will at least get frustrated and not use the hotspot, if not suspicious about what this step is for. One way that an attacker can potentially get the user to accept the certificate is to masquerade it as a program that checks the computer if it meets the requirements for connecting to the hotspot.

This attack is different from the other two attacks. In the corporate proxy attack scenario, the company is in charge of what the employees can and cannot do on company devices. The configurations and certificate are pushed to client devices and the user cannot do much about it. In the Internet proxy service attack scenario, the user chooses to use the proxy server for the benefits that it provides so the user accepts the terms of using the proxy service. In this attack, an attacker wants to gain access to a user's information without the user's suspicion, so it tries to look for ways to autoconfigure proxy settings without the user knowing and tricking the user to accept its certificate.

There are several variations to the attack. One of them is a DHCP starvation attack [46]. This allows the attacker to make the above attack work on legitimate hotspots also by taking over IP address management. The idea of the DHCP starvation attack is to exhaust the DHCP address pool of the legitimate access points by flooding the DHCP server with IP address requests. Once the DHCP server responds to all of these requests, the legitimate DHCP server will have run out of available IP addresses and will not respond to further requests. An attacker can take the responses as a result of the flooding and can now hand out these IP addresses to new clients that want to connect to the hotspot. It can add other DHCP options to, for example, inject proxy settings onto clients' computers or change DNS server IP addresses to attacker-selected ones.

Another variation is to use malware or an exploit. An attacker can write a malware that masquerades as a legitimate application and have the user to download it. All the attacker needs the malware to do is to set the proxy if necessary and add its certificate to

the list of trusted certificates, which can be done at first launch. After this, the application is just like any other legitimate application because no additional work needs to be done to carry out the attack. When an unsuspecting user downloads the application, it will override the settings on the device and the operating system or browser will read that setting and forward traffic to the attacker's proxy server. The attacker can also release a custom browser. That way, no settings need to be changed on the operating system, and as long as the user uses the attacker's browser, traffic will be forwarded to the attacker's proxy.

4.4 Captured Data

Once the encrypted data gets intercepted by the proxy server and decrypted, all the plaintext can be read. If an attacker is in control of the proxy server, then the attacker gets access to lots of personal data. This section shows some of the data that can be captured on various websites.

A dummy set of login credentials were used, namely an email of "test1234@test1234.com" and password of "testingpassword" and that is used on a number of popular websites to login. The login will of course fail but the username and password still needs to be submitted to the destination server for authentication so capturing the login details shows that a proxy server can capture such information that users normally assume is encrypted. Figures 4.9, 4.10, 4.11, 4.12, 4.13, 4.14 and 4.15 show the captured login credential submission for Facebook, Twitter, LinkedIn, Google, Live, iCloud and CIBC respectively. The captured credentials are highlighted in yellow for each of the figures. Note that for the CIBC login, a user logs in with its debit card number which is fixed length and its password is limited to twelve characters so the login credentials were truncated.

For the purposes of demonstrating the attack, only the login credentials were captured. However, an attacker can definitely intercept more information if he chooses. Information such as credit card numbers and anything else that can be displayed on a webpage can potentially be intercepted by an attacker with this attack. This was actually verified during the experiment but a screenshot of the result is omitted to protect confidential banking information.

```
POST /login.php?login_attempt=1 HTTP/1.1
Host: www.facebook.com
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Proxy-Connection: keep-alive
Accept-Encoding: gzip, deflate
Accept-Language: en-us
Content-Type: application/x-www-form-urlencoded
Origin: https://www.facebook.com
Content-Length: 146
Connection: keep-alive
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9) AppleWebKit/537.71 (KHTML, like Gecko) Version/7.0 Safari/537.71
Referer: https://www.facebook.com/
DNT: 1
Cookie: _e_0JdD_0=%5B%220JdD%22%2C1383165320840%2C%22act%22%2C1383165320840%2C1%2C%22https%3A%2F%2Fwww.facebook.com%2Flogin.php%3Flogin_attempt%3D1%22%2C%22form%22%2C%22submit%22%2C%22bluebar%22%2C%22r%22%2C%22%2F%22%2C%7B%22ft%22%3A%7B%7D%2C%22gt%22%3A%7B%7D%7D%2C0%2C0%2C0%2C%227usxq9%22%2C17%5D; act=1383165320840%2F1; datr=Y8BKUue1HQFIwthIw_rsfQ1t; fr=0ofzm0y3IvTFqiEcN.AWXpkz140NfyWguxlY4jjkGl4TA.BSTaNo.79.FJN.AWU1tpAZ; locale=en_US; lu=RAKGkxrzdtLEj4oa017A_BvA; reg_fb_gate=https%3A%2F%2Fwww.facebook.com%2F; reg_fb_ref=https%3A%2F%2Fwww.facebook.com%2F

lsd=AVovhtTK&email=test1234%40test1234.com&pass=testingpassword&default_persistent=0&timezone=240&lgnrnd=133511_zoUa&lgnjs=1383165311&locale=en_US
```

Figure 4.9: Facebook login

```

POST /sessions HTTP/1.1
Host: twitter.com
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Proxy-Connection: keep-alive
Accept-Encoding: gzip, deflate
Accept-Language: en-us
Content-Type: application/x-www-form-urlencoded
Origin: https://twitter.com
Content-Length: 208
Connection: keep-alive
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9) AppleWebKit/537.71 (KHTML, like Gecko) Version/7.0 Safari/537.71
Referer: https://twitter.com/
DNT: 1
Cookie: __utma=43838368.90843851.1381328629.1383101084.1383164902.11;
__utmb=43838368.3.10.1383164902; __utmc=43838368; __utmv=43838368.lang
%3A%20en; __utmz=43838368.1382717777.9.7.utmcsr=platform.twitter.com|
utmccn=(referral)|utmcmd=referral|utmctt=/widgets/follow_button.
1382126667.html;
_twitter_sess=BAh7CToPY3JlYXRlZF9hdGwrC0dfDQtCAToHawQiJWRkMGQ5NTljMmZi
ZDhk
%250AZTdjOTFjZjk5ZDIzYWI1ZjEzOgxcj3JmX2lkIiU2ZmU4ODZlNTA0ZTg1NTc1%250A
OGI3MTUxZTNiMzI1NzYzNSIKZmxhc2hJQzonQWN0aW9uQ29udHJvbGxlcjo6%250ARmxhc
2g60kZsYXNoSGFzaHsABjoKQHVzZWR7AA%253D
%253D--726ddc2a6cd0c317994bc2aa4f98db9622b936ab; dnt=1;
guest_id=v1%3A138132862695373823; remember_checked=0;
remember_checked_on=0; twll=l%3D1382717782;
external_referer=wbKK5bxUNDvtbUb7vTCRRc50Fx78YaBEWI0MBNo1fmgBdZahd1VFd
oZsiCKltm4A%7C1

session%5Busername_or_email%5D=test1234%40test1234.com&session
%5Bpassword
%5D=testingpassword&return_to_ssl=true&scribe_log=&redirect_after_logi
n=%2F&authenticity_token=352fa45969d1c3182b2b2f7c6e335982652461e7

```

Figure 4.10: Twitter login

```

POST /uas/login-submit HTTP/1.1
Host: www.linkedin.com
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Proxy-Connection: keep-alive
Accept-Encoding: gzip, deflate
Accept-Language: en-us
Content-Type: application/x-www-form-urlencoded
Origin: http://www.linkedin.com
Content-Length: 230
Connection: keep-alive
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9) AppleWebKit/537.71 (KHTML, like Gecko) Version/7.0 Safari/537.71
Referer: http://www.linkedin.com/
DNT: 1
Cookie:
bscookie="v=1&20131002185940819766f3-089f-4c08-8b0a-20e3c42faf83AQFPRZRjCc7Q0-dJuPxof1JVuFifw9Gk"; JSESSIONID="ajax:0704957546333493666";
_chartbeat2=ql8rwfpw4g5we19.1380740387514.1381970769227.1000000000000001;
_chartbeat_uuniq=1; L1e=203e6005;
leo_auth_token="GST:UNrUnZYjwY_aiE4vuGmUA9vfyYEDTd0J3EmdSSB7id6GZhbB5zUs36:1383165824:ac413218216f46c604e3970aa979e38275488673";
visit="v=1&M"; X-LI-IDC=C1; __qca=P0-1318933571-1380740390121;
__utma=23068709.72961629.1383165830.1383165830.1383165830.1;
__utmb=23068709.2.10.1383165830; __utmc=23068709;
__utmv=23068709.guest; __utmz=23068709.1383165830.1.1.utmcsr=(direct)|utmccn=(direct)|utmcmd=(none); _lipt="0_7-khuUbudTgq_3fWmaEu2gX80iLAa7u6BUQLUMKDj0aapf1qvq9BnFxvoJvASvQX4NclEch42i2pHJXBkBlfs1cMIOZQz2zu1zbRRSafpCyddDFdq0en9t4ZVUbPso5sJJIZCAn2KVYW_b01fpb0rc4vR7nwR8jK2iPQbe06chAmLmpBltx9PmG1sT0tl3F";
bcookie="v=2&8a509696-aa6c-4384-8798-e0ffa7b0b7f2"; lang="v=2&lang=en-us"; lidc="b=LB83:g=15:u=1:i=1383146493:t=1383232893:s=2973429160"

isJsEnabled=true&source_app=&session_key=test1234%40test1234.com&session_password=testingpassword&signin=Sign+In&session_redirect=&trk=&csrfToken=ajax%3A0704957546333493666&sourceAlias=0_7r5yezRXCiA_H0CRD8sf6Dh0jTKUNps5xGTqeX8EEoi

```

Figure 4.11: LinkedIn login

```

POST /ServiceLoginAuth HTTP/1.1
Host: accounts.google.com
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Proxy-Connection: keep-alive
Accept-Encoding: gzip, deflate
Accept-Language: en-us
Content-Type: application/x-www-form-urlencoded
Origin: https://accounts.google.com
Content-Length: 724
Connection: keep-alive
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9) AppleWebKit/537.71 (KHTML, like Gecko) Version/7.0 Safari/537.71
Referer: https://accounts.google.com/ServiceLoginAuth
DNT: 1
Cookie: GALX=96HM2okz5rk;
GAPS=1:6t09kIj1pxuye4ue3IT8ceJYElMSw:yd6RvCSljqenreqp;
GoogleAccountsLocale_session=en_GB;
GMAIL_LOGIN=T1383165141690/1383165141690/1383165234503;
GMAIL_RTT=1226;
NID=67=jmX_9YIufLKZpWBle62_AltQqRZyGFwZXTZVm7qMk9nZ2YP-xToTZN6LSwMPKtlWvjGYV6eYtLn15x-vo2hyhjtKNLzyqoF1nPNMzcmUV2Uyl9FAyXgQmbRdBanp1Lr2nCcjuF63aVwdDD4ghg;
PREF=ID=24fe12addcb25203:U=38b31bf183c0a776:FF=0:LD=en:TM=1380629856:LM=1380654789:S=DEeNNWJK1zLVW9eD

GALX=96HM2okz5rk&continue=https%3A%2F%2Fmail.google.com%2Fmail%2F&service=mail&rm=false&ltmpl=default&ssc=1&ss=1&checkedDomains=youtu
ube&checkConnection=youtube%3A1599%3A1&pstMsg=1&_utf8=
%E2%98%83&bgresponse=
%21A0KtmMI2NzmSNEQCIicE6pCAUAIAAAD0UgAAAqAN9E8nb-
BeaQpkyZpzZSghfqsDCXkY3DwKtQEG9_5KpyNB0bRvPC5ecdYqp7_Mpa9pylgUIiULngMN
PH7sMsfjDsJRjNZoR0aBJGiuPuVFx8jU4iFskLDE-
Maxe75CjlbQjRXg4fanUTv29A58QnUyp8KIwaNaWdYCGLGPPH_auQf0F8wCjICfvhJtKsT
-xPtTHfdSvQF2R55nXx3fJKhBS6jdsR5wPWGk8JKS7yJKTF-
l9M0zMTMJat_yma0kDiB0oLgbKNnDpaQ8n_080Bh4sCji-
tnSBPW5yUq5Kfsax0&pstMsg=1&dnConn=&checkConnection=youtube
%3A1363%3A1&checkedDomains=youtube&Email=test1234%40test1234.com&Passw
d=testingpassword&signIn=Sign+in&PersistentCookie=yes&rmShown=1

```

Figure 4.12: Google login

```

POST /ppsecure/post.srf?
wa=wsignin1.0&ct=1383165398&rver=6.1.6206.0&sa=1&ntprob=-1&wp=MBI_SSL_
SHARED&wreply=https:%2F%2Fmail.live.com%2F%3Fowa%3D1%26owasuffix%3Dowa
%252f&id=64855&snsc=1&cbcxt=mail&bk=1383165399 HTTP/1.1
Host: login.live.com
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/
*;q=0.8
Proxy-Connection: keep-alive
Accept-Encoding: gzip, deflate
Accept-Language: en-us
Content-Type: application/x-www-form-urlencoded
Origin: https://login.live.com
Content-Length: 516
Connection: keep-alive
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9) AppleWebKit/
537.71 (KHTML, like Gecko) Version/7.0 Safari/537.71
Referer: https://login.live.com/login.srf?
wa=wsignin1.0&ct=1383165398&rver=6.1.6206.0&sa=1&ntprob=-1&wp=MBI_SSL_
SHARED&wreply=https:%2F%2Fmail.live.com%2F%3Fowa%3D1%26owasuffix%3Dowa
%252f&id=64855&snsc=1&cbcxt=mail
DNT: 1
Cookie: MSP0K=$uuid-fc7b9248-4a0c-4667-8ea2-0b0b77e5304b;
CkTst=G1383165401134; MSPReq=lt=1383165399&co=1&id=64855;
mbox=check#true#1383165463|session#1383165402815-547487#1383167263|
PC#1383165402815-547487.20_25#1384375005;
wldidperf=throughput=4&latency=847&FR=L&ST=1383165415296

login=test1234%40test1234.com&passwd=testingpassword&SI=Sign
+in&type=11&PPFT=CgpWat0KtzDQGjd%21frBlXnnjGIGBoDT7d6811p*5JB80vTTLxm
%21FwASLVSL
%21%21ISwoCqSpPhPUsatRHpja0SNYdQxAu1arMgBpW546Uz3TSNvyQF9HCfHedRK0b9*
Q%21ocD2IT40MrDrmYDG3hQV*EB
%21fT4*dEwfMjKe3rkmhL0rQ83utjeUxsFrWt0v7JuIJCjEAVqATHn*Dby0wXEA*UvKYTh
AuxkFASJmRwxXYij696KrWRJt*wMC%216o1KJid04A
%24%24&PPSX=Passp&idsbho=1&sso=0&NewUser=1&LoginOptions=3&i1=0&i2=1&i3
=13868&i4=0&i7=0&i12=1&i13=0&i14=603&i15=5904&i17=0&i18=__Login_String
s%7C1%2C__Login_Core%7C1%2C

```

Figure 4.13: Live login

```
POST /setup/ws/1/login?clientBuildNumber=1T.
111208&clientId=A09C1A79-3909-4D72-9E06-1ADAD43F947F HTTP/1.1
Host: setup.icloud.com
Accept: */*
Proxy-Connection: keep-alive
Accept-Encoding: gzip, deflate
Accept-Language: en-us
Content-Type: text/plain
Origin: https://www.icloud.com
Content-Length: 88
Connection: keep-alive
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9) AppleWebKit/
537.71 (KHTML, like Gecko) Version/7.0 Safari/537.71
Referer: https://www.icloud.com/
DNT: 1
Cookie: NSC_q04-
tfuvqtfswjdf=ffffffff1296081045525d5f4f58455e445a4a422971

{"apple_id":"test1234@test1234.com","password":"testingpassword","exte
nded_login":false}
```

Figure 4.14: iCloud login

4.5 Detection and Mitigation of Attack on SSL Using Proxy Servers

4.5.1 Detection

If the attack is successful, then it is fairly hard to notice any issues. This is because the active part of the attack only needs to happen at the beginning, that is, setting the proxy to point to the attacker's server, and accepting the certificates. After this is done, the operating system or browser reads the configuration and applies them by forwarding traffic to the proxy server, which is the passive part of the attack. The forwarding of traffic to a configured proxy is normal behaviour. It does not break any protocol or is not part of an exploit so everything looks just the way it is supposed to on the user's device.

There may be small things that a user may notice that can cause suspicion, especially when a user does not normally use a proxy server. On the client side, a user may notice that some websites not loading properly. This happens on websites with Java applets that require a direct connection, and Java applets fail to load if a proxy server is used. There


```
POST /olbtxn/authentication/SignOn.cibc HTTP/1.1
Host: www.cibconline.cibc.com
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.9; rv:25.0)
Gecko/20100101 Firefox/25.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
DNT: 1
Referer: https://www.cibconline.cibc.com/olbtxn/authentication/
PreSignOn.cibc?inline=true&locale=en_CA&random=0.9629369235851564
Cookie: cibc.locale.transient=en_CA;
JSESSIONID=JrQYSxXBRNvSl0znxNchTnf1YcM7mzJj1KG6vxVGzwpTy31K53PD!
1719438066;
securityUID=JrQYSxXBRNvSl0znxNchTnf1YcM7mzJj1KG6vxVGzwpTy31K53PD!
1719438066!1383192481721; trackCookie=99370212151893010; locCookie=t;
CIBC_Marketing0=lang%3Aen_CA; CIBC_Contrast=0; CIBC_TextSize=0;
atgPlatoStop=1; toggleContactInfo=true; ssoToken=; ssoIndicator=;
ssoSessionId=; fs_nocache_guid=8DBA495C220B6382BA62484825AAB7D7;
showtrusteer=1; checker=exit
Connection: keep-alive
Content-Type: application/x-www-form-urlencoded
Content-Length: 182

isPersistentCookieDisabled=0&reload=false&selIndex=-1&popupcheck=false
&securityUID=&inlineDeleteCard=&ingressURL=%2Fca
%2Fpersonal.html&newCardNumber=test1234&pswPassword=testingpassw
```

Figure 4.15: CIBC login

may be a higher delay than usual if a proxy server is used since two separate connections are created, although the difference may not be significant enough to be noticeable. Lastly, if the proxy server is under heavy load, then slowdowns may be significant enough for the user to notice. To detect the attack, a tech-savvy user would use Wireshark to sniff the traffic coming in and out of the client device. On the server side, the administrator can analyze the IP's visiting the website. If a lot of traffic is coming from a single IP then there may potentially be a problem, although if a proxy server is behind a router then traffic would come from a single IP anyways so this method is not foolproof.

4.5.2 Mitigation

The attack is fairly easy to stop once it is detected, unless a malware or exploit is used. Since the attack works by setting the proxy server and accepting a certificate, stopping the attack simply involves removing the proxy settings and the certificate. There are several ways to stop the client from using the proxy server. The simplest thing to do is to remove proxy settings from the operating system and/or browser by going into their settings. However, if the client device is a corporate device, then IT policies may prevent the user from making modifications. In this case, the user can prevent the proxy from working by blocking the proxy from the operating system's firewall or separate firewall if available. The best thing to do in this case, as already mentioned, is to not use corporate devices for personal use. The last thing that one can do is to stop using automatic proxy configuration so that there will be no surprises. Again, on a corporate device, this workaround may not be possible.

4.6 Summary of Attack on SSL Using Proxy Servers

In this chapter, three different ways of compromising SSL traffic using proxies were analyzed and two of them were implemented. Devices that were part of the network infrastructure were utilized to assist in the attack by setting additional parameters that are not frequently used but are implemented and part of the protocol standard. In the corporate proxy case, the settings were forced onto the client devices so users have no choice but to accept them. In the Internet proxy service case, the user chooses to use the service so they accept the benefits and risks, likely unaware of the consequences. In the wireless hotspot case, the attacker tries to trick the user into accepting the proxy settings and certificate. This goes to show that the attack can work in an environment where protocols are working exactly

the way it is suppose to, where users are not aware of the consequences of using a service, and where an attacker tries to trick or force users.

The reason that the attack works is because of the opposing purposes of SSL and proxy servers. SSL provides end-to-end encrypted connections while proxy servers prevents end-to-end connections. So, using SSL with proxies means breaking up SSL connections into two separate ones, where the client and proxy server share one connection, and therefore one pair of keys, and the proxy server and the destination server share another connection, and therefore another pair of keys. Traffic from the client to the proxy server is decrypted at the proxy server and re-encrypted and sent to the destination and vice versa, so the proxy server serves as a man-in-the-middle that can intercept all SSL traffic. An attacker can use this for malicious purposes if it can trick or force users to use its own proxy server. A company can force these settings on its devices so the users have no choice, and the company is able to read all of its employees' activities that are supposed to be encrypted over SSL.

To avoid traffic from being read by proxies, there are a few things that users can do. Users should be aware of their company policy and not use company devices for personal use. Users should make sure automatic proxy configuration is turned off to prevent an attacker from autoconfiguring proxy settings for them. Lastly, users should never trust a certificate blindly and should always be vigilant of applications downloaded from the Internet.

Chapter 5

Conclusions and Future Work

This chapter summarizes the contributions of this thesis and suggests potential future research.

5.1 Contributions

In Chapter 3, an analysis was first done on four major location service providers to see the format of their location query and what unique information are sent to the service provider. Then the environment in which the attack was performed and the equipment used was described. The implementation and results of the attack were shown next. The implementation used USRP software defined radios for jamming and an actual WiFi access point to pose as fake access points. The results of the attack showed that all four major location service providers on which the attack was performed were vulnerable to the attack. To the author's knowledge, this was the first time that four major location service providers were attacked together and the results gathered in a single document. An analysis was performed on the effects of the attack on location services and WiFi infrastructure. A lot of services nowadays rely on location services and some major functionality would be lost if location service is compromised while some applications would lose some valuable information but can still operate without that information. An analysis was also done on the effects of the attack on WiFi infrastructure as legitimate access points are compromised and fake access points are put in place. This analysis showed a more serious threat to WiFi infrastructure as a whole. There are actually three main reasons why the attack is a serious problem. First, since a regular user of WiFi would not be able to notice such an attack, an attacker can potentially gain lots of confidential information by performing such an

attack on an area with lots of WiFi traffic. Second, while the equipment used for the experiment consists of USRPs that costs thousands of dollars and an access point that cost over one hundred dollars, a survey of devices that are available for purchase on the Internet shows that WiFi jammers can be purchased for several hundred dollars and a less powerful access point could be used that cost approximately half the price used for the experiment. The USRPs and the embedded system acting as an access point were used only because they were readily available to the author. This shows that the attack can be carried out with relatively low cost. Lastly, the attack can be performed without a lot of technical knowledge. A WiFi jammer can be used to jam WiFi signals with some batteries or power supply and the flip of a switch, and access points available to regular consumers are fairly easy to set up, making the attack much more accessible to a wider range of people. For those who have a little more technical knowledge, they can build their own jammer from open source designs and cheap components, and re-use existing computers to act as fake access points. A number of countermeasures were proposed. These include using access point information from multiple WiFi bands which increases the cost for the attacker, using information from multiple location service technologies to verify one another which increases the cost for the attacker by a lot, and using fingerprinting of unique information around WiFi access points which increases the cost for the attacker dramatically. These countermeasures will not necessarily mitigate attacks completely but using these countermeasures will make the cost and complexity of the attack much higher. Given the increased adoption and value provided by the use of location services, more research should be conducted to ensure security while remaining convenient for the user.

In Chapter 4, an analysis was first performed on the effect of passing SSL traffic through a proxy server. After making the observation that SSL traffic needs to be decrypted at the proxy server and thus can be used in a man-in-the-middle attack, a number of attack scenarios were analyzed that allows a potential attacker to carry out a man-in-the-middle attack against SSL using proxy servers. The first scenario was Internet proxy servers, where the proxy server is run by some service provider or possibly device vendor on the Internet. In this scenario, the user either knowingly sets up the service on the device along with certificates or the device is preloaded with the configuration and certificates, so the user needs to be aware of the benefits and costs of using the service or device. The second scenario was the corporate proxy. In an organization, devices are managed by the IT department. Any setting could be determined and forced onto the device by the IT department and there is nothing that the user can do about it, so users should avoid using company devices for sensitive personal matters. The last scenario analyzed was wireless hotspots. This is a very common scenario nowadays with many stores and restaurants offering free WiFi. The provider, or more likely, an attacker can sit on the existing network,

and perform the attack. The attacker can use social engineering techniques to get users to accept its proxy configuration and certificates, use proxy autoconfiguration to force proxy settings on the user and trick the user to accept the certificate, or use malware to set proxy configuration and install certificates without the user's knowledge. The kinds of information that an attacker can get access to were shown next, and demonstrates that a lot of sensitive information, including login credentials and banking information, inside SSL traffic can be compromised using this attack. Some methods to detect and mitigate the attack were introduced. For detection, small things such as websites not loading properly, some third-party applets not working or slower than normal loading times may be a cause for concern. For those with more technical knowledge, Wireshark can be used to see where traffic is going to. For mitigation, the user needs to remove the proxy configuration and certificate if possible. If not, then the user would have to block the proxy server. Also, users should disable proxy autoconfiguration since this sets the proxy automatically for the user without any interaction and can be an avenue for attack.

5.2 Future Work

5.2.1 Extending Attack on WiFi-based Location Services

The author believes that the attack on WiFi-based location service can be extended to cellular-based location services. The method used for the WiFi-based attack was to jam legitimate access points then set up fake access points. To adapt this to the cellular case, an attacker would jam cellular base stations and set up fake base stations. The jamming part could be done with USRPs again but with a different daughterboard to match the cellular frequency that the attacker wants to jam. A cellular jammer could also be used. A fake base station can be set up using OpenBTS [13], which is an open-source GSM base station that can run on a modern computer with USRP. There are also commercially available base stations available on the market but their cost prohibits most people from using them. If the attack is successful on cellular-based location services, then three methods for determining location have been compromised and shows the urgency of creating more secure location service experience. There are major issues that need to be overcome to perform the attack. The cost of the attack increases a lot since setting up fake base stations require expensive hardware. The complexity of attack also increases a lot since setting up OpenBTS is not a trivial task and requires a lot of technical knowledge. Also, cellphones that implement location services are generally smartphones that are multi-band, meaning they are able to use many cellular bands, so to jam legitimate base stations

involves jamming all these frequencies. Lastly, cellular spectrum is licensed so performing this attack on a live environment is very likely illegal.

5.2.2 Extending Attack on SSL Using Proxy Servers

In this thesis, an analysis on the types of scenarios where encrypted SSL traffic could be compromised using proxy servers was performed and some of the scenarios implemented, specifically, the corporate proxy server scenario and social engineering scenario with proxy autoconfiguration. In future work, more implementations of the attack can be done or possibly combined with other works. For example, malware can be written to trick users into downloading something that poses as a legitimate application but in fact configures proxy settings and installs certificates. Proxy autoconfiguration can be used as an alternate implementation to ARP poisoning in `sslstrip` [54] to replace HTTPS links on HTTP webpages with HTTP links.

In addition to more implementations, an analysis of Internet proxy servers should be performed. The reason is that service providers claim to provide many features with its service, such as anonymity and also do not store data about its users, and more. However, no one outside of the service provider really knows how these services work. The fact is that the service providers are the man-in-the-middle and is able to read all the traffic that passes through their service, and the mere possibility that the service provider can read the data provides potential for abuse. An analysis of how the service providers work will provide a better understanding of the benefits and costs of using these services and puts the service providers' claims to the test.

References

- [1] Apple (Canada) - iPhoto - New full-screen views, emailing photos, and more.
- [2] Designing Wi-Fi in the 5 GHz band — Metageek.
- [3] echoecho.me - find : meet : chat.
- [4] Foursquare.
- [5] iOS 6: About Reminders.
- [6] iOS: Using Pssbook.
- [7] Manage wireless network profiles.
- [8] My Location - Toolbar Help.
- [9] OpenWrt.
- [10] Orbicule — Undercover — Mac — How it works.
- [11] OS X Mountain Lion: Choose preferred Wi-Fi networks.
- [12] Radio Channel Frequencies.
- [13] rangepublic.
- [14] Scapy.
- [15] Smartwatch by Metawatch.
- [16] sslmeat - A HTTP and SSL man-in-the-middle proxy - Google Project Hosting.
- [17] UP by Jawbone — Activity Tracker — Band + App for a Healthier You.

- [18] Weather Apps & Services - The Weather Network.
- [19] Welcome to the Tomato USB web site - TomatoUSB.
- [20] Wholesale Wifi Jammer-Buy Wifi Jammer lots from China Wifi Jammer wholesalers on Aliexpress.com.
- [21] www.dd-wrt.com — Unleash Your Router.
- [22] Unapproved Draft Standard for Information Technology- Telecommunications and information exchange between systems- Local and metropolitan area network- Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications. (This document reflects the combining of the 2003 Edition of 802.11 plus the 802.11g, 802.11h, 802.11i and 802.11j Amendments) (Revision of IEEE Std 802.11-1999) (Superseded by P802.11-REVma_D8.0). *IEEE Std P802.11-REVma/D7.0*, pages –, 2006.
- [23] Specification of the Bluetooth System. Technical report, Bluetooth SIG Inc., June 2010.
- [24] Metro - About Metro - News - Metro Transit Police report increase in electronic device thefts, urge public caution, August 2013.
- [25] What Is My IP Address. What is an ip address?, 2013.
- [26] Nadhem J. AlFardan, Daniel J. Bernstein, Kenneth G. Paterson, Bertram Poettering, and Jacob C. N. Schuldt. On the security of rc4 in tls. In *Proceedings of the 22Nd USENIX Conference on Security, SEC'13*, pages 305–320, Berkeley, CA, USA, 2013. USENIX Association.
- [27] H. Amir. Why johnny can't surf (safely)? attacks and defenses for web users. *Computers & Security*, 28(1-2):63–71, 2009.
- [28] arkasha and bobzilla. WiGLE - Wireless Geographic Logging Engine - Plotting WiFi on Maps.
- [29] G. Ayres, R. Mehmood, K. Mitchell, and N.J.P. Race. Localization to enhance security and services in wi-fi networks under privacy constraints. In R. Mehmood, E. Cerqueira, R. Piesiewicz, and I. Chlamtac, editors, *Communications Infrastructure. Systems and Applications in Europe*, volume 16 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, pages 175–188. Springer Berlin Heidelberg, 2009.

- [30] M.Z. Barahim, M.R. Doomun, and N. Joomun. Low-cost bluetooth mobile positioning for location-based application. In *Internet, 2007. ICI 2007. 3rd IEEE/IFIP International Conference in Central Asia on*, pages 1–4, 2007.
- [31] S. Capkun, M. Cagalj, and M. Srivastava. Secure localization with hidden and mobile base stations. In *INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings*, pages 1–10, 2006.
- [32] L. Chen and G. Gong. Chapman & Hall/CRC cryptography and network security. Boca Raton, FL : CRC Press.
- [33] D.D. Coleman and D.A. Westcott. *CWNA Certified Wireless Network Administrator Study Guide*. Wiley Publishing, Inc., Indianapolis, Indiana, 2004.
- [34] M. Conti, N. Dragoni, and S. Gottardo. Mithys: Mind the hand you shake - protecting mobile devices from ssl usage vulnerabilities. *CoRR*, abs/1306.6729, 2013.
- [35] Wikipedia contributors. Dynamic Host Configuration Protocol, 2013.
- [36] Wikipedia contributors. Proxy server, 2013.
- [37] P. Dandumont. Don't trust geolocation ! — le journal du lapin.
- [38] A. El-Rabbany. *Introduction to GPS : the Global Positioning System*. Boston, MA : Artech House, 2006.
- [39] Nadhem J. Al Fardan and Kenneth G. Paterson. Lucky thirteen: Breaking the tls and dtls record protocols. *2012 IEEE Symposium on Security and Privacy*, 0:526–540, 2013.
- [40] F. Ferreri, M. Bernaschi, and L. Valcamonici. Access points vulnerabilities to dos attacks in 802.11 networks. In *Wireless Communications and Networking Conference, 2004. WCNC. 2004 IEEE*, volume 1, pages 634–638 Vol.1, 2004.
- [41] L. Fried. Wave Bubble.
- [42] A. Godber and P. Dasgupta. Countering rogues in wireless networks. In *Parallel Processing Workshops, 2003. Proceedings. 2003 International Conference on*, pages 425–431, 2003.
- [43] C. Gonyea. DNS: Why It's Important & How It Works — Dyn Blog, 2010.

- [44] F.J. Gonzalez-Castano and J. Garcia-Reinoso. Bluetooth location networks. In *Global Telecommunications Conference, 2002. GLOBECOM '02. IEEE*, volume 1, pages 233–237 vol.1, 2002.
- [45] T.E. Humphreys, B.M. Ledvina, M.L. Psiaki, B.W. O’Hanlon, and Jr. Kintner, P.M. Assessing the spoofing threat: Development of a portable gps civilian spoofer. In *Proceedings of the 21st International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS 2008)*.
- [46] M. Husameldin, S. Khaled, and I. Youssef. Mitigation of DHCP starvation attack. *Computers and Electrical Engineering*, 38(5):1115–1128, 2012.
- [47] Cisco Systems Inc. Basic Networking: What You Need to Know - Cisco Systems, 2013.
- [48] Comodo Group Inc. How Firewalls Work - What is a Firewall Does — Comodo, 2013.
- [49] Wildpackets Inc. Ethernet Packets and Protocols, 2013.
- [50] A. Jeffries. Technology/WPAD - Squid Web Proxy Wiki, 2010.
- [51] D.-I. Kim and R.A. Scholtz. A random spreading code assignment scheme for centralized spread-spectrum packet radio networks. In *Military Communications Conference, 1991. MILCOM '91, Conference Record, Military Communications in a Changing World., IEEE*, pages 132–136 vol.1, 1991.
- [52] R. Lemos. Cybercriminals Likely to Expand Use of Browser Proxies – Dark Reading, 2013.
- [53] X. Li, Y. Chen, J. Yang, and X. Zheng. Achieving robust wireless localization resilient to signal strength attacks. *Wireless Networks*, 18(1), 2012.
- [54] M. Marlinspike. New tricks for defeating ssl in practice. In *Proceedings of BlackHat 2009, DC*, 2009.
- [55] M. Marra. windows - What is Active Directory and how does it work? - server fault, 2012.
- [56] A Mcneil. Build your own WiFi jammer.
- [57] C. Null. Cities search for solution to phone-theft epidemic — PCWorld, June 2013.

- [58] M.B. Pursley. *Introduction to Digital Communications*. Electrical Engineering Series. Prentice Hall, 2005.
- [59] Oppliger R., Hauser R., and Basin D. Ssl/tls session-aware user authentication - or how to effectively thwart the man-in-the-middle. *Computer Communications*, 29(12), 2006.
- [60] S.B. Roosa and S. Schultz. The "certificate authority" trust model for ssl: A defective foundation for encrypted web traffic and a legal quagmire. *Intellectual Property & Technology Law Journal*, 22(11):3 – 8, 2010.
- [61] P. Sathukhan, N. Chatterjee, A. Das, and P.K. Das. A scalable location-based services infrastructure combining gps and bluetooth based positioning for providing services in ubiquitous environment. In *Internet Multimedia Services Architecture and Application(IMSAA), 2010 IEEE 4th International Conference on*, pages 1–6, 2010.
- [62] A.W. Scott and R. Frobenius. *RF Measurements for Cellular Phones and Wireless Data Systems*. Hoboken, N.J. : IEEE : Wiley & Sons, 2008.
- [63] S. Seccia. Transit Police warn of e-device thefts in city - News - Burnaby now, July 2011.
- [64] M.K. Simon, J.K. Omura, and R.A. Scholtz. *Spread Spectrum Communications Handbook, Electronic Edition*. McGraw-Hill telecom: Engineering. McGraw-Hill, 2001.
- [65] M. Strasser, B. Danev, and S. Čapkun. Detection of reactive jamming in sensor networks. *ACM Trans. Sen. Netw.*, 7(2):16:1–16:29, September 2010.
- [66] M. Strasser, C. Pöpper, S. Čapkun, and M. Cagalj. Jamming-resistant key establishment using uncoordinated frequency hopping. In *Security and Privacy, 2008. SP 2008. IEEE Symposium on*, pages 64–78, 2008.
- [67] L. Thomas. What happens when a computer joins an Active Directory domain? - Server Fault, 2009.
- [68] N.O. Tippenhauer, C. Pöpper, K.B. Rasmussen, and S. Čapkun. On the requirements for successful GPS spoofing attacks. In *Proceedings of the 18th ACM conference on Computer and communications security, CCS '11*, pages 75–86, New York, NY, USA, 2011. ACM.

- [69] N.O. Tippenhauer, K.B. Rasmussen, C. Pöpper, and S. Čapkun. Attacks on public WLAN-based positioning systems. In *Proceedings of the 7th International Conference on Mobile Systems, Applications, and Services*, MobiSys '09, pages 29–40, New York, NY, USA, 2009. ACM.
- [70] A.J. Viterbi. *CDMA: Principles of Spread Spectrum Communication*. Addison-Wesley Wireless Communications Series. ADDISON WESLEY Publishing Company Incorporated, 1995.
- [71] X. Wang, J. Luo, M. Yang, and Z. Ling. A potential http-based application-level attack against tor. *Future Generation Computer Systems*, 27(1):67–77, 2011.
- [72] M.Z. Win and R.A. Scholtz. Ultra-wide bandwidth time-hopping spread-spectrum impulse radio for wireless multiple-access communications. *Communications, IEEE Transactions on*, 48(4):679–689, 2000.
- [73] L. Xiao, H. Dai, and P. Ning. Jamming-resistant collaborative broadcast using uncoordinated frequency hopping. *Trans. Info. For. Sec.*, 7(1):297–309, February 2012.
- [74] W. Xu, W. Trappe, Y. Zhang, and T. Wood. The feasibility of launching and detecting jamming attacks in wireless networks. In *Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing*, MobiHoc '05, pages 46–57, New York, NY, USA, 2005. ACM.