

Dissertation

by Carl James Oatley

FILE	17640_CARL_JAMES_OATLEY DISSERTATION_78196_475639537.DOCX (15.36M)		
TIME SUBMITTED	21-APR-2016 11:48AM	WORD COUNT	15704
SUBMISSION ID	56188919	CHARACTER COUNT	89718

VIRTUALISED WIRELESS NETWORK PENETRATION TESTING

By
Carl J Oatley

Supervisor: Dr Carlene E-A Campbell
Second Marker: Dr Kapilan Radhakrishnan

Project submitted as part of the requirements
for the award of BSc (Hons) Computer
Networks

April 2016

Declaration Form

I, ... declare that I am the sole author of this Project; that all references cited have been consulted; that I have conducted all work of which this is a record, and that the finished work lies within the prescribed word limits.

This work has not previously been accepted as part of any other degree submission.

Signed:

Date:

Form of Consent

I, ... hereby consent that my Project, submitted in candidature for the B.Sc.(Hons) Computer Networks degree, if successful, may be made available for inter-library loan or photocopying (subject to the law of copyright), and that the title and abstract may be made available to outside organisations.

Signed :

Date :

Copyright Acknowledgement

I acknowledge that the copyright of this project report, and any product developed as part of the project, belong to University of Wales Trinity Saint David, Swansea.

Abstract

The experimental hypothesis of this study is to gain an advanced level of knowledge regarding several attacks that can be conducted on a wireless network, in the attempt to gain illegal access to that network. The knowledge gained from implementing these attacks will provide a greater understanding of the techniques required to prevent the reoccurrence of those attacks in the future. The key objective was to meet this initial hypothesis by implementing five different attacks, overall enabling a broader coverage of the techniques used to secure wireless networks. 4 virtual machines were created using the VMware workstation 12 virtualisation suite on a host Windows 7 PC. An additional VM was also installed in the same manner on a Windows 8.1 host laptop, which was used as the attacking device. A reconnaissance network mapping attack using Nmap provided information of the devices connected to the network. Following this, WEPs 64 and 128-bit encryption was cracked using Wifite, which generated a large number of IVs which were then used to find faults with the network keys for each. WPA was then attacked using a dictionary attack that was compared to information found when de-authentication and re-authentication occurred by a legitimate client. A soft AP attack was then initiated, allowing unsuspected clients to connect, thus retrieving sensitive data. EvilAP Defender was authorised on a client Linux machine and successfully detected the rogue AP. TCP/SYN flooding DoS attack was implemented on the default gateway and an authorised client to assess the effects of the attack and to study the embedded feature found within the Windows OS. Overall, the five attacks were carried out successfully, as well as the defensive techniques required to prevent them, therefore strengthening the security of the wireless network supporting the initial hypotheses of this experiment. Future recommendations include the use of different attacking tools and defence techniques in order to draw comparison between the various applications.

Contents

Declaration Form	i
Form of Consent	i
Copyright Acknowledgement	i
Abstract.....	ii
Contents	iii
List of Figures	v
Acknowledgements.....	vi
Chapter 1 - Introduction	1
Chapter 2 – Research & Review of Literature	3
Introduction.....	3
Wireless Networks	3
Network Penetration	5
Virtualisation	16
Conclusion	17
Chapter 3 – Research and Development Methodology	19
Research Methodology.....	19
Development Methodology	20
Chapter 4 – Design and Implementation.....	22
Design	22
Network Structure	22
Router	24
Virtual Machines	25
Reconnaissance	26
WEP	27
WPA	27
Soft AP	28
TCP SYN Flooding	30

Implementation	30
Physical Network	30
Reconnaissance	30
WEP	31
WPA	33
Soft AP	35
TCP/SYN Flooding	37
Chapter 5 – Evaluation	39
Reconnaissance	39
WEP	39
WPA	40
Soft AP	40
TCP/SYN Flooding	41
Chapter 6 - Conclusions and Recommendations.....	42
Conclusions	42
Recommendations.....	43
Reflection	44
References	46
Appendices	52

List of Figures

Figure 1 – A diagram that represents the physical design of the network topology	22
Figure 2 – The VMware Workstation 12 home screen which provides a list of the virtual machines and their associated operating systems that were created on the Zoostorm PC.....	23
Figure 3 – The VMware Workstation 12 home screen located on the Toshiba Satellite Pro host Laptop that shows the unauthorised Linux VM	24
Figure 4 – Router configuration of the SSID and security options being performed in the Google Chrome browser.....	25
Figure 5 – Selecting the virtual disk to be split into multiple files as a security precaution during the creation of the VMs	26
Figure 6 – The implementation of WEP using both 64 bit and 128 bit encryption keys generated using the passphrase “password”	27
Figure 7 – A diagram that represents how the unauthorised VM creates an imitation of the authentic WLAN so that unsuspecting clients can connect to it and then communicate through its bridged connection to the legitimate AP	29
Figure 8 – The ping command being issued before and after the deactivation of the firewall on all the Windows VMs.....	30
Figure 9 - IV capturing times recorded when cracking WEP	32
Figure 10 - The number of IVs captured after the time had elapsed with the total captured and the variance between the two	33
Figure 11 - A detailed list of detected wireless APs along with important characteristics	34
Figure 12 – Aircrack-ng produced the network key following a dictionary attack	35
Figure 13 – Network connection settings obtained from when connected to the soft AP	36
Figure 14 – A visual confirmation of the username and password used when attempting to access a social network account	37

Acknowledgements

I would like to thank the University of Wales Trinity St David for the opportunity to study here and for allowing to carry out a research project in a subject I have become increasingly interested in. I would also like to thank Carlene Campbell, John Rees and Kapilan Radhakrishnan as well as all the staff and my fellow students for their continued support and guidance throughout my time at the University and more specifically throughout the writing of my dissertation. The help and support both emotionally and financially I have received from my family and friends has been overwhelming and fundamental to my studies.

Chapter 1 - Introduction

This experimental study has been developed for the purpose of creating a wireless network that can withstand a variety of attacks that may be performed by possible intruders attempting to gain illegitimate access to said network. The Wireless Local Area Network (WLAN) will contain three nodes, a single server and two clients, and from its inception will be an Open Network or Hotspot. This implies that there is no security of any kind on the WLAN, thus allowing any device outside of the network to connect, unhindered and with no reassurances that they are permitted to be on the network. Needless to say, it would be completely impractical and extremely dangerous for a home or a business to surrender instantaneous and comprehensive access of its entire infrastructure without some security procedures in position.

Throughout this study there will be an investigation into five wireless network attacks that are used by hackers and penetration testers to bypass the security measures of WLANs that have some level of security in position. The experiment will then proceed to use such attacks on the Open Network. Each attack shall be undertaken separately and persistently to gain a greater understanding of how they work. Once there is a sufficient understanding of an attack, security measures will then be introduced to the network that will either neutralise or repel such an attack. These attacks will be initiated by a forth node located on the exterior of the network, which is laden with the various software types used within hacking/penetration testing situations.

The experiment will be implemented within a virtualised environment as opposed to that of a physical setting. The virtual environment will be created and contained via specialised virtualisation software that is installed on a desktop PC. The virtualisation software allows virtual nodes (or images) to be created by allocating some of the specifications of the desktop PC (which it is running on) to said nodes, thus allowing them to function in the same manner as a physical PC.

Following this introductory chapter, the remainder of this report will be broken down into five different chapters (chapter 2-6). In chapter 2 the review of relevant literature is documented and assessed, while chapter 3 will address the research and developmental methodologies of this study. Chapter 4 discusses

the design and implementation procedures of the experiment while also providing the results which will then be evaluated in chapter 5. The final chapter (6) will conclude what the study has produced and also provide recommendations for future studies.

Chapter 2 – Research & Review of Literature

Introduction

This chapter will look at past literature that has influenced the standards of security that are present in modern-day wireless networks. Firstly, a comparison of wired and wireless networks will investigate which has been deemed to be the less secure of the two. This section will also look at the current protocols that are being used and how and why they came to fruition.

The second section of this chapter will look at a variety of penetration testing or hacking techniques used presently and how said attacks can bypass security measures of a network. Throughout this study while referring to both penetration testers and hackers as a collective, the term 'attacker' will be used. There will also be studies into the different types of software being used to initiate these attacks and how they differ.

The third section will explore the physical and virtualised environments in which attacks are carried out while also assessing the various versions of said virtual environments. The study looks to review literature on wireless network security attacks that have been undertaken in both environments. The findings will then be evaluated for the purpose of discovering which would be the most suitable environment to run the experiment.

Wireless Networks

Wireless technology has become a fundamental part of modern society, with portable devices such as laptops and MacBooks seemingly being commonplace. As a result of this and the ever increasing popularity of handheld devices (such as Smart Phones) and the emergence of the Internet of Things (IoT) (sensor nodes for example), the demand for wireless networks has increased exponentially. In an article written by Press [1] for Forbes in 2014, it was highlighted that research carried out by ABI research [2] estimated that by the end of that same year there would be 16 billion wirelessly connected devices active worldwide. This figure is expected to increase further by 2020 to around 41 billion [1, 2] with the compulsion for wireless networks to increase correspondingly. With such an extraordinary escalation, the security of wireless

networks will continue to be the main priority for all those tasked with the implementation and preservation of said networks.

There are many advantages to wireless networks that have fuelled the dramatic increase in its popularity over the traditional wired networks. The foremost benefit to wireless networks is the mobility aspect that users inherit. They can access the networks content or the Internet from any location so long as they remain within the perimeter of the network. Unfortunately, in addition to this, as previously mentioned the primary benefit of wireless networks is similarly their ultimate handicap, this being security. Moreover, the ease of accessibility to wireless networks becomes a significant security risk. Unlike wired networks, where an intruder would have to physically connect to the network in an attempt to access it, wireless networks are not so direct. An intruder could attempt to access the network from anywhere as long as they are in range of an Access Point (AP) meaning they do not need to be inside of the building [8].

The security of wireless networks, which operates using the 802.11 standard, has been an issue since the turn of the millennium. Fluhrer *et al* [3] in 2001, presents a technique which cracked the portion of said standard known as Wired Equivalent Privacy (WEP). The paper provided proof that there was a weakness within the RC4 Key Scheduling Algorithm as it would generate a high number of weak keys. Once they had collected a significant amount of these keys, they were then able to decipher the algorithm. This attack later became known as the FMS attack. Following this discovery came many other attacks (such as KoreK, Chopchop and PTW) that would eventually deem WEP as being completely impractical which resulted in WEP being replaced by Wi-Fi Protected Access (WPA) in the 802.11 standard.

Although WPA replaced WEP as the security feature of the 802.11 standard it was not an original design. It is, in fact, a modified version of the WEP security structure that preceded it. Although WPA possesses TKIP (Temporal Key Integrity Protocol) and EAP (Extensible Authentication Protocol) in addition to an MIC (Message Integrity Checker), it still retains the same RC4 algorithm found in WEP. A study by Tews and Beck [4] found that an attacker can crack WPA by using a modification of the Chopchop attack that was designed to crack WEP. This method of attack would not give network access but it would however give the attacker access to confidential data. In contrast, a study by

Moskowitz [5] suggests that WPA can be cracked through the use of dictionary attacks when directed at WPA use of a Pre-Shared Key (PSK). With the before mentioned issues with WPA, the development of WPA2 was introduced thereafter.

WPA2 remains the security protocol of choice for the 802.11 standard (although WEP and WPA are still available). With the new protocol being rid of the flawed RC4 cipher stream and replaced with the AES (Advanced Encryption Standard) block cipher. Although WPA2 has been deemed to be secure, it is still susceptible to certain DoS (Denial of Service) attacks to the MAC layer, offline dictionary attacks used against the PSK and also internal attacks [6].

One such internal attack is referred to as 'Hole 196' which exploits a weakness in the WPA2 protocol. The vulnerability was discovered by security researchers at AirTight [48] when they inspected the 1232-page IEEE 802.11 Standard documentation. Concealed within page 196 of said document is a declaration that WPA2s use of group temporal key (GTK), which is shared across all sanctioned clientele, can be abused to initiate internal attacks such as launching MITM and DoS attacks. An insider (authorised user) can inject spoofed GTK-encrypted packets onto a network. This would allow them to sniff and also decrypt other user's data while also scanning their devices for vulnerabilities and installing malware onto them.

During an investigation into advanced stealth MITM (ASMITM) attacks, Agarwal *et al* [49] successfully exploited the Hole 196 vulnerability by using stealth MITM (SMITM) combined with wireless DoS (WDoS) attacks (the result is an ASMITM). This investigation used two open-sourced software's in Madwifi driver and WPA supplicant to practically exploit the vulnerability, concluding that the attack could not be prevented which corresponds with the outcome stated within [48].

Network Penetration

A penetration test (also referred to as a pen test or ethical hack) encompasses a tester (pen tester) and their attempts at bypassing a networks security features so that they may obtain the resources of said network (e.g. databases or sensitive data). The pen tester would do so without any prior knowledge of traditional access information such as usernames and passwords, thus

replicating the circumstances as that of an unethical hacker [7]. The primary difference between a pen tester and a hacker is the permission granted to the tester to attempt to penetrate the network [7, 8]. The pen tester uses all the tools and techniques used by hackers while remaining vigilant of any boundaries set by the client that may breach any integrity or privacy requirements [8].

Before any attempt to gain access to any network both pen testers and hackers must first do some reconnaissance work to gain a greater understanding of the network they are aiming to infiltrate. All the attacks used to penetrate a network fall within one of two categories, passive attacks and active attacks. Generally reconnaissance work is firmly associated with passive attacks due to the nature in which they are conducted. Sniffing [9] is one such passive attack which allows an attacker to monitor, capture and analyse network traffic that is not encrypted. An attacker can use any information found throughout this process to infiltrate the network or cause the network to become dissolute.

There are countless varieties of open-sourced software's readily accessible on the Internet for attackers to download that would provide them with the tools required to perform such an attack. The Kali Linux [10] suite alone provides an attacker with proximately 600 tools that deal with all aspects of penetration testing.

In a study by Troya *et al* [11] the use of the extremely popular Wireshark software was used to analyse management and control frames captured while examining the vulnerabilities in the 802.11i standard. The examination used Wireshark to compare the frames captured from two different scenarios for comparison. The study also displays the compatibility Wireshark has with other tools as it analyses frames captured by Kismet.

Once the attacker has gained a sufficient quantity of information from the passive attacks they may then use said information to perform active attacks. Research shows that wireless networks are susceptible to Denial of Service (DoS) attacks and Man in the Middle (MITM) attacks, although there are others that can also be executed. A MITM attack is initiated when an attacker intercepts a single connection between two communicating nodes and creates two connections, thus allowing the communication to exist through them. The

MITM attack will grant the attacker the power to not only read and alter any data passing through them but it allowing them to insert any new data into the communication. This is done without either nodes knowledge or consent [12] and is one of the two conditions required to make a MITM attack. The other is referred to as ID spoofing, where an attacker is also able to impersonate a legitimate node of the network [13]. The sniffing attack discussed previously is a form of a MITM or session hijacking.

Another attack that falls within the session hijacking classification would be Sidejacking. Its functionality is very similar to Sniffing and it even involves using the same tools, but it differs in the data it is trying to locate. Sidejacking exploits a lack of encryption in Cookies that are left by Websites once the user has progressed through the SSL encrypted login process [14]. Cashion and Bassiouni [15] in 2011 tell of the increasing risk of Sidejacking due to the popularity of Cookies and unsecure wireless hotspots. The paper further reinforces previous findings regarding the actions of Sidejacking set out above and also suggests a means of preventing the attack. They present a protocol for securing Cookies over a wireless network, while also addressing future works which includes a browser plug-in and PHP script for Web developers.

During the same year a paper by Kumar [16] proposed the introduction of a three-tier authentication technique which verifies a session between the host and the server. This system uses the local storage feature of HTML5 (Hyper Text Markup Language) to overcome cookie vulnerabilities through client-side JavaScript and server-side logic. The JavaScript is used in the third tier of verification where it is used to extract a secret key found within the local storage and then forwards said key to the server, highlighting the problem with method. For the three-tier verification to work there must be support for JavaScript as without it the key cannot be retrieved. Not all sites are subject to Sidejacking [16, 17] due to them functioning solely on a secure network. PayPal are one such site that due to the financial nature of their business operates in this manner. For communication between a client and server to be deemed as secure the HTTPS tag must be displayed within the URL, siting that the communication is encrypted.

While Sidejacking uses software to intercept communication, an evil-twin attack (which is also a session hijacking attack) requires hardware to be integrated into

a network. An evil-twin attack occurs when an attacker introduces an additional access point onto a network unbeknown to the victim. Once the evil-twin AP has been successfully incorporated onto the network it then proceeds to steal the identity of a genuine AP found on the network. So that the evil-twin AP can impersonate the genuine AP it must mask its BSSID and SSID as that of the genuine AP. Once this is complete, the evil-twin AP presents itself as a legitimate AP on the network and allows users to connect. When the users connect to the evil-twin AP it can then begin to sniff the frames they send. The evil-twin can then either forward said frames onto the genuine AP or drop the frame should it contain the data the attacker requires.

An investigation by Bhatia *et al* [18] assesses the difficulties in not only identifying an evil-twin AP but also discovering its location. They create an algorithm which detects which AP is legitimate and which is the evil-twin. Accompanying the algorithm are four-square antennas attached to each receiver in the network which will be used to distinguish a location for the evil-twin. Although the algorithm constructed within the investigation could successfully identify the evil-twin AP in every instance, their attempts at finding the location of it were not. While the algorithm is not expensive to implement, one would suspect that ensuring all receivers within a network that would have to be equipped with four-square antennas would be expensive.

Alternatively, Mustafa and Xu [19] proposed a method of detecting evil-twin AP's via the client-side as opposed to the receiver-side suggested by Bhatia *et al* [18]. Within their proposal [19] they introduce a detection mechanism referred to as CETAD (Client end Evil Twin Access point Detector) which is an application installed on a client's device. The application itself finds discrepancies within the evil-twin AP's network setup compared to that of legitimate APs on that network. In a typical network topology, all legitimate APs will have the same SSID and also similar network parameters such as the ISP (Internet Service Provider) and Global IP addresses. When an evil-twin is inserted into the topology, it does not portray the same network setup as the legitimate APs thus allowing the application to detect it. The benefits of such a proposal result in there being no need for any additional hardware or software to be installed within the network infrastructure and it is compatible with all wireless devices.

A more recent study by Nakhila *et al* [20] in 2015 proposes an alternative client-side defence mechanism to that offered by CETAD [19]. The method of detection presented by [20] focuses more on the evil-twin AP using a cellular connection as opposed to the traditional method of accessing the Internet through the legitimate AP. The attacker using a cellular connection to connect to the Internet would result in a change to the gateway (from the gateway which a legitimate AP would use). This defence mechanism detects the change of gateway and would proceed to break the SSL/TCP connection to the server and inform the user of the evil-twin AP. This method of defence is limited as it would only work should the attacker opt to use an alternative means of connecting (such as cellular) to the Internet. However, this kind of defence method would function alongside some time delayed detection such as the one suggested by Yang *et al* [21].

An evil-twin attack is one of the biggest security threats to a network (especially a Hotspot) however it does involve connecting an additional access point to the network infrastructure. This would be extremely noticeable to the public eye should an attacker be sitting in a coffee shop with an AP by their side. So another attack that would be less noticeable (physically) would be the soft AP (software access point) attack. [22, 23] This technique functions identically to that of an evil-twin in regards to how it implements itself onto a network by copying the identity of a legitimate AP and allowing users to connect to it. It does differ however in its physical appearance. Where an evil-twin uses an access point, the soft AP uses a laptop, desktop or wireless enabled mobile devices (mobile phones, tablets etc.) that have software installed which allows them to act as wireless access points. [23] Although there is software readily available over the Internet that can be downloaded (Wireshark for example), many operating systems installed on the devices previously mentioned come pre-equipped with access point capabilities. For example, Windows 7 has a Virtual Wi-Fi feature integrated within it which allows any device with a WNIC (Wireless Network Interface Card) that uses it to portray an access point. This is also true for the two largest mobile operating systems, iOS and Android. They too have the capability of allowing the user to activate a Personal Hotspot or Portable Hotspot which provides the user with the ability of allowing other devices to connect to their network.

The attacks discussed in the paper so far function in a deceptive manner as they do not broadcast their intentions to their victims. However, a denial of service (DoS) attack is not so misleading. DoS attacks use brute force to flood a network with traffic which results in legitimate users attempting to access the network thus depriving them of the content they require [24]. Not all DoS attacks are the same, but they can all be separated into two distinct classes based on the method of their attack. Some DoS attacks concentrate their efforts on application weaknesses as they direct specifically crafted packets or abusive workloads aimed at the applications vulnerabilities resulting in crashes or reboots of the entire system.

While this form of DoS attack generates from a single machine, a distributed denial of service (DDoS) attacks the infrastructure and transpires via multiple machines. These machines are referred to as zombie machines that an attacker has hijacked meaning that there only need be a single attacker that controls all the machines. Each machine floods a network with packets in conjunction with the other zombie machines thus draining the bandwidth, CPU cycles or the TCP buffers [25]. This is commonly referred to as a direct DDoS attack as the attacker uses the zombie machines to flood the victim's network directly. An alternative DDoS attack would be a reflective DDoS attack which uses not only the zombie machines to attack a network, but also other websites. In a reflective DDoS attack, the attacker instructs the zombie machines to send request packets to various IP hosts that will respond to any packet request (also known as Internet reflectors). The Internet reflectors then reply to these requests, however the source IP address of the request packets are spoofed and direct the responses of said Internet reflectors to the victims network, thus flooding the network with traffic [26].

Research has shown that the proposed best means of counteracting the DDoS attacks resides on the ISP domain routers (Edge routers) whether that be a direct or a reflective attack. Defensive measures such as Sardana *et al* [27] suggested a honeypot based approach which deals with DDoS attacks by luring them into a false server planted into the network to use entropy to capture the attentiveness of a distribution along with behavioural detection. Additionally, the Pairing Based Filtering (PF) proposed by Al-Duwairi [26] pairs request and reply packets at the edge router, thus preventing drastic bandwidth consumption by

filtering out the attacker reply packets. The major issue with having these DDoS defence techniques in place is dependent purely on ISP cross-network collaboration. All ISPs would be required to reassess their network infrastructure resulting in additional costs for something which they are not obligated to ensure.

The DoS and DDoS attacks that have been addressed above are primary attacks on wired networks. However there are additional threats concerning wireless networks in regards to DoS attacks. Whereas wired DoS attacks either focus on an application weakness or at the infrastructure itself, wireless DoS attacks concentrate on either signal jamming or bypassing protocols in the MAC-layer.

Jamming is one of the most effective ways in which to prevent communication between nodes. An attacker, in this case a *Jammer*, can interfere with the communication channels by omitting a signal that can either interfere or disregard the MAC protocols. There are various attacks available at a jammers disposal and they can all be categorised into four types [28, 29]:

- Constant – a radio signal constantly omits random bits to the channel (before said channel becomes idle) while having complete disregard for MAC-layer protocol.
- Deceptive – a constant injection of regular packets to the channel would result in a legitimate correspondent being misled into believing the packets are authentic and would remain in a receptive state.
- Random – by alternating between a predetermined timescale of jamming and sleeping modes, the random jammer can opt to act as either a constant or deceptive jammer while being considerate of its own energy consumption.
- Reactive – a reactive jammer only transmits when the channel is active (not idle) thus targeting the reception of a message. Reactive jammers are not as energy efficient as the random jammer but they are deemed to be more discreet than their counterparts.

There has been great interest into this topic of research and how to detect and prevent RF jamming. The method of attacks dates back to post WLAN technology where RF jamming was used in World War II (WWII) [30] and possibly even before then. However, as the WLAN landscape grew so too did

the need for defence against jamming attacks. In 2004, Xu *et al* [31] proposed two methods of detection and defence against (at that point in time) an unstoppable attack. The paper addresses the different attacks to the MAC and PHY-layers of a wireless network and how through the use of sensing times (for the MAC-layer) and noise levels (for the PHY-layer) on the channel they can be detected. They also suggest that to mitigate any DoS attacks on a wireless medium, the users should either change their transmission frequency (channel surfing) or move to a new location (spatial retreats). In both cases it simply requires the legitimate user to completely avoid the channel on which the attack is being deployed, suggesting it is more of a circumvention technique as opposed to a standalone defence mechanism. Although these methods of defence are effective they are not always the most efficient means of dealing with an attack due to the user constantly having to move (either physically or digitally) whenever an attack occurs.

A more recent study in 2010 by Misra *et al* [32] furthered the research carried out by Xu *et al* [31] and introduced a Honeynode detection strategy to coexist with a modified version of the channel surfing algorithm created by Xu. A Honeynode functions in the same manner as a honeypot (as previously discussed) and from here the modified channel surfing algorithm can function. There are slight differences between the algorithms proposed in the two papers, with the algorithm in [32] claiming to be slightly more efficient when dealing with low intensity attacks. Other than the introduction of a Honeynode detection system, both papers are generally similar.

Although RF jammers are extremely effective attacks they are also dependent on additional hardware. There are other software-based DoS attacks that penetrate a wireless network by evading the MAC-layer protocols. One such example would be TCP-SYN flooding which attacks TCPs three-way handshake connection process. Unlike a legitimate TCP connection where a client sends a request to a server, which then responds with a SYN acknowledgement (ACK), which sequentially prompts an ACK response from the client, this process does not occur during a TCP-SYN flooding attack. During TCP-SYN flooding [33] an attacker issues a SYN request that has a spoofed source IP address to the victim's server. A SYN ACK response will be generated and sent to the spoofed IP address which results in there being no corresponding ACK from the client,

thus creating an incomplete connection or a half-open connection. This connection will remain in said state for approximately 75 seconds after which the connection times-out. All half-opened connections are then stored and maintained in the server's limited memory which once full will cause the server to drop all connection requests.

Defence mechanisms such as SYNkill [34], SYN Cookies [35] and SYN Cache [36] are a select few that have been guarding networks against TCP-SYN flooding for decades. These defence methods are all implemented at the victims server or at the server firewall, however they give no indication of where the attacks are sourced. A more recent development by Saravanan & Gowri Shankar [37] uses a combination of four detection methods which includes TCP flags, ports, ICMP feedback and route tracing to distinguish the origin of an attack. The multipurpose mechanism can detect an attack in a very short period of time, while also producing an extremely high standard of accuracy while differentiating between legitimate SYN traffic and SYN flooding traffic.

In contrast, a case study by Lim & Kim [38] assesses the effects of having increased levels of security, causing performance degradation on the server nodes they are implemented. The paper proposes a method of dynamically altering the security level on the server node in conjunction with the threat level on its neighbouring nodes. This technique will allow the server node the opportunity to alter its defences, whether that is security parameter reconfiguration or commissioning the appropriate prevention/avoidance procedures to mitigate specific attacks. This prevention strategy allows the node to dynamically combat an attack (depending on its level of threat) while maintaining minimal performance degradation.

There are also attacks that function by taking advantage of the Internet Control Message Protocol (ICMP). ICMP attacks are some of the most basic, yet effective, DoS attacks available. An ICMP attack uses an ICMP Ping [39, 40], which is a technique used by network administrators used to check network connectivity, to flood the target network. This is referred to as the Ping Flood attack and should this method be used in a DDoS scenario then the attacker would have a large number of devices sending ICMP Pings to the victim, thus causing their network to crash. This works due to an ICMP Echo server being a necessity on a computer and therefore when an ICMP Echo Request message

is received by a computer, it must respond accordingly with an ICMP Reply message.

In addition to the ICMP Ping attack there is also an ICMP Base Land attack or Smurf attack [40, 41]. The Smurf attack functions in the same manner as the ICMP Ping attack where it relies upon the ICMP Echo Request message, however the content within that message is different. Within a Smurf ICMP Echo Request message the source IP address is spoofed to that of the victims own IP address, therefore the automatically generated ICMP Echo Reply message will also be sent to the victim, thus exhausting their computing resources.

The defence against ICMP attacks are relatively straight forward. Most Firewalls already have ICMP attack defence mechanisms embedded although they may not completely protect the network. A supplementary method of defence can be employed on the networks router by disabling its ability to forward traffic to broadcast addresses [41]. In addition, a paper by Junior & Kumar [40] looks at the DoS defence mechanisms built-in to Apple's Lion and Microsoft's Windows 7 Operating Systems (OSs). They investigate how each OS responds when confronted by both an ICMP Ping and Smurf attack and monitor their responses. The investigation revealed that the Windows 7 OS was the greater of the two as it preserved the processor of the device it was operating on.

There are other forms of attacks that do not necessarily fall into the MITM and DoS attack categories but they do have the ability to infiltrate a network in order to launch them. One such attack is the Address Resolution Protocol (ARP) Cache Poisoning (commonly referred to as ARP Poisoning) which utilises weaknesses within the MAC-layer. ARP is the process of resolving IP address and MAC address coupling for outbound packets. The MAC address and IP address is then stored within the ARP cache and used to forward inbound traffic to that specific machine. If the IP address does not exist within the ARP cache then ARP will broadcast the IP address over the network, requesting that the machine with that IP address reply with its MAC address. When there is a reply, the ARP cache is then updated to show that that IP address is coupled with the MAC address provided by the ARP reply [42]. Traditionally, an ARP Poisoning attack can only be initiated while the attacker is connected to the same LAN as their target. This means that the network must consist of switches, HUBs and

bridges. These are found within wired networks which make an ARP Poisoning attack very limited in regards to its deployment and effectiveness within this infrastructure; however this is not the case within wireless networks. Due to the wireless access point functioning as a transparent bridge to the MAC-layer, they inadvertently provide attackers with an opportunity to conduct ARP Poisoning attacks. The effects of these attacks are not just limited to the machines that are within the WLAN but also any machine that is connected to the wired backbone of the network [43].

A basic defence method for ARP Poisoning would be to remove the dynamic aspect from the ARP protocol. By providing static ARP entries into the ARP cache an attacker can no longer use ARP spoofing. Although this is the safest method of protection it not feasible within a large network due to the constant updating that an administrator would be required to do [44].

A more dynamic approach by Nam *et al* [45] in 2013 called GMR-ARP was created to counteract ARP Poisoning attacks on WLANs. This method was the third in a trilogy of ARP Poisoning mitigation techniques proposed by Nam *et al* which proceeds the proposed MR-ARP in 2010 [46] and EMR-ARP in 2012 [47]. GMR-ARP is the evolutionary combination of MR-ARP and EMR-ARP where it provides the security for wired networks provided by MR-ARP to coexist with a modified version of EMR-ARP designed for wireless networks. GMR-ARP maps MAC and IP addresses through an improved voting system which relies on the responses of neighbouring nodes. The new GRM-ARP method proposed has improved on the issues of unreliability due to computation power differences in neighbouring nodes that surrounded EMR-ARP. GMR-ARP opts not to use the computerised puzzle that EMR-ARP uses and also partially drops any votes from neighbouring nodes which have rapid replies, thus providing a more reasonable voting scheme.

There are also some measures such as Snort or DecaffeintID that can be installed on the client's device that can detect an ARP Poisoning attack. However, such software is not compatible with mobile devices meaning they have a limited spectrum in which then can defend. It should also be noted that installing software on devices within a large network is not practical [48].

Virtualisation

With the cost of acquiring specific hardware and software to create a network being extraordinarily high and the ability to implement them being difficult, the prospects of creating a network is slim. One possibility to counteract these obstacles is through virtualisation software. In an article by Marshall [50] he tells how this specialised software is designed to allow users multiple OSs (guest OSs) to function in full isolation of each other even though they are all operational on the same device. There are various virtualisation software manufacturers operational at present with Davis & Lowe [51] bestowing VMware's vSphere as the most commonly used hypervisor in all data centres. [50] However regardless of the manufacturer, the virtualisation can only be created by performing one of three very different techniques. They are full virtualisation (which uses binary translation), paravirtualisation (OS assisted) , which improves performance and efficiency by communicating between hypervisor and guest OS, or hardware assisted virtualisation (which opts not to use either of the previous techniques by operating at a lower level).

While there are three different forms of virtualisation techniques there are also two distinct hypervisor types. A hypervisor is what allows multiple OSs to exist and function simultaneously on a single machine. It is a low-level programme which assigns assets by using a thin layer of code in either software or firmware. The two hypervisor types are referred to as Type 1 and Type 2 and operate in an entirely diverse manner to each other. Type 1 hypervisors are known commonly as bare metal or native hypervisors due to them being directly active on the system hardware. In contrast Type 2 hypervisors are installed pieces of software that require the host OS for them to function [52]. Type 2 hypervisors are in effect the easier of the two to acquire and implement as they are compatible with most existing OSs and do not require additional hardware.

In a report published in 2007 by Gammage & Shiffler [53] they predicted that by 2011 the number of virtual machines (VMs) in commission could raise to a staggering 846 million. However, it is not stated which of the three versions would be the more favoured in regards to their popularity and usage. There are numerous papers that involve the use of virtualisation software in their investigations. Papers such as that by Mulfari *et al* [54] investigates how Oracle's VirtualBox, in conjunction with noVNC (virtual network computing),

reacts when used as a support mechanism for assistive tools in cloud computing. Although the use of noVNC caused several of the assisted tools to become inoperative, the VirtualBox environment it used had no effects on the results. Another paper by Bechtsoudis & Sklavos [55] uses VMware to create VMs that host different OSs for the purpose of generating a testbed of penetration tests to test the weaknesses within a network. This paper shows that not only is virtualisation software flexible in regards to how a user can setup their topology but also how it can use that virtual topology as a practical environment to assess various networking attacks.

An additional paper by Petrovic & Schiper [56] compares the fault-tolerances two different open-sourced hypervisors (KVM and Xen) have on a systems development and performance. In the investigation they conclude that the hypervisors do not greatly affect the system performance, however KVM's reliability on the host OS does harvest substantial advantages in regards to the reuse of code and the overall development. Such an experiment does publicise that virtual environments do have an advantageous and disadvantageous effect on the system on which it is being deployed.

Conclusion

Since their inception, wireless networks have been subject to countless investigations into their security due to the lack of consideration into the incorporation of security during the development process [57]. There have been three different encryption methods established in WEP, WPA and WPA2 with the aforementioned being cracked and the latter having several issues. Wireless networks are prone to MITM and DoS attacks regardless of the encryption being used, however WPA2 is susceptible to insider attacks using the Hole 196 found in its protocol. By exploiting this inadequacy, a network can then be breached and opened up to other attacks. Although there are multiple attacks that range from the passive aspects, such as eavesdropping to the more active attacks such as brute force password cracking attacks, some are more effective than others. ARP poisoning is one such attack that has numerous open-sourced software available on the Internet that would allow such an attack to commence. If implemented efficiently an ARP attack can become a catalyst for a MITM or a DoS attack. It should also be said that no attack is commenced without proper reconnaissance being undertaken

beforehand. An attacker requires an in-depth knowledge of the infrastructure they are attempting to infiltrate. For them to gain such knowledge and to attack said infrastructure the Kali Linux suite is the ideal tool. It gifts the attacker access to over 600 tools which are some of the most powerful hacking tools available today (such as Wireshark, Nmap and Aircrack).

Chapter 3 – Research and Development Methodology

This chapter contains the methodologies for both the research and development practices. The philosophy, approach and the three strategies that personalise the research methodology for this study will all be evaluated and discussed.

Research Methodology

This study will use positivism as its philosophy in conjunction with a deductive approach to the research methodology. As this is an experimental study the research produces statistical data which is recognised as quantitative data and as a result is one of the many features of the positivistic paradigm. The experiment itself will be performed within an environment that has been tailored for the study and is subsequently a manufactured artificial environment. The study looks to test existing theories on wireless network attacks and mitigation techniques as opposed to generating a theory (an inductive approach and a feature of phenomenology) and is therefore considered to be a deductive approach. It is known that research using a positivistic philosophy would usually progress to use a deductive approach as is the case with this study.

As previously mentioned, quantitative data is a feature of positivistic philosophy and also has a significant standing within a deductive approach. This, coupled with the substantial quantity of literature readily available on this topic and the more specific characteristics of this study (such as the different attacks), highlight why a deductive approach is the expected selection for this method. This in turn leads on to the three strategic methods that are a necessity to conducting a thorough research methodology.

The first strategy that must be used is a documentation review which provides a means of comparison for the final results of the study, while also ascertaining the key researchers and theories within the domain of computer networks and security. It will also reveal whether or not the project holds any relevance and practicality.

The second strategy would be the experimental strategy due to the nature of the study being conducted, with said study looking to evaluate how an attack accesses a network by implementing security measures within the network. This coincides with the rule of an experimental strategy whereby the

manipulation of one variable influencing change in another can then be measured.

The final strategy is the use of a case study, due to the intensive investigations into the processes of attack and defence techniques used to penetrate and secure wireless networks. The information gathered from understanding how an attack affects the network can then be used to prevent it in the future.

Development Methodology

The experimental hypothesis of this study is to gain an advanced level of knowledge regarding several attacks that can be conducted on a wireless network, in the attempt to gain illegal access to that network. The knowledge gained from implementing these attacks will provide a greater understanding of the techniques required to prevent the reoccurrence of those attacks in the future. The key objective was to meet this initial hypothesis by implementing five different attacks, overall enabling a broader coverage of the techniques used to secure wireless networks.

The experiment itself was hosted on a Zoostorm 7872-0492/A desktop equipped with a dual core Intel Core i3 4130 processor and an Asus H81M-E motherboard. It possesses a 1 TB hard drive, DDR3 16 GB of Random Access Memory (RAM) which operates at a frequency of 16000 MHz and supports a 64-bit operating system.

The host had a Microsoft Windows 7 Home Premium 64-bit edition operating system and housed VMware's Workstation 12 virtualisation suite as the environment in which the experiment will be synthesized. The guest operating systems consisted of a single Windows Server 2012 R2 (installed on the networks server), two Windows 10 education editions (installed on each of the authorised clients) and a single Linux Debian 7 64-bit (installed as an authorised node). The Linux machine had the Kali penetration suite 2.0 installed also. There are numerous websites such as www.offensive-security.com [58] that allow a user to download custom images of Kali Linux that are pre-configured to be installed on VMware virtual machines (VM) such is the case in this experiment. Each VM was assigned 60 GB of storage space and 2 GB of RAM upon their creation. All the devices were then paired with their own TP-

Link 300Mbps Mini Wireless NUSB adapter (model number TL-WN823N) so that they could connect to the wireless network.

Once the devices had been created they were then connected to the wireless network which used a Netgear – DG834G v3, 54Mbps wireless ADSL modem router (serial number – 1JX173BF15A10). The router would be used as the default gateway and the point of communication between the authorised devices. It would also be the point where the unauthorised device would try to penetrate the network. The unauthorised device would be a direct reproduction of the Linux machine previously discussed with its own external WLAN card that was also previously mentioned. However, this device would be hosted on a Toshiba Satellite Pro with an Intel dual-core, i5-4210U processor and DDR3 8 GB of RAM. The OS that would be used to house the VMware virtual machine was Windows 8.1.

Chapter 4 – Design and Implementation

This chapter is tasked with addressing any design decisions made throughout the creation of this experiment. It will cover all aspects of the design from the physical network which is represented by the diagram shown in Figure 1 (below), to each of the 5 attacks, while providing justifications as to why certain aspects have been included and others have been omitted or neglected. The chapter will also look at the implementation processes of the topics discussed in the design and address their effects and any issues that arose.

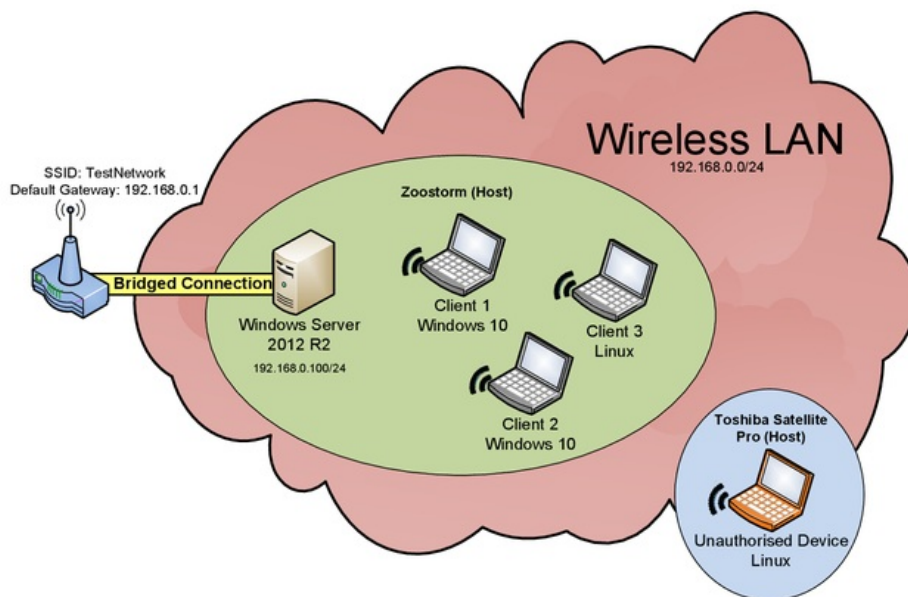


Figure 1 – A diagram that represents the physical design of the network topology

Design

Network Structure

There were several aspects that had to be amalgamated in the designing process of the network topology to ensure it would function correctly and to emulate the diagram in Figure 1. Firstly, there were four VMs created and housed on the Zoostorm PC (referred to as host PC) which had VMware

Workstation 12 installed on it (the installation of the virtualisation suite and the VMs will be discussed later in this section). Below (Figure 2) is a list of the VMs created along with the associated OS and will be referred to as the authorised VMs from the remainder of this paper.

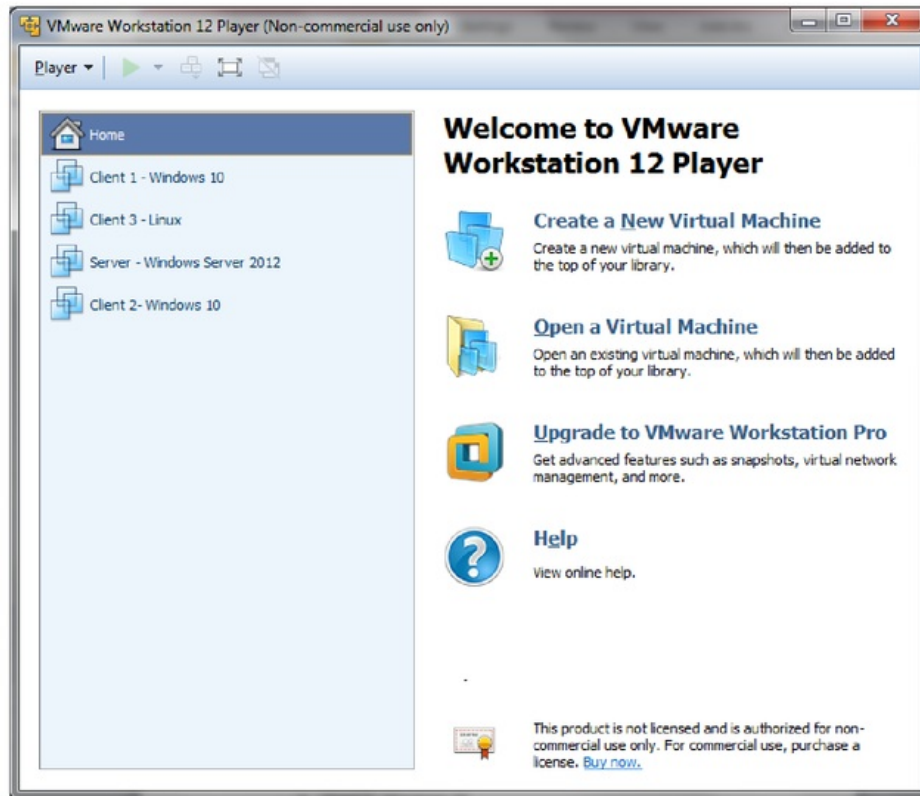


Figure 2 – The VMware Workstation 12 home screen which provides a list of the virtual machines and their associated operating systems that were created on the Zoostorm PC

With the exception of the Server, which had a bridged connection to the host PCs wireless LAN (WLAN) card, all other three client machines were paired with an external WLAN card to prevent communication occurring through the internal WLAN card of the host PC.

Secondly, once the authorised machines were created on the host PC, the unauthorised Linux VM (referred to as unauthorised VM) was installed on the Toshiba Satellite Pro Laptop (referred to as host Laptop). This was created in the same manner as the authorised VMs and is represented below in Figure 3.

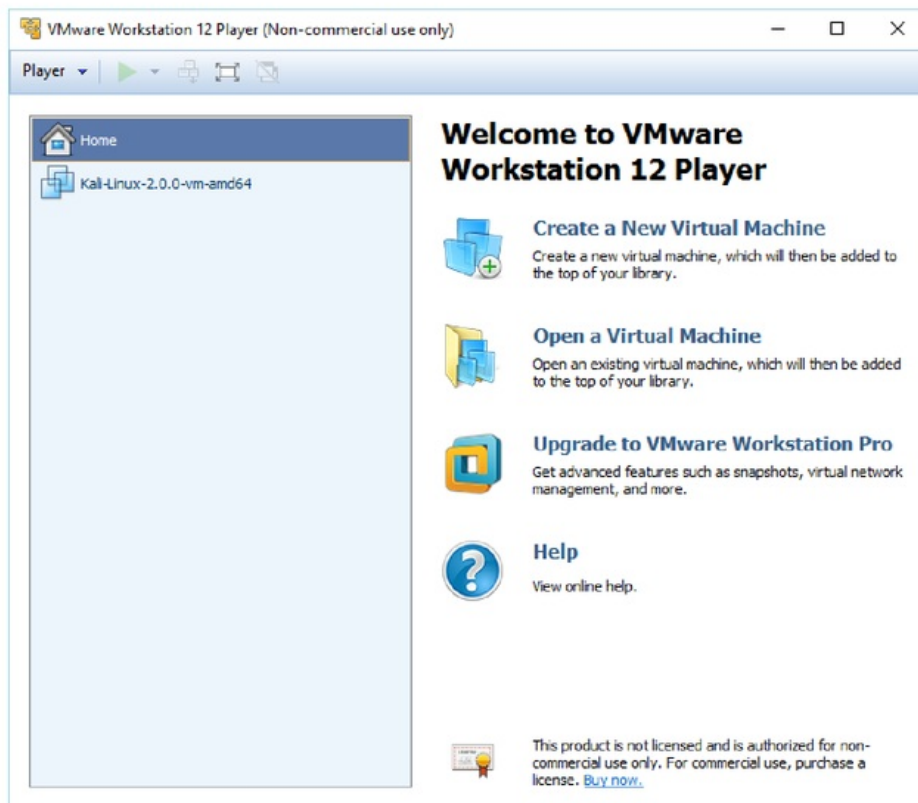


Figure 3 – The VMware Workstation 12 home screen located on the Toshiba Satellite Pro host Laptop that shows the unauthorised Linux VM

Although the host PC did have sufficient memory and storage to house and run a fifth virtual machine, it did not have the desired number of USB ports to embrace all the external WLAN cards required for each device. While the external WLAN cards were not a necessity for the authorised virtual machines, it was for the unauthorised virtual machine. This was due to the prerequisite for packet injection and a supplementary wireless interface to launch several of the attacks. The inclusion of the external WLAN cards also gave the VMs a more realistic appearance in regards to their hardware specifications in comparison to real-world network topologies.

Router

The inclusion of a wireless access point was obligatory as the topology required wireless communication to exist between all VMs (authorised and unauthorised), and the VMs existing on multiple devices. The AP itself was a redundant home router and therefore required to be restored to its factory

settings before being reconfigured for the purpose of this experiment. It was assigned an SSID of TestNetwork and operated on channel 6. On its original configuration all security and encryption measures were disabled to emulate an open network.

All router configurations were accessed and completed using the Google Chrome web browser and by entering the default gateway IP address into the URL (see Figure 4). As the experiment progressed the security and encryption settings were altered and will be discussed in greater detail later in this section.

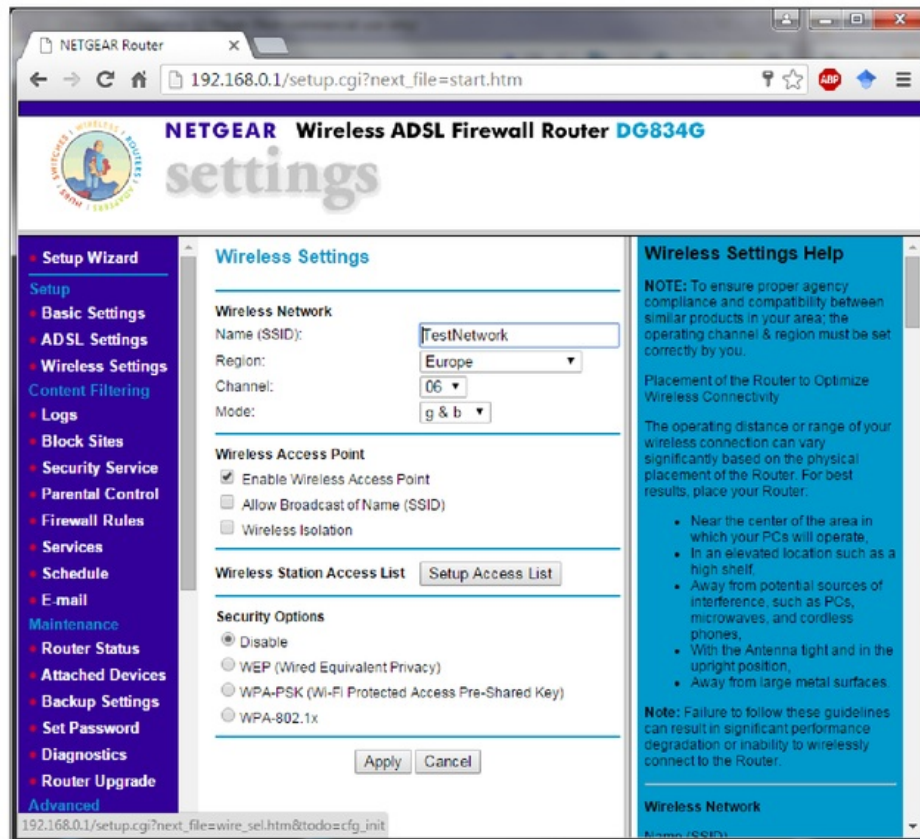


Figure 4 – Router configuration of the SSID and security options being performed in the Google Chrome browser

Virtual Machines

On creation of the virtual machines themselves the decision was made to split the virtual disk on all machines into multiple files as opposed to storing them as a single file (Figure 5). This was a safety precaution in the event of a fault occurring on the PC and the virtual machines having to be moved to another

device. However, by selecting this option there was a possibility of performance degradation occurring on large disk spaces.

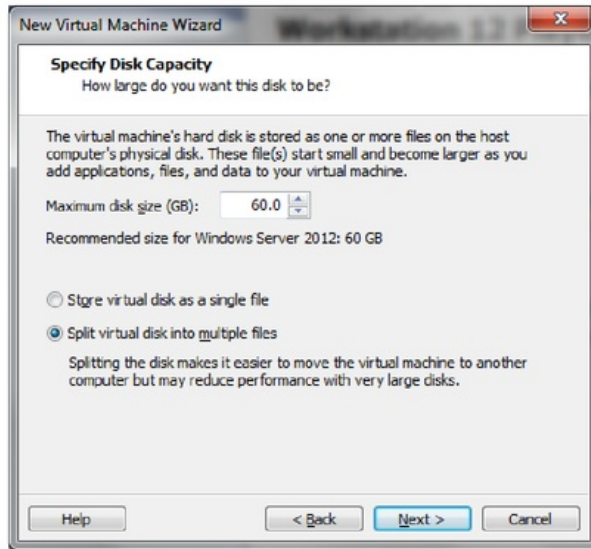


Figure 5 – Selecting the virtual disk to be split into multiple files as a security precaution during the creation of the VMs

Once all the virtual machines had been created and were operational they were then connected to the open network that was previously created. All connecting clients obtained their IP addresses dynamically using the routers DHCP function that held a pool of IP addresses in the range of 192.168.0.2 – 192.168.0.254, with a class C subnet of 255.255.255.0. The server however was assigned a static IP address of 192.168.0.100/24 to imitate real-world network configuration standards.

Reconnaissance

With the network having no security or encryption, any device with wireless capabilities can access the network. All network traffic therefore is travelling through the network in plain text meaning usernames and passwords can be viewed by anybody with the knowhow. However, username and password will be ascertained at a later stage in a separate attack. Therefore, for the reconnaissance attack network mapping was initiated for the purpose of gaining further information on the other devices connected to the network.

WEP

The WEP security feature possessed two possibilities of encryption to generate a network key, 64 bit and 128 bit encryption. In order to draw a comparison between the differing encryptions, both were attacked, recording the amount of IVs captured before cracking as well as the time taken to crack each key. There was also the option to create a unique key or generate a key by means of a passphrase. In this instance the decision was made to generate a key using the passphrase “password”, as it was a more efficient process due to the key generator producing a selection of four keys (shown in Figure 6).

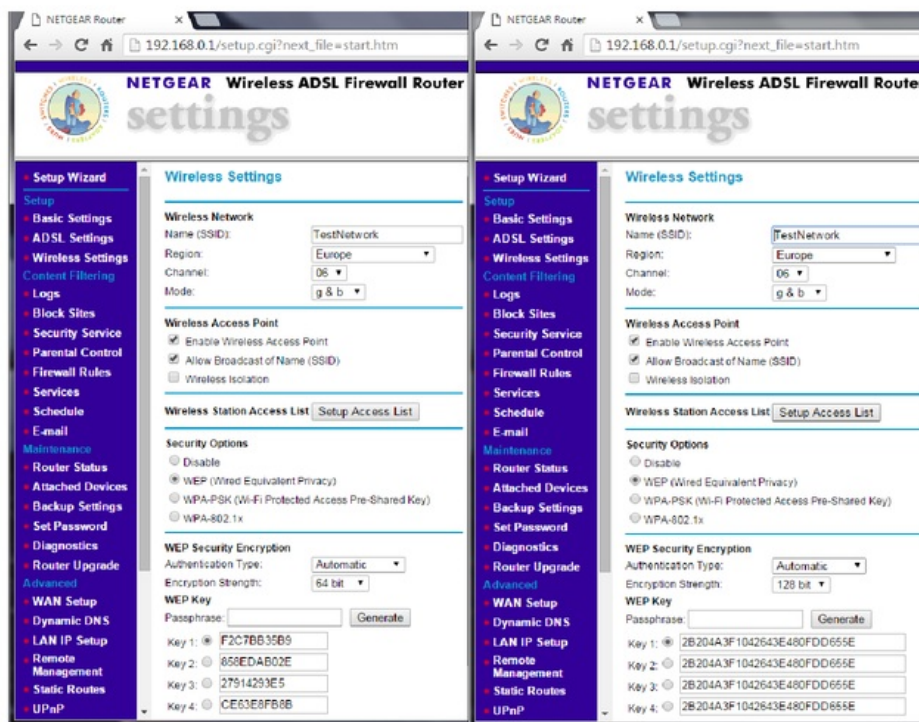


Figure 6 – The implementation of WEP using both 64 bit and 128 bit encryption keys generated using the passphrase “password”

WPA

Having bypassed WEP, WPA was then implemented onto the router. WPA does not support key generation and therefore a network key was applied. WPA enforces minimum and maximum key lengths of 8 and 63 so for the purpose of this experiment the network key “password” was used to in accordance with the minimum key length rule.

Once WPA was functioning, a dictionary attack was then launched using a list of 1,000 words. For the experiment to persist the network key was manually entered into the list of dictionary words that contained all lowercase words with no numerical values included. The reason for the limitations in regards to the length and complexity of the wordlist was due to a more complex word list and network key, resulting in an increase in the time it would have taken to crack the key. As the purpose of the experiment was a means of assessing the impact attacks have on a wireless network, the time in which it took to crack the key would have been irrelevant.

Although a time comparison could have been made between two network keys with different complexities, the results would have also been immaterial as the time taken to crack a key is solely dependent on where the word is positioned within the wordlist.

Soft AP

While the unauthorised VM could use the external WLAN card to monitor and penetrate the network; it also had the additional benefit of being able to impersonate the TestNetwork wireless AP. The external WLAN card was masked with the same SSID as the legitimate AP and would then allow unsuspecting clients to connect to it. Once connected to the Soft AP, each device would be issued with an IP address created for the Soft AP, thus allowing all communications between clients to traverse through it (see Figure 7). With the Soft AP in place, information such as usernames and passwords could be retrieved.

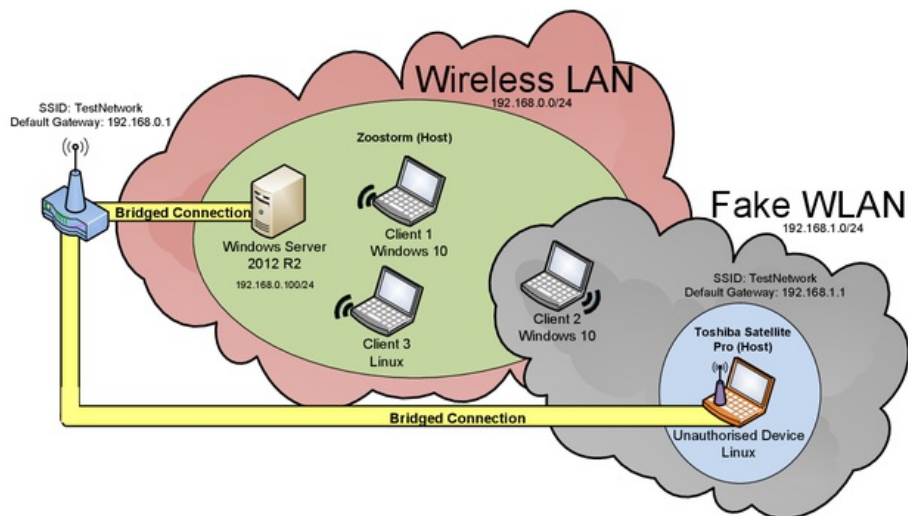


Figure 7 – A diagram that represents how the unauthorised VM creates an imitation of the authentic WLAN so that unsuspecting clients can connect to it and then communicate through its bridged connection to the legitimate AP

As shown in Figure 7, for the Soft AP to function appropriately it also had to be connected to a legitimate router. If for instance the Soft AP had been used to filter internal traffic between those devices connected to it and those connected to the legitimate AP so not to raise suspicion, then it would have been connected to the TestNetwork AP. However, the experiment looked to gain personal details of those devices looking to access the Internet and therefore the Soft AP was connected to an authentic home router with an Internet connection. So this could transpire, the Linux virtual machine (the Soft AP) bridged a connection to the hosts Laptop's internal WLAN card and by proxy connected to the authentic router.

To defend against a Soft AP, client side software tasked with monitoring the network to detect any Rouge AP, either software or hardware, was installed. The software was run on the authorised Linux machine and would alert the administrator to any discrepancies found. This method of defence was chosen as it was an effective yet simple approach that provided protection and detection against a difficult attack.

TCP SYN Flooding

To gain a greater understanding of the effects the TCP/SYN Flooding DoS attack had on the network, it was performed on the default gateway of the network and the Server virtual machine. The attacks themselves were not conducted simultaneously, they were performed separately so that the effects of the attack on the default gateway could be witnessed and the security features that prevent the attack on the Server addressed. The reason that the attack was performed on the two devices was due to an embedded security feature executed within the Windows OS installed on the Server machine. This feature will be discussed in greater detail at a later stage of this paper.

Implementation

Physical Network

Once all the authorised devices had been created and subsequently connected to the LAN, communication between the devices had to be established. In order to establish whether there were lines of communication present, the ping command was used. The results showed that the ICMP packets were dropped as they could not reach their destination which was what the expected outcome would be. This was due to the active Windows firewall on all devices preventing the ICMP packets from being received by the device they were tasked with protecting. So that ICMP communications between the LAN devices could exist, the firewall on each device was deactivated and then the ping command issued once more. As expected there was a reply from the recipient of the ping thus proving communication exists between the authorised VMs (see Figure 8).



```
Microsoft Windows [Version 10.0.10586]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Users\Client 1>ping 192.168.0.18

Pinging 192.168.0.18 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 192.168.0.18:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

C:\Users\Client 1>

C:\Users\Client 1>ping 192.168.0.18

Pinging 192.168.0.18 with 32 bytes of data:
Reply from 192.168.0.18: bytes=32 time<1ms TTL=128
Reply from 192.168.0.18: bytes=32 time<1ms TTL=128
Reply from 192.168.0.18: bytes=32 time<1ms TTL=128
Reply from 192.168.0.18: bytes=32 time<1ms TTL=128

Ping statistics for 192.168.0.18:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 1ms, Average = 0ms

C:\Users\Client 1>
```

Figure 8 – The ping command being issued before and after the deactivation of the firewall on all the Windows VMs

Reconnaissance

A reconnaissance attack on an unsecure network can yield many results depending upon the method research. In this project, Nmap was used to map

the devices that were connected to the network. The scan sort out and was successful in discovering a variety of information on these devices such as OS, open ports, IP addresses and MAC addresses. This kind of attack provides valuable information when gathering information on a network and it is relatively easy to deploy. An example of the results from this attack can be found in appendix 1. Defence against an attack such as this would be to employ security and encryption methods, which will prevent connection from unwanted users and also prevent sensitive data traveling around the network in plain text.

WEP

There are various applications that use numerous different techniques within the Kali suite that can be used to crack WEP encryption. For this experiment the decision was made to use the Wifite suite over the more popular Aircrack-ng suite. The primary reason for doing so was to ensure diversity regarding the suites being used and Aircrack-ng will feature later in the project. Wifite itself is very modest due to the simplicity of its execution, while its convenience not only allows it to cracks WEP, but it also detects the surrounding wireless networks using the encryption.

Once the TestNetwork had been detected and selected, Wifite gave the option to specify how many IVs would need to be gathered before cracking would commence. As stated previously, WEP would be cracked on both the 64 bit and 128 bit encryption keys, so for both these attacks the IV collection figure was set to 20,000. From this point forward these attacks will be referred to as attack A (64 bit key) and attack B (128 bit key). The time both keys (A and B) took to reach this figure was recorded using Wifite's integrated countdown clock (set at 10 minutes) and then compared to establish which would be cracked the quickest. The times recorded can be seen below in Figure 9.

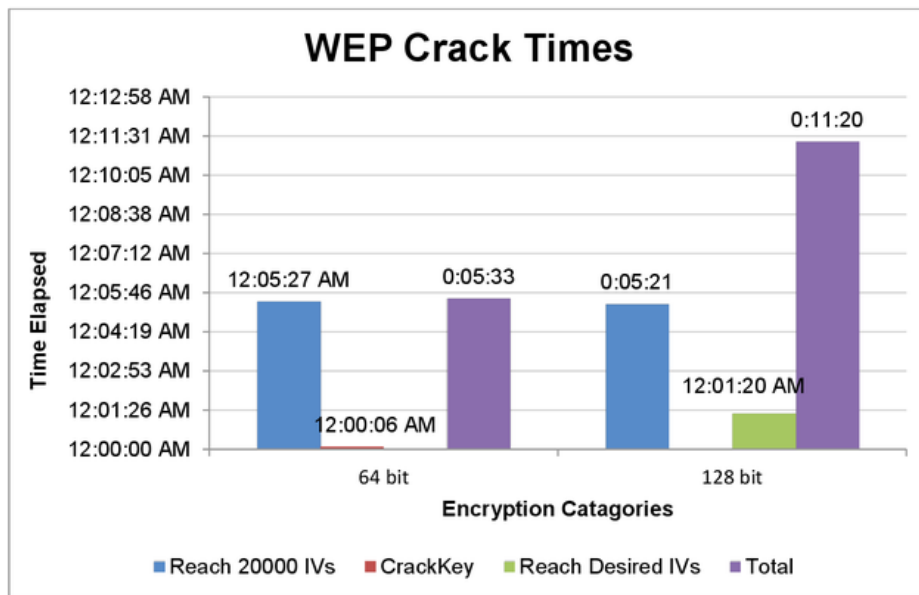


Figure 9 - IV capturing times recorded when cracking WEP

As expected the A was cracked in a shorter period of time than the B. It took just over double the time to crack the B than it did to crack the A, although the attack on B revealed that 20,000 IVs were captured faster than the attack on A. Figure 1 also suggests that A may have been cracked by capturing less IVs than the warranted 20,000 minimum limit set, as on completion of the IV capturing no other IVs were required, which resulted in A being cracked within 6 seconds.

Unfortunately, the time taken between the required number of IVs being captured and the cracking of B could not be recorded due to Wifite's 10 minute countdown clock elapsing before the process was completed. It did however record the number of IVs captured as the time elapsed (shown in Figure 10) and this figure was then used to calculate the time it took to reach the desired number of IVs required to crack B (shown in Figure 9).

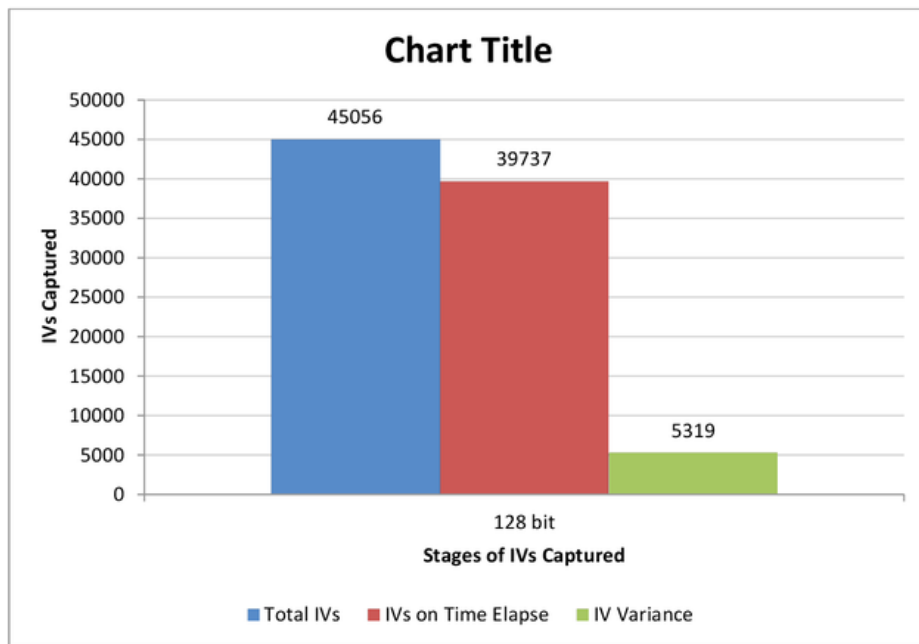


Figure 10 - The number of IVs captured after the time had elapsed with the total captured and the variance between the two

WPA

Where WEP was cracked by capturing IVs, WPA was cracked using a brute force dictionary attack via the Aircrack application. Unlike Wifite, Aircrack has a more complex and diverse usability due to it using various tools within the application. Such tools include Airmo-ng which was used to place the interface hosting the external WLAN card into monitoring mode and also giving it an interface ID. A list of active processes was then displayed and all were subsequently terminated to avoid future issues while using some of the other tools required for this attack.

A tool that would have been affected would have been Airdump-ng which was then used on the monitoring interface. Airdump-ng is a detection tool that provided a list of the wireless networks within its vicinity and several other important characteristics that were crucial in the progression of this attack (see Figure 11).

CH 9][Elapsed: 1 min][2016-03-02 12:07

BSSID	PWR	Beacons	#Data, #/s	CH	MB	ENC	CIPHER	AUTH	ESSID
00:18:4D:F6:01:AE	-35	204	0	0	6	54	WPA	TKIP	PSK TestNetwork
00:1E:2A:13:63:C2	-68	100	19	0	11	54e	WPA2	CCMP	PSK NETGEAR
0C:D6:B0:45:B9:DC	-82	66	12	0	1	54e	WPA2	CCMP	PSK TALKTALK45B9D5
68:A0:F6:63:0C:F8	-97	61	0	0	7	54e	WPA2	CCMP	PSK TALKTALK630CF8
4C:17:EB:67:E6:8A	-97	3	0	0	11	54e	WPA2	CCMP	PSK SKY7E689

BSSID	STATION	PWR	Rate	Lost	Frames	Probe
(not associated)	AC:22:08:8F:9E:9E	-49	0 - 1	0	12	
(not associated)	A8:88:08:1D:35:FF	-67	0 - 1	0	27	NETGEAR
(not associated)	80:D2:1D:38:CC:7F	-73	0 - 1	0	2	NETGEAR
(not associated)	F0:24:75:54:42:86	-83	0 - 1	0	7	NETGEAR
00:18:4D:F6:01:AE	60:E3:27:1E:80:71	-49	0 - 1	0	1	

Figure 11 - A detailed list of detected wireless APs along with important characteristics

From this list, the network was then identified and added to the monitoring interface using its BSSID and the channel it was operating on. Once successfully added to the monitoring interface, Airdump-ng then displayed an additional list of all the BSSIDs for the devices connected to the TestNetwork AP, which was also stored in an information file. With Airdump-ng continuously running in the background, Aireplay-ng was then used to de-authenticate a client from the AP so that re-association could occur. The client was selected using its BSSID and thusly monitored until it established a 4-way handshake with the AP. This process can be seen within Appendix 2.

The final phase of this attack was to use the wordlist (discussed previously) and the information file created by Airdump-ng with the Aircrack-ng tool. Aircrack then ran through the wordlist checking each word against the information obtained during the re-association process that was stored in the information file. As anticipated the network key was eventually paired with the matching word within the list and the password was revealed (see Figure 12).

```
[00:00:00] 72 keys tested (541.81 k/s)

Current passphrase: password
KEY FOUND! [ password ]
KEY FOUND! [ password ]
Master Key      : F0 78 6C 79 2E F5 8E CE B0 F1 9D 56 12 46 B4 A5
                  A1 A8 F6 DB 41 28 88 C8 69 6C E3 FA B4 25 B8 B7

Transient Key   : AC DB CD 85 22 B3 C4 94 25 FD 6C 82 91 48 47 12
                  87 10 6F FC C7 81 FC 12 30 9C 7A C9 EA D7 32 8A
                  76 88 F4 B6 6D 8C 12 3C 28 D5 B3 DF F9 24 F6 02
                  CC 8F DE D9 23 6D A1 01 65 D0 88 D4 19 C0 76 D9

EAPOL HMAC     : 28 19 6D B8 D9 DF 2F 9D 4E 2B 94 98 84 91 35 37
```

Figure 12 – Aircrack-ng produced the network key following a dictionary attack

The only issue that arose during this attack was the implementation of the network key into the wordlist. For reasons unknown, when the word was first entered into the list it could not be detected. The second attempt was done through the command prompt and was successful. When the attack was complete, two matches were found within the wordlist (see Figure 12). Although this is not a problem in regards to finding the key, it would cause an increase in the time taken to find the key if this were to happen regularly and within a larger wordlist.

Soft AP

The first stage of configuring a Soft AP was to ensure that there was a DHCP server installed on the unauthorised Linux machine to ensure the creation of an illegitimate IP address scheme to be assigned to the connecting devices. Following this, the DHCP server was then configured by declaring its parameters that contained details such as the network IP address and subnet mask, default gateway and the SSID, within a de-configuration file. All configurations associated with this process can be found in appendix 3.

Once the DHCP server had been configured it was then time to begin the attack process. The initial aspect of the attack was to enable monitoring on the external WLAN card using Airmmon-ng (as discussed previously) so that Airbase-ng could then be initiated. Airbase-ng created the soft AP and a tunnel interface by proxy. This would act as the routing feature between the devices and the network router and therefore had to be configured so that this procedure could occur. Within this configuration the illegitimate network was declared, the conversion between illegitimate IP addresses and legitimate IP address was

acknowledged and the redirection of traffic through a different port along with several others. The configuration performed in Airbase-ng was then concluded by linking said configurations to the de-configuration file previously created and configured and then the DHCP server was started (full configuration can be seen in appendix 4). SSL strip was configured as a visual legitimacy feature for users connected to the Internet by including the lock symbol in the browser while the Ettercap suite was activated as a sniffing tool for the traffic being filtered through the interface.

A client machine was used to firstly verify that the soft AP had been created, and then secondly, check that it functioned correctly. The client detected two TestNetwork SSIDs (the authorised and the unauthorised APs) proving that the soft AP was indeed available (see appendix 5). It was then connected to the soft AP and as expected, obtained an IP address along with all other compulsory information from its DHCP pool (as shown in Figure 13).

Property	Value
Connection-specific DN...	TestNetwork
Description	Realtek RTL8192CU Wireless LAN 802
Physical Address	60-E3-27-1E-80-71
DHCP Enabled	Yes
IPv4 Address	192.168.1.3
IPv4 Subnet Mask	255.255.255.0
Lease Obtained	03 March 2016 19:35:11
Lease Expires	03 March 2016 19:45:11
IPv4 Default Gateway	192.168.1.1
IPv4 DHCP Server	192.168.1.1
IPv4 DNS Server	192.168.1.1
IPv4 WINS Server	
NetBIOS over Tcpip En...	Yes
Link-local IPv6 Address	fe80::a55a:8677:b36f:c8a2%12
IPv6 Default Gateway	
IPv6 DNS Server	

Figure 13 – Network connection settings obtained from when connected to the soft AP

The client was then used to simulate an attempt at accessing a social media account to verify that Ettercap was operational by capturing the login details entered on the client machine. The results proved successful as the details were captured and presented on the terminal in real-time and can be viewed in Figure 14.



```
HTTP : 31.13.90.36:80 -> USER: user@facebook.com PASS: password INFO: http://w
ww.facebook.com/
CONTENT: lsd=AVr6f2Zm&email=user%40facebook.com&pass=password&default_persistent
=0&timezone=&lgnrdim=&lgnrmd=024201_rh8b&lgnjs=n&locale=en_GB
```

Figure 14 – A visual confirmation of the username and password used when attempting to access a social network account

To defend against the soft AP, EvilAP Defender was run on the authorised Linux machine along with the installation of MySQL, Python and its libraries. The EvilAP Defender suite could run in one of two modes, Learning and Normal. Learning mode was the initial mode to be run in this experiment as it was the mode which provided information on the surrounding wireless network APs. The legitimate TestNetwork AP was identified and stored into the suites authorised AP list for comparison when running in Normal mode. Normal mode was then used to detect any rogue AP that had infiltrated the network by comparing the SSIDs and their respective BSSIDs to that of those within its authorised list. As expected there was a duplication of SSIDs, however there was a discrepancy with the BSSID counterpart and therefore EvilAP Defender identified that a fake AP was on the network (see appendix 6).

Unfortunately this approach does not prevent a rogue AP from accessing the network and it only notifies the administrator of an infringement when the suite is run, therefore it does not monitor the network constantly. The identification process is slightly limited in that the rogue AP would only be detected if the BSSIDs were not identical to those APs found within the list of authorised APs. It did however provide enough information and the aptitude to defend the network by implementing a DoS attack on the rogue AP, thus preventing any devices from connecting or communicating with the intruding AP.

TCP/SYN Flooding

The TCP/SYN flooding attack was initiated using MSF (Metasploit Framework) console feature. It allowed the target to be identified via its IP address, alongside the spoofed IP address and the target port. The attack was done on two separate occasions with the first attacking the default gateway of the network. Once executed, the DoS attack prohibited any new attempted connections to the network (see appendix 7) and also any communication between those devices already connected to the AP (see appendix 8).

The attack was then stopped and performed on the Server machine which has an embedded defence mechanism against this attack. Unfortunately, the feature could not be disabled and therefore the attack did not have any effect on the Server whatsoever. Microsoft does allow for device monitoring that provides information when a SYN attack has been attempted. Regrettably this could not be performed as the router also defends against such attacks and therefore the monitoring of the Server returned no results as the attack was prevented by the router.

Chapter 5 – Evaluation

The overall aim of this experiment was two-fold, firstly by conducting several attacks that are often performed by hackers in an attempt to gain illegal access to a wireless network. Secondly, as each attack was implemented, knowledge was gained regarding the construction of each attack and its effects on the network, in order to understand which techniques would be required to successfully defend against them.

The wealth of past research describes the insecurities surrounding wireless technologies, with the findings from the current experiment reiterating this. The attacks chosen to be implemented in this experiment sought to reflect the broad spectrum of attacks used within real-world scenarios. This decision was made in order to understand the variety of different outcomes that can result from such attacks, rather than focusing specifically on a sole aspect. As a result of this, the conclusions obtained are wide reaching and varied with regards to the primary aim of this experiment. An evaluation of each of the attacks and their subsequent defences carried out will be discussed below.

Reconnaissance

Through implementing Nmap, a mapping of the network and its connected devices was successful. The application returned a range of devices along with some detailed information regarding each such as their OSs and service packs installed, the name of the device, and the workgroup they were associated with. The attack itself could have been extended further by searching available ports, however it did function as expected which showed in its results. Following this, WEP was installed on the router.

WEP

As discussed in the Research and Review of Literature chapter, WEP is extremely insecure due to the number of weak keys it generates. This was upheld during this experiment as the Wifite application captured IVs and used them to crack the encryption and produce the APs key. The number of IVs captured varied between the levels of encryption (64 and 128 bit) and therefore so did the time taken to crack each. Though the attack produced comparable data between the two levels of encryption, a comparison between the data found amongst two different applications which conduct this attack could also

be performed. As a defence against this attack, WEP was replaced with WPA as the security and encryption mechanism of the network.

WPA

As discussed in chapter 2, WPA is vulnerable to dictionary attacks. This experiment not only supports the conclusions found within the past literature but it also discovered a fault with the attack itself. Dictionary attacks use text files filled with words or alphanumeric sequences (referred to as words) to crack passwords. Dependent upon where or if the password is situated within these text files is a direct correlation with the time it takes to crack a password. The results of this experiment showed that duplications of data within the files can occur and thus effect the time it takes to cipher through the document in search of the correct word. Consequently, studying the effects of such an occurrence where duplicate words were strategically positioned before the known password would also have given this experiment extra depth. Due to hardware limitations WPA could not be replaced with WPA2 which would have prevented this attack from occurring.

Soft AP

The Soft AP attack functioned exactly as anticipated by broadcasting itself as the legitimate AP, allowing clients to connect to it and then subsequently issuing network information (such as IP addresses, subnet masks and a default gateway) to those clients. In return, the Soft AP was then granted access to the traffic that those clients produced, which contained sensitive data for a social networking account. This corresponds with the literature previously discussed and highlights the extraordinary security threat to wireless networks in general, while also providing a great insight into the ease in which one could be created to act as a mobile Hotspot. Although there are various methods available to detect and prevent a rogue AP, most require additional hardware or software and therefore increases the cost of network maintenance. This experiment looked at a client side software that detects rogue Aps, however it was limited in its functionality as it would only detect when prompted. It did provide a means of prevention however this was not explored and should be considered as an avenue of future investigations.

TCP/SYN Flooding

As previously discussed, this attack became slightly complicated to investigate due to the defensive features contained within the Windows OS and the APs firewall. An attack was initiated successfully to provide an insight into the effects it has on a network which corresponds to past research, the process for defence was not so forthright. Though the outcome was as expected with the attack being prevented, the practice of the methods of defence could not be addressed due to the integrated security features that are integrated into the Windows OSs and the APs firewall

In conclusion, the experiment provides an overarching understanding of the attacks and security methods surrounding wireless networks, however it does not provide in-depth knowledge regarding generalised areas. An advantage of concentrating on specific categories of attacks is it may allow for greater understanding of how to completely defend against all aspects and techniques hackers may use. For example, rather than preventing a single DoS attack, greater knowledge into their wide variety specifically would allow for defence against all attacks encompassing the DoS category.

There are potential practical limitations within the current experiment with regards to the virtualisation software used. As the VMs are assigned portions of the specification capabilities of the host, only a limited number of machines could be created. As a result, the size of the network is limited and therefore the attacks undertaken can only be assessed on a small scale. For example the findings of this experiment may not be entirely reflective of the outcomes on a larger scale network as there would be more devices, AP's and ultimately a larger security threat. An additional practical limitation relates to the router used throughout the experiment. Due to age restrictions the implementation of WPA2 encryption could not be performed and therefore certain security aspects could not be incorporated. Consequently, future directions may include the use of an updated router that is used for business as opposed to personal use.

Chapter 6 - Conclusions and Recommendations

Conclusions

This experiment looked at performing network penetration techniques used by hackers and penetration testers that would highlight the security flaws found within wireless networks. The purpose of this was to then gain a wider understanding of the techniques required to detect and prevent against future attacks and therefore creating a more secure network. For this to evolve, virtual machines were created within the VMware virtualisation software and then connected as authorised devices to the network that had no security applied from its inception. In addition, a separate virtual machine was created to act as an imposter tasked with performing the attacks.

A total of 5 separate attacks were carried out resulting in a range of versatility with regards to the security features implemented on the network. Firstly a reconnaissance attack was performed using Nmap that provided information about devices connected to the network. At this point WEP encryption was enforced on the network to prevent unauthorised access. To test the insecurities that surrounded WEP, the Wifite suite was used which allowed the detection of the WEP network and then subsequently performed the attacks which cracked its 64 bit and 128 bit encryption levels in less than 6 and 12 minutes respectively. This led to the omission of WEP and the inclusion of WPA as the networks updated security encryption. Similarly to WEP, WPA also had its own insecurities that would also be identified and emphasised. Aircrack-ng was used to perform a brute force dictionary attack that revealed the password required to access the network. Unfortunately, due to the practical limitations of the outdated router, WPA2 could not be employed to counteract this weakness.

A MITM attack in the form of a Soft AP was then performed which allowed unsuspecting devices to connect to an AP that was impersonating the networks legitimate AP. This allowed the monitoring of all the communications between the devices and the Internet and as a result provided a user's login information for a social networking site. The use of the EvilAP Defender software on the client-side was used for detection and notification of the rogue APs existence on the network, by comparing the information on the surrounding wireless networks with a list comprised of legitimate APs. The final attack saw the TCP/SYN

Flooding DoS attack being performed on the default gateway (so that the effects could be seen) and on the Server client to show how the Windows OSs have integrated defences to prevent such an attack.

Recommendations

There are multiple avenues for further research regarding this experiment. Network security is not limited to the specific attacks generated in this experiment. For example, there are numerous ways of cracking WPA that are more efficient than a dictionary attack which can often take longer than found in this experiment. The use of the Reaver suite would provide a different and more efficient approach to attacking the flaws of WPA. The flaws found within WPA2 that were covered in the literature review (e.g. 'Hole 196') could also be addressed once implemented. Using different approaches to detecting and preventing rogue APs could be addressed as the EvilAP Defender does not prevent rogue APs from integrating onto the network. And although other preventative measures have been discussed, they may open up avenues of law suits due to the illegality of performing DoS attacks on hardware that are not under your position. For the physical properties, conducting such an experiment on a larger scale would provide a better understanding of the necessary security measures required to prevent the attacks on a more complex topology.

Reflection

Initially, when deciding which attacks would be appropriate to run in this experiment, I felt I did not have the breadth of knowledge to enable me to make an informed decision. However, through research into the relevant literature surrounding this topic I realised the variety of attacks that I could perform in this experiment, as well as those I would not be able to perform due to practical limitations. If I were to recreate a similar experiment now, I would initiate the attacks first because I have the knowledge available to do so and then write the literature review to specifically reflect these attacks.

How I perceived the experiment to be in the beginning did not reflect how it was in fact carried out. Originally, I thought it would all take place on a PC and would therefore be a relatively straightforward process, however that wasn't the case. I began to realise I required additional hardware (external WLAN cards and a wireless router) and although I had prior knowledge about the various different attacks, I did not have any experience of instigating these attacks or defending against them. I was previously taught the theoretical process of different attacks on wireless networks, however I had no prior practical experience of physically carrying these out. Therefore, I found transferring the theory into practice challenging. I had also not used a Linux machine prior to this experiment and as a result I was not competent in understanding its features and how to navigate through its operating system. Consequently, it took me longer to understand how to implement attacks using the command line interface that is associated with running Linux applications. As I am more familiar with Windows, where different applications are launched on separate interfaces, it took me a while to familiarise myself with Linux's unique approach to this. In future, if I were to carry out a similar project, I feel I have now acquired the appropriate skills and knowledge to use the Linux system to a higher standard than I previously could at the beginning of this project.

I now feel I have a better understanding of the functionality of virtual machines and their limitations which I did not have at the start of this process, mainly in regards to how they function within a wireless network. In addition, I discovered that external WLAN cards are required when performing specific attacks because the host's internal WLAN card does not support the packet injection features. It took a lot of research into this to understand why the attacks could

not be performed via the host's internal WLAN card, thus wasting a lot of time. So if I were to recreate an experiment such as this, I would now understand such equipment to be a necessity when using virtual machines.

The practical application has broadened my realisation of how insecure wireless networks are and how easily they can be manipulated to gain access or information. Although the process of doing this was a lot more complicated than I initially thought, I now have improved knowledge and therefore find it easier to understand and implement. As a result of this, I can now see how easy it can be for the average person to replicate or carry out these attacks themselves. It has made me more aware of the insecurities surrounding wireless networks, more specifically for hotspots, as well as how easily someone can create a fake one in order to gain information.

If I were to carry out future research into this area, I would look into more specific attacks rather than carrying out a broad range of these attacks in order to gain a greater understanding of the variety of attacks within a specific category. I would focus more on a man in the middle attack as it can be less detectable compared to DoS attacks. In hindsight, the network mapping attack did not reveal anything surprising to me and I would probably have gone for a traffic analysis attack if I were to do this again. I expect this may possibly reveal more interesting information such as usernames and passwords and unencrypted texts that could've been manipulated.

References

1. Press G. Internet of Things By The Numbers: Market Estimates And Forecasts - Forbes [Internet]. 2014 [cited 2015 Oct 20]. Available from: <http://www.forbes.com/sites/gilpress/2014/08/22/internet-of-things-by-the-numbers-market-estimates-and-forecasts/>
2. The Internet of Things Will Drive Wireless Connected Devices to 40.9 Billion in 2020 | ABI Research [Internet]. [cited 2015 Oct 20]. Available from: <https://www.abiresearch.com/press/the-internet-of-things-will-drive-wireless-connect/>
3. Fluhrer S, Mantin I, Shamir A. Weaknesses in the key scheduling algorithm of RC4. In: Selected areas in cryptography [Internet]. Springer; 2001 [cited 2015 Oct 26]. p. 1–24. Available from: http://link.springer.com/chapter/10.1007/3-540-45537-X_1
4. Tews E, Beck M. Practical attacks against WEP and WPA. In: Proceedings of the second ACM conference on Wireless network security. ACM; 2009. p. 79–86.
5. Weakness in Passphrase Choice in WPA Interface - Wi-Fi Networking News [Internet]. [cited 2015 Oct 27]. Available from: http://wifinetnews.com/archives/2003/11/weakness_in_passphrase_choice_in_wpa_interface.html
6. Waliullah M, Moniruzzaman ABM, Rahman MS, others. An Experimental Study Analysis of Security Attacks at IEEE 802.11 Wireless Local Area Network. International Journal of Future Generation Communication and Networking. 2015;8(1):9–18.
7. Pearson WR, Lipman DJ. Improved tools for biological sequence comparison. Proceedings of the National Academy of Sciences. 1988;85(8):2444–8.
8. Henry KM. Penetration Testing: Protecting Networks and Systems. IT Governance Ltd. ISBN 978-1-84928-372-4.
9. Common Types of Network Attacks [Internet]. [cited 2015 Nov 10]. Available from: <https://technet.microsoft.com/en-us/library/cc959354.aspx>
10. What is Kali Linux ? | Kali Linux [Internet]. [cited 2015 Nov 11]. Available from: <http://docs.kali.org/introduction/what-is-kali-linux>
11. Troya AS, Astudillo JJ, Romero CG, Saenz FG, Diaz J. Vulnerability detection in 802.11 i wireless networks through link layer analysis. In:

- Communications (LATINCOM), 2014 IEEE Latin-America Conference on [Internet]. IEEE; 2014 [cited 2015 Nov 11]. p. 1–6.
12. Man-in-the-middle attack - OWASP [Internet]. [cited 2015 Nov 11]. Available from: https://www.owasp.org/index.php/Man-in-the-middle_attack
 13. Chen Z, Guo S, Zheng K, Yang Y. Modeling of man-in-the-middle attack in the wireless networks. In: Wireless Communications, Networking and Mobile Computing, 2007 WiCom 2007 International Conference on [Internet]. IEEE; 2007. p. 2255–8.
 14. Kotkar A, Nalawade A, Gawas S, Patwardhan A. Network Attacks and Their Countermeasures. Network [Internet]. 2013 [cited 2015 Nov 11];1(1). Available from: <http://core.ac.uk/download/pdf/26774787.pdf>
 15. Cashion J, Bassiouni M. Robust and low-cost solution for preventing sidejacking attacks in wireless networks using a rolling code. In: Proceedings of the 7th ACM symposium on QoS and security for wireless and mobile networks [Internet]. ACM; 2011. p. 21–6.
 16. Kumar V. Three Tier Verification Technique to foil session sidejacking attempts. In: Internet (AH-ICI), 2011 Second Asian Himalayas International Conference. IEEE; 2011. p. 1–4.
 17. Sood A, Enbody R. The state of HTTP declarative security in online banking websites. Computer Fraud & Security. 2011;2011(7):11–6.
 18. 1. Bhatia P, Laurendeau C, Barbeau M. Solution to the wireless evil-twin transmitter attack. In: Risks and Security of Internet and Systems (CRiSIS), 2010 Fifth International Conference. IEEE; 2010. p. 1–7.
 19. Mustafa H, Xu W. CETAD: Detecting evil twin access point attacks in wireless hotspots. In: Communications and Network Security (CNS), 2014 IEEE Conference. IEEE; 2014. p. 238–46.
 20. Nakhila O, Dondyk E, Amjad MF, Zou C. User-side Wi-Fi Evil Twin Attack detection using SSL/TCP protocols. In: Consumer Communications and Networking Conference (CCNC), 2015 12th Annual IEEE. IEEE; 2015. p. 239–44.
 21. Chao Yang, Yimin Song, Guofei Gu. Active User-Side Evil Twin Access Point Detection Using Statistical Techniques. IEEE Transactions on Information Forensics and Security. 2012 Oct;7(5):1638–51.
 22. Cisco Adaptive Wireless Intrusion Prevention Service Configuration Guide, Release 5.2 - wIPS Policy Alarm Encyclopedia [Cisco Adaptive Wireless

IPS Software] - Cisco [Internet]. [cited 2015 Nov 13]. Available from: http://www.cisco.com/c/en/us/td/docs/wireless/mse/3350/5-2/wlPS/configuration/guide/msecg_wlPS/msecg_appA_wlPS.html#wp1166170

23. Lanze F, Panchenko A, Ponce-Alcaide I, Engel T. Hacker's Toolbox: Detecting Software-Based 802.11 Evil Twin Access Points. In: 12th Annual IEEE Consumer Communications & Networking Conference. IEEE; 2015.
24. McDowell M. Understanding Denial-of-Service Attacks | US-CERT [Internet]. 2013 [cited 2015 Nov 14]. Available from: <https://www.us-cert.gov/ncas/tips/ST04-015>
25. Beitollahi H, Deconinck G. Analyzing well-known countermeasures against distributed denial of service attacks. *Computer Communications*. 2012 Jun;35(11):1312–32.
26. Al-Duwairi B, Manimaran G. Distributed packet pairing for reflector based DDoS attack mitigation. *Computer Communications*. 2006 Aug;29(12):2269–80.
27. Sardana A, Joshi RC, Kim T, Jang S. Deciding optimal entropic thresholds to calibrate the detection mechanism for variable rate DDoS attacks in ISP domain: honeypot based approach. *Journal of Intelligent Manufacturing*. 2010 Oct;21(5):623–34.
28. Wang L, Wyglinski AM. A combined approach for distinguishing different types of jamming attacks against wireless networks. In: *Communications, Computers and Signal Processing (PacRim)*, 2011 IEEE Pacific Rim Conference. IEEE; 2011. p. 809–14.
29. Xu W, Trappe W, Zhang Y, Wood T. The feasibility of launching and detecting jamming attacks in wireless networks. In: *Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing*. ACM; 2005. p. 46–57.
30. Signal Jamming and its Modern Applications | Editor IJSR - Academia.edu [Internet]. [cited 2015 Nov 17]. Available from: http://www.academia.edu/4700714/Signal_Jamming_and_its_Modern_Applications
31. Xu W, Wood T, Trappe W, Zhang Y. Channel surfing and spatial retreats: defenses against wireless denial of service. In: *Proceedings of the 3rd ACM workshop on Wireless security*. ACM; 2004. p. 80–9.

32. Misra S, Dhurandher SK, Rayankula A, Agrawal D. Using honeynodes for defense against jamming attacks in wireless infrastructure-based networks. *Computers & Electrical Engineering*. 2010 Mar;36(2):367–82.
33. Wang H, Zhang D, Shin KG. Detecting SYN flooding attacks. In: *INFOCOM 2002 Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies Proceedings*. IEEE; 2002. p. 1530–9.
34. Schuba CL, Krsul IV, Kuhn MG, Spafford EH, Sundaram A, Zamboni D. Analysis of a denial of service attack on TCP. In: *Security and Privacy, 1997 Proceedings, 1997 IEEE Symposium*. IEEE; 1997. p. 208–23.
35. A. Zuquete, Sept 2002, "Improving the functionality of SYN cookies," in *Proceedings of 6th IFIP Communications and Multimedia Security Conference* pp. 57–77.
36. J. Lemon, Feb.2002 "Resisting SYN flooding DoS attacks with a SYN cache," in *Proceedings of USENIX BSDCon'2002*, pp.89–98.
37. Kumarasamy S, Gowri Shankar A. An Active Defense Mechanism for TCP SYN flooding attacks. *arXiv preprint arXiv:12012103* [Internet]. 2012 [cited 2015 Nov 18]; Available from: <http://arxiv.org/abs/1201.2103>
38. Lim S-H, Kim J-H. Dynamic Security Level Changing Strategy Using Attack Predictions-Case Study of TCP SYN Attacks. In: *IT Convergence and Security (ICITCS), 2014 International Conference*. IEEE; 2014. p. 1–4.
39. Bogdanoski M, Risteski A. Wireless network behavior under icmp ping flood dos attack and mitigation techniques. *International Journal of Communication Networks and Information Security (IJCNIS)*. 2011;3(1).
40. Junior RB, Kumar S. Apple's Lion vs Microsoft's Windows 7: Comparing Built-In Protection against ICMP Flood Attacks. *Journal of Information Security*. 2014;05(03):123–35.
41. DoS Attack (DDOS,Ping Flood,Smurf,Fraggle,SYN Flood,Teardrop) [Internet]. [cited 2015 Nov 18]. Available from: http://webcache.googleusercontent.com/search?q=cache:http://www.ids-sax2.com/articles/PreventDosAttacks.htm&gws_rd=cr&ei=6tZMVsGxNMbsUpmosvAL
42. Address Resolution Protocol [Internet]. [cited 2015 Nov 19]. Available from: <https://technet.microsoft.com/en-us/library/cc940021.aspx>
43. Fleck B, Dimov J. Wireless access points and arp poisoning. Online document [Internet]. 2001 [cited 2015 Nov 19]; Available from:

<http://bandwidthco.com/whitepapers/netforensics/arp-rarp/Wireless%20Access%20Points%20and%20ARP%20Poisoning.pdf>

44. Sharma D, Khan O, Aggarwa K, Vaidya P, "A New Approach to Prevent ARP Spoofing", In Proceedings of International Journal of Innovative Technology and Exploring Engineering (IJITEE) ISSN: 2278-3075, Volume-3, Issue-1, June 2013.
45. Nam SY, Djuraev S, Park M. Collaborative approach to mitigating ARP poisoning-based Man-in-the-Middle attacks. Computer Networks. 2013 Dec;57(18):3866–84.
46. Nam SY, Kim D, Kim J. Enhanced ARP: preventing ARP poisoning-based man-in-the-middle attacks. Communications Letters, IEEE. 2010;14(2):187–9.
47. Nam SY, Jurayev S, Kim S-S, Choi K, Choi GS. Mitigating ARP poisoning-based man-in-the-middle attacks in wired or wireless LAN. EURASIP Journal on Wireless Communications and Networking. 2012;2012(1):1–17.
48. WPA2 Hole196 Vulnerability [Internet]. [cited 2015 Nov 9]. Available from: <http://www.airtightnetworks.com/WPA2-Hole196>
49. Agarwal M, Biswas S, Nandi S. Advanced Stealth Man-in-The-Middle Attack in WPA2 Encrypted Wi-Fi Networks. IEEE Communications Letters. 2015 Apr;19(4):581–4.
50. Marshall D. Understanding Full Virtualization, Paravirtualization, and Hardware Assist. VMWare. 2007;1–14.
51. Davis D, Lowe SD. 2015 State of SMB IT Infrastructure Survey Results. ActualTech Media; 2015.
52. What's the difference between Type 1 and Type 2 hypervisors? [Internet]. [cited 2015 Nov 25]. Available from: <http://searchservvirtualization.techtarget.com/feature/Whats-the-difference-between-Type-1-and-Type-2-hypervisors>
53. Gammage B, Shiffler III G. Report Highlight for Dataquest Insight: PC Virtualization Forecast Scenarios. Gartner Research. 2007 Aug 8.
54. Mulfari D, Celesti A, Villari M, Puliafito A. Using Virtualization and noVNC to Support Assistive Technology in Cloud Computing. IEEE; 2014. p. 125–32.
55. Bechtsoudis A, Sklavos N. Aiming at higher network security through extensive penetration tests. Latin America Transactions, IEEE (Revista IEEE America Latina). 2012;10(3):1752–6.

56. Petrovic D, Schiper A. Implementing Virtual Machine Replication: A Case Study Using Xen and KVM. IEEE; 2012. p. 73–80.
57. Perez E. 802.11 i (How we got here and where are we headed). SANS Institute [Internet]. 2004 [cited 2015 Nov 25]; Available from: <http://cracking8hacking.com/cracking-hacking/Ebooks/Wireless/SHELF/paper1467.pdf>
58. Kali Linux Custom Images Download - Offensive Security [Internet]. [cited 2016 Feb 2]. Available from: <https://www.offensive-security.com/kali-linux-vmware-virtualbox-image-download/>

Appendices

1.

```
Host script results:
  _nbstat: NetBIOS name: CARLS_PC, NetBIOS user: <unknown>, NetBIOS MAC: ac:22:0b:8f:9e:9e (Asustek Computer)
  _smb-os-discovery:
    OS: Windows 7 Home Premium 7601 Service Pack 1 (Windows 7 Home Premium 6.1)
    OS CPE: cpe:/o:microsoft:windows_7::sp1
    Computer name: Carls_PC
    NetBIOS computer name: CARLS_PC
    Workgroup: WORKGROUP
    System time: 2016-04-09T20:27:01+01:00
  _smb-security-mode:
    account_used: guest
    authentication_level: user
    challenge_response: supported
    message_signing: disabled (dangerous, but default)
  _smbv2-enabled: Server supports SMBv2 protocol
```

2.

```
root@kali: ~
File Edit View Search Terminal Help

root@kali:~# aireplay-ng -0 0 -a 00:18:4D:F6:01:AE wlan0mon
04:32:47 Waiting for beacon frame (BSSID: 00:18:4D:F6:01:AE) on channel 6
NB: this attack is more effective when targeting
a connected wireless client (-c <client's mac>).
04:32:48 Sending DeAuth to broadcast -- BSSID: [00:18:4D:F6:01:AE]
04:32:48 Sending DeAuth to broadcast -- BSSID: [00:18:4D:F6:01:AE]
04:32:49 Sending DeAuth to broadcast -- BSSID: [00:18:4D:F6:01:AE]
04:32:50 Sending DeAuth to broadcast -- BSSID: [00:18:4D:F6:01:AE]
04:32:50 Sending DeAuth to broadcast -- BSSID: [00:18:4D:F6:01:AE]
04:32:51 Sending DeAuth to broadcast -- BSSID: [00:18:4D:F6:01:AE]
04:32:52 Sending DeAuth to broadcast -- BSSID: [00:18:4D:F6:01:AE]
04:32:52 Sending DeAuth to broadcast -- BSSID: [00:18:4D:F6:01:AE]
04:32:53 Sending DeAuth to broadcast -- BSSID: [00:18:4D:F6:01:AE]
04:32:55 Sending DeAuth to broadcast -- BSSID: [00:18:4D:F6:01:AE]
04:32:56 Sending DeAuth to broadcast -- BSSID: [00:18:4D:F6:01:AE]
^Z

root@kali: ~
File Edit View Search Terminal Help

CH 6 || Elapsed: 6 mins || 2016-03-03 04:34 || WPA handshake: 00:18:4D:F6:01:AE
BSSID PWR RXQ Beacons #Data, #/s CH FD ENC CIPHER AUTH ESSID
00:18:4D:F6:01:AE -33 90 3767 214 0 6 54 . WPA TKIP PSK TestNetwork
BSSID STATION PWR Rate Lost Frames Probe
00:18:4D:F6:01:AE 60:E3:27:1E:80:71 -41 54 -24 0 187
```

3.

```
authoritative;
default-lease-time 600;
max-lease-time 7200;
subnet 192.168.1.0 netmask 255.255.255.0 {
  option routers 192.168.1.1;
  option subnet-mask 255.255.255.0;
  option domain-name "TestNetwork";
  option domain-name-servers 192.168.1.1;
  range 192.168.1.2 192.168.1.40;
}
```

4.

```
root@kali:~# ifconfig at0 192.168.1.1 netmask 255.255.255.0
root@kali:~# ifconfig at0 mtu 1400
root@kali:~# route add -net 192.168.1.0 netmask 255.255.255.0 gw 192.168.1.1
root@kali:~# echo 1 > /proc/sys/net/ipv4/ip_forward
root@kali:~# iptables -t nat -A PREROUTING -p udp -j DNAT --to 192.168.0.1
root@kali:~# iptables -P FORWARD ACCEPT
root@kali:~# iptables --append
iptables v1.4.21: unknown option "--append"
Try 'iptables -h' or 'iptables --help' for more information.
root@kali:~# iptables --append FORWARD --in-interface at0 -j ACCEPT
root@kali:~# iptables --table nat --append POSTROUTING --out-interface eth0 -j MASQUERADE
root@kali:~# iptables -t nat -A PREROUTING -p tcp --destination-port 80 -j REDIRECT --to-port 10000
root@kali:~# dhcpd -cf /etc/dhcpd.conf -pf /var/run/dhcp.pid at0
bash: dhcpd: command not found
root@kali:~# dhcpd -cf /etc/dhcpd.conf -pf /var/run/dhcp.pid at0
Internet Systems Consortium DHCP Server 4.3.1
Copyright 2004-2014 Internet Systems Consortium.
All rights reserved.
For info, please visit https://www.isc.org/software/dhcp/
Config file: /etc/dhcpd.conf
Database file: /var/lib/dhcp/dhcpd.leases
PID file: /var/run/dhcp.pid
/etc/dhcpd.conf line 4: subnet 192.168.1.1 netmask 255.255.255.0: bad subnet number/mask combination.
subnet 192.168.1.1 netmask 255.255.255.0
8:27 Trying to set MTU on wlan0mon to 1800
8:27 Access Point with BSSID 68:E3:27:1E:C1:DE started.
```

5.

WiFi



TestNetwork
Secured



NETGEAR
Secured



TALKTALK45B9D5
Secured



TestNetwork
Open
Other people might be able to see info that you send over this network



Connect automatically

6.

```
CH 10 ][ Elapsed: 18 s ][ 2016-04-02 13:02
master
BSSID          PWR Beacons  #Data, #/s  CH  MB  ENC  CIPHER AUTH ESSID
00:18:4D:F6:01:AE -47    56        0    6  54  . WPA  TKIP  PSK  TestNetwork
00:1E:2A:13:63:C2 -64    30        7    0  11  54e WPA2  CCMP  PSK  NETGEAR
0C:D6:BD:45:B9:DC -70    12        0    0   4  54e WPA2  CCMP  PSK  TALKTALK45B9D5

BSSID          STATION          PWR   Rate    Lost   Frames  Probe
00:1E:2A:13:63:C2 30:3A:64:AE:3D:1E -1    2e- 0    0        1
00:1E:2A:13:63:C2 FC:C2:DE:64:8B:EF -26   0 -24    0        1
00:1E:2A:13:63:C2 80:D2:1D:38:CC:7F -67   54e- 1    0        2

Entering Normal Mode ...

===== Check Result =====

Fake AP with different MAC Detected!

(BSSID - SSID - PWR - Channel - Cipher - Privacy - Auth)
(60:E3:27:1E:C1:0E - TestNetwork - '-97' - 11 - - OPN - )

Deauth time: 0
Preventive Mode is not enabled

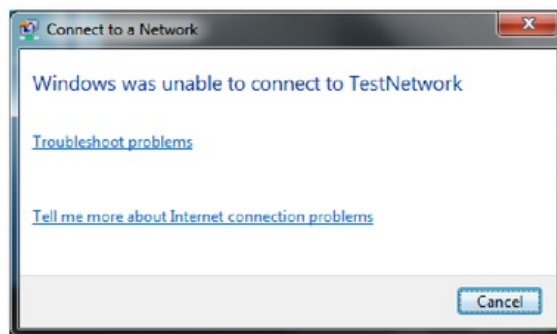
=====

Cannot send alert. Admin email not found!
Configure admin notification from Learning Mode

PHY      Interface      Driver      Chipset
phy1     wlan0mon        rtl8192cu   Realtek Semiconductor Corp. RTL8192CU 802.11n WLAN Adapter
(mac80211 station mode vif enabled on [phy1]wlan0)
(mac80211 monitor mode vif disabled for [phy1]wlan0mon)

root@kali:~#
```

7.



8.



Hmm, we can't reach this page.

Try this

- Make sure that you've got the right web address: <http://192.168.0.1>
- [Refresh the page](#)
- [Search for what you want](#)