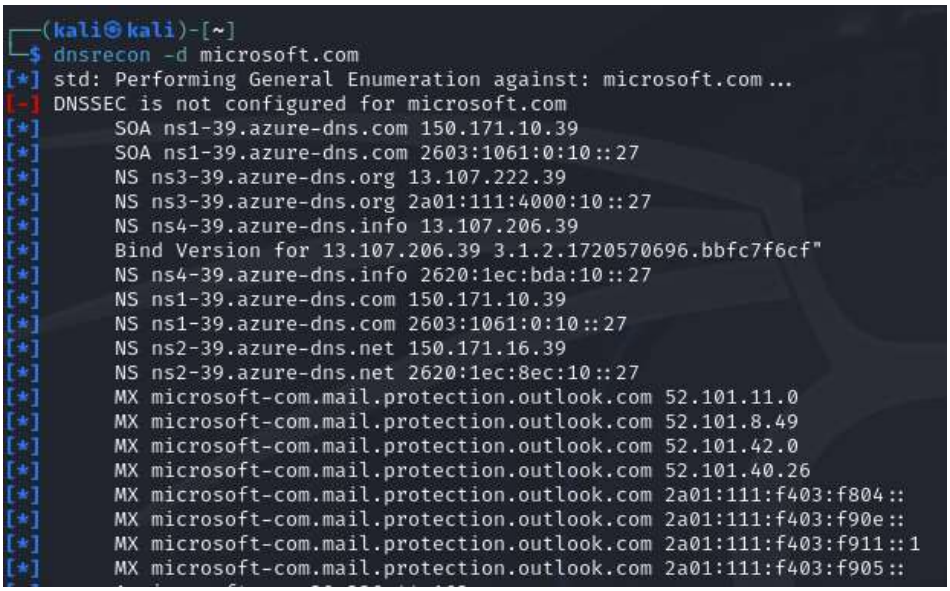
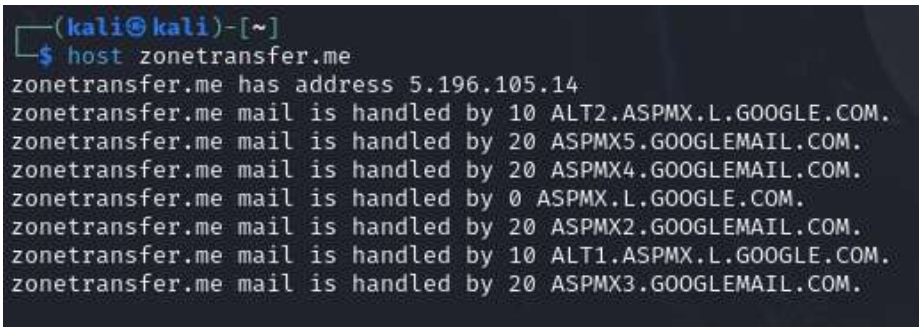
This report details the activities performed for active reconnaissance and enumeration using various tools in Kali Linux. The exercises focus on DNS enumeration to gather information about target domains, zone transfer requests to identify potential misconfigurations in DNS servers and Nmap scanning for network mapping and service detection. These techniques are essential for identifying vulnerabilities in network infrastructures and assessing the overall security posture of an organization.

So, if we wanted to get the DNS records of a website: microsoft.com, you could use the following command: dnsrecon -d microsoft.com




This is only a partial list of the output. The full output is much longer. Here you can see the SOA or Start of Authority DNS record. This is a type of DNS record that contains important information about a DNS zone. You can see at the top the SOA is ns1-39.azure-dns.com with an IP address of 150.171.10.39. It has a corresponding Name Server or NS further down. You can also see that it goes up to 4 Name Servers, the fourth being ns4-39.azure-dns.info with an IP address of 13.107.206.39.

For zone transfer requests, you could query zonetransfer.me



This output shows the DNS records for the domain zonetransfer.me. Using the host command, we were able to obtain the following types of DNS resource records: Address Record which maps the domain name zonetransfer.me to the IP address 5.196.105.14 and Mail Exchange Records which specify the mail servers responsible for receiving email on behalf of the domain and their priorities.

To enumerate the Name Servers, you could use the command “host -t ns zonetransfer.me”

```
┌──(kali㉿kali)-[~]
└─$ host -t ns zonetransfer.me
zonetransfer.me name server nsztm2.digi.ninja.
zonetransfer.me name server nsztm1.digi.ninja.
```

Here, you can see that two Name Servers were picked up: nsztm1.digi.ninja and nsztm2.digi.ninja

You can attempt to zone transfer using the command “host -l zonetransfer.me <name server>.” In this case, I will use nsztm1.digi.ninja

```
┌──(kali㉿kali)-[~]
└─$ host -l zonetransfer.me nsztm1.digi.ninja
Using domain server:
Name: nsztm1.digi.ninja
Address: 81.4.108.41#53
Aliases:

zonetransfer.me has address 5.196.105.14
zonetransfer.me name server nsztm1.digi.ninja.
zonetransfer.me name server nsztm2.digi.ninja.
14.105.196.5.IN-ADDR.ARPA.zonetransfer.me domain name pointer www.zonetransfer.me.
asfdbbox.zonetransfer.me has address 127.0.0.1
canberra-office.zonetransfer.me has address 202.14.81.230
dc-office.zonetransfer.me has address 143.228.181.132
deadbeef.zonetransfer.me has IPv6 address dead:beaf::
email.zonetransfer.me has address 74.125.206.26
home.zonetransfer.me has address 127.0.0.1
internal.zonetransfer.me name server intns1.zonetransfer.me.
internal.zonetransfer.me name server intns2.zonetransfer.me.
intns1.zonetransfer.me has address 81.4.108.41
intns2.zonetransfer.me has address 167.88.42.94
office.zonetransfer.me has address 4.23.39.254
ipv6actnow.org.zonetransfer.me has IPv6 address 2001:67c:2e8:11::c100:1332
owa.zonetransfer.me has address 207.46.197.32
alltcpportsopen.firewall.test.zonetransfer.me has address 127.0.0.1
vpn.zonetransfer.me has address 174.36.59.154
www.zonetransfer.me has address 5.196.105.14
```

Here we can see that there might be some sensitive information within this output. Some possibilities for this might be internal.zonetransfer.me (IPADDR 127.0.0.1). "internal" suggests an internal network, which shouldn't be publicly visible. Also vpn.zonetransfer.me (IPADDR 174.36.59.154). "VPN" could be a target for attackers trying to gain unauthorized access.

You can use the dnsenum command to gather detailed DNS information about a domain. It can retrieve the IP addresses associated with the domain, list the name servers, identify the mail servers responsible for handling email, attempt to perform zone transfers to get all DNS records from the domain's DNS servers, use Google scraping and brute-force techniques to find subdomains, and more.

```
┌──(kali㉿kali)-[~]
└─$ dnsenum zonetransfer.me
dnsenum VERSION:1.3.1

-----   zonetransfer.me   -----


Host's addresses:
__________________

zonetransfer.me.                         5856     IN    A        5.196.105.14


Name Servers:
______________

nsztm1.digi.ninja.                       9457     IN    A        81.4.108.41
nsztm2.digi.ninja.                       9456     IN    A        34.225.33.2


Mail (MX) Servers:
___________________

ALT1.ASPMX.L.GOOGLE.COM.                 293      IN    A        209.85.203.26
ASPMX3.GOOGLEMAIL.COM.                   293      IN    A        64.233.184.27
ALT2.ASPMX.L.GOOGLE.COM.                 293      IN    A        64.233.184.26
ASPMX5.GOOGLEMAIL.COM.                   293      IN    A        142.250.153.27
ASPMX4.GOOGLEMAIL.COM.                   293      IN    A        142.250.27.27
ASPMX.L.GOOGLE.COM.                      292      IN    A        142.251.163.26
ASPMX2.GOOGLEMAIL.COM.                   293      IN    A        209.85.203.26
```

This is a small portion of the output. The whole output is very long.

Nmap or Network Mapper is an open-source tool used for network discovery and security auditing. It is widely used to map out a network, discover hosts and services, and identify potential security vulnerabilities.

We can use our Metasploitable server to run some test Nmap commands.

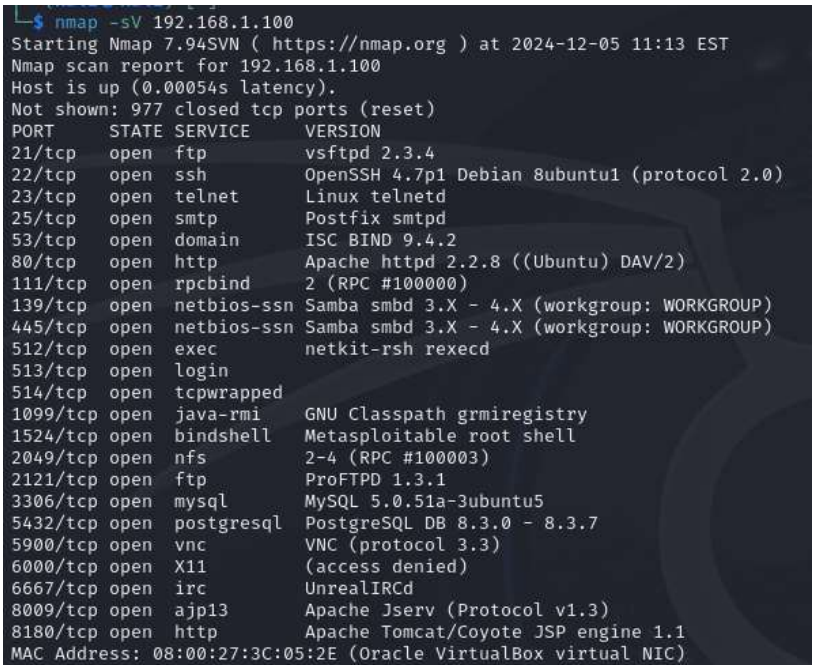
A TCP SYN scan is a technique used to identify open ports on a target system. Use the command nmap –sS <IP Address>.

```
┌──(kali㉿kali)-[~]
└─$ nmap -sS 192.168.1.100
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-12-05 11:05 EST
Nmap scan report for 192.168.1.100
Host is up (0.00065s latency).
Not shown: 977 closed tcp ports (reset)
PORT     STATE SERVICE
21/tcp   open  ftp
22/tcp   open  ssh
23/tcp   open  telnet
25/tcp   open  smtp
53/tcp   open  domain
80/tcp   open  http
111/tcp  open  rpcbind
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
512/tcp  open  exec
513/tcp  open  login
514/tcp  open  shell
1099/tcp open  rmiregistry
1524/tcp open  ingreslock
2049/tcp open  nfs
2121/tcp open  ccproxy-ftp
3306/tcp open  mysql
5432/tcp open  postgresql
5900/tcp open  vnc
6000/tcp open  X11
6667/tcp open  irc
8009/tcp open  ajp13
8180/tcp open  unknown
MAC Address: 08:00:27:3C:05:2E (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 0.30 seconds
```

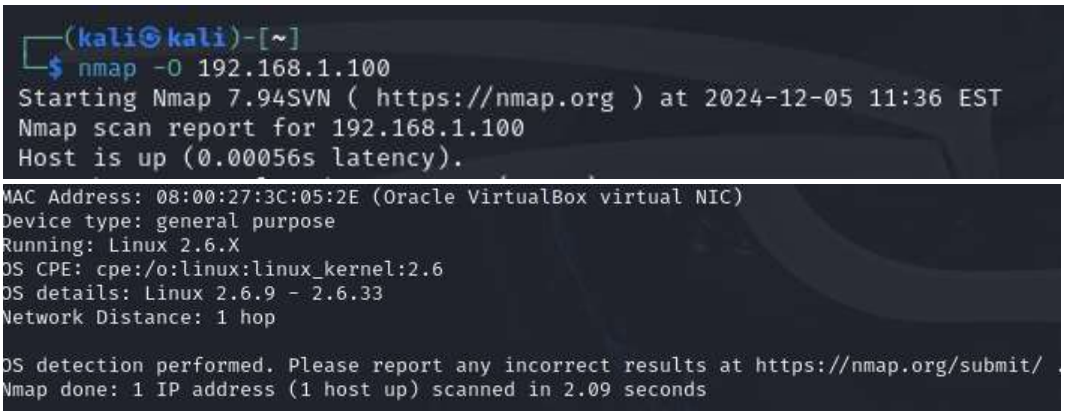
Here, we can see that 23 ports are open. Some ports here may jump out at you as needing to be closed. Some include port 23/tcp for telnet. Telnet is an unencrypted protocol and should be replaced by SSH. Also, port 1524/tcp for ingreslock. This is often used as a backdoor and should be closed unless specifically needed.

A version scan is used to detect the software and version information of services running on open ports. This can help identify vulnerabilities related to specific software versions. Use the command nmap –sV <IP Address>.



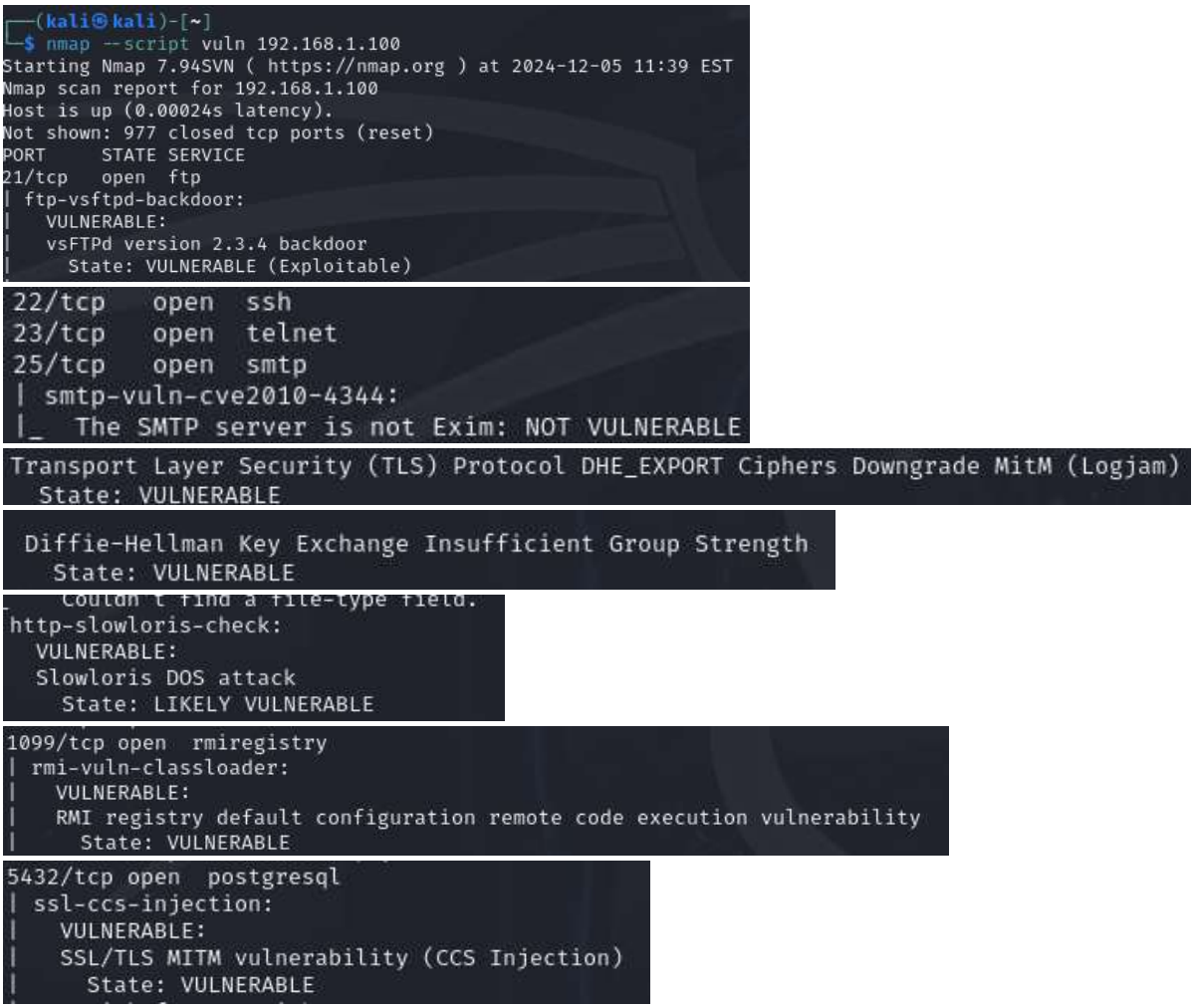

You can notice that this shows almost the same output as the previous command, but it also shows the version of each service in use.

An OS detection scan is used to identify the operating system running on a target host. Nmap does this by analyzing the responses from the target to various probes and comparing them to a database of known OS signatures. Use the command nmap –O <IP Address>.



This shows the target host is running Linux with a kernel version between 2.6.9 and 2.6.33. This also provides the network distance (number of hops) to the target, in this case, 1 hop.

The “--script vuln” option in Nmap utilizes the Nmap Scripting Engine or NSE to detect vulnerabilities on the target system by running scripts from the "vuln" category.




Here are some snippets from the output. As you can see, it scans the ports and shows their state, whether vulnerable or not. This command can provide valuable insights into the security posture of a network and help identify areas that need remediation.

Active reconnaissance and enumeration activities using tools like DNSRecon, DNSEnum, Nmap, and Host provided valuable insights into the target environment's security. Critical information about domain configurations, live hosts, and potential vulnerabilities were uncovered. Thorough reconnaissance in penetration testing revealed misconfigurations and security weaknesses that could be exploited. This project highlighted the importance of implementing robust security measures to protect sensitive information and maintain network integrity.