

CREDENTIAL DUMPING

p. EJD0fK
0B2ubAl0pOPnnmQ1sZH
n8lRRCgb7yJZOQ4eXJtPj1FKsy6
INvywLTK
FZSJWwfA90
qWV4J2ymxcN
hQabFIE4jXIOS3UIK
CMP2VBWIsI1Xgq4Akj1
nnGGERTaOWYMzF
mYxfyiV7HsBHeWTsl
uXZ5ETN5
Es
0B2ubAl0pOPnnmQ1sZH
n8lRRCgb7yJZOQ4eXJtPj1FKsy6
QZC8UamuCKxl9cRPwXfOkRhHh6rwoKa
EDxQbTTZidiQxpA3FLicgntG8ZfuHdnDWg8w
WqnA93TSskasKJtP3KTCOs9vYYIQEU4C8Gkmfq
AtglTKMdFtzLbgmz0OnlyQDSMHCwk1S33575Ppal2
YCHu34Lv0br6vpkeuR1GFtRyXojm4LZMY6Avr25u6yT
3ay4VkBbqlezNEVRiAqNzAAJEP9jPtvHOesb3nHI8EBY
8VqeGHjpVD5aGGISX47Y9k8NLMNg0nCXzyjTABt92ye
2k4W6bsXZCOK12XHNJotvuaSV2Ke7b8HM4d2MK92c
8RZKpqVzncMGxLWyoQKTCysRZ4DdKiti5452GLwdf17
oDWbYkfZHILAYuDfclotMjkOKgLPsLjwjwix9K8Jd15pO2
3lI8ci2vAvIKJIVJaLYgDRk6zdTRPbXki8X4Bx1TTBiyNg
wzXW0AOFT3bYseLPhu4A1w3V2ZsZyhLFe6Tji3jC60P
EJEgXKA2p1fqL8ysNgXBQewfDumaM6eE w4vj VA
ZZjkG HIDn Ksm
mPTI fG9C XhMK
MZvd smSEA6hN8yEftM bUuD
v UIIDbl3K smguVBWeE8 Caop8DsKkV0 ZqR9l6Vs G
L5uHv QQTpjspL7I26TJjJQAC U3txgjM7sZ6ESeQrCK UC4pb
Pas397W94CmOjhmZr1 5pls5cQSL6JjxbTffytQVbq4QABp08Z4f97Vu 2ubAl0pOPnnmQ1sZH
oP67kpw50t1PPqiUhGJmS NdZSjBG1xMaNiBh8xMbzoIMm3MaJ3c5rX RRCgb7yJZOQ4eXJtPj1FK
3oQzonJ5ad6aDk7i4qtSi0 4mLINvywLTK1sTQiK3K5fS uCKxl9cRPwXfOkRhHh6
LS2v5kqjpWNXZKNKI0 DFZSJWwfA90C2k6Lwz8o pA3FLicgntG8ZfuHdn
H73OPcyZMUP 3KTCOs9vYYIQ
dp6FjCbjbw QDSMHCw
RZZ7iY2 FtRyXojm
NQ JEP9

INTERNAL MONOLOGUE

Contenido

Introducción	3
Explotación	3
Explotación del imperio PowerShell	4
Descifrado de Hash.....	4
Conclusión	4

Introducción

Mientras se realizan operaciones del Equipo Rojo, es posible encontrarse con un escenario en el que el atacante no puede utilizar Mimikatz. Esto podría deberse a que casi todos los programas antivirus o de malware detectarán la presencia de Mimikatz tan pronto como llegue a la máquina de destino. Este es el escenario en el que un atacante puede realizar un ataque de monólogo interno. Para realizar este ataque, se requirió una herramienta desarrollada por Elad Shamir de Missing Link Security.

Al estar en contacto con el Mecanismo de seguridad de Windows, estará familiarizado con NetNTLM. Es un protocolo basado en respuesta a desafíos que se utiliza donde Windows no puede aplicar la autenticación basada en Kerberos. En este método, el servidor envía un desafío de 8 bytes con el hash NTLM como clave al usuario. El hash es un hash MD4 de la contraseña del usuario. Hay dos versiones de NetNTLM. Ambos son vulnerables. La versión 1 de NetNTLM se introdujo hace bastante tiempo y actualmente está deshabilitada de forma predeterminada.

En un sentido general, el ataque de degradación se realizó contra el propio Mimikatz. Después de la explotación de la máquina objetivo, el atacante, ya sea usando Mimikatz o manualmente, puede editar claves de registro como LMCompatibilityLevel con valores como 0,1,2 que pueden hacer que el dispositivo comprometido use la versión anterior o degradada de NTLM para interactuar con otros servidores SMB y puede llevar a cambiar a otros usuarios y servidores.

Sin embargo, en este ataque que se describe en la demostración, no se utiliza Mimikatz y, en cambio, el atacante invoca una llamada a un procedimiento local desde una aplicación en modo de usuario al paquete de autenticación NTLM a través del SSPI. Esto calcula la respuesta NetNTLM que analizamos anteriormente en el contexto del usuario que inició sesión. El ataque inhabilita inherentemente los controles preventivos NetNTLMv1, luego extrae todos los tokens de inicio de sesión que no son de red de los procesos actualmente en ejecución y se hace pasar por los usuarios asociados. Para cada usuario suplantado, NTLM SSP invoca localmente una respuesta NTLMv1 al desafío elegido y luego restaura los valores originales de las claves de registro analizadas anteriormente. Ahora el hash capturado se puede descifrar con la herramienta de tu preferencia, como John the Ripper o Hash Cat.

GitHub: [monólogo interno](#)

Explotación

Tiene la opción de compilar el ejecutable usted mismo obteniendo los binarios de GitHub. Sin embargo, para esta demostración, descargaremos el ejecutable.

[Descargar InternalMonologue.exe](#)

Después de descargar el ejecutable, supongamos que el atacante tiene el punto de apoyo inicial en la máquina objetivo. Es necesario transferir el ejecutable a la máquina de destino y ejecutarlo con ciertos parámetros. El parámetro Degradar debe tener el valor "verdadero" para degradar la versión. Luego, el parámetro Threads también debe contener el valor verdadero y, finalmente, para realizar la suplantación, el valor del parámetro Impersonate también debe ser verdadero. Al ejecutar con éxito el ejecutable, el atacante puede extraer con éxito el hash v1 degradado del usuario objetivo, como se demostró.

```
InternalMonologue.exe -Degradar verdadero -Subprocesos verdadero -Suplantar verdadero
```

```
C:\Users\raj\Downloads>InternalMonologue.exe -Downgrade true -Threads true -Impersonate true
raj: :DESKTOP-ATNONJ9:5018402148e15a8d77cb22dd46f1449a2791416b73ee9c3d:5018402148e15a8d77cb22dd46f1449a2791416b73ee9c3d:1122334455667788
```

www.hackingarticles.in

Explotación del imperio PowerShell

Si el atacante decide comprometer la máquina objetivo a través de PowerShell Empire y tiene un agente activo, entonces puede realizar un ataque de degradación directamente desde PowerShell Empire. Dentro de las credenciales, PowerShell Empire tiene un módulo llamado `invoke_internal_monologue` que esencialmente realiza el mismo ataque que el ejecutable discutido anteriormente. Este método no implica transferir un ejecutable y ejecutarlo en la máquina de destino, lo que lo hace mucho más sigiloso.

usar credenciales del módulo/`invoke_internal_monologue`
ejecutar

```
(Empire: 293VMKUL) > usemodule credentials/invoke_internal_monologue
(Empire: powershell/credentials/invoke_internal_monologue) > execute
[*] Module is not opsec safe, run? [y/N] y
[*] Tasked 293VMKUL to run TASK_CMD_WAIT
[*] Agent 293VMKUL tasked with TASK_CMD_WAIT
[*] Tasked agent 293VMKUL to run module powershell/credentials/invoke_internal_monologue
(Empire: powershell/credentials/invoke_internal_monologue) >
raj::.:5018402148e15a8d77cb22dd46f1449a2791416b73ee9c3d:5018402148e15a8d77cb22dd46f1449a2791416b73ee9c3d:1122334455667788
```

Descifrado de hash

En ambas variantes de ataques que se realizaron anteriormente, se descubrió que el hash para el usuario `raj` era el mismo y ahora hay dos formas en las que se puede usar este hash. En primer lugar, el atacante puede utilizar directamente el hash para iniciar sesión realizando un Pass the Hash. Pero si el atacante quiere, puede descifrar el hash utilizando a John el Destripador. Almacene el hash extraído del usuario `raj` en un archivo en el escritorio de nuestro Kali Linux y asígnele un nombre hash. Luego, utilizando John the Ripper para describir el formato como NetNTLM como se muestra a continuación, se puede observar que el hash se puede descifrar. Se descubrió que el hash era la contraseña "123".

`john --format=netntlm hash --show`

```
(root@kali)-[~/Desktop]
# john --format=netntlm hash --show
raj:123.:5018402148e15a8d77cb22dd46f1449a2791416b73ee9c3d:5018402148e15a8d77cb22dd46f1449a2791416b73ee9c3d:1122334455667788
1 password hash cracked, 0 left
```

Conclusión

A veces, ideas tan simples como degradar la versión del mecanismo de autenticación pueden resultar peligrosas. Como este ataque no requiere ninguna herramienta que esté dirigida a varios mecanismos defensivos, puede pasar desapercibido y obtener esas credenciales. Esto es una prueba de que la seguridad está en constante evolución y que la única forma de adelantarse a un atacante es pensar como tal.

ÚNETE A NUESTRO PROGRAMAS DE ENTRENAMIENTO

