uc3m | Universidad Carlos III de Madrid

Máster Universitario en Ciberseguridad

2018-2019

Trabajo Fin de Máster

"Ataques para evaluar Active Directory en Windows Server 2019."

Borja Lorenzo Fernádez

Tutor

Andrés Marín López

Agradecimientos

Resumen

Directorio Activo, del inglés *Active Directory (AD)*, es el servicio de directorio proporcionado por Microsoft y que tiene como finalidad principal la gestión de manera eficiente y centralizada la información y los recursos de una empresa. En la actualidad, Active Directory es utilizado por la mayoría de las organizaciones a nivel mundial y se considera una de las partes fundamentales para el correcto funcionamiento de una empresa. La información gestionada por Active Directory permite gestionar usuarios como pueden ser empleados, clientes, proveedores y que éstos puedan localizar los dispositivos, recursos y servicios distribuidos por la red como pueden ser ordenadores, servidores, impresoras, bases de datos, etc. Como se puede deducir, debido a la importancia que tiene Active Directory dentro de una organización y la información que gestiona le sitúan en uno de los principales objetivos para atacantes y ciberdelincuentes. Comprometer el servicio Active Directory supone un gran problema de seguridad para una empresa si el sistema principal de gestión se ve comprometido. Es por ello, que en los últimos años ha aumentado considerablemente el ataque a Active Directory cuya finalidad es hacerse con el control de la empresa y comprometer la seguridad de la información. Con el fin de contribuir al desarrollo, el trabajo realizado se centra en el análisis de las principales amenazas o ataques que pueden comprometer Active Directory gestionado por la última versión lanzada por Microsoft: Windows Server 2019. Esto se ha logrado mediante la creación de un laboratorio de pruebas, de manera local, que ha permitido la creación de una empresa ficticia y la revisión de una batería de los principales ataques analizados.

Palabras Clave:

Microsoft Active Directory, Domain Controller, Kerberos, Ciberseguridad, Windows Server, Pentesting, Red Team

Índice general

Índice de figuras

Índice de tablas

1. Introducción

En los últimos años, con el desarrollo intrínseco de la tecnología y el aumento masivo de información, empresas y organizaciones a nivel mundial se han visto en la necesidad de disponer de sistemas y/o servicios que les permitan administrar de una manera lógica, estructurada y eficaz tanto los usuarios como la información y recursos distribuidos en la red que disponen para el correcto funcionamiento del negocio. Microsoft Active Directory [1] se presenta como una solución efectiva a esta problemática. Para ello, Microfost ha implementado un servicio de directorio a través de una base de datos distribuida que permite a los usuarios localizar y administrar los recursos en red que dispone la organización. Estos recursos engloban bases de datos, sistemas de ficheros, aplicaciones web, servidores, impresoras, etc. Además, Active Directory sirve para gestionar la autenticación y autorización de dichos recursos, es decir, permite administrar qué usuarios pueden, o no, acceder a dichos recursos.

En la actualidad, Active Directory es la solución elegida por más del 90 % de las empresas y organizaciones a nievel mundial [2] para la gestión y administración de los recursos e información de una empresa. Esto supone que la amplia mayoría de atacantes elijan Active Directory como el objetivo, o *target*, principal en el ciclo de vida de un ataque dirigido a una organización. La finalidad principal es comprometer la infraestructura, obtener información confidencial o realizar ataques de Ransomware para estorsionar o sacar beneficio económico. Esto se puede observar en la manera de atacar de los principales grupos de ciberdelincuentes o grupos organizados como puede ser APT28, APT29, Cobal Strike, etc. o en los últimos ataques de Ransomware como WannaCry, NotPetya, MBR-ONI [3] [4] [5] que ponen los servicios de Active Directory en el punto de mira y centro de sus ataques. Además, la aparición de vulnerabilidades críticas y el desarrollo de herramientas cada vez más sofisticadas posibilitan considerablemente la explotación afectando a la seguridad de Active Directory.

Estas carácterísticas hacen que Active Directory sea un importante objeto de estudio para investigadores y equipos tanto de *Red Team* como *Blue Team*. El trabajo realizado ha abordado este problema y presenta como objetivo principal la revisión, análisis y prueba de alguna de las principales amenazas que ponen en grave riesgo la seguridad de Active Directory en la última versión de Windows Server.

1.1. Estado del Arte

Active Directory fue lanzado por primera vez en 1999 con el Sistema Operativo *Windows 2000 Server Edition*. Desde entonces, proteger, mantener actualizado y crear una infraestructura sólida y segura ha sido uno de los principales objetivos de Microsoft. Para ello, se ha instaurado una política de actualizaciones semestrales con plazo de servicio de 18 meses [6] que corrige las vulnerabilidades encontradas y propone nuevas implementaciones que mejoren tanto el uso como la seguridada.

Para el desarrollo de este proyecto se ha utilizado la última versión disponible: Windows Server 2019 1903 como Domain Controller para la gestión y administración de Active Directory. Windows Server 2019 está basado en la versión más estable y optimizada de Windows Server 2016 y se han añadido mejoras considerables que se pueden consultar en [7] y que destacan, en términos de seguridad, la implementación de un sofisticado antivirus para la protección de amenzas: *Windows Defender Advanced Threat Protection (ATP)*, un nuevo conjunto de funciones para la identificación y prevención de intrusiones: *Windows Defender ATP Exploit Guard* y novedades en la seguridad con *Software Defined Networking (SDN)* introducido en versiones anteriores.

Por otro lado, en cuanto a la experimentación, se ha utilizado una topología de Active Directory que permite evaluar satisfactoriamente los ataques analizados. Aunque existen multitud de ataques diferentes y variadades de los mismos, se ha considerado los siguientes ataques para delimitar el límite del proyecto realizado:

- **Pass-The-Hash**
- **NTLM Relay**
- **Overpass-The-Hash**
- **Pass-The-Ticket**
- **Golden/Silver Ticket**
- **Kerberoast**

La gran mayoría de estos ataques sobre Active Directory, no son específicos de este si no que aprovechan debilidades en los protocolos de autenticación utilizados. En la actualidad, los protocolos utilizados principalmente son: Microsoft NTLM y Keberos Version 5 Protocol. Por este motivo, ambos protocolos se van a detallar y analizar en los capítulos posteriores.

1.2. Objetivos

Como se ha comentado anteriomente, el objetivo principal de este trabajo es la revisión y análisis de las principales amenazas que pueden comprometer la seguridad de Active Directory. Para ello, es necesario la recreación de un laboratorio de pruebas que permita replicar dichos ataques.

Como se ha comentado anteriomente debido a la importacia del caso de estudio, este proyecto tiene como objetivo la adquisición de conocimiento sobre la infraestructura Active Directory como punto de partida tanto para otro tipo de investigaciones dejándolas así como trabajo futuro además de la aplicación de la topología con nuevos Domain Controllers, servidores, etc. Por otro lado, el conocimiento de los principales ataques y como está organizado es de gran importacia hoy en día, permitiendo así una correcta implmentación que minimice los riesgos a los que está sometida una organización.

Por último, este trabajo tiene como objetivo establecer las pautas y directrices para la creación de un laboratorio que permita tanto a *Pentesters* o profesionales de la seguridad ofensiva para realizar ejercicios simulados de *Red Team* en entornos controlados como a administradores de sistemas o equipos de *Blue Team* para probar nuevas configuraciones o realizar simulaciones de actualizaciones o mejoras en un entorno simulado.

1.3. Organización del Proyecto

El presente documento se divide en 6 capítulos, en los cuales en primera instancia se detallan los aspectos a teneren cuenta relacionados con Active Directoy, se detalla el laboratorio implementado, las pruebas que se van a realizar, la experimentación realizada así como los resultados obtenidos durante el trascurso:

2. Autenticación y autorización en Windows

Este capítulo aborda la metodología utilizada por los Sistemas Windows para desarrollar la autenticación y posterior autorización de un usuario u objeto. Con este fin, se profundizará en el inicio de sesión interactivo, independientemente si es de forma local o remoto, la gestión de las credenciales introducidas por el usuario hasta su posterior validación y finalmente, la comprobación de si ese usuario u objeto tiene permisos suficientes para ejecutar la acción que está solicitando.

2.1. Autenticación vs autorización

La mayoría de ataques y vulnerabilidades que amenazan Sistemas Windows y por consecuencia Active Directory reside en la forma de autenticación y autorización de estos. Por lo tanto, es importante conocer los procesos y procedimientos involucrados así como las diferencias entre autenticación y autorización [8]:

Por un lado, la autenticación consiste en la verificación de la identidad de un usuario, dicho con otras palabras, que el sistema de autenticación, como puede ser a la hora de iniciar sesión, asegure de que el usuario es quien dice ser. Esta verificación se puede realizar, por ejemplo, con un secreto o contraseña conocido únicamente por el usuario y que será validada posteriormente.

Por otro lado, cuando se habla de autorización se refiere a establecer y delimitar qué recursos son los que puede acceder el usuario en cuestión o grupos de usuarios. Por ejemplo, establecer que los usuarios administradores puedan acceder a carpetas compartidas con información confidencial. La verificación de que un usuario puede realizar la acción que está solicitando realizar, como puede ser el acceso a un dispositivo, recurso, etc.

2.2. Escenarios de autenticación

Para utilizar equipos basados en Windows es necesario disponer de una cuenta válida independientemente de si se solicita acceder a un equipo localmente o en red. Por lo tanto, Windows provee tecnología de control de acceso para determinar tanto si un usuario es quién dices ser, es decir, el proceso de autenticación como para gestionar si dicho usuario tiene los permisos necesarios para acceder al recurso o dispositivo que está solictando. A continuación se va a enumerar los posibles casos en los que se solicitará la autenticación de un usuario [9]:

2.2.1. Inicio de sesión interactivo (interactive logon)

Este escenario corresponde al inicio de sesión principal en sistemas basados en Windows por lo que se detallará en las secciones siguiente de este capítulo, ocurre cuando un usuario accede a una cuenta de usuario local o a una cuenta de dominio para iniciar sesión en un equipo.

Se produce un inicio de sesión de forma local cuando un usuario tiene acceso físico al equipo y este no está unido a ningún dominio o cuenta de usuario en Active Directory. Este inicio de sesión requiere disponer de una cuenta de usuario en el administrador de cuentas de seguridad del inglés *Security Account Manager (SAM)* donde se comproborá si las credenciales almacenadas son iguales a las credenciales proporcionadas por el usuario. Este inicio de sesión permite acceder al usuario a los recursos de Windows del equipo local.

Inicio de sesión en dominio ocurre cuando un usuario accede a una cuenta de usuario en Active Directry. Para ello, es necesario que el equipo disponga de una cuenta de dominio de Active Directory y está conectado físicamente a la red. Esto le permite tener acceso tanto a los recursos locales como los recursos proporcionados por el dominio (carpetas compartidas, servicios, etc.).

Además, también se produce un inicio de sesión interactivo cuando un usuario accede de manera remota a un equipo a través del protocolo de escritorio remoto del inglés *Remote Desktop Protocol (RDP)*. Las credenciales son enviadas al equipo donde se está intentado conectar yeste es el que procede a su posterior validación.

2.2.2. Inicio de sesión a través de aplicaciones o servicios

Este escenario ocurre cuando una aplicación o un servicio solicita que un usuario inicie sesión para acceder a los recursos que ofrece dicha aplicación o servicio. Como se puede observar en la figura 2.1 al ejecutar el comando RunAs que lanza la aplicación *cmd.exe* como el usuario *Cliente01*, este nos solicita la contraseña de dicho usuario. Además, Windows gestiona las credenciales para aplicaciones y servicios que no requieren la interacción de un usuario.

Los sistemas basados en Windows, implementan inicio de sesión único conocido como *Single Sign-On (SSO)* [10]. El objetivo principal de SSO es que sólo que haya que introducir las credenciales de un usuario una única vez, para acceder a cualquier recurso que necesite autenticación (en vez de introducir las credenciales cada vez). Como se verá a continuación, Windows guarda de manera local en memoria dichas credenciales en el subsistema *Local Security Authority (LSA)*.

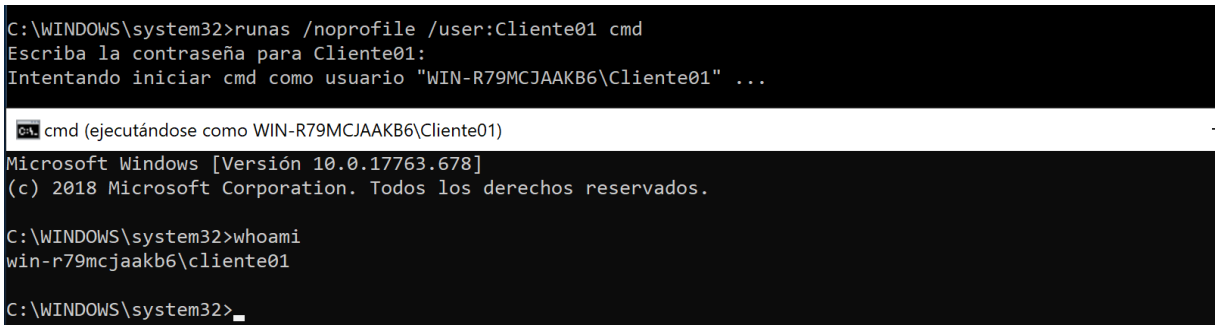

Fig. 2.1. Autenticación al ejecutar el comando RunAs.

2.2.3. Inicio de sesión en red

El inicio de sesión en red del inglés *Network Logon* ocurre una vez el usuario es correctamente autenticado en un equipo a través de alguno de los procesos explicados anteriormente e intenta acceder a cualquier servicio de red. Este proceso suele ser invisible al usuario a no ser que sea necesario otras credenciales.

2.2.4. Otros escenarios

Existen otros escenarios de inicio de sesión como puede ser “Inicio de sesión a través de Smartcard” que requiere el uso del protocolo Kerberos o “Inicio de Sesión biométrico” donde se utiliza un dispositivo para obtener las credenciales biométricas como puede ser la huella digital y se compara con las credenciales almacenadas durante la creación de la cuenta.

2.3. Inicio de sesión interactivo

El proceso de autenticación a través de inicio de sesión interactivo del inglés *Interactive Logon*, a diferencia del inicio de sesión en red o *Network Logon*, es llevado a cabo por el proceso *WinLogon* que se encarga de recoger las credenciales introducidas por el usuario y su posterior validación. Un usuario que inicia sesión en un equipo ya sea localmente o un inicio de sesión en red introduce el usuario y la contraseña (denominado credenciales de usuario) y sirve para verificar la identidad del usuario. Por otro lado, cuando se inicia sesión a través de una Smart Card *(Smart Card Logon)* las credenciales están almacenadas en el chip de la tarjeta y estas son leídas por un dispositivo externo y el usuario introduce el *Personal Identification Number (PIN)*.

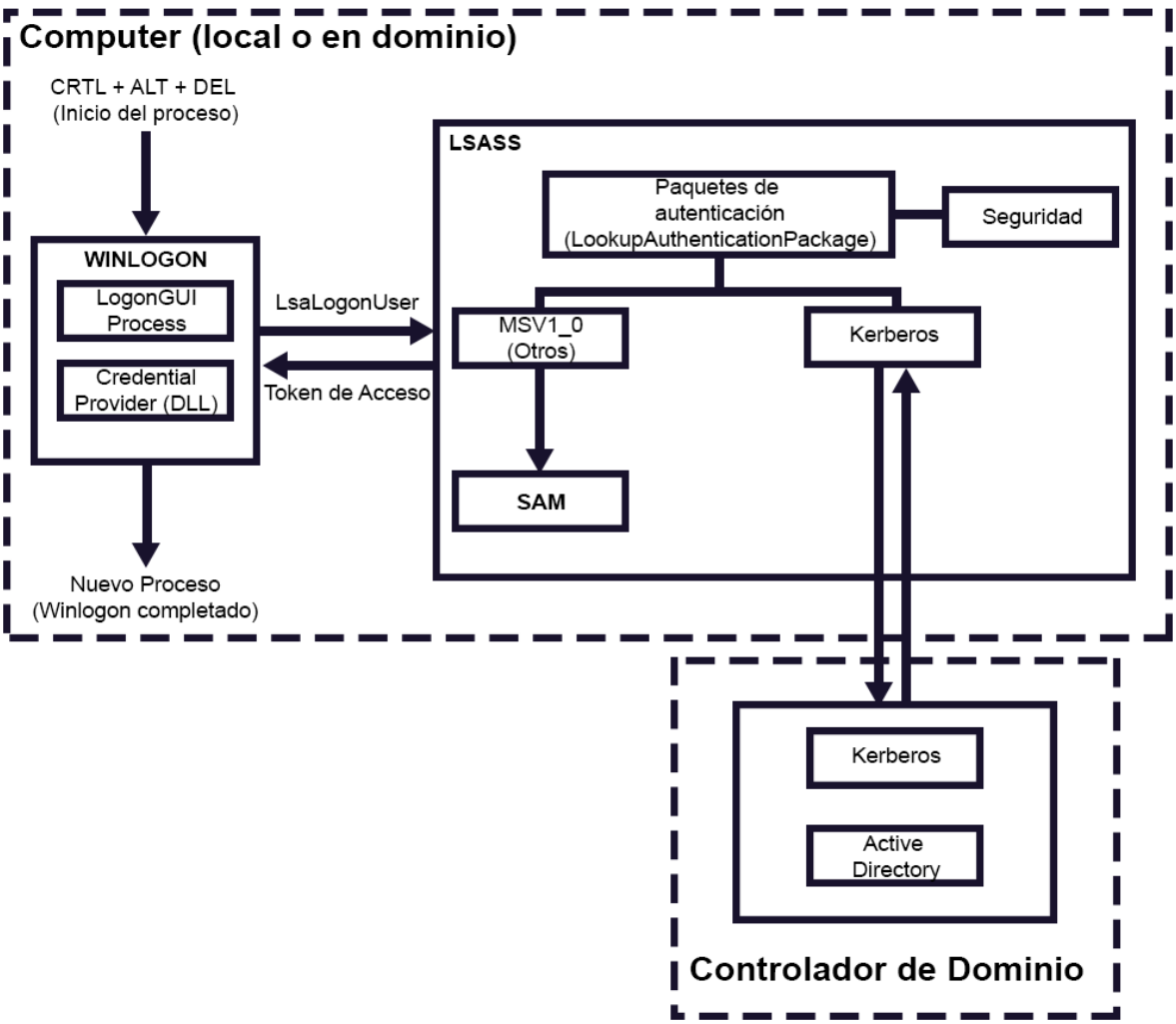

Fig. 2.2. Proceso de inicio de sesión interactivo (WinLogon).

2.3.1. Proceso de inicio de sesión interactivo (WinLogon)

WinLogon.exe es el proceso encargado de coordinar y administrar el inicio de sesión interactivo. Además, este proceso también se encarga de gestionar el *logout*, lanzar los procesos necesarios para la autenticación de un usuario, cambiar las contraseñas, bloquear y desbloquear un equipo y proporcionar la seguridad necesaria para que ningún otro proceso pueda acceder a información sensible cuando estos procedimientos se están llevando a cabo.

Como se puede ver en la Figura 2.2 el proceso de inicio interativo consta de varias fases [11]:

1. En primer lugar, el proceso de inicio de sesión comienza con una secuencia denominada *Secure Attention Sequence (SAS)*, esta secuencia es *CTRL + ALT + DEL* por defecto e inicia el proceso *WinLogon.exe*.

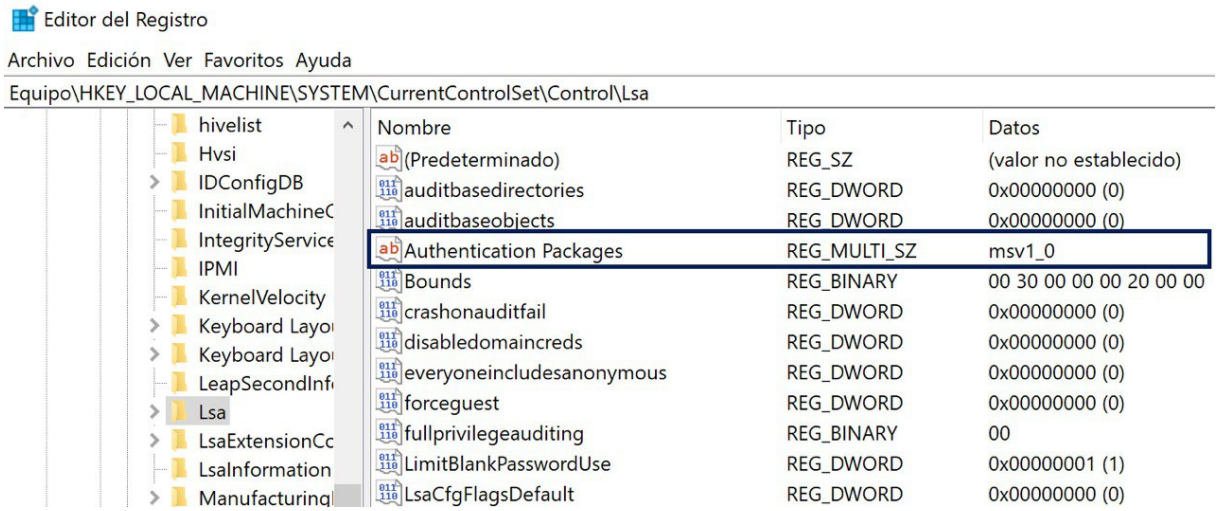

Fig. 2.3. Clave de registro sobre los paquetes de autenticación.

2. *WinLogon.exe* es el ejecutable encargado de gestionar el inicio de sesión interactivo. Para ello, inicializa el proceso *LogonUI.exe* cuya finalidad es proporcionar la interfaz de usuario por defecto y recoger las credenciales introducidas.

3. *LogonUI.exe* es el proceso encargado de solicitar, enumerar y mostrar al usuario la interfaz con las credenciales necesarias para la autenticación. Para ello, consulta los diferentes *Credential Providers* [12] que dispone el sistema. Los *Credential Providers* son las bibliotecas de enlace dinámico, del inglés *Dynamic-Link Library (DLL)* que se encargan de proporcionar la información necesaria, manejar la comunicación y la lógica con las entidades de autenticación externas y serializar y empaquetar las credenciales correctamente. Estas DLLs se encuentran en [1] o en [2] (Si se trata de un inicio de sesión con Smart Cards).

4. Una vez introducidas las credenciales, WinLogon.exe se comunica con el proceso LSASS a través de la función *LsaLookupAuthenticationPackage*. Esta función tiene como objetivo obtener los paquetes de autenticación disponibles en el sistema a través de la clave de registro [3] como se puede observar en la Figura 2.3.

5. Posteriormente, se envían las credendiales a través de la función *LsaLogonUser*. Si algún paquete de autenticación autentica al usuario el proceso continua, en cambio, si ningún paquete indica que se ha iniciado sesión correctamente el proceso acaba. Los paquetes de autenticación usados por Windows por defecto son MSV1_0 y Kerberos, ambos se detallarán en el siguiente capítulo de la literatura.

6. Cuando se trata de un inicio de sesión en Dominio, LSA utiliza el proceso *Netlogon.exe*, estre proceso se encarga de mantener un canal de comunicación seguro entre el

[1] %SystemRoot%\System32\authui.dll
[2] %SystemRoot%\System32\SmartcardCredentialProvider.dll
[3] HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Lsa

equipo y el controlador de dominio (*Domain Controller*) y pasar las credenciales a través de este.

7. Una vez autenticado, el proceso LSASS comprobará en la base de datos de políticas locales si el usuario autenticado tiene los permisos suficientes para realizar la acción que está solicitando. Si el inicio de sesión no coincide el proceso de autenticación acaba y LSASS elimina cualquier estructura de datos creada y lo notifica a WinLogon. Si el acceso está permitido, LSASS agrega los IDs de seguridad correspondientes, busca en la base de datos los permisos asociados a los usuarios del mismo grupo del SID del usuario, los añade al token de acceso (*Access Token*) y crea dicho token que será enviado a Winlogon con un mensaje de inicio de sesión correcto.

8. Por último, Winlogon mira en el registro [4] y crea un proceso con el valor que haya contenido en el registro. El valor por defecto es *Userinit.exe* que carga el perfil del usuario autenticado.

Una vez definido el proceso de inicio interactivo a grandes rasgos, se va a pasar a detallar los componentes mencionados que forman parte de dicho proceso.

2.3.2. Local Security Authority (LSA)

Local Security Authority [13] se encarga de validar el acceso a los objetos, comprobar si un usuario tiene permisos suficientes y generar mensajes de auditoría. Es decir, LSA se encarga de las siguientes acciones:

- Autenticar y registrar los usuarios en un sistema local, es decir, se encarga del proceso visto anteriormente.
- Administrar la politica de seguridad local de un sistema, del inglés *Local Security Policy*.
- Proporcionar los servicios necesarios tanto para la autenticación de un usuario, como para la generación de los tokens de acceso correspondientes.
- Gestionar los servicios necesarios para mantener la relación entre nombres y SIDs.

Local Security Authority Subsystem Service (LSASS)

El proceso *Local Security Authority Subsystem Service (LSASS)* se encarga de instanciar las politicas de seguridad en el sistema, realizar un seguimiento de las políticas de seguridad de las cuentas activas, modificar credenciales y crear tokens de acceso y almacenar las credenciales de los usuarios activos del sitema, esto permite que un usuario no tenga que

[4] HKLM\SOFTWARE\Microsoft\Windows NT\Current Version\Winlogon\Userinit

introducir las credenciales cada vez que accede a un recurso *(Single Sign-On)* [10].

Este proceso es de gran interés para los atacantes ya que LSASS puede almacenar credenciales como Tikets de Kerberos, Hashes NT, Hashes LM o credenciales con algoritmos de cifrado débiles que se puede obtener la contraseña en texto claro.

LSASS almacena las credenciales de las sesiones activas, estas credenciales se alamacenan cuando un usaurio realiza alguna de las siguientes acciones:

- Se inicia sesión localmente (o remotamente a través de *Remote Desktop Protocol (RDP)*).
- Se ejecuta un proceso o tarea usando el comando *RunAs*.
- Se ejecuta un servicio de Windows que necesite mayores privilegios de los actuales y se requiere autenticación.
- Se ejecuta una tarea programada para la que es necesario autenticarse.
- Se ejecuta una tarea local usando una herramienta de administración remota.

Logon Sessions

Logon Sessions [14] [15] es una estructura de datos que representa un *Security Principal* [16]. Una entidad de seguridad del inglés *Security Principal* corresponde a cualquier entidad que se puede autenticar en un sistema basado en Windows como puede ser usuarios, grupos de usuarios o procesos ejecutados en un contexto de seguridad de usuarios o grupos de usuarios.

Cuando un usuario inicia sesión de forma satisfactoria el proceso LSA crea una *Logon Session* que será utilizada para la creación del token de acceso y s eincrementará la referencia al número de sesiones creadas. Esta referencia también es aumentada cuando se duplica el token, cuando un usuario ejecuta procesos en nombre de otro usuario, etc.

Para listar las *Logon Sessions* en un sistema se puede utilizar el comando *logonsessions* de *Windows Sysinternals* [17] como se puede ver en la Figura 2.4.

2.3.3. Security Support Provider Interface (SSPI)

Aunque no se ha mencinado anteriormente, *Security Support Provider Interface (SSPI)* [18] es una API que permite que una aplicación pueda utilizar varios modelos de seguridad, es decir, abstrae las llamadas necesarias en el proceso de autenticación y permite que

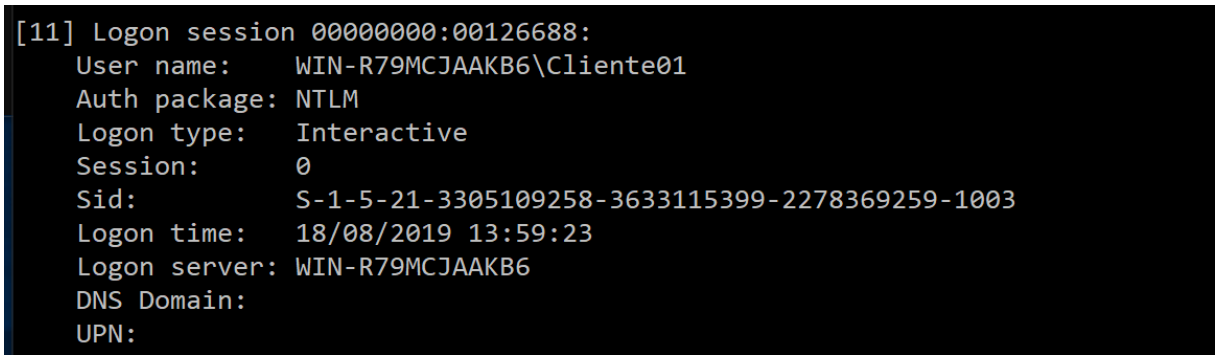

Fig. 2.4. Salida del comando Logonsessions.

una aplicación lleve a cabo un proceso de autenticación sin especificar los protocolos de autenticación, denominados paquetes de autenticación que se verán en detalle en la siguiente sección de este capítulo.

En primer lugar, se negocia el protocolo a utilizar, para que el proceso se complete correctamente ambas máquinas deben aceptar mismo *Security Support Provider (SSP)*. Un SSP es un DLL que implementa SSPI y permite la ejecució de los paquetes de autenticación. Cada paquete proporciona un "mapeo.[en]tre las llamadas de funciones SSPI de una aplicación y las funciones de un modelo de seguridad.

Los principales SPP son:

- Kerberos.

```
%SystemRoot\%\Windows\System32\kerberos.dll
```

- NT Lan Manager (NTLM): NTLMv1 y NTLMv2.

```
%SystemRoot%\Windows\System32\msv1_0dll
```

- Digest.

```
%SystemRoot%\Windows\System32\Wdigest.dll
```

- Schanell.

```
%SystemRoot%\Windows\System32\Schannel.dll
```

- Negotiate.

```
%SystemRoot%\Windows\System32\lsasrv.dll
```

Negotiate

Microsoft Negotiate [19] es un SSP que actua como intermediario entre la API SPPI y otro SSP. Cuando una aplicación requiera algún tipo de autenticación, se envía una petición a Negotiate con los armunetos necesarios (parámetros, credenciales, SSPs a utilizar...), este lo examinará y pasará la petición al SSP correspondiente que llevará a cabo la autenticación.

Actualmente, Neogtiate elige entre Kerberos y NTLM. Seleccionará el primero simpre y cuando haya conexión entre las dos partes implicadas en el proceso y el usuario haya especificado el Service Principal Name (SPN), un User Principal Name (UPN), o una cuenta de NetBIOS. En cambio, si se trata de una autenticación local utilizará NTLM.

2.3.4. Security Account Manager (SAM)

Security Account Manager (SAM) corresponde a una base de datos que almacena localmente la información sobre las cuentas de usuario y grupos de usuarios (identificador de usuario, nombre de usuario y hash de la contraseña). Esta información es consultada por el proceso LSA a la hora de autenticar a un usuario de forma local comparando el hash de la contraseña introducida por el usuario y el hash de la contraseña contenido en base de datos SAM. Esta base de datos corresponde con el fichero:

```
%SystemRoot%\Windows\System32\config\SAM
```

2.4. Autorización

Una vez completado correctamente el proceso de autenticación, se procede a comprobar si el usuario tiene los privilegios necesarios para realizar la acción que se está solicitando ejecutar, como puede ser acceder a un equipo, accedere a un recurso, etc.

2.4.1. Access tokens

Cuando el proceso LSA verifica la autenticación del usuario, se crea un *Access Token*, este objeto describe el contexto de seguridad de un proceso o de un hilo (*thread*) [20]. Para SSPI se denomina contexto de seguridad a una estructura de datos que contiene datos relevantes de seguridad como puede ser la clave de sesión o la duración de dicha sesión. Cada proceso ejecutado por un usuario dispone de una copia del token de acceso de ese usuario. Windows utiliza estos tokens para identificar a un usuario cuando ejecuta un hilo que necesite privilegios de ese usuario. La información contenida en un *Access Token* es:

- El SID de la cuenta del usuario en cuestión.
- El SID del grupo de usuarios de los que el usuario es miembro.
- El SID de la session (logon session).
- Los privilegios del usuario y del grupo de usuarios al que pertenece.
- El SID del grupo primario.
- El *Discretionary Access Control List* (DACL) [21] por defecto que se utiliza cuando el usuario crea un proceso sin especificar el descriptor de seguridad.
- La procedencia del token de acceso.
- Si el token de acceso es primario o es una suplantación.
- Una lista de SIDs restrictivos.
- Niveles de suplatación actuales.
- Otras estadísticas.

Cuando un administrador local inicia sesión en una máquina, se crean dos tokens diferentes: Un token primario que contendrá el contexto de seguridad del usuario y un token de administrador. Esto es debido a la política de Windows de mínimo privilegio posible, esto significa que el sistema usará por defecto el token primario cuando un proceso o hilo interactue con un *Securable Objects* [22]. Esto es importante a la hora de entender *User Account Control (UAC)*.

2.4.2. User Account Control (UAC)

User Account Control (UAC) [23] [24]sirve para controlar cuando se están usando privilegios de adminstración. Como se ha comentado anteriormente al iniciar sesión desde una cuenta administrativa, se crean dos tokens de usuario: uno denominado *full access token* y un token secundario denominado *filtered access token*. Esto permite que los procesos ejecutados por este usuario se ejecuten con el segundo token siempre y cuando no necesiten privilegios de administración y así cumplir la política de mínimo privilegio posible. Esto lo podemos observar cuando se completa un inicio de sesión válido, el proceso *Explorer.exe* se ejecuta con el *filtered access token*. Cuando un proceso concreto necesita privilegios de administración, se requerirá una elvación de privilegios realizando una elevación de UAC. Como se puede ver en la Figura 2.5 se ha ejecutado una terminal *(cmd.exe)* con privilegios administrativos y aparece el mensaje de Control de Cuentas de Usuario para elevar de privilegios.

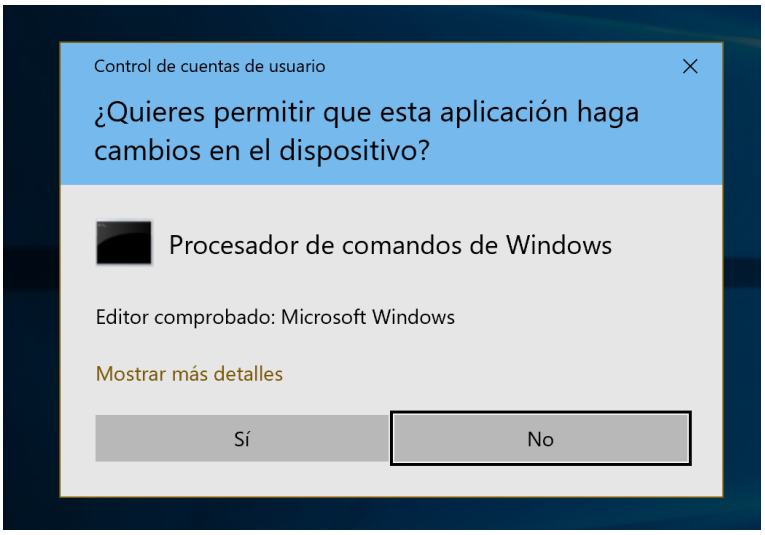

Fig. 2.5. Control de Cuentas de Usuario al ejecutar cmd.exe

Para controlar qué procesos necesitan privilegios especiales o cuales no, los Sistemas Windows hacen uso de *Mandatory Integrity Control* [25], un mecanismo para controlar el acceso a los *Securable Objects* [22], para ello, el MIC utiliza niveles de integridad para evaluar el acceso a un proceso. Estos niveles son: *untrusted, low, medium, high, system e installer* [26] [27]. Esto implica que un objeto con integridad baja (low) no puede escribir en un objeto de integridad medio.

- **Untrusted:** Son aquellos procesos iniciados de forma anónima, como puede ser procesos lanzados desde cuentas de Invitados.
- **Low:** Nivel de integridad usado por defecto para la interacción con internet. Cuando se lanza *Internet Explorer* se utiliza este modo, por lo tanto, todos los archivos y procesos asociados a este se le asignan un nivel de integridad bajo.
- **Medium:** Nivel de integridad usado por la mayoría de los procesos, es el nivel designado por defecto siempre y cuando no se especifique explícitamente un nivel inferior o superior.
- **High:** Nivel de integridad destinado para aquellos procesos que necesitan privilegios de administración. Los objetos con este nivel de integridad solicitarán una elevaciñon de privilegios a través de UAC.
- **System:** Nivel de integridad reservado para los objetos del sistema, estos objetos engloban el kernel de Windows y servicios del *core*.
- **Installer:** Nivel de integridad especial utilizado para la instalación de software. Este nivel es igual o superior a todos los niveles anteiores, lo que permite que este nivel puede desinstalar los demás objetos.

3. Paquetes de autenticación

Esta sección detalla en profundidad los paquetes de autenticación utilizados en sistemas basados en Windows. Los paquetes de autenticación son *Dynamic-Link Libraries (DLLs)* lanzadas por el proceso LSA durante un inicio de sesión que se encargan de analizar y validar las credenciales introducidas por el usuario, crear una nueva *logon session* y pasar la información al proceso LSA para que este cree el *Access Token* correspondiente si la validación ha sido correcta. Windows permite la carga de multiples paquetes de autenticación lo que permite que LSA siporte multiples procesos de inicio de sesión diferentes. En este capítulo se van a detallar los paquetes de autenticación utilizados por defecto: MSV1_0 y Kerberos. Además, se va a detallar la forma que tiene Windows de almacenar las contraseñas en el sistema.

3.1. Windows hashes

Los Sistemas Windows, en lugar de almacenar las contraseñas en texto plano, algo que sería un gran problema de seguridad utilizan los siguientes algoritmos de hash [28]:

3.1.1. Hashes Lan Manager (LM)

El algoritmo Lan Manager (LM) para realizar la función hash de las contraseñas almacenadas en Windows fue una de las primeras implementaciones que desarrolló Windows para mantener cifradas las contraseñas. Hoy en día está práctiacamente en desuso y desde 2017 se recomienda desactivar la opción de que se guarden las credenciales de esta forma [29].

Algoritmo

El algoritmo de hash utilizado realiza el siguiente procedimiento:

- Convertir todos los carácteres a letras mayúsculas.
- Añadir un padding de carácteres nulos hasta que tenga una longitud de 14 carácteres.
- Dividir la contraseña en dos partes de 7 carácteres cada una.
- Crear dos DES keys para cada parte.
- Cifrar a través de DES las partes anteriores con el string "KGS!@#$ %".
- Concatenar ambos strings.

3.1.2. Hashes NT

Los hashes NT, también conocidos como hashes NTLM, es la forma que utiliza atualmente Windows para alamcenar las contraseñas de los usuarios del sistema. Estos hashes están almacenados en la SAM si se trata de un equipo local o en el fichero NTDS del Active Directory si se trata de un equipo en dominio. A través de la obtención de este tipo de hashes se puede realizar un ataque de Pass-The-Hash (se detallara en los siguientes capítulos).

Algoritmo

Windows encodea la contraseña del usuario con UTF-16 Little Endian y posteiormente realiza un hash con el algoritmo MD4:

- MD4(UTF-16-LE(password))

3.2. MSV1_0

MSV1_0 [30] es el paquete de autenticación proporcionado por Windows e implementa la familia de protoclos Lan Manager versión 1 y 2 (LM y NT) y Net Lan Manager versión 1 y 2 (NTLMv1 y NTLM v2) [31].

Este paquete de autenticación soporta tanto inicio de sesión de forma local como inicio de sesión para cuentas y servicios en dominios. El paquete MSV1_0 ejecuta una aquitectura cliente/servidor, es decir, el cliente es el que recibe las credenciales (username y el hash de la contraseña) y las valida frente al servidor.

Cuando se ejecuta localmente cliente y servidor están representados por la misma máquina que se encarga de recoger las credenciales proporcionadas por el usuario a través de los *Credential Providers* y compararlas con las credenciales introducidas por el usuario cuando creó la cuenta y que están almacenadas en la SAM, si ambas contraseñas son iguales el proceso de autenticación es correcto.

En inicio de sesión en dominio el cliente representa la máquina local y el servidor representa el Domain Controller donde está configurado Active Directory. El cliente recoge las credenciales y las pasa por el canal de comunicación seguro creado por el proceso *Winlogon.exe* y las comunica con la instancia de MSV1_0 ejecutada en el Domain Controller. El cliente delega la comprobación de las credenciales al Domain Controller, esto se denomina *Pass-Through*. La instancia de MSV1_0 del Domain Controller realiza la validación de de las credenciales comprobando la información recibida con los datos almacenados el la base de datos del Domain Controller y devuelve la información a la instancia ejecutada en local. Si la validación ha sido correcta, el paquete MSV1_0 local devuelve la información

al proceso LSA local.

Windows ha implementado los protocolos de desafío/respuesta NTLMv1 y NTLMv2 para la intercambiar las credenciales introducidas por el usuario entre la máquina local y el Domain Controller en lugar de intermbiar las credenciales directamente. A continuación se van a detallar ambos protocolos.

3.2.1. Net NT Lan Manager Versión 1 (Net-NTLMv1)

3.2.2. Net NT Lan Manager Versión 2 (Net-NTLMv2)

3.3. Kerberos

Bibliografía

[1] Microsoft, "Introducción a active directory." https://support.microsoft.com/es-es/help/196464, Octubre 2000.

[2] C. Truran, "Active directory: The crown jewels for insider attacks." https://www.scmagazineuk.com/active-directory-crown-jewels-insider-attacks/article/1473390, Febrero 2018.

[3] M. Bresman, "Wannacry, notpetya, mbr-oni and friends: Tales of wiper attacks and active directory destruction." https://www.semperis.com/blog/wannacry-notpetya-wiper-attacks-active-directory, Abril 2018.

[4] M. J. Schwartz, "Hydro hit by lockergoga ransomware via active directory." https://www.bankinfosecurity.com/hydro-hit-by-lockergoga-ransomware-via-active-directory-a-12207, Marzo 2019.

[5] C. Cimpanu, "Norsk hydro ransomware incident losses reach $40 million after one week." https://www.zdnet.com/article/norsk-hydro-ransomware-incident-losses-reach-40-million-after-one-week, Marzo 2019.

[6] Microsoft, "Windows server release information." https://docs.microsoft.com/en-us/windows-server/get-started/windows-server-release-info, Mayo 2019.

[7] Microsoft, "What's new in windows server 2019." https://docs.microsoft.com/es-es/windows-server/get-started-19/whats-new-19, Abril 2019.

[8] Microsoft, "Windows authentication concepts." https://docs.microsoft.com/es-es/windows-server/security/windows-authentication/windows-authentication-concepts, Octubre 2016.

[9] Microsoft, "Windows logon scenarios." https://docs.microsoft.com/es-es/windows-server/security/windows-authentication/windows-logon-scenarios, Octubre 2016.

[10] Microsoft, "Single sign-on in windows 2000 networks." https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-2000-server/bb742456(v=technet.10), Septiembre 2009.

[11] B. Catlin, P. Yosifovich, J. Hanrahan, M. Russinovich, A. Ionescu, and D. Solomon, *Windows Internals: User Mode*. Windows internals ; Part 1, Microsoft Press, 2017.

[12] Microsoft, "Credential providers in windows 10." https://docs.microsoft.com/en-us/windows/win32/secauthn/credential-providers-in-windows, Mayo 2018.

[13] Microsoft, "Lsa authentication," Mayo 2018. https://docs.microsoft.com/en-us/windows/win32/secauthn/lsa-authentication.

[14] Microsoft, "Lsa logon sessions." https://docs.microsoft.com/en-us/windows/win32/secauthn/lsa-logon-sessions, Mayo 2018.

[15] K. Brown, *The .NET Developer's Guide to Windows Security (Microsoft Net Development Series)*. Addison-Wesley Professional, 2004.

[16] Microsoft, "Entidades de seguridad." https://docs.microsoft.com/es-es/windows/security/identity-protection/access-control/security-principals, Abril 2017.

[17] Microsoft, "Windows sysinternals." https://docs.microsoft.com/es-es/sysinternals, Septiembre 2017.

[18] Microsoft, "Sspi." https://docs.microsoft.com/en-us/windows/win32/secauthn/sspi, Mayo 2018.

[19] Microsoft, "Microsoft negotiate." https://docs.microsoft.com/en-us/windows/win32/secauthn/microsoft-negotiate, Mayo 2018.

[20] Microsoft, "Access tokens," 2018. https://docs.microsoft.com/es-es/windows/win32/secauthz/access-tokens.

[21] Microsoft, "Access control lists," 2018. https://docs.microsoft.com/es-es/windows/win32/secauthz/access-control-lists.

[22] Microsoft, "Securable objects," 2018. https://docs.microsoft.com/es-es/windows/win32/secauthz/securable-objects.

[23] Microsoft, "User account control," 2018. https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2008-R2-and-2008/cc731416(v=ws.10).

[24] Microsoft, "Cómo funciona el control de cuentas de usuario," 2018. https://docs.microsoft.com/es-es/windows/security/identity-protection/user-account-control/how-user-account-control-works.

[25] Microsoft, "Mandatory integrity control," 2018. https://docs.microsoft.com/en-us/windows/win32/secauthz/mandatory-integrity-control.

[26] T. Bradley, "Introduction to windows integrity control," 2007. https://www.symantec.com/connect/articles/introduction-windows-integrity-control.

[27] Microsoft, "Modify an object label," 2017. https://docs.microsoft.com/en-us/windows/security/threat-protection/security-policy-settings/modify-an-object-label.

[28] P. Gombos, "Lm, ntlm, net-ntlmv2, oh my!." https://medium.com/@petergombos/lm-ntlm-net-ntlmv2-oh-my-a9b235c58ed4, Febrero 2018.

[29] Microsoft, "Seguridad de red: no almacenar valor de hash de lan manager en el próximo cambio de contraseña." https://docs.microsoft.com/es-es/windows/security/threat-protection/security-policy-settings/network-security-do-not-store-lan-manager-hash-value-on-next-password-change, Abril 2017.

[30] Microsoft, "Msv1_0 authentication package." https://docs.microsoft.com/en-us/windows/win32/secauthn/msv1-0-authentication-package, Mayo 2018.

[31] Microsoft, "Ntlm overview." https://docs.microsoft.com/es-es/windows-server/security/kerberos/ntlm-overview, Octubre 2016.