

<http://pegasus.javeriana.edu.co/~PA1710-1-APTTool>

PROTOTIPO DE HERRAMIENTA PARA DETECTAR CAMBIOS DE ENTORNO
EN SISTEMAS OPERATIVOS SEGÚN COMPORTAMIENTOS IDENTIFICADOS
DE AMENAZAS PERSISTENTES AVANZADAS (APT)

Autor:

Oscar Andrés Montenegro Bocanegra

PONTIFICIA UNIVERSIDAD JAVERIANA
FACULTAD DE INGENIERIA
MAESTRÍA EN INGENIERÍA DE SISTEMAS Y COMPUTACIÓN
BOGOTÁ, D.C.
2017

<http://pegasus.javeriana.edu.co/~PA1710-1-APTTool>

PROTOTIPO DE HERRAMIENTA PARA DETECTAR CAMBIOS
DE ENTORNO EN SISTEMAS OPERATIVOS SEGÚN
COMPORTAMIENTOS IDENTIFICADOS DE AMENAZAS
PERSISTENTES AVANZADAS (APT)

Autor:

Oscar Andrés Montenegro Bocanegra

MEMORIA DEL TRABAJO DE GRADO REALIZADO PARA CUMPLIR UNO
DE LOS REQUISITOS PARA OPTAR AL TÍTULO DE
MAGÍSTER EN INGENIERÍA DE SISTEMAS Y COMPUTACIÓN

Director

Ing. Rafael Vicente Páez Méndez PhD.

Comité de Evaluación del Trabajo de Grado

Ing. María Isabel Serrano Gómez MSc.

Ing. Ricardo Herrera Hernández MSc.

Página web del Trabajo de Grado

<http://pegasus.javeriana.edu.co/~PA1710-1-APTTool>

PONTIFICIA UNIVERSIDAD JAVERIANA
FACULTAD DE INGENIERIA
MAESTRÍA EN INGENIERIA DE SISTEMAS Y COMPUTACIÓN
BOGOTÁ, D.C.
Mayo, 2017

**PONTIFICIA UNIVERSIDAD JAVERIANA
FACULTAD DE INGENIERIA
MAESTRÍA EN INGENIERIA DE SISTEMAS Y COMPUTACIÓN**

Rector Magnífico

Jorge Humberto Peláez, S.J.

Decano Facultad de Ingeniería

Ingeniero Jorge Luis Sánchez Téllez

Director Maestría en Ingeniería de Sistemas y Computación

Ingeniera Ángela Cristina Carrillo Ramos

Director Departamento de Ingeniería de Sistemas

Ingeniero Efraín Ortíz Pabón

Artículo 23 de la Resolución No. 1 de Junio de 1946

“La Universidad no se hace responsable de los conceptos emitidos por sus alumnos en sus proyectos de grado. Sólo velará porque no se publique nada contrario al dogma y la moral católica y porque no contengan ataques o polémicas puramente personales. Antes bien, que se vean en ellos el anhelo de buscar la verdad y la Justicia”

AGRADECIMIENTOS

Agradezco a mis padres y mi hermana por su amor, paciencia y por ser la voz de apoyo en los momentos oportunos.

También, agradezco a mi Director Ing. Rafael Vicente Páez Méndez PhD por su interés, apoyo y dedicación durante la realización con el Trabajo de Grado.

A mis amigos por su agradable compañía, creer en mí y ser otro motivo para seguir adelante

Contenido

| | |
|--|-----------|
| INTRODUCCIÓN | 1 |
| I – DESCRIPCIÓN GENERAL DEL TRABAJO DE GRADO..... | 3 |
| OPORTUNIDAD, PROBLEMÁTICA Y ANTECEDENTES..... | 3 |
| 1.1 <i>Oportunidad o problemática</i> | 3 |
| 1.2 <i>Impacto deseado</i> | 3 |
| DESCRIPCIÓN DEL PROYECTO | 4 |
| 1.3 <i>Visión global</i> | 4 |
| 1.4 <i>Objetivo general</i> | 4 |
| 1.5 <i>Objetivos específicos</i> | 4 |
| 1.6 <i>Metodología propuesta</i> | 4 |
| II – MARCO TEÓRICO / ESTADO DEL ARTE | 9 |
| 1. AMENAZAS PERSISTENTES AVANZADAS (APT) | 9 |
| 1.1 <i>Definición</i> | 9 |
| 1.2 <i>Fundamentos</i> | 10 |
| 1.3 <i>Características de las APT</i> | 14 |
| HERRAMIENTAS DE SEGURIDAD | 17 |
| 1.4 <i>Sistemas de Detección de Intrusos (IDS)</i> | 17 |
| 1.5 <i>Sistemas de Prevención de Intrusos (IPS)</i> | 18 |
| 1.6 <i>Firewalls</i> | 19 |
| 1.7 <i>Antivirus</i> | 19 |
| 1.8 <i>Honeypots</i> | 20 |
| TRABAJOS RELACIONADOS EN EL ÁREA | 20 |
| III – DESARROLLO DEL TRABAJO | 23 |
| 1. FASE DE ENTENDIMIENTO DEL CONTEXTO DE LA PROBLEMÁTICA..... | 23 |
| 1.1 <i>Actividades de documentación</i> | 23 |
| 1.2 <i>Simulación de ataques investigados</i> | 34 |
| FASE DE ENTENDIMIENTO DE LOS DATOS | 35 |
| FASE DE MODELADO DE LOS DATOS | 36 |
| 1.2 <i>Reducción de dimensionalidad</i> | 36 |
| 1.3 <i>Algoritmos aplicables</i> | 39 |
| FASE DE EVALUACIÓN..... | 43 |
| FASE DE DISEÑO Y DESARROLLO DEL PROTOTIPO..... | 44 |

| | | |
|-----|--|-----------|
| 1.4 | <i>Diseño del prototipo</i> | 44 |
| 1.5 | <i>Fase de implementación</i> | 47 |
| 1.6 | <i>Riesgos</i> | 48 |
| 1.7 | <i>Pasos para determinar si un dato que refleja un cambio de comportamiento es señal de un ataque o no</i> | 49 |
| | PRUEBAS DEL PROTOTIPO..... | 50 |
| | IV – RESULTADOS OBTENIDOS | 52 |
| 1. | PROTOTIPO DESARROLLADO | 52 |
| 1.1 | <i>Prototipo modo cliente:</i> | 52 |
| 1.2 | <i>Herramienta en modo servidor</i> | 54 |
| | CONSIDERACIONES ADICIONALES | 55 |
| | PAQUETES INCLUIDOS: | 56 |
| | RESULTADOS DE LAS PRUEBAS..... | 56 |
| | V – CONCLUSIONES Y TRABAJOS FUTUROS | 60 |
| 1. | CONCLUSIONES | 60 |
| 2. | TRABAJOS FUTUROS..... | 60 |
| | VI – REFERENCIAS Y BIBLIOGRAFÍA | 61 |
| | VII - ANEXOS | 66 |
| | GLOSARIO | 66 |
| | ANEXO 2: CARTA DE AUTORIZACIÓN DE LOS AUTORES..... | 68 |
| | ANEXO 3: DESCRIPCIÓN TRABAJO DE GRADO..... | 71 |

LISTA DE TABLAS

| | |
|---|----|
| Tabla 1. Evolución archivos componentes Duqu [40] | 31 |
| Tabla 2. Vectores de los ataques APT investigados y que fueron utilizados en los laboratorios para obtener datos para el modelamiento | 35 |
| Tabla 3. Comparación tiempos de lectura bases de datos obtenidas | 43 |
| Tabla 4. Comparación potencial de instancias correctamente clasificadas por cada algoritmo | 44 |
| Tabla 5. Comparación de errores redes neuronales y SVM..... | 44 |
| Tabla 6. Tabla estimación riesgos iniciales para la implementación del prototipo | 49 |

LISTA DE ILUSTRACIONES

| | |
|---|----|
| Ilustración 1. Composición tareas bajo metodología CRISP-DM. Adaptado de Chapman [11]..... | 5 |
| Ilustración 2. Fases de la metodología CRISP-DM [11]..... | 7 |
| Ilustración 3. Resumen conceptos Amenazas Persistentes Avanzadas (APT) | 16 |
| Ilustración 4. Modelo de monitoreo basado en red de genes propuesto por Wang et Al [6]. | 21 |
| Ilustración 5. Funcionamiento de framework de detección usando honeypots y NIDS [30] | 22 |
| Ilustración 6. Nombre de la ruta del payload del cual salió el nombre de Operación Aurora [33] | 24 |
| Ilustración 7. Correo utilizado para engañar a la víctima [11]..... | 24 |
| Ilustración 8. Infraestructura utilizada para recolección de datos | 34 |
| Ilustración 9. Estructura base de datos inicial | 36 |
| Ilustración 10. Interpretación columna cadena de datos BD inicial | 36 |
| Ilustración 11. Proceso de reducción bases de datos | 39 |
| Ilustración 12. Trazo de hiperplano para delimitar datos entre dos categorías [50].. | 42 |
| Ilustración 13. Algoritmos utilizados tras obtención, tratamiento de datos y evaluación de algoritmos para pronósticos..... | 44 |
| Ilustración 14. Topología diseñada como sistema para implementar el prototipo a construir | 46 |
| Ilustración 15. Diagrama de componentes prototipo BWCare..... | 47 |
| Ilustración 16. Mapa de calor utilizado para calificar riesgos del prototipo de la aplicación | 49 |
| Ilustración 17. Pasos que se ejecutan en nuevo caso para determinar si es señal de ataque o no | 50 |
| Ilustración 18. Ubicación archivo ejecutable herramienta BWCare | 52 |
| Ilustración 19. Conexión herramienta BWCare cliente con servidor..... | 52 |

| | |
|---|----|
| Ilustración 20. Pestaña principal prototipo de la herramienta versión cliente | 53 |
| Ilustración 21. Pestaña "Configuración" prototipo de la aplicación versión cliente ... | 54 |
| Ilustración 22. Ventana mostrando novedad en carpeta monitoreada | 54 |
| Ilustración 23. Ubicación archivo ejecutable herramienta BWCare Server..... | 55 |
| Ilustración 24. Pantalla historial en versión servidor | 55 |
| Ilustración 25. Notificación creación de archivo | 56 |
| Ilustración 26. Notificación eliminación de archivo | 56 |
| Ilustración 27. Exploit que posee la herramienta Metasploit respecto al Ataque Aurora | 57 |
| Ilustración 28. Exploit elegido para atacar la vulnerabilidad ms10_002 | 57 |
| Ilustración 29. Configuración del atacante para ejecutar exploit de Aurora | 58 |
| Ilustración 30. La víctima ingresa a página maliciosa creada por exploit de Operación Aurora | 58 |
| Ilustración 31. Desde la máquina del atacante, se envía el backdoor para ingresar a la máquina de la víctima..... | 58 |
| Ilustración 32. Sesión de la víctima abierta desde el servidor..... | 59 |
| Ilustración 33. Notificación de novedad en máquina de la víctima | 59 |

ABSTRACT

The objective of this job consists on develop the prototype of a tool for detecting the possibility of occurrence of an informatic attack in a system conformed by a Local Area Network (LAN), knowing the behavior changes caused by an Advanced Persistent Threat attack into a computer. This prototype doesn't pursue the objective of replacing the Information Security tools that exists until today, but complementing them. These detections will be achieved with the help of the Factorial analysis method followed by the Support Vectorial Machines (SVM) algorithm and the methodology used was CRISP-DM because it conceives in an integral way: From understanding the context of the problematic to getting a tool that mitigates the impact. Also, this methodology was useful because the amount of data recollected was too big for comprehend the problem and modelling it.

RESUMEN

El objetivo de este Trabajo de Grado consiste en desarrollar el prototipo de una herramienta para detectar la posibilidad de ocurrencia de un ataque informático en un sistema conformado por una red LAN, teniendo en cuenta los cambios de comportamiento causados por ataques de tipo Amenaza Persistente Avanzada (APT) sobre los computadores de dicha red. Esta herramienta no tiene la pretensión de reemplazar a los productos de Seguridad Informática que hasta el momento existen, sino de complementarlos. Estas detecciones se consiguen con ayuda del método de análisis factorial y el algoritmo de Máquinas de Soporte Vectorial (SVM). La metodología utilizada para ejecutar el proyecto fue CRISP-DM ya que concibe la problemática de forma integral: Desde el entendimiento del contexto del problema hasta conseguir una herramienta que pueda mitigar el impacto correspondiente. También, esta metodología fue útil porque se recolectó una cantidad de datos demasiado grande para comprender el problema y modelarlo.

RESUMEN EJECUTIVO

Con el paso del tiempo, los ataques informáticos se han convertido en actos más organizados, mejor enfocados y cada vez más peligrosos. Un ejemplo de esto son las Amenazas Persistentes Avanzadas (*Advanced Persistent Threat* - APT) que son ataques más sigilosos que se toman su tiempo para analizar la víctima que desean atacar e identificar cuáles son las vulnerabilidades más importantes que se pueden encontrar para después efectuar el ataque. Este tipo de ofensivas tienen como propósito robar información o perturbar la operación de la víctima y pueden estar patrocinados por el Gobierno [1].

Uno de los temas más complicados con las Amenazas Persistentes Avanzadas radica en que se utilizan múltiples vectores de ataque y en cualquier orden, haciendo que sea muy difícil de rastrear a tal punto que ninguna herramienta de Seguridad Informática lo pueda detectar, de manera que contrarrestarlo sea una tarea aún más complicada.

Varias empresas de Seguridad Informática e investigadores ya han propuesto soluciones, pero todavía no alcanzan una efectividad suficiente para mitigar estos ataques. Entre ellos, se han propuesto *frameworks* o herramientas apoyadas en elementos especiales de seguridad como *honeypots*.

En este trabajo se pretende atacar la problemática desde un punto de vista diferente: detectar cambios de comportamiento (también conocidos como variaciones de entorno) en un sistema, conformado por una red LAN teniendo como referencia la información de los elementos (archivos, carpetas, registros, puertos) que fueron afectados por los ataques APT de mayor magnitud hasta el momento. Para esto, se realizó un prototipo de una herramienta informática de nombre "*BWCare*" que detectará los cambios mencionados anteriormente con ayuda de uno o más algoritmos. Adicionalmente, el prototipo debe cumplir con otras labores adicionales que también son síntoma de un ataque tipo APT como lo es el inflado de archivos (Que se conoce como una técnica donde un archivo crece de manera desmesurada en muy poco tiempo).

Vale la pena indicar que los cambios de comportamiento o variaciones de entorno del sistema operativo que se desean capturar son la creación, eliminación, modificación sobre archivos, carpetas y registros e incluso el bloqueo de un puerto

El prototipo será un complemento para las demás herramientas de seguridad en lugar de reemplazarlas. Asimismo, el construir este prototipo tampoco implica que vaya a solucionar el problema de las APTs ya que se trata de una herramienta detectiva, mas no correctiva.

Los principales beneficiados son las áreas de IT de las compañías, especialmente Administradores de Sistema y Administradores de red, quienes van a tener más información para poder fortalecer su infraestructura a partir de todas las novedades que la herramienta vaya registrando.

Este proyecto fue efectuado bajo la metodología CRISP-DM (*Cross Industry Standard Process for Data Mining*) que comprende todas las fases necesarias para encontrar una solución adaptada al problema que se quiere resolver partiendo desde el contexto y la información existente. El proceso seguido fue el siguiente:

- Entendimiento del problema: Análisis de todo lo que hay en torno a un ataque tipo APT, es decir, significado, características, modo de operación, quienes ejecutan este tipo de ataques, etc.
- Entendimiento de los datos: En esta fase se obtuvieron la mayor cantidad de datos posibles para posteriormente generar un modelo. Esto consistió en dos tareas: La primera fue la documentación los ataques tipo APT de mayor impacto, del cual se recopilaron todas las entradas afectadas, es decir, archivos, carpetas, registros, puertos, direcciones IP, drivers y servicios. Los ataques investigados fueron Stuxnet, Duqu y Operación Aurora. La segunda tarea fue la elaboración de simulaciones de los vectores utilizados por estos ataques del cual también se obtuvieron otros datos importantes como tipos de archivos utilizados y carpetas preferidas para infectar.
- Generación de un modelo de datos: Esta fase consistió en la construcción de un modelo de datos que fue el insumo para comprobar cuál fue el mejor algoritmo que ayuda a determinar si un dato nuevo puede ser señal de ataque informático o no. Para esto, y ante la cantidad de datos recolectados, se requirió hacer una labor de reducción de dimensiones que consistió en dos tareas: Agrupar los datos obtenidos de la fase anterior en familias y la aplicación de la técnica de análisis factorial.
- Evaluación del modelo: A partir del modelo de datos que recolecta los cambios de comportamientos encontrados, se aplicaron varios algoritmos para comprobar cuál de ellos generaba los mejores resultados. El algoritmo seleccionado se incluyó en el desarrollo del prototipo con el fin de determinar si un cambio de comportamiento detectado sobre un dato del sistema operativo en un computador puede ser señal de un ataque informático.
- Diseño y desarrollo del prototipo: El prototipo presenta una arquitectura cliente-servidor y para apoyarse en la implementación del algoritmo se contó con el paquete Weka. El lenguaje elegido para desarrollar el prototipo es Java por las prestaciones gráficas para mostrar la información, la facilidad para conectarse con Weka y su extensibilidad.
- Pruebas: Se realizaron tres pruebas:
 - Básica: Se realizaron modificaciones, creaciones, eliminaciones y procedimiento de inflado a archivos y carpetas identificadas como críticas después del análisis de los ataques tipo APT.
 - Identificación de comportamientos producidos por ataques tipo APT investigados.

- Prueba en red: El servidor logró captar exitosamente los cambios tomados por las máquinas cliente.

Las aplicaciones basadas en cambios de entorno ofrecen una alternativa para complementar las herramientas de seguridad en su misión de monitorear posibles ataques, especialmente aquellos tan difíciles de detectar como los APT. Sin embargo, todavía no es suficiente para tener una protección apropiada ante este tipo de ataques: ayudar al personal de las compañías e incluso particulares para tomar conciencia de los peligros que hay en la red, no confiar en mensajes sospechosos y siempre comunicar al personal idóneo podrían contribuir a que este tipo de amenazas no tengan un impacto tan fuerte.

INTRODUCCIÓN

En los últimos años, la frecuencia y los daños causados por los ataques informáticos han ido en aumento y han tocado varias industrias: Algunas de ellas son las finanzas, salud, gobierno, militares, entretenimiento, entre otras, y éstas han perdido dinero por perjuicios sociales, económicos e incluso políticos. Según datos de los Laboratorios de McAfee, se estima que el costo anual que la economía global tiene que asumir por concepto de cibercrimen es de US\$400 billones [2] [3].

La tendencia de las amenazas informáticas indica que no dan muestra de decrecer, todo lo contrario, siguen presentándose y muestran una evolución en sus metodologías, objetivos y herramientas a un ritmo considerable [4]: Esto se da especialmente cuando se habla de las amenazas informáticas avanzadas donde los atacantes son cada vez más estructurados y los ataques que arman están mejor planeados.

El reflejo de esta evolución de ataques informáticos son las Amenazas Persistentes Avanzadas (APT) que son ataques realizados por personas o equipos técnicamente muy hábiles y que utilizan en conjunto varios vectores de ataque (por ejemplo, técnicas de evasión, disfrazar ataques y el engaño o Ingeniería Social) para alcanzar sus objetivos que generalmente son: establecer su posicionamiento dentro de la infraestructura de tecnología para filtrar información hacia el exterior en cualquier momento o entorpecer la operación de un sistema informático [5].

Ante este panorama, la actitud de las organizaciones ante este tipo de eventos ha mostrado alguna mejoría ya que las áreas gerenciales han puesto mayor interés sobre estos incidentes y han invertido en el acompañamiento para tratarlos (según /SACA, en 2014 la carencia de acompañamiento era del 75% y para 2015 se redujo a 66% [4]), pero el tema sigue siendo preocupante porque las áreas de IT aún no distinguen correctamente entre una APT y una amenaza convencional: Los medios utilizados para detectar estos dos tipos de amenazas son los mismos (Antivirus, Firewall, IDS, etc.), pero un ataque tipo APT tiene la característica de estar especialmente diseñado para evadir estos productos con el fin de conseguir sus objetivos [4] [6] [7].

En este trabajo se pretende atacar la problemática desde un punto de vista diferente: detectar cambios de comportamiento en un sistema conformado por un grupo de computadores en una red LAN. Dichos cambios están basados en los elementos que fueron afectados por los ataques tipo APT de mayor impacto. Para realizar estas detecciones, se toma el algoritmo de Máquinas de Soporte Vectorial (SVM) con ayuda de un tratamiento de datos previo realizado con Análisis Factorial, además de tener un elemento adicional para capturar archivos que han crecido de manera desmesurada en poco tiempo (también conocido como archivos inflados).

Los principales beneficiados son las áreas de IT, especialmente Administradores de Sistema o Administradores de red, que obteniendo una relación de los cambios que terminaron reconociéndose como anómalos, pueden tener una visión más clara para

tomar decisiones con miras de fortalecer el perímetro de seguridad de la infraestructura tecnológica que tienen a cargo.

El orden en que se presenta este trabajo es el siguiente: En primer lugar, se realiza una descripción general del proyecto donde se enuncia el objetivo general, los objetivos específicos, el impacto y la metodología propuesta. En el capítulo II, se hace una reseña teórica de los ataques tipo APT, herramientas de seguridad habitualmente utilizadas en una infraestructura tecnológica y unos trabajos relacionados con el tema. En el capítulo III se presenta el desarrollo del trabajo con la metodología relacionada y en el capítulo IV se presentan los resultados obtenidos, es decir, la herramienta construida y las pruebas realizadas. Finalmente, se mencionarán las conclusiones y trabajos futuros.

I – DESCRIPCIÓN GENERAL DEL TRABAJO DE GRADO

En este capítulo, se presenta el contexto general que está enmarcado este proyecto. Basado en los antecedentes e hitos que se han dado en este tipo de ataques, se pretende hacer una formulación del problema con su respectiva justificación y dar una perspectiva del impacto que podría dar este trabajo.

Oportunidad, problemática y antecedentes

1.1 Oportunidad o problemática

En los últimos años, se han presentado casos en donde la estructura de los ataques informáticos ha cambiado de una manera notoria: Ataques como *Stuxnet*, *Duqu*, *Oc-tubre Rojo* o *Flame* son algunos ejemplos que han logrado llamar la atención por su magnitud, la cantidad e importancia de sus víctimas y los daños causados, cuyas consecuencias no sólo pueden llegar a ser económicas, también pueden llegar a comprometer la sostenibilidad de organizaciones e incluso, afectar la integridad de personas [8]. Un tipo de estos nuevos ataques se les conoce como Amenazas Persistentes Avanzadas (*Advanced Persistent Threats* o APT).

La estructura con que se arma un ataque tipo APT es tan compleja que además de estar diseñada para evadir los productos de Seguridad Informática existentes, aún no se ha conseguido un elemento efectivo para tratar estos casos [8]: ya se están dando a conocer *frameworks*, aplicativos con módulos especiales, pero todavía cuentan con muchas falencias [9] y en algunos casos, necesitan complementos por los cuales hay que pagar un monto económico adicional para tener una funcionalidad apropiada [10].

Ante esta situación, se propone el prototipo de una herramienta que basado en cambios de comportamientos identificados en los ataques tipo APT que más impacto han tenido y con ayuda de uno o más algoritmos, identifique la posibilidad de que un computador dentro de una red LAN sufra un ataque informático. El desarrollo de este prototipo surge como una oportunidad de complementar a las herramientas de Seguridad Informática sin desplazarlas, pero la idea no es desarrollar una herramienta que detecte APTs o sea mejor que las ya existentes porque es una herramienta detectiva, no correctiva.

1.2 Impacto deseado

Con el desarrollo del prototipo mencionado, se pretende que un Administrador de Sistema, Administrador de red, o encargados del área de IT de una compañía, puedan obtener un historial de los cambios que terminaron reconociéndose como anómalos (que son indicios de potenciales ataques informáticos) y con esto puedan tener una visión más clara de las medidas para fortalecer el perímetro de seguridad de la infraestructura tecnológica que tienen a cargo.

Descripción del proyecto

1.3 Visión global

En este proyecto se propone el prototipo de una herramienta que trabaja de forma complementaria a un producto de seguridad (IDS/IPS, antivirus, etc.) detectando un posible ataque informático a partir de la variación en el entorno que opera un Sistema Operativo. Dicha detección se calcula con ayuda de uno o varios algoritmos conociendo los comportamientos que se han identificado en ataques tipo Amenaza Persistente Avanzada (APT).

Hay que aclarar que cuando se refiere al término sistema, se está hablando de un conjunto de máquinas que están trabajando dentro de una red LAN.

1.4 Objetivo general

Desarrollar el prototipo de una herramienta para detectar cambios de entorno en un sistema operativo, tomando como base los comportamientos identificados en ataques tipo Amenaza Persistente Avanzada (APT).

1.5 Objetivos específicos

1. Realizar un análisis para definir los cambios de comportamiento más frecuentes en los sistemas ocasionados por los ataques de tipo Amenaza Persistente Avanzada (APT).
2. Seleccionar un algoritmo que pueda facilitar la detección de cambios en el comportamiento del sistema (entendiendo el sistema como un conjunto de máquinas que están trabajando dentro de una red que presta diferentes tipos de servicio: servicios web, bases de datos, etc.).
3. Realizar el diseño de la herramienta utilizando como base el algoritmo de detección seleccionado y los cambios de comportamientos identificados en los sistemas que recibieron un ataque de tipo APT.
4. Implementar la herramienta según el diseño obtenido.
5. Validar la herramienta con un caso de estudio.

1.6 Metodología propuesta

Para la consecución del objetivo general y de los objetivos específicos, se efectúa la metodología CRISP-DM (*Cross Industry Standard Process for Data Mining*) que es una guía de proyectos descritos en términos jerárquicos en los cuales por cada fase se detallan un grupo de tareas genéricas que, a su vez, tendrán por dentro unas tareas especializadas y que, dentro de ellas, se ejecutarán unos procesos específicos. Esta metodología es especialmente utilizada para proyectos relacionados con Minería de Datos [11] [12].

CRISP-DM fue propuesta en 1999 por un grupo de empresas conformadas, entre otras, por Teradata, SPSS, Daimler-Chrysler y OHRA como el fin de tener guía metodológica para todo proyecto que tenga en cuenta tareas de Minería de Datos [12].

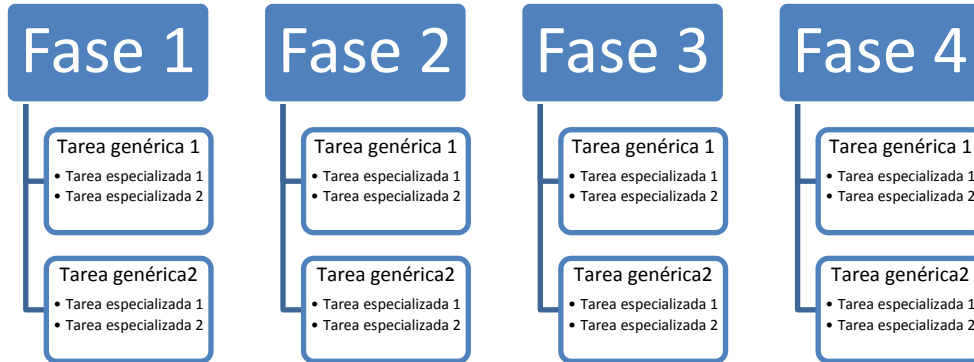


Ilustración 1. Composición tareas bajo metodología CRISP-DM. Adaptado de Chapman [11]

Todo proyecto estructurado bajo esta metodología cuenta con las siguientes fases:

- Entendimiento del negocio: Consiste en entender los objetivos y requerimientos del proyecto basados en la actividad que ejecuta el negocio y traducir el problema a una definición de minería de datos [11].

Aquí, los puntos claves son identificar los procesos críticos en el área que presenta el problema a resolver, saber qué información existe al inicio del proceso, qué información se obtiene durante su desarrollo y cómo se puede enriquecer la ejecución haciendo uso de la información que se va recopilando [11] [12].

Para identificar posibilidades de enriquecimiento del proceso a mejorar, se pueden utilizar mecanismos como análisis costo/beneficio (donde se identifican los costos de ejecución del proyecto, los costos de mantenimiento y los beneficios que se pueden percibir) o matrices de prioridades (donde se comparan los proyectos a partir de una o varias características como importancia, dificultad, etc., a las cuales se pone un valor de una escala terminada) [13].

Después de identificar las propuestas para enriquecer procesos del negocio en los que tienen problemas, se identifica qué es lo que espera el cliente o usuario. Para esto es importante saber cuáles son los entregables y los objetivos a cumplir [13].

- Entendimiento de los datos: Es el proceso donde se recolectan los datos, se describen y se evalúa su calidad con el fin de tener datos completos o agregados para hacer una tarea de minería más completa [11].

- Preparación de los datos: Consiste en construir el conjunto de variables a partir de la selección y limpieza de datos. En la selección, es importante tener claro el motivo de incluir o no algún dato. Después de tener los datos seleccionados y limpios, se inicia la construcción de una vista minable que describa el problema [11] [13].
- Modelado: En esta fase, se procede a seleccionar una o más técnicas para modelar los datos. Además de lo anterior, se genera un conjunto para hacer entrenamiento y pruebas de los datos obtenidos en la fase de entendimiento con el fin de calibrar el modelo y sus parámetros. En este punto, es posible combinar técnicas: Por ejemplo, si se va a hacer un modelo descriptivo, primero se podría aplicar la técnica de *clustering* y después reglas de asociación; pero si se va a trabajar con un modelo predictivo, primero podría aplicarse alguna técnica de clasificación y sobre uno de los casos aplico otra técnica o algoritmo [11] [13].

Cuando se trabaja con casos predictivos, se deben tener en cuenta 3 conjuntos de datos [12]:

- Set para entrenar el modelo: Casos resueltos. Por ejemplo, de 100 casos existentes, se pueden usar 80 para calibrar el modelo obtenido.
 - Set para probar el modelo: Casos resueltos sin respuestas. De los 100 casos existentes, se pueden tomar los 20 restantes para probar el modelo. Con esto se comprueba la precisión del modelo incluyendo casos de falsos positivos y falsos negativos.
 - Set para evaluar el modelo: No tiene casos resueltos. Si la predicción obtenida ocurre o no.
- Evaluación: Se evalúa si el modelo resultante cumple con los objetivos del negocio. Estos resultados pueden darse en forma de matriz de confusión en donde se obtienen los falsos positivos y falsos negativos. Para considerar el modelo bueno, la diagonal que corresponde a verdaderos positivos y falsos negativos deberían tener valores altos (alrededor del 50%, exactamente 50% no se va a dar) [11].
 - Despliegue (*deploy*): Implementación del modelo y procesos futuros, es decir, cómo se va a desplegar el modelo para el negocio y cómo se va a mantener (cada cuanto voy a actualizar el modelo, si mensualmente, anualmente, cada 4 años, etc.) [11]

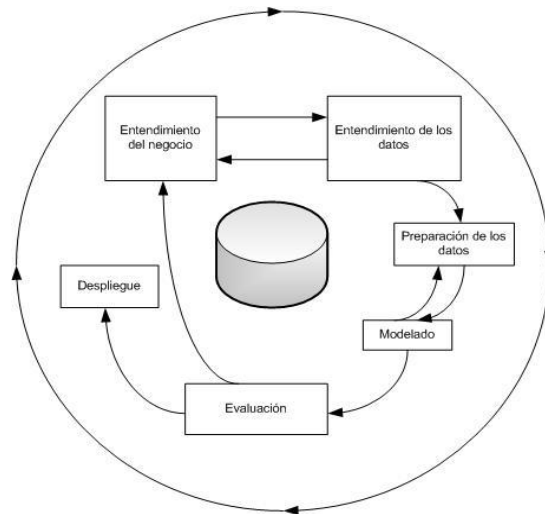


Ilustración 2. Fases de la metodología CRISP-DM [11]

Basados en los objetivos mencionados en las secciones 2.2 y 2.3, el planteamiento para este proyecto aterrizado bajo la metodología CRISP contará con las siguientes fases:

- Entendimiento del negocio: Esta fase comprende el marco teórico/estado del arte en donde se tratan los conceptos de Amenazas Persistentes Avanzadas, herramientas de seguridad y trabajos relacionados en el área (sección II).
- Entendimiento de los datos: Esta fase se divide en dos tareas: La primera, la lectura de la documentación técnica de los tres ataques tipo APT de mayor impacto que hasta el momento se han presentado [14]: Duqu, Stuxnet y Operación Aurora. La segunda tarea es la realización de simulaciones de los vectores o métodos utilizados por los ataques anteriormente mencionados sobre un sistema operativo que opera normalmente y no haya sido infectado.
- Preparación y modelado de los datos: Se especifica el tratamiento a los datos obtenidos de las tareas realizadas en la fase anterior (entendimiento de los datos) de manera que lo que se haya captado permita una creación del modelo de manera clara. A partir de este tratamiento previo, se selecciona el algoritmo que ayudará a realizar la clasificación que un dato nuevo sea señal de un ataque informático o no. De los datos obtenidos en las fases anteriores, se hará un proceso de entrenamiento y otro de pruebas.
- Evaluación: Se evalúa si los algoritmos candidatos para hacer la clasificación según el modelo obtenido tienen una capacidad de determinación apropiada para ser implementada, es decir, si su valor es superior al 50%. Asimismo, se observa de estos algoritmos identificados, cuál de ellos clasifica mejor.

- Despliegue: En esta fase, se realiza el desarrollo del prototipo adaptando el modelo ya probado y calibrado. Dentro de esta fase se incluyen las siguientes tareas:
 - a. Diseño de la aplicación: Se indican cuáles son los componentes que se van a incluir en el desarrollo de la herramienta. Aquí se harán las siguientes actividades:
 - i. Desarrollo de diagramas de diseño: Componentes, secuencia, casos de uso.
 - ii. Validación de los diagramas con el cliente.
 - b. Implementación del prototipo según el diseño establecido en el inciso anterior: Éste debe estar en capacidad de notificar el dato o elemento del sistema operativo que presentó algún cambio e indicar si es señal de ataque o no. La metodología de desarrollo que se utilizará será en espiral debido a que además de requerirse mucha comunicación con el cliente, se requiere hacer una planeación de cuáles son los elementos que se requieren para adaptar la aplicación a cada vector de ataque que se desea detectar. Dentro de esta tarea se realiza:
 - i. Aclaración de requerimientos
 - ii. Desarrollo la herramienta.
 - iii. Retroalimentación con el cliente.
- Pruebas: Aplicación del prototipo implementado a un plan de pruebas determinado. Este plan de pruebas tiene contemplado:
 - a. Planeación: Preparación de actividades para comprobar la respuesta de la herramienta ante los escenarios planteados, los cuales son:
 - i. Pruebas de adición, eliminación y modificación de variables del entorno (archivos, carpetas, puertos, registros, etc.) e inflado de archivos.
 - ii. Prueba con la herramienta McAfee Evader [15] con la cual se utiliza uno de sus vectores para que pueda identificarse el cambio generado en una máquina víctima y la probabilidad de que éste sea señal de un ataque.
 - iii. Prueba con la herramienta Metasploit de Kali [16] para utilizar uno de los vectores de los ataques investigados, se capturen los cambios en el sistema operativo y se indique si dichos cambios son señal de un ataque informático.
 - b. Ejecución del plan de pruebas.
 - c. Retroalimentación.

II – MARCO TEÓRICO / ESTADO DEL ARTE

En los últimos años, se ha podido identificar que el número de amenazas cibernéticas ha ido en crecimiento hasta tal punto que ellas mismas se han ido cambiando y evolucionando. Dichos cambios se han logrado determinar no sólo en las técnicas y herramientas que utilizan, sino también en los propósitos y las víctimas que pretenden afectar. Ahora, estas amenazas cibernéticas se han vuelto una de las preocupaciones de entidades gubernamentales, financieras, militares, diplomáticas e incluso nucleares que miran con más interés cómo cuidar su información para que no sea robada, cambiada o, por lo menos, accedida por personal no autorizado [17].

Una de las amenazas cibernéticas que más ha tomado fuerza son las Amenazas Persistentes Avanzadas (APT). Sus ataques han sido el centro de atención en todo el mundo por la magnitud, su versatilidad y las entidades que han afectado.

En este capítulo, se tratan los fundamentos teóricos más importantes de acuerdo al desarrollo de este trabajo. Inicialmente, se hace una reseña sobre las Amenazas Persistentes Avanzadas (APT) indicando su definición, características y fundamentos. Posteriormente, se sigue con una reseña de las herramientas de seguridad más importantes que han tratado el tema de las APT bien sea por detección o para conseguir algún otro resultado. Finalmente, se relacionan algunos trabajos importantes en la investigación de soluciones ante este tipo de ataques.

1. Amenazas Persistentes Avanzadas (APT)

1.1 Definición

El desafío de proteger la información de una compañía se vuelve una labor cada vez más ardua porque los ataques informáticos son cada vez más complejos y eficientes, que son producto de una mejor estructuración de su estrategia y objetivos.

Reflejo de lo anterior son las Amenazas Persistentes Avanzadas (APT), que se definen como "Adversarios con niveles sofisticados de pericia y recursos que permiten, haciendo uso de múltiples vectores de ataque (por ejemplo, cibernético, físico y el engaño), generar oportunidades para alcanzar sus objetivos, que habitualmente son establecer y extender su posicionamiento dentro de la infraestructura de tecnología de las organizaciones con fines de filtrar información hacia el exterior continuamente, o socavar o impedir aspectos importantes de una misión, un programa o una organización, o ubicarse en una posición que le permita hacerlo en el futuro." [6]

Estos ataques cuentan con la ventaja de hacer uso de muchos recursos para hacer su ejecución, es decir, no cuentan con un solo vector para ejecutar su ataque, sino que juntan varios elementos con el fin de extraer la información de la víctima.

Un atacante involucrado en este tipo de acciones cuenta con las siguientes motivaciones para hacer estas ejecuciones [18]:

- El deseo de acceder a propiedad intelectual sensible y tener canales confiables para poder ingresar a ellos cuando se requiera.
- No están interesados en ataques rápidos en donde obtienen datos. Su foco se centra más en estar dentro del ambiente de la víctima de manera tranquila y silenciosa armando los mensajes necesarios para tantear las vulnerabilidades más probables de atacar.
- Persistencia y previsión: No desfallecer en el objetivo cuando un ataque no es exitoso y siempre tener un plan de contingencia en caso de una posible detección.
- El atacante quiere tener acceso a largo plazo a los datos de su víctima.

Ahora, hay que entender que una APT no se trata de ataques organizados por activistas políticos, no necesariamente tienen que ser siempre sofisticados y tampoco tiene como intención primaria promover crímenes. Una APT, a veces, está patrocinados por el Gobierno con fines de cuidarse política y económicamente [1] y pueda que el ataque no sea de por sí avanzado o sofisticado, puede simplemente aprovechar debilidades y errores en la estructura informática y de información de la víctima con el fin de robar datos (especialmente propiedad intelectual). De hecho, el término “sofisticado” se acomoda más al atacante que ya está más estructurado y es más organizado en sus ejecuciones [19].

1.2 Fundamentos

1.2.1 Ciclo de vida de un ataque tipo APT

Cuando el atacante ha elegido su víctima, inicia un ciclo de vida para armar el ataque tipo Amenaza Persistente Avanzada. Dicho ciclo de vida tiene las siguientes fases [17] [20]:

- Reconocimiento: El atacante recopila toda la información importante acerca de la víctima sin dejar ninguna fuente sin visitar. Puede apoyarse en redes sociales, intranet, Internet, etc.
- Entrega: Se establece un punto de apoyo o una intrusión inicial en el sistema llegando a contactar la víctima con base en la información obtenida en la fase de reconocimiento. Esto se consigue por medio de una o varias técnicas de como *phishing*, ingeniería social, *DNS Spoofing*, ataque “*Man in the Middle*”, uso de memorias USB, *exploits*, ataques basados en la Web, entre otras.
- Explotación: Una vez el ataque cumple con el cometido de ingresar en la red de la víctima, se instala un *malware* y se crea una conexión que le permita vincular el equipo comprometido con un servidor externo a la infraestructura

de la víctima llamado *Command and Control* (o C&C), con el cual puede ejecutar las instrucciones. Para conseguir esto, el atacante suele utilizar un *backdoor* o puerta trasera para establecer la comunicación y el *malware* a utilizar debe tener la capacidad para ocultarse.

- Operación: El atacante se mueve en la red de la víctima identificando servidores que contengan información sensible, cuentas de usuarios que tienen acceso a estos recursos, buscar unidades mapeadas, etc., y a partir de esta exploración, se define la estrategia para recolectar los datos y exportar la información al servidor C&C.
- Extracción de datos: El atacante obtiene los permisos suficientes para tener acceso a aquellos recursos que tienen información sensible (esto implica que se debió haber realizado escalamiento de privilegios). Además, empieza a crear copias en otros lados para tener información de respaldo.
- Explotación continua: Se extraen los datos y exportarlos a un lugar fuera del área donde permanece la víctima. El envío de estos datos se hace de forma cifrada a través de protocolos como FTP o HTTP al servidor C&C.



Figure 1. Ciclo de vida de una Amenaza Persistente Avanzada. Adaptado de Giura [20]

1.2.2 Vías de infección

Como se especificó anteriormente, los ataques tipo APT tienen la ventaja de tener varios vectores de ataque para ejecutar su labor y tiene sentido obedeciendo a que el

proceso conlleva mucho tiempo y atraviesa por muchas etapas. Cada vector es elegido de acuerdo al análisis de los datos obtenidos en el proceso de recolección y en el objetivo que el atacante desea perseguir.

En líneas generales, Holguín *et Al.* [17] propone dos grandes vías de infección en los ataques tipo APT: la ingeniería social y los *exploits* los cuales se detallarán a continuación:

1.2.2.1 Ingeniería Social

La Ingeniería Social es el grupo de prácticas que utiliza el atacante para engañar a la víctima con el fin de conseguir su objetivo, que bien puede ser, acceder y/o tomar información privilegiada de la víctima (personal y empresarial) que se puede dar visitando un enlace, abriendo un documento que le fue enviado por correo electrónico o accedido desde una memoria USB, permitir el paso a un desconocido a las instalaciones físicas de la empresa de la víctima o incluso, acceder al perfil del usuario en una determinada red social. Esta técnica se utiliza especialmente en la fase de reconocimiento [17]. Dentro del contexto de la Ingeniería social, hay varias técnicas que se utilizan para obtener información, entre ellas:

- Perfil en redes sociales: Son las cuentas relacionadas con la potencial víctima en las redes sociales a las cuales está afiliado. En él se muestra información personal y en algunos casos, su historial laboral. Dependiendo del nivel de cercanía a esta persona en dicha red, se puede obtener más información. El uso de las redes sociales también se presta para crear perfiles falsos que tienen el fin de engañar a la potencial víctima acercándose a ella y obtener algo de información útil para poder decidir qué vector podría utilizar para atacar.
- Programación neurolingüística: Aprovechar la influencia del lenguaje para manipular la conducta mental y emocional de las personas y así obtener información.
- Lenguaje corporal: Estudiar la expresión corporal de la persona que van a engañar con el fin de que el atacante pueda entender el tono del lenguaje que debería utilizar para poder extraerle información.
- Comunicación verbal: Acercamiento verbal a la potencial víctima. Generalmente, se utilizan pretextos.

El atacante también aprovecha situaciones del contexto para extraer información que resulta importante. Por ejemplo, en una compañía, resulta fácil extraer información de la secretaria, de un empleado inconforme o incluso, de trabajadores de una empresa tercerizada o también conocidos como contratistas.

La Ingeniería Social utiliza los siguientes métodos en combinación con las técnicas anteriormente mencionadas para realizar sus ataques justo después de haber accedido al equipo de la víctima:

1.2.2.1.1 Infeción por malware

Esta infección puede ser:

- A través de sitios web: Consiste en que el equipo de la potencial víctima se contagia con solo visitar una página web a la cual el atacante había inyectado previamente un script entre su código HTML. El navegador del equipo de la víctima estará obligado a hacer nuevas peticiones a otros servidores controlados por el atacante quienes intentarán explotar las vulnerabilidades del navegador para lograr descargar el malware.
- *Phishing*: Es el envío de un correo electrónico suplantando la identidad de un conocido de la potencial víctima. En dicho correo electrónico, se encuentra incorporado un enlace a un sitio malicioso para que este personaje pueda acceder a él o también puede haber un documento adjunto que al ser abierto hace que la máquina se infecte.
- Redes y aplicaciones *Peer-to-peer (P2P)*: En una red P2P, dos o más computadores se pueden conectar en red y compartir recursos sin necesidad de un servidor dedicado. A diferencia de los modelos cliente-servidor, las redes P2P descentralizan los recursos de una red, es decir, en lugar de dejar la información compartida en un solo lugar, ésta puede ser localizada en cualquier dispositivo de la red [21].

Una aplicación P2P, permite a un dispositivo final, conocido como *peer*, funcionar simultáneamente como cliente para una transacción o como servidor para otra dentro de una misma sesión [22]. Una aplicación P2P puede utilizarse en redes P2P, redes cliente-servidor o a través de Internet [21].

Bajo este escenario, el usuario víctima puede estar descargando un archivo o en programas que supuestamente son legítimos, pero que en realidad posee código malicioso o hay un “asistente” que recoge cierta información de la víctima [17]. Esta situación se da porque las redes P2P normalmente no utilizan cuentas de usuario, permisos o monitores centralizados, razón por la cual es difícil implementar políticas de seguridad [21].

- Software no licenciado: Son copias de los programas que los usuarios quieren tener sin pagar por una licencia. Estos programas se pueden adquirir por medio de páginas web o redes P2P, pero la dificultad se encuentra en el hecho de que no hay garantía que estas aplicaciones estén libres de *malware*.

Por lo general, dichas aplicaciones sin licencia vienen acompañadas de otras adjuntas diseñadas para evadir el proceso de licenciamiento. Entre dichos programas se encuentran los *cracks*, *keygens*, etc.

1.2.2.1.2 Medios físicos

Otro de los métodos que utiliza la Ingeniería social para obtener información de las víctimas es por medio de los medios físicos, los cuales son dispositivos extraíbles (memorias USB, discos duros, tarjetas, CDs, DVDs y demás) que se conectan a cualquiera de los equipos de una red para cumplir su cometido. Dentro de estos medios, también se tienen en cuenta los dispositivos móviles tanto empresariales como personales, que pueden ejercer el rol de atacante o de víctima. De hecho, ya se han dado casos de dispositivos hechos en fábrica con malware instalado [17].

1.2.2.2 Exploits

Los *exploits* son herramientas que aprovechan vulnerabilidades a partir de los navegadores y programas de los clientes conectados para permitir la ejecución de código que no es confiable o también para obtener información [23]. Para que las víctimas puedan caer en estas herramientas, existen diversas técnicas que van desde el envío de correos electrónicos con enlaces maliciosos hasta incluir código en páginas Web legítimas que permita redirigir los usuarios al dominio del atacante.

1.3 Características de las APT

Básicamente, las Amenazas Persistente Avanzadas (APT) tienen tres características principales que coinciden con sus siglas en inglés [20] [6]:

- Avanzada (A): Atacantes con habilidad y bien entrenados que utilizan todo el espectro de tecnologías de intrusión a la red y técnicas tradicionales de inteligencia. Aquí también entra el hecho que el ataque debe ser lo suficientemente sigiloso para no ser identificado por un antivirus o un IDS.
- Persistente (P): Contar con la resistencia de mantener el ataque por largos periodos de tiempo y tener habilidad para responder a las medidas puestas por los sistemas objetivos (es decir, que pueden reorganizarse y volver a armar un ataque en caso de que el ataque originalmente establecido haya fallado)
- Amenaza (T): Intención del atacante de causar daño y crear pérdidas interrumpiendo servicios y pérdida de datos.

Adicionalmente, hay unas características que complementan a este tipo de ataques:

- Premeditada: El atacante estudia a sus potenciales víctimas, y cuando ya encuentra el objetivo a atacar, empieza a analizar sus vulnerabilidades para después desplegar el respectivo ataque [17]. Un ejemplo de lo anterior son las personas con cargo estratégico en una compañía o personas que poseen cuentas con perfil de Administrador en el sistema de la empresa en la que trabajan (Incluye analistas de alto rango, Administradores de red, Gerente de IT, etc.). Por lo general, este tipo de personajes no saben que están siendo

analizados para ser atacados y son los predilectos por la importancia de información que manejan.

- Los ataques APT se concentran en organizaciones, no necesariamente gubernamentales [19].
- El tiempo no es factor para implementar los ataques: Para el atacante, no importa si se toma semanas o meses para encontrar a la víctima predilecta y otro tiempo más para armar el ataque. La idea es que sea lo menos ruidoso y lo más efectivo posible.
- Los atacantes esperan conseguir beneficios a corto plazo, cosa que se puede esperar en un atacante de propósito general. [17].
- Así como los atacantes son persistentes, también lo son sus ataques. De hecho, una de sus ventajas consiste que, en caso que el ataque sea detectado, el atacante tiene planes de contingencia que le permita reorganizarse y atacar nuevamente [20].
- Por otro lado, este tipo de ataques están bien financiados ya que necesita disponer de tiempo y recursos para llevarse a cabo, además de que pueden haber organizaciones involucradas [17].
- El atacante está enfocado en todos los activos que son importantes para la organización, especialmente, los datos [19].
- También, se puede decir que los niveles de detección de los ataques tipo APT son muy bajos porque usan firmas únicas o son muy difíciles de correlacionar con los ataques conocidos. Esto se debe a que están diseñados para evadir soluciones *antimalware* y Sistemas de Detección de Intrusos (IDS) [17].

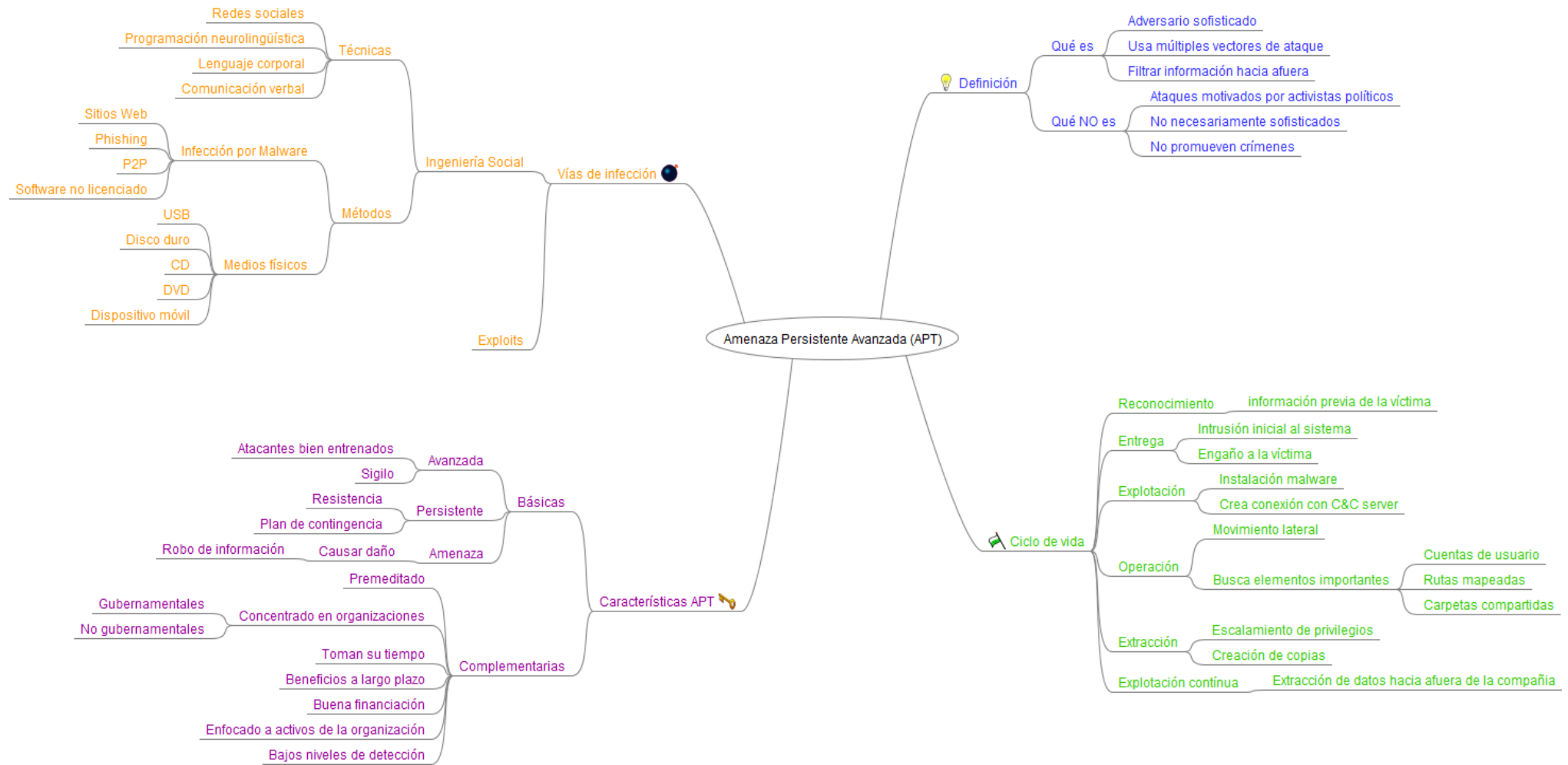


Ilustración 3. Resumen conceptos Amenazas Persistentes Avanzadas (APT)

Herramientas de seguridad

1.4 Sistemas de Detección de Intrusos (IDS)

Un Sistema de Detección de Intrusos (IDS) se conoce como un sistema de *Hardware* o *Software* que automatiza el proceso de monitoreo de eventos que ocurren en un computador o red en busca de señales de intrusión, actividades de usuarios no autorizados y la ocurrencia de malas prácticas. Esta herramienta está diseñada especialmente para vigilar los ataques de parte de personas que están fuera del sistema [23].

Dentro de los Sistemas de Detección de Intrusos, se encuentran:

1.4.1 Sistemas de Detección de Intrusos basados en red (NIDS)

Los Sistemas de Detección de Intrusos basados en Red (*Network Intrusion Detection System* - NIDS) se encargan de analizar el tráfico de la red completa examinando los paquetes individualmente y detectando aquellos que potencialmente son maliciosos [24]. Los NIDS tienen dos componentes:

- Un sensor: Situado en un segmento de la red, la monitorea en busca de tráfico sospechoso
- Una consola: Recibe las alarmas del sensor o sensores y dependiendo de la configuración reacciona a las alarmas recibidas.

Los NIDS funcionan a partir de:

- Detección de firmas: Examina pasivamente todo el tráfico de red que pasa por sus puntos y examina cada paquete TCP/IP contra las firmas de ataques conocidos: si detecta que hay un paquete sospechoso, la conexión con la dirección IP que está enviando los paquetes es cortada, el paquete se libera y la alarma se levanta. Son útiles para monitorear ataques conocidos y asegurar que ninguno de ellos va a ser exitoso en infringir el sistema. Su gran desventaja está en que no es tan efectivo detectando *malwares* nuevos o bastante modificados de manera que su firma no se pueda comparar. Otra desventaja de los NIDS basados en firmas radica en que son ciegos a cualquier ataque que no tenga firma [24].
- Detección de anomalías: Son sistemas que construyen modelos estadísticos o de línea base para el tráfico que monitorean y levantan una alerta si hay un evento que se muestra diferente de manera significativa de estos modelos. Uno de los métodos más comunes de detección de anomalías de red consiste en observar tráfico por “conformidad” de tráfico TCP/IP e incluso de tráfico de capas superiores como HTTP o SMTP. Este modelo de detección tiene la desventaja en definir los umbrales de lo que es “correcto”, lo que es “anómalo”,

hasta donde se delimitan las fronteras para identificar si un evento es de alguno de estos dos tipos y lo complejo que es para que estos sistemas se adapten y aprendan [24].

- Híbrido: Los sistemas híbridos toman las mejores características de los dos sistemas anteriormente mencionados y los une en un solo sistema en un intento por responder a las debilidades que cada uno posee: Utiliza las firmas por su flexibilidad y rapidez en el procesamiento y utiliza las anomalías para detectar el tráfico sospechoso. Sin embargo, estos sistemas producen muchas falsas alarmas puede ser por construcción imprecisa de firmas o por mala configuración de los sensores [23].

1.5 Sistemas de Prevención de Intrusos (IPS)

Los Sistemas de Prevención de Intrusos (IPS) se conocen como sistemas que poseen las mismas capacidades de los IDS, pero con el ingrediente extra de intentan detener un ataque activo o un problema de seguridad [23].

1.5.1 Sistemas de Prevención de Intrusos basados en red

Los IPS basados en red se caracterizan por prevenir los eventos sospechosos. Esto se logra ubicando el dispositivo en la línea del tráfico que es monitoreado. Cada paquete es revisado y si no levanta ninguna alerta basado en una firma o en un umbral de anomalía se admite, en caso contrario, el paquete se descarta y levanta la alerta [23].

La gran desventaja de este tipo de sistemas radica en que confían demasiado en firmas estáticas, no pueden examinar tráfico encriptado y no son tan eficientes cuando se utilizan en redes de alta velocidad [23].

1.5.2 Sistemas de Prevención de Intrusos basados en host

Es un sistema que utiliza la misma estrategia de firmas que los IDS e IPS basados en red para proteger y analizar lo que otros procesos en el sistema están haciendo a alto nivel de detalle. Involucra comunicación entre procesos, llamadas a sistema, tráfico de red y patrones de comportamiento para actividades sospechosas. Cuenta con la ventaja de ofrecer una protección mejorada ante ataques del día cero que no son muy conocidos [25].

Como todos los procesos basados en firmas, son efectivos cuando dependen de este esquema. Adicionalmente, requieren de mucho procesamiento y por lo tanto, no son muy útiles en sistemas de mucha carga de procesos [25].

1.6 Firewalls

Los firewalls son responsables de controlar el acceso entre dispositivos como computadores, redes y servidores. Su propósito es el de filtrar paquetes: inspeccionan cada uno de ellos independientemente si son de entrada o de salida. Sin embargo, el rol que están cumpliendo los firewalls ha evolucionado y ahora son capaces de hacer otras labores como puertas de enlace a nivel de la aplicación, lo que significaba que podía inspeccionar el interior de los paquetes más allá del encabezado TCP para entender qué labor estaba realizando [26].

Su funcionamiento inicia cuando llega un paquete y el firewall aplica una política o regla para determinar la acción apropiada que puede pasar por aceptar el paquete o rechazarlo y guardar el resultado en un log o historial. Cada regla consiste en un conjunto de tuplas donde cada una de ellas corresponde a un campo de cabecera y hay otros cinco campos para el paquete de Internet: protocolo, dirección y puerto de origen, dirección y puerto de destino. De los campos anteriormente mencionados, la cabecera se compara secuencialmente con los campos de la regla: si hay un “match”, se ejecuta la acción asociada [26] [23].

Los firewalls pueden ser clasificados basados en donde estén implementados. Según lo anterior, pueden clasificarse en host y en red: Los firewalls en host son aplicaciones que hacen una protección de múltiples elementos en el sistema (pueden hacer NAT), servicios QoS, entre otros. Una de sus grandes ventajas es la capacidad de actualizarse automáticamente. Los firewalls de red son aplicaciones diseñadas para proteger todos los computadores de una red interna, por lo tanto, debe ser capaz de manejar altos anchos de banda y procesar paquetes a alta velocidad [23].

Los firewalls presentan como principales desventajas proporcionar un solo perímetro de defensa, dependen mucho de las reglas, si un agresor lo pasa su valor como defensa se pierde y además las actualizaciones, que en el caso específico de los firewall en red, suelen ser muy costosos [26].

Adicionalmente, ya se conoce de la existencia de los firewalls de próxima generación (*Next Generation Firewall* - NGFW) los cuales se caracterizan por soportar niveles adicionales de granularidad de políticas y control, control de aplicaciones, IPS, anti-malware, seguridad a correos electrónicos, entre otros. A pesar de estas capacidades adicionales, este tipo de firewalls castigan un poco más el performance o rendimiento del dispositivo sobre el cual trabajan debido a la gran cantidad de reglas que almacenan [27].

1.7 Antivirus

Los antivirus tienen la función de detectar la presencia de malware, removerlo y proteger el host de futuras infecciones. La tarea de la detección también está en minimizar falsos positivos (falsas alarmas) y falsos negativos (*malware* perdido). Estas detecciones se realizan comparando los bytes de los archivos con firmas conocidas que poseen unos patrones de bytes específicos [28].

Otra de las tareas importantes de los antivirus es la distribución de sus firmas cuando un nuevo *malware* es capturado: luego de ser analizado, sus características quedan plasmadas en una firma la cual es difundida a los demás hosts a manera de actualizaciones. Sin embargo, las debilidades de esta técnica radican en la dependencia a las firmas, las cuales no son efectivas ante un *malware* muy nuevo o que ha pasado por muchas modificaciones [28].

Otro enfoque que han tomado los antivirus es el basado en comportamientos que consiste en enfocarse lo que el *malware* intenta hacer. Esta aproximación no utiliza las reglas para hacer las detecciones, sino que utiliza las reglas heurísticas. Sin embargo, definir hasta qué punto un comportamiento es normal o es anómalo resulta ser muy complicado y poder determinarlo es una tarea aún más difícil [23].

1.8 Honeypots

Es una técnica que tiene como objetivo aprender los comportamientos que pueden asumir los atacantes. Para esto, atraen los ataques a un *host* que es vulnerable. Esta técnica cobra aún más importancia si los ataques se logran identificar cuando ya son inminentes, a pesar de que no habría mucho tiempo para tomar una estrategia de defensa.

Todo *honeypot* tiene funciones diferentes a la de un PC común y corriente. En primer lugar, no puede ser utilizado para servicios legítimos o tráfico. En segundo lugar, el *honeypot* está encomendado a tareas de monitoreo y de tomar registro de las actividades captadas, y en tercer lugar, el *honeypot* debe estar aislado de la red real: si estuviera en ella, su función sería en vano y podría generar oportunidades para que un atacante haga un daño real a la infraestructura [23].

Trabajos relacionados en el área

Wang *et Al.* [6], proponen un *framework* basado en una red de genes con el fin de describir los patrones de comportamiento semánticos de las aplicaciones de red. Para conseguir esto, utilizaron tecnologías como *Big Data*, *Cloud Computing* y una combinación entre análisis inverso de protocolos de red y procesamiento de flujos de datos.

Haciendo uso de las tecnologías anteriormente mencionadas, se compararon comportamientos de la red estudiada con los de las redes de genes para determinar las acciones relevantes de la primera respecto a las aplicaciones de red conocidas y descubrir aquellos comportamientos que son anormales y así proveer la base para crear el *framework* que logre identificar y seguir los ataques tipo APT.

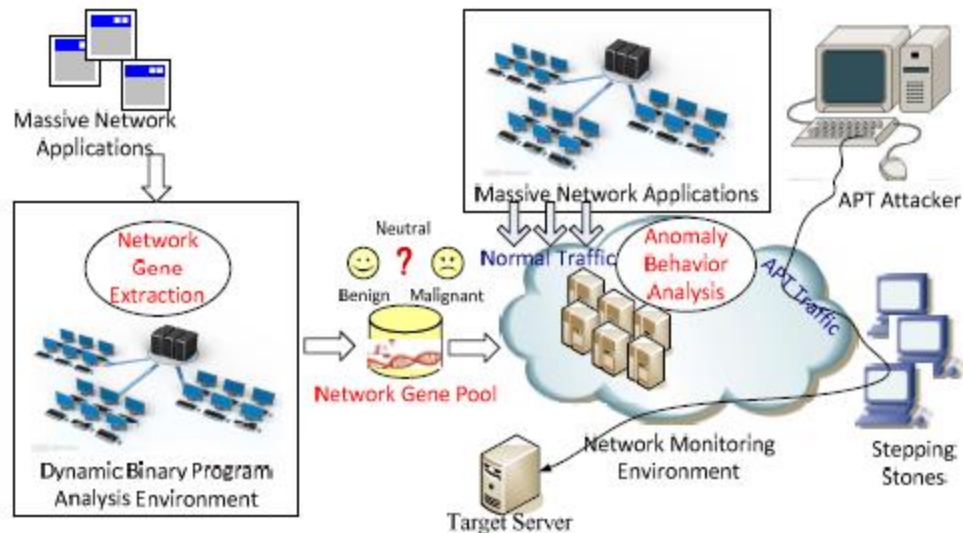


Ilustración 4. Modelo de monitoreo basado en red de genes propuesto por Wang et Al [6].

Sin embargo, durante la ejecución del procedimiento llegaron a un punto en donde se realizó un análisis manual que terminó siendo ineficiente y causó que fuera imposible cubrir la necesidad de detectar una APT en caso de que existiesen una gran cantidad de aplicaciones de red en segundo plano.

Chandran *et Al.* [29] propusieron un modelo de clasificación para detección de APTs. Esta clasificación se hace bajo el algoritmo de *Random Forest* que consiste en tener un número de árboles de decisión (que se forman a partir de unos datos de entrenamiento) y combinarlo con la técnica de *bagging* con el fin de obtener predicciones de alta precisión. Se necesitaron grandes cantidades de datos en largos periodos para obtener los árboles de clasificación y de pruebas. Las pruebas realizadas generaron clasificaciones cercanas al 98%.

Zhao *et Al.* [7] realizó un trabajo sobre detección de *malware* tipo APT y dominios de servidores *Command and Control* (Servidores C&C) basados en análisis de tráfico y DNS maliciosos en el cual sitúan un sistema en el punto de salida de una red de la organización. El tráfico analizado correspondiente a una dirección IP sospechosa se realizó a partir de técnicas basadas en firmas y anomalías.

Los Laboratorios de Kaspersky continuamente presenta información de ciber-campañas y *malware* APT. Sus reportes incluyen dominios y servidores C&C con los que se contactan los *malware* APT [14].

Trend Micro propone su herramienta *Deep Security* que se caracteriza por detectar, analizar y actuar en tiempo real frente a las amenazas avanzadas. Esta plataforma cuenta con utilidades como *antimalware*, firewall basado en *host*, inspección de *logs*, supervisión de puertos y protocolos, protección al correo electrónico, interfaz para presentación de reportes en tiempo real, entre otras. Maneja una versión para administradores y otra versión como inspector. La principal desventaja de esta herramienta está en su funcionamiento depende de un agente especial de VMware sobre el cual se implementan estos servicios y por el cual se tiene que pagar un monto económico adicional [10].

Por otro lado, Saud e Islam [30] proponen un framework que trabaja con un *honeypot* y un IDS de red (NIDS) para alertar al administrador sobre los eventos que pueden darse sobre su red. La idea es que con el uso de una herramienta de engaño como lo es el *honeypot*, se pueda identificar los recursos que más son atacados y se pueda establecer políticas más acordes para detectar estos ataques. Para la implementación, se creó un ambiente que se compone de tres máquinas virtuales de las cuales una será el *honeypot* y otra será el NIDS. La alarma se levanta cuando un servicio aparentemente legítimo se ejecuta en el *honeypot* y es solicitado por un intruso, esta alarma le llega al Administrador del Sistema que tomará la acción pertinente.

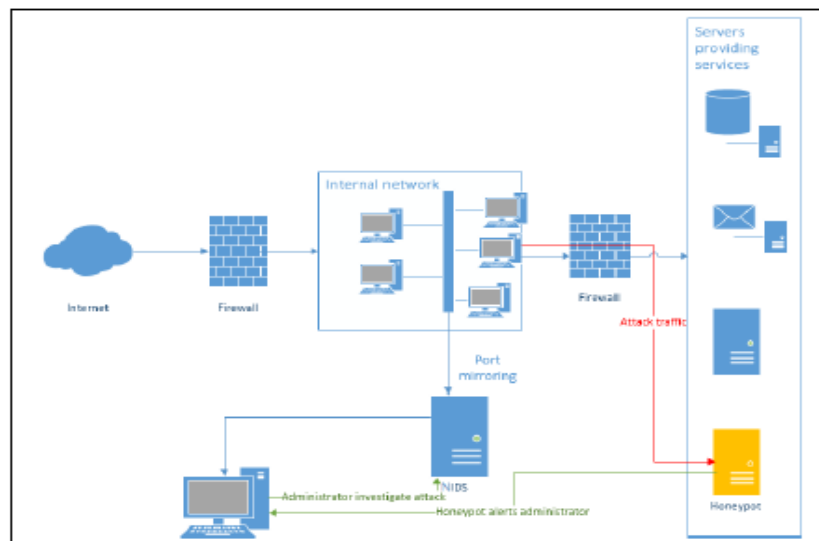


Ilustración 5. Funcionamiento de framework de detección usando honeypots y NIDS [30]

III – DESARROLLO DEL TRABAJO

En este capítulo se ve el detalle de las actividades que se propusieron en la metodología (sección 1.2.4). Todo el proceso que se sigue hasta la obtención del prototipo parte de la hipótesis de saber si existen características que se puedan asociar entre el comportamiento de los elementos del sistema operativo y la posibilidad de ocurrencia de un ataque APT.

Para comprobar lo anterior, se inicia con la fase de entendimiento cuyo núcleo es la obtención de los datos. Luego, se indican la preparación de los datos, obtención del modelo, evaluación y el despliegue que es la forma en que se adapta el algoritmo al prototipo a construir.

1. Fase de Entendimiento del contexto de la problemática

1.1 Actividades de documentación

En esta tarea se explican tres de los ataques tipo APT más importantes, de los cuales se obtienen las entradas (archivos, registros, drivers, etc.) que fueron el insumo para formar un repositorio o base de datos utilizado por el algoritmo elegido y que viene dentro de la herramienta desarrollada para detectar si un evento nuevo es síntoma de un ataque tipo APT o no. Los tres ataques que se detallaron son: Operación Aurora, Stuxnet y Duqu.

1.1.1 Operación Aurora

La Operación Aurora se conoce como un ataque informático que ocurrió desde el segundo semestre de 2009 y tenía el objetivo de robar la propiedad intelectual de activos de información de las compañías que eran señaladas como sus víctimas. Este ataque hace parte de la ciberguerra que se presume fue realizada desde China y que tenía como objetivo 34 empresas entre las que se encuentran Google, Adobe, Symantec, Yahoo, Dow Chemical y Juniper [31].

El nombre de "Aurora" se debe a la designación de la ruta de un archivo encontrado en el payload que fue seguido a un hacker relacionado con la operación [32].

```

00 00 00 00 00 00 F0 3F 00 00 00 00 00 00 20 40 .....?..... @
00 01 80 46 75 3D A7 3F D4 8B 0A 3F 15 EF C3 3E ...Fu=?.?..?..>
F3 04 35 3F 00 00 00 00 00 00 00 00 00 00 00 00 ..5?.....
65 2B 30 30 30 00 00 00 00 00 00 C0 7E 01 50 41 e+000.....".PA
00 00 00 00 FF FF 47 41 49 73 50 72 6F 63 65 73 .....GAIisProces
73 6F 72 46 65 61 74 75 72 65 50 72 65 73 65 6E sorFeaturePresen
74 00 00 00 4B 45 52 4E 45 4C 33 32 00 00 00 00 t...KERNEL32...
31 23 51 4E 41 4E 00 00 31 23 49 4E 46 00 00 00 1#QNAN...1#INF...
31 23 49 4E 44 00 00 00 31 23 53 4E 41 4E 00 00 1#IND...1#SNAN...
52 53 44 53 91 82 FE 94 29 AB E5 42 A6 53 10 A8 RSDS...).B.S...
D2 04 69 98 10 00 00 00 66 3A 5C 41 75 72 6F 72 ...i.....f:\Auror
61 5F 53 72 63 5C 41 75 72 6F 72 61 56 4E 43 5C a_Src\Aurora\NVC\
41 76 63 5C 52 65 6C 65 61 73 65 5C 41 56 43 2E Avc\Release\AVC\
70 64 62 00 94 4D 03 10 00 00 00 00 00 00 00 00 pdb...M.....
FF FF FF FF 00 00 00 00 00 00 00 00 54 21 03 10 .....T!..
00 00 00 00 00 00 00 00 00 00 00 00 01 00 00 00 .....
    
```

Ilustración 6. Nombre de la ruta del payload del cual salió el nombre de Operación Aurora [33]

La Operación Aurora se concentra en dos procesos principales para conseguir su objetivo: Primero, la infección de un equipo de la compañía víctima por medio del envío de un correo con un link adjunto que explota una vulnerabilidad en Internet Explorer. Segundo, la instalación de un *backdoor* para que el atacante pueda ingresar al equipo infectado y extraer la información. Dichos procesos se consiguen por medio de las siguientes tareas que especifica McAfee [9]:

1. La víctima (empleado de la compañía) recibe un email con un link o un mensaje instantáneo desde una fuente conocida:

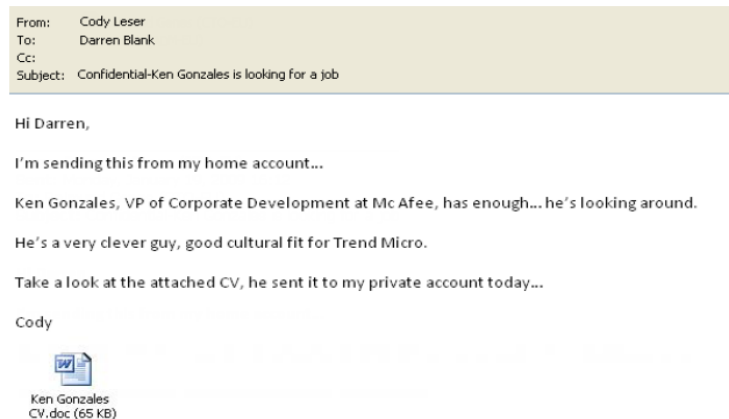


Ilustración 7. Correo utilizado para engañar a la víctima [11]

2. El usuario da clic en el link que dirige a la víctima a una página web en Taiwán, que contiene *payload* malicioso escrito en *Javascript*.
3. El navegador del usuario descarga el código *Javascript* malicioso que incluye un *exploit* de día cero que aprovechará una vulnerabilidad de Internet Explorer.

4. El *exploit* descarga un archivo binario disfrazado como imagen desde los servidores C&C que apuntan a una página web en Taiwán y que ejecutan el *payload* malicioso.
5. El *payload* instala un *backdoor* o puerta trasera conectada al servidor C&C en Taiwán.
6. El atacante accede al equipo de la víctima y queda disponible para que descargue fuentes de propiedad intelectual (por ejemplo, fórmulas, patentes, *copyright*, etc.) y archivos ubicados en sistemas de gestión de configuración de software (SCM).

Según McAfee [34], hay 3 elementos que forman parte del proceso de infección del equipo que será víctima y la creación del *backdoor* o puerta trasera para el acceso por parte del atacante:

- *Exploit* Comele [35]: Es un *exploit* que crea un script "hecho a mano" que aprovecha una vulnerabilidad de Internet Explorer causando una conexión de nombre `hxxp://demo[remove].jpg` de donde se descarga un archivo binario con nombre `Roarur.dr`, el cual se guarda como el ejecutable `%Application Data%\a.exe`, por ejemplo, `C:\Documents and Settings\User\Application Data\a.exe`. En el mismo directorio, `a.exe` se desenscripta y queda como el archivo `b.exe` que quedará en la misma carpeta.
- `Roarur.dr` [36]: Es un troyano que descarga el archivo `Roarur.dll` en el equipo de la víctima.
- `Roarur.dll` [37]: Es el troyano que instala el malware en el equipo de la víctima y deja el *backdoor* instalado para que el atacante pueda ingresar.

Los sistemas operativos afectados por el ataque fueron Windows 2000, XP, Server 2003 y 2008, Vista y Windows 7 [38].

Las entradas afectadas por la Operación Aurora son:

- Archivos:
 - `%Application Data%\a.exe`
 - `%Application Data%\b.exe`
 - `Rasmon.dll`
 - `DFS.bat`
 - `Securmon.dll`: Inyecta código en el servicio `SVCHOST.EXE` para ver si los archivos `Acelpvc.dll` y `VedioDriver.dll` se encuentran en el equipo de la víctima.
 - `A0029670.dll`
 - `Acelpvc.dll` (la presencia de este archivo no implica que haya una infección)
 - `AppMgmt.dll`

- VedioDriver.dll (la presencia de este archivo no implica que haya una infección)
- %Temp%\c_1758.nls
- %Temp%\[RANDOM FILE NAME]
- Registros:
 - HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\RaaS [% random 4 chars %] "ImagePath" = %SystemDir%\svchost.exe -k netsvcs
 - HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\RaaS [% random 4 chars %] "Start"= 02, 00, 00, 00
 - HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\RaaS [% random 4 chars %] \Parameters"ServiceDll" = %SystemDir%\rasmon.dll
 - HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\WindowsNT\CurrentVersion\SvcHost\ "netsvcs"
 - HKEY_LOCAL_MACHINE\Software\Sun\1.1.2\ "IsoTp"
 - HKEY_LOCAL_MACHINE\Software\Sun\1.1.2\ "AppleTik"
- Servicios:
 - Svchost.exe
- Accesos a páginas:
 - 360.home[removed].com
 - blog1.serve[removed].com
 - google.home[removed].com
 - ftp2.home[removed].com
 - update.our[removed].com
 - yahooo.8866.org
 - sl1.homelinux.org
 - 360.homeunix.com

1.1.2 Stuxnet

Stuxnet es conocido por ser un ataque que fue diseñado con el objetivo de reprogramar los Sistemas de Control Industriales (ICS o también conocidos como SCADA), modificando los Controladores de Programación Lógicos (PCL) para hacerlos trabajar de la manera que el atacante prefiera y escondiendo los cambios realizados en caso que el operador del ICS llegase a tener alguna sospecha. Según Symantec, es conocido por ser uno de los ataques más complejos por el entorno en el cual opera, el objetivo a cumplir y la variedad de componentes que utiliza que son *exploits* del día cero, *rootkits* para Windows y PCLs, técnicas de evasión de antivirus, procesos de inyección de código, funciones para infectar redes, actualizaciones de aplicaciones P2P e interfaces *Command and Control* [39].

Para la ejecución de la fase de reconocimiento, se necesitaba tener el esquema de funcionamiento de los ICS los cuales podían ser tomados por un empleado o podían ser obtenidos por una de las versiones iniciales de Stuxnet. Con estos documentos, se tenía el conocimiento para desarrollar una versión del ataque más a la medida, incluyendo el manejo de *drivers* que estuvieran firmados digitalmente para evitar ser rastreados.

Para la fase de infección, se utilizaron dispositivos USB para ingresar al sistema y para expandirse por la LAN de la víctima se utilizaron ataques de día cero y los recursos compartidos de red [40], esto con el fin de encontrar máquinas que manejaran el software Step 7 de Siemens, que eran los que controlaban el código en el PLC [39]. Toda la información de equipos infectados, llegaba al atacante por medio de un servidor C&C server desde el que podía manejar la forma de propagación del *malware*.

La fase de expansión de Stuxnet se ejecuta de 4 maneras:

- Memoria *flash*: Stuxnet aprovechó la vulnerabilidad MS10-046, que consistió en permitir la ejecución remota de código si se muestra un icono de acceso directo especialmente diseñado. A partir de esto, se podría conseguir elevar el nivel de permisos de usuario hasta conseguir el de usuario local. Este tipo de expansión afecta los sistemas operativos Windows XP, Vista, 7 y Server 2003 y 2008 [41].
- Red compartida: Se consiguió a partir de la explotación de las vulnerabilidades MS10-073 (Win32k.sys) y MS10-092 (Programador de tareas) con el fin de elevar privilegios. En el primero, el atacante utilizaba referencias del *kernel* para tal cometido facilitando la instalación de programas, ver, cambiar o eliminar datos, e incluso crear cuentas nuevas con todos los permisos [42]. En cuanto al segundo, Windows estaba validando de manera incorrecta si las tareas existentes en el Programador se estaban ejecutando en un marco de seguridad apropiado. Esto lo aprovechaba el atacante para ejecutar código arbitrario en el sistema local [43]. Dichas vulnerabilidades pudieron ser corregidas si tenían activadas las actualizaciones automáticas.
- Vulnerabilidad RPC (MS08-067): Esta vulnerabilidad consistió en el tratamiento incorrecto de las solicitudes RPC en la ejecución remota de código en el servicio de servidor de Windows que causaba una sobrecarga en el buffer. El atacante podía aprovechar esta característica para ejecutar código con privilegios de SYSTEM [42] [44].
- Vulnerabilidad del *Print Spooler* (MS10-061): Esta vulnerabilidad consistió en la ejecución remota de código dado el caso que un atacante realice una soli-

cidad de impresión a un sistema vulnerable que tenga una interfaz de administrador de trabajos por RPC. Esta ejecución de código remota se da porque cuando está habilitada la opción de compartir impresoras, no se validan los permisos de acceso al *spooler*, permitiendo al atacante crear archivos en el directorio de sistema y ejecutar código arbitrario, todo a partir de la solicitud de impresión especialmente diseñada [45].

El núcleo de Stuxnet fue un archivo de extensión DLL que contenía el código de la amenaza y trabajaba en conjunto con un *stub* que posee las instrucciones a ejecutar (“*exports*”) y unos recursos para llevarlas a cabo. Estos tres componentes son muy importantes para el resto del proceso que ejecuta este ataque.

Para la fase de instalación, inicialmente se verificaba la versión instalada de Stuxnet (si la hay), la versión del antivirus y el proceso más vulnerable de ser inyectado. Posteriormente, se revisaba si había permisos de administrador: si los había se iniciaba la inyección de código, si no, se iniciaba con ataques del día cero para escalar privilegios. En este proceso, se instalaron los drivers *Mrxcls.sys* y *Mrxnet.sys* para esconder archivos e inyectar código [39].

Luego de la instalación en uno de los computadores de la red, Stuxnet utilizaron las aplicaciones P2P, las carpetas compartidas, el *Print Spooler*, vulnerabilidades en el bloque de envío de mensajes de Windows Server (*Server Message Block - SMB*) y/o medios removibles para expandirse [41] [42].

Las entradas que fueron afectadas por Stuxnet son:

- Archivos:
 - Archivo DLL de gran tamaño que originalmente está empaquetado (pesa aproximadamente 1,18 MB)
 - ~WTR4141.tmp
 - Win32k.sys
 - MrxNet.sys
 - MrxCls.sys
 - %SystemRoot%\inf\oem7A.PNF
 - %SystemRoot%\inf\mdmcpq3.PNF
 - %SystemRoot%\inf\oem6C.PNF
 - %SystemRoot%\inf\mdmeric3.PNF
 - %Windows%\help\winmic.fts
 - %System%\winsta.exe
 - kerens32.dll
 - Kernel32.dll
 - Ntdll.dll
 - mdmcpq3dd.pnf
 - Archivos con extensión .LNK de tamaño 4171 bytes.

- Archivos de tipo ~WTRxxxx.tmp de 4kb a 8 Mb
- %Windir%\System32\shell32.dll
- jmidebs.sys
- s7apromx.dll
- mfc42.dll
- msvcrt.dll
- ccprojectmgr.exe
- cc_alg.sav
- db_log.sav
- cc_tag.sav.
- xutils\listen\xr000000.mdx
- xutils\links\s7p00001.dbf
- xutils\listen\s7000001.mdx
- %Temp%\~dfXXXX.tmp ubicado en la carpeta de archivos temporales
- lssas.exe (system process)
- avp.exe (Kaspersky)
- mcshield.exe (McAfee VirusScan)
- avguard.exe (AntiVir Personal Edition)
- bdagent.exe (BitDefender Switch Agent)
- UmxCfg.exe (eTrust Configuration Engine from Computer Associates International)
- fsdfwd.exe (F-Secure Anti-Virus suite)
- rtvscan.exe (Symantec Real Time Virus Scan service)
- ccSvcHst.exe (Symantec Service Framework)
- ekrn.exe (ESET Antivirus Service Process)
- tmproxy.exe (PC-cillin antivirus software from TrendMicro)
- Registros:
 - HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\MRxCls\”ImagePath” = “%System%\drivers\mrxccls.sys”
 - HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\MrxCls\Data
 - HKEY_LOCAL_MACHINE\Software\Siemens\Step7, value: STEP7_Version
 - HKEY_LOCAL_MACHINE\Software\Siemens\WinCC\Setup, value: Version
- Procesos:
 - Lsass.exe
 - Winlogon.exe
 - Svchost.exe
 - S7tgotpx.exe (Maneja el software Step 7)
 - CCProjectMgr.exe
- Servicios:
 - Print spooler

- s7tgotpx.exe
- services.exe
- RPC server
- Accesos a páginas:
 - www.mypremierfutbol.com
 - www.todaysfutbol.com

1.1.3 Duqu

De acuerdo con las investigaciones realizadas por el Laboratorio de Criptografía y Sistemas de Seguridad de la Universidad de Budapest (Crysys) [46] y Symantec [47], se puede decir que Duqu fue una amenaza precursora a Stuxnet, que tiene por objetivo recopilar datos de una víctima como parte del proceso de reconocimiento (o inteligencia), de manera que se facilite la realización de un ataque a éste o a un tercero. Su nombre se debe a que los archivos temporales creados por el *keylogger* estaban precedidos con la cadena de caracteres -DQ [46].

El ataque fue reportado por el Laboratorio de Criptografía y Sistemas de Seguridad de la Universidad de Budapest (Crysys) en el mes de octubre de 2011, indicando ser una amenaza similar a Stuxnet. El troyano fue expandido como un documento de Word adjunto a un correo electrónico [48].

Duqu estaba compuesto de los siguientes elementos:

- Archivo "*dropper*": Archivo de word que explotaba una vulnerabilidad del día cero de Windows.
- Instalador: Revisaba que el gusano haya quedado instalado en el equipo afectado. En caso de no estar infectado, procedía a instalar el gusano.
- Archivo Driver: Punto inicial en el que Duqu comenzaba la infección.
- Archivo DLL: Inyectaba su código en el servicio que carga los drivers.
- Archivo de configuración: Contenía las directivas de los archivos a afectar con determinada acción.
- *Keylogger*: Capturaba datos insertados por el usuario que pueden ser importantes pensando en atacar otra víctima.
- Aplicaciones anexas: Aplicaciones que eran descargadas de un servidor Command and Control (C&C server). Estas aplicaciones anexas eran el programa de robo de información, módulo de reconocimiento y el módulo de expansión de tiempo de vida del gusano.

Tanto el driver como el archivo DLL y el archivo de configuración han presentado variantes relacionadas con la eliminación de código, se presentaban en archivos nuevos y no se sobrescribían sobre los archivos existentes. La evolución de estos archivos se presenta en la siguiente tabla:

| Driver | DLL | Archivo de configuración |
|---------------|--------------|--------------------------|
| Jminet7.sys | Netp191.pnf | Netp192.pnf |
| Cmi4432.sys | Cmmi4432.pnf | Cmi4464.pnf |
| Nfred965.sys | Netf1.pnf | Netf2.pnf |
| Nred961.sys | Iddr021.pnf | |
| Adp55xx.sys | Ird182.pnf | |
| Adpu321.sys | Netq795.pnf | |
| Iastor451.sys | | |
| Allide1.sys | | |
| Iraid18.sys | | |
| Noname.sys | | |
| Igdkmd16b.sys | | |

Tabla 1. Evolución archivos componentes Duqu [40]

Duqu operaba de la siguiente manera [48]:

- Apenas se abría el documento adjunto de Word en un correo, se disparaba un *exploit* que primero revisaba si Duqu estaba instalado en el equipo afectado. Para esto, hacía uso del registro de Windows. Si el registro se encontraba, la ejecución del gusano terminaba. En caso contrario, se descriptaba el driver (Jminet7.sys) y a su vez, éste descriptaba el archivo que contenía el archivo DLL (Netp191.pnf) y se esperaba a que el equipo reiniciara. Apenas se cumplía el reinicio de la máquina, el archivo Netp191.dll se inyectaba sobre services.exe.
- Para comunicarse con el servidor C&C (Command and Control), desde el archivo Netp191.pnf se miraba si el equipo atacado estaba conectado a Internet, se inyectaba a sí mismo dentro del proceso que estaba manejando el navegador (puede

ser Explorer.exe, IExplore.exe o Firefox.exe) y en el producto de seguridad instalado (bien puede ser avp.exe, Mcshield.exe, avguard.exe, bdayent.exe, umxcfg.exe, fsdfwd.exe, rtvscan.exe, ccsvchst.exe, ekrn.exe, tmproxy.exe o ravmond.exe).

- Luego de las anteriores verificaciones, se extraía el netp191.zdata.mz que contenía el *payload* DLL que ejecutaba la conexión con el C&C server y junto con un archivo de configuración y otro archivo DLL completaban la operación mencionada utilizando el protocolo HTTP (puerto 80) o HTTPS (puerto 443) e iban dirigidas a las direcciones IP 206.183.111.97, 77.241.93.160, 123.30.137.117 ó a la dirección IP de un computador que ya estaba infectado. Para la comunicación con otro computador infectado, se utilizaba el protocolo HTTP.

Las entradas afectadas por el ataque Duqu fueron:

- Archivos:
 - Jminet7.sys
 - Cmi4432.sys
 - Nfred965.sys
 - Nred961.sys
 - Adp55xx.sys
 - Adpu321.sys
 - lastor451.sys
 - Allide1.sys
 - Iraid18.sys
 - Noname.sys
 - lgdkmd16b.sys
 - Netp191.pnf
 - Cmimi4432.pnf
 - Netf1.pnf
 - lddr021.pnf
 - lrd182.pnf
 - Netq795.pnf
 - Netp192.pnf
 - Cmi4464.pnf
 - Netf2.pnf
 - nep191_res302.dll
- Registros:
 - HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet Settings\Zones\4\“CF1D”
 - HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\JmiNET3

- HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\JmiNET3\FILTER
- Procesos:
 - lssas.exe (system process)
 - avp.exe (Kaspersky)
 - mcshield.exe (McAfee VirusScan)
 - avgguard.exe (AntiVir Personal Edition)
 - bdagent.exe (BitDefender Switch Agent)
 - UmxCfg.exe (eTrust Configuration Engine from Computer Associates International)
 - fsdfwd.exe (F-Secure Anti-Virus suite)
 - rtvscan.exe (Symantec Real Time Virus Scan service)
 - ccSvcHst.exe (Symantec Service Framework)
 - ekrn.exe (ESET Antivirus Service Process)
 - tmproxy.exe (PC-cillin antivirus software from TrendMicro)
 - ravmond.exe
 - Explorer.exe
 - IExplore.exe
 - Firefox.exe
 - Pccntmon.exe
 - netp191.zdata.mz
 - ntdll.dll
 - sort[random].nls
 - keylogger.exe
 - Archivo .tmp con el prefijo ~DQ en el nombre en la carpeta %TEMP%
- Servicios:
 - services.exe
 - svchost.exe
- Protocolos y puertos
 - HTTP (80)
 - HTTPS (443)
- Carpetas:
 - %TEMP%
 - %System%\Drivers
 - %SystemDrive%\inf\Netp191.dll
 - User/... /Appdata/Local/Temp
 - Documents and Settings/.../Local data/temp
- Accesos a páginas / direcciones IP:
 - 206.183.111.97
 - 77.241.93.160

- 123.30.137.117

1.2 Simulación de ataques investigados

Posteriormente, se realizó una serie de laboratorios haciendo uso de algunos vectores de ataque utilizados por Stuxnet, Duqu y Operación Aurora para infectar una serie de computadores e identificar qué elementos fueron infectados y que elementos se encontraban en las máquinas después de ser infectadas y que potencialmente tenían relación con alguno de los ataques investigados. En otras palabras, se recolectaron datos (carpetas, archivos, registros, etc.) en el mismo computador, pero en dos estados diferentes: no infectado (también conocido como estado seguro) e infectado (también conocido como estado inseguro).

La infraestructura utilizada para realizar los laboratorios consistió en simular una red LAN (con máquinas virtuales Windows 7 y Windows 10 montadas sobre VirtualBox) a la cual se conectaba un computador externo (máquina que trabajaba con el sistema Kali Linux la cual tiene la ventaja de tener un repositorio de vectores para simular un ataque real y algunas herramientas para hacer medidas).



Ilustración 8. Infraestructura utilizada para recolección de datos

Los vectores utilizados desde el computador atacante (Kali Linux) sobre los equipos víctimas fueron [16]:

- MS10-046: Vulnerabilidad utilizada por Duqu y Stuxnet para insertar archivos con extensión LNK en el computador de la víctima para utilizar rutas largas de las unidades de disco las cuales eran más difíciles de predecir en caso de tener que leer medios removibles. [40] [47]
- MS08-067: Vulnerabilidad utilizada por Stuxnet y Duqu para tratamiento incorrecto de las solicitudes RPC en la ejecución remota de código en el servicio de servidor de Windows causando sobrecarga en el buffer y así ejecutar código no deseado [42] [44]
- MS 10-002: Vulnerabilidad utilizada por Operación Aurora en donde se podía permitir la ejecución remota de código si un usuario de Internet Explorer visitaba una página web diseñada de manera especial por un atacante [38].
- Sobrecarga del búfer: Variante de la MS08-067.
- Persistent backdoor: Crea una puerta trasera ligada a varios procesos que se encuentran en ejecución. Si no hay procesos disponibles, se infecta el ejecutable de Notepad.

| | Stuxnet | Duqu | Aurora |
|---------------------|---------|------|--------|
| Persistent Backdoor | | | X |
| MS 10-046 | X | X | |
| MS 08-067 | X | X | |
| MS 10-002 | | | X |
| Sobrecarga búfer | | X | |

Tabla 2. Vectores de los ataques APT investigados y que fueron utilizados en los laboratorios para obtener datos para el modelamiento

Fase de entendimiento de los datos

Luego de ejecutar el proceso de recolección de datos entre lectura y laboratorios, se encontraron 438.750 datos entre archivos, registros, puertos, servicios y procesos que estaban operando antes de iniciar el vector de ataque (entorno seguro) y después de la respectiva ejecución (entorno inseguro). Basado en lo anterior, se formó una base de datos inicial compuesta de 85 columnas compuestas de:

- Dato capturado
- Tipo de dato: Archivo, puerto, registro o carpeta.
- Cadena de datos: Cadena que identifica el estado del dato antes y después de ejecutar el vector de ataque. Esta cadena se compone de 81 dígitos utilizados de la siguiente manera:
 - 0: Si el dato capturado no estaba en el estado seguro o después de ejecutar el vector.
 - 1: Si el dato capturado se encontraba en el estado seguro.
 - 2: Si el dato capturado se encontraba después de ejecutar el vector de ataque.
- 40 estados seguros: Se mira si el dato se encuentra en el sistema operativo antes de ejecutar el vector de ataque. Si se encuentra, toma el valor 1, si no se encuentra, toma el valor 0.
- 40 estados inseguros: Se mira si el dato se encuentra en el sistema operativo después de ejecutar el vector de ataque. Si se encuentra, toma el valor 2, si no se encuentra, toma el valor 0.
- Columna que incluye los datos encontrados en las lecturas.
- Variable “Atacado o no”: Variable categórica dicotómica que toma los valores Si o No y que hace de variable dependiente de la base de datos. Consiste en ver si el dato observado guarda alguna relación con los datos identificados como atacados en la investigación bibliográfica de la tarea anterior. Si existe una relación (Por ejemplo, que en un archivo o una llave del registro de Windows tenga una ruta muy similar a la de un dato infectado por un ataque tipo

más reducida, pero sin perder la información que la base de datos original contenía. Para esto, se recurrió a las técnicas de reducción de dimensionalidad. Las etapas que se efectuaron fueron:

1.2.1 Familias de datos

Esta primera etapa se realizó con la intención de reunir los registros de la base de datos en grupos basados en la similitud de su estructura en el campo “Cadena de datos”. Esto significa que se crearon familias de datos según su presencia antes y después de ejecutar los vectores de ataque. Al realizar esta tarea, se obtuvieron 403 familias reduciendo la base de datos original (que era de 438.750 x 85) a una de 403 x 85.

1.2.2 Análisis factorial

La siguiente tarea de esta fase consistió en la aplicación del análisis factorial que consiste en formar grupos homogéneos de variables (llamados factores) a partir de un puñado de éstas que se correlacionan entre sí procurando independencia entre ellos. Su propósito es encontrar un número mínimo de factores que expliquen la mayor cantidad posible de información contenida en los datos originales [49].

En líneas generales, el análisis factorial se trata de descomponer el problema de reducir la dimensionalidad de un conjunto de datos en dos partes [50]:

$$\begin{aligned} \text{Varianza total} &= \text{Varianza común o compartida} \\ &+ \text{Varianza debido a errores de medición} \end{aligned}$$

Ecuación 1. Varianza en Análisis Factorial

Matemáticamente, lo anterior se puede expresar en el hecho que se quiere traducir p variables originales del estudio x_1, \dots, x_p en función de un número m factores comunes $m < p$ (que explica la varianza común entre las variables originales), más un factor específico o único (varianza debido a los errores de medición) [51], dejando una serie de ecuaciones de la forma [49]:

$$\begin{aligned} x_1 &= a_{11}F_1 + a_{12}F_2 + \dots + a_{1m}F_m + e_1 \\ x_2 &= a_{21}F_1 + a_{22}F_2 + \dots + a_{2m}F_m + e_2 \\ &\dots \\ x_p &= a_{p1}F_1 + a_{p2}F_2 + \dots + a_{pm}F_m + e_p \end{aligned}$$

Ecuación 2. Sistema de ecuaciones Análisis factorial

Donde X es el vector de las variables originales, a es la correlación entre la variable p y el factor m , F es el vector de factores comunes y e es el vector de factores únicos.

El requisito fundamental para aplicar esta técnica está en que el conjunto de variables utilizado debe generar factores que sean ortogonales porque garantizan la independencia entre ellos. En otras palabras, lo que se pretende conseguir es que el primer

factor sea independiente del segundo, este último del tercero y así sucesivamente [52].

Para ejecutar el análisis factorial, se siguen los siguientes pasos:

- **Elaborar una matriz de correlaciones:** Se obtiene una matriz en la que se ubican las correlaciones entre todas las variables consideradas. Para comprobar que la matriz obtenida tiene la correlación apropiada, se pueden aplicar métodos como el determinante de la matriz de correlaciones, el test de esfericidad de Bartlett, el índice Kaiser – Meyer – Olkin, coeficiente de correlación parcial, coeficiente de correlación anti-imagen o también la diagonal de la matriz de correlación anti-imagen [49].
- **Extraer factores iniciales:** Para este fin, se utilizan métodos como Análisis de Componentes Principales (tradicionalmente es el más utilizado), ejes principales, mínimos cuadrados no ponderados, mínimos cuadrados generalizados, factorización por imágenes o método alfa [52].
- **Ejecutar una rotación a los factores iniciales extraídos:** Este proceso se realiza porque es complejo hacer una interpretación con matrices de cargas factoriales que no son adecuadas. Para esto, se aplican métodos como Varimax (Minimiza el número de variables con cargas altas en un factor, mejorando así la interpretación de factores), Quartimax (Tiene como objetivo tener correlaciones elevadas en cada variable con pocos factores. Para eso, se maximiza la varianza de las cargas factoriales al cuadrado de cada variable en los factores), Equamax (híbrido de las dos anteriores donde se intenta hacer una optimización por variables y por factores simultáneamente), Oblimin (Minimiza la interpretabilidad de los factores y qué tan ortogonales son) y Promax (Altera el resultado de una rotación hasta crear una solución con cargas factoriales cercana a la estructura ideal) [52].
- **Obtención de puntuaciones factoriales:** Basado en el nuevo conjunto de variables obtenido, se mira qué valores toma cada individuo o unidad experimental en cada uno de los factores. Para esto, se utilizan métodos como el de regresión, Bartlett o Anderson-Rubin [52].
- **Validación:** Para validar el modelo obtenido, se analiza la bondad de ajuste (diferencias entre las correlaciones de la matriz de entrada y la matriz factorial) y la generalidad de los resultados (Comprobar los resultados del primer análisis factorial realizando nuevos análisis factoriales sobre nuevas muestras extraídas de la población) [52].

En la fase de extracción de factores iniciales, se utilizó el método de Análisis de Componentes Principales (PCA) que consiste en condensar la información que se describe en un conjunto de variables mediante un grupo más reducido que son combinaciones lineales del primero. La idea de este método considera que si existe una pérdida de información, sea la menor posible e incluso, su resultado se pueda representar en gráficas [51] [53].

Luego de aplicar todas las fases del análisis factorial, se obtuvo una matriz de 65 factores que indicaban la relación sobre cada familia de datos por 81 filas.

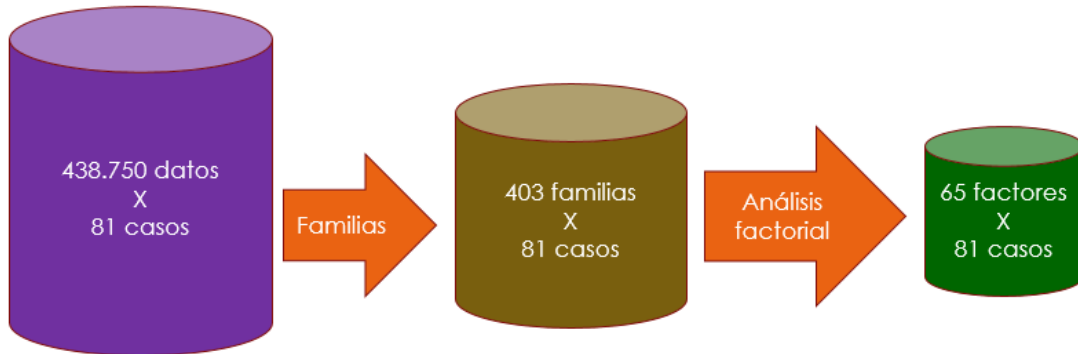


Ilustración 11. Proceso de reducción bases de datos

1.3 Algoritmos aplicables

Para encontrar el mecanismo para hacer determinaciones, se necesitaba que dicho elemento cumpla con los siguientes requerimientos:

- Capacidad de clasificar los casos nuevos en un grupo de riesgo (Si es ataque o No es ataque).
- Clasificar un nuevo caso en una variable categórica dependiente sabiendo que su valor es dicotómico (Si es ataque o No).
- Trabajar con la posibilidad de ruido generado por casos nuevos.

Las principales alternativas que se tuvieron en cuenta fueron:

- **Regresión logística:** El modelo de regresión logística, también conocida como discriminación logística, se utiliza para resolver problemas de clasificación en modelos de predicción para variables categóricas (por ejemplo, mucho, poco o nada), especialmente si son dicotómicas (si-no, A-B, etc.). Su objetivo es estimar la probabilidad de ocurrencia de un suceso o hecho dicotómico y_j (variable dependiente) basado en los factores que le pueden afectar (variables independientes) y que están catalogados como $X = (x_1, x_2, \dots, x_n)$. Por cada evento, la suma de todas estas probabilidades igual a 1. La regresión logística forma parte de los modelos de probabilidad no lineal que se caracterizan en que la función utilizada para predecir esté entre los valores comprendidos entre 0 y 1 [50].

Para un hecho nuevo que se quiera determinar la ocurrencia que puede tomar un valor 1 o un valor 0, se tienen unas probabilidades determinadas: si la ocurrencia es del primer tipo mencionado, la probabilidad sería de $p(y = 1|X)$, y la probabilidad para el segundo evento sería de $p(y = 0|X) = 1 - p(x)$.

La ecuación equivalente a suponer las probabilidades de un evento para un evento A o para su contrario B (conociendo que se trata el problema como

solución dicotómica, es decir, solo tomará uno de dos valores) ambas en función de las variables X es [54]:

$$p(x) = \frac{e^{\beta_0 + \beta'x}}{1 + e^{\beta_0 + \beta'x}}$$

Ecuación 3. Probabilidad de ocurrencia ante evento 1 en un modelo logístico con solución dicotómica

Un elemento importante en los modelos de regresión logística es el “ratio odd” o ratio de probabilidades que indica la preferencia que ocurra un hecho (por ejemplo, puede tomar el valor A) frente a que no ocurra (puede tomar el valor B). Matemáticamente, se entiende como el cociente entre las probabilidades de ocurrencia de los únicos dos valores que puede tomar la variable dependiente [51]:

$$O_i = \frac{p_i}{1 - p_i}$$

Ecuación 4. Ratio de probabilidades de eventos [51]

El ratio de probabilidad tiene un valor más significativo cuando se comparan las variables dependientes o explicativas frente a los eventos u observaciones que posee el caso. Esto facilita la interpretación del modelo según la variación que pueda presentar alguno de sus parámetros e inferir a qué tipo de eventos le podría beneficiar estos cambios respecto a otros. Matemáticamente, esto se explica bajo la siguiente ecuación:

$$e^{\beta h} = \frac{O_i}{O_j}$$

Ecuación 5. Cociente de ratio de probabilidades entre dos eventos [51]

- **Redes neuronales:** Es una técnica que se caracteriza por tener las habilidades de los algoritmos de aprendizaje y clasificación. Se compone de un grupo de neuronas donde cada una de ellas tiene un conjunto de entradas (datos que recibe) y una salida generada a partir de la implementación de una función con parámetros ajustables. La característica principal de las redes neuronales radica en que son algoritmos de caja negra, es decir, son aquellos que al principio se entregan unas variables y al final devuelve un resultado sin indicar cómo llegó al valor dado [55].

Cada una de las neuronas se va calibrando automáticamente, haciendo que los datos de entrada tomen mayor importancia ya que entre más datos se tengan, habrá mayor calidad en el entrenamiento de las neuronas y pueden reflejar resultados de manera más fiable a la realidad. Una vez entrenada la red, se emplean los datos originales para ser clasificados [55].

Este algoritmo posee variaciones como:

- *Backpropagation*: Este mecanismo funciona a partir del aprendizaje de los ejemplos y calibrar continuamente los pesos de las neuronas de manera que al terminar el entrenamiento y comience a funcionar con datos reales se puedan obtener salidas o resultados confiables. Al iniciar el proceso de entrenamiento, la tasa de errores será muy grande, pero a medida que se calibre la red, se obtendrán errores cada vez más pequeños hasta llegar a un valor que ya no tendrá variación o será insignificante [50].
- Aprendizaje supervisado: Es un aprendizaje iterativo que se utiliza para realizar tareas muy similares y recibir una retroalimentación frecuente y detallada. En cada iteración, la neurona va aprendiendo los datos de entrada si son correctos o no y la acción que debe ejecutar. La idea bajo este tipo de algoritmo es aprender a predecir acciones futuras basado en una comparación con ejemplos ya conocidos. Cualquier discrepancia entre el caso de estudio y los ejemplos ya estudiados se asume como el error para modificar el sistema, ajustar las neuronas y a la siguiente iteración, volver a realizar el cálculo [50].
- Máquinas de soporte vectorial (SVM): Es una técnica de clasificación formulada entre Vapnik y Cortes en 1995 basada en el uso de hiperplanos en espacios de muy alta dimensionalidad [56]. Este problema de clasificación consiste en tener un conjunto de n datos $S = (x_1, y_1), \dots, (x_n, y_n)$, donde cada x_i forma parte de los casos que se desean estudiar y que pertenecen a un espacio de entrada X (también X se conoce como vector), y cada y_i pertenece a $\{-1, 1\}$ e indica la categoría que pertenece x_i .

Los objetivos que tienen las SVM son:

- Encontrar un hiperplano de separación óptimo en un contexto no paramétrico. Un hiperplano en un espacio D -dimensional se puede expresar por medio de la ecuación $h(x) = \langle w, x \rangle + b$, donde w es el vector de pesos donde cada valor indica la importancia o contribución a la regla de clasificación, $\langle w, x \rangle$ es un producto escalar y b es el sesgo o *bías* el cual define el umbral de decisión. Para hallar el mejor hiperplano o aquel que se considera como óptimo, se tiene que recurrir a tres conceptos [51]:
 - Margen máximo que consiste en elegir aquel hiperplano que esté a la misma distancia de los ejemplos más cercanos de cada una de las categorías (es decir, encuentre un punto neutro entre las categorías sin importar si una es más numerosa que la otra).
 - Margen geométrico (γ) que consiste en la maximización de la distancia mínima entre los ejemplos del conjunto de datos y el hiperplano.
 - Vectores soporte que delimita al hiperplano respecto a los puntos más cercanos a cada una de las categorías.
- Clasificar un nuevo dato a partir de la posición del hiperplano.

- Elegir una función núcleo o función kernel que permita mapear los datos obtenidos a un espacio de características más alto en términos dimensionales, de manera que se pueda obtener una mayor separación posible entre las categorías [50].

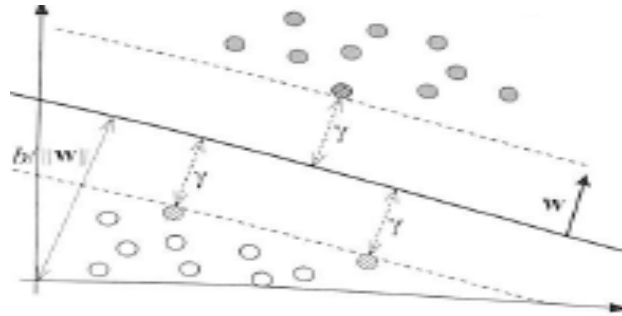


Ilustración 12. Trazo de hiperplano para delimitar datos entre dos categorías [50]

La regla de clasificación se puede expresar como $f(x) = \text{signo}(h(x))$, donde la función signo toma un valor 1 si $x \geq 0$, o puede tomar el valor -1 si $x < 0$. Basado en lo anterior, se pretende encontrar un hiperplano que separe las categorías que se forman con los puntos de X y cumplir la condición dada por la ecuación $(w \cdot z) + b = 0$ [50].

Para los casos de clasificación, las SVM maneja dos casos:

- Casos linealmente separables:

También conocido como SVM de margen máximo, parte de la hipótesis que consiste en que los casos de pueden ser separados por un hiperplano. Para conseguirlo, se mapea el espacio de las variables X a un espacio de mayor dimensionalidad, llamado $z = \varphi(x)$, y se busca el hiperplano óptimo que permita hacer la división entre los x_i casos de la siguiente manera [56]:

$$f(x_i) = \text{signo}((w \cdot z) + b) = \begin{cases} 1 & y_i = 1 \\ -1 & y_i = -1 \end{cases}$$

Ecuación 6. Clasificación casos linealmente separables en una SVM

Sin embargo, este caso presenta las siguientes restricciones [50]:

- La naturaleza del clasificador: Como es lineal, no representa de manera ideal muchos problemas.
- Se necesita que el conjunto de datos sea linealmente separable, lo cual no tiene darse siempre y tampoco es fácil de conseguir. De hecho, la existencia de ruido hace que el problema no conciba como linealmente separable o tampoco conviene manejarlo de esa manera.

- Casos linealmente no separables (o de margen blando):

Los casos linealmente no separables parten del hecho que no todos los casos reales se pueden separar con una línea recta. Por tal motivo, para hacer más tolerante los errores que pueden darse en la clasificación, se utilizó una variable de holgura ξ y un parámetro de regularización C aplicado al vector de pesos w , haciendo que el problema del hiperplano óptimo se redefina a minimizar $\frac{1}{2}w \cdot w + C \sum_1^n \xi_i$ sujeto a $y_i(w \cdot z_i + b) \geq 1 - \xi_i \quad i = 1 \dots n$ [50] [57].

Fase de evaluación

Inicialmente, para comprobar que sí fue necesario utilizar la fase de reducción de dimensionalidad, se intentó hacer las labores de clasificación entre la base de datos original y la base de datos factorial utilizando la herramienta estadística Weka [58]. El resultado de la lectura de ambas bases de datos se resume en la siguiente tabla:

| Base de datos utilizada | Tiempo de lectura de la base de datos |
|--|---------------------------------------|
| Base de datos original (438.750x85) | No terminó |
| Base de datos puntos factoriales (81x65) | 2 segundos |

Tabla 3. Comparación tiempos de lectura bases de datos obtenidas

Como la base de datos original ni siquiera pudo terminar de ser leída, no se podían ejecutar labores de clasificación ni de predicción. Por lo tanto, sí fue necesario aplicar las labores de reducción de dimensionalidad.

De los algoritmos seleccionados en la fase anterior, se volvió a utilizar la herramienta Weka sobre la base de datos de puntos factoriales con el fin de identificar cuál de ellos tiene el mejor índice de clasificación (instancias correctamente clasificadas en Weka). Al ejecutar cada uno de los algoritmos se obtuvieron los siguientes resultados:

| Algoritmo | Potencial de clasificación calculado |
|---------------------|--------------------------------------|
| Redes neuronales | 75% |
| SVM | 75% |
| Regresión logística | 50,6% |

Tabla 4. Comparación potencial de instancias correctamente clasificadas por cada algoritmo

Como se puede evidenciar, hay una igualdad entre las redes neuronales y la máquina de soporte vectorial. Para encontrar un desempate, se analizaron los índices de error que posee cada algoritmo:

| Algoritmo | Error medio absoluto | Error medio cuadrado | Error absoluto relativo |
|------------------|----------------------|----------------------|-------------------------|
| Redes neuronales | 0,263 | 0,4708 | 52,35% |
| SVM | 0,25 | 0,5 | 49,76% |

Tabla 5. Comparación de errores redes neuronales y SVM

Basado en los anteriores resultados, se confirma que el algoritmo que se utilizará para clasificar casos nuevos es la máquina de soporte vectorial.

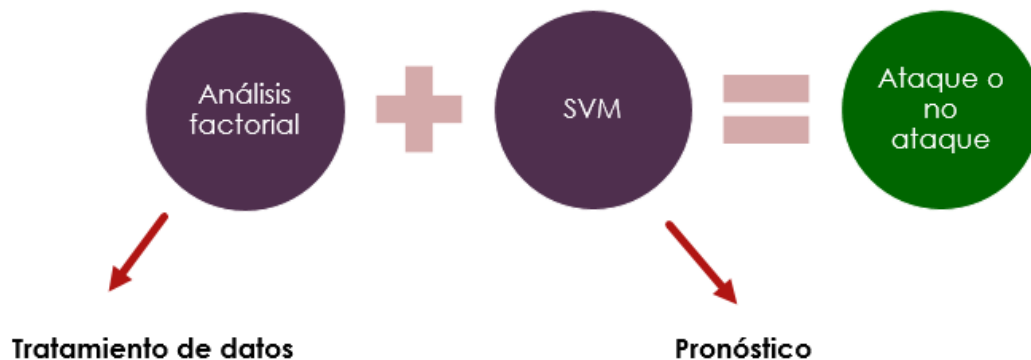


Ilustración 13. Algoritmos utilizados tras obtención, tratamiento de datos y evaluación de algoritmos para pronósticos

Fase de diseño y desarrollo del prototipo

1.4 Diseño del prototipo

Para representar la arquitectura, se emplea la descripción creada por Philippe Kruchten: "4 + 1 Vistas", que basado en cinco vistas concurrentes se resalta un elemento o relación diferente del sistema. Según Kruchten [59], las vistas en que aparece en la herramienta son :

- Vista de Casos de uso o escenarios: Describe el conjunto de casos de uso que representan las abstracciones de los requerimientos más relevantes. Los casos de uso utilizados para el prototipo son:
 - Consultar historial de actividades anómalas.
 - Almacenar cambios de comportamientos anómalos en el historial.
 - Desplegar cambios de comportamientos anómalos.
 - Comunicar cambios de comportamiento anómalos.

- Vista Lógica: Esta vista se encarga de explicar los requerimientos funcionales, es decir, lo que el sistema debe brindar en términos de servicios a los usuarios.

Para efectos del prototipo, se utiliza un estilo arquitectónico cliente-servidor, que se caracteriza por la interacción que un usuario, estando como cliente, tiene con otra unidad de cómputo (computador central o servidor) que le proporciona uno o varios servicios. Después de esta interacción, el computador que hace el rol de cliente recibe los resultados y los presenta al usuario final. Los dos roles están separados de la siguiente manera:

- Rol servidor:
 - Almacenar de actividades anómalas: Todo cambio de comportamiento que se registre en cualquiera de los computadores de la red LAN se almacena en un log de actividades general o global que tendrá el servidor.

- Rol cliente:
 - Detectar actividades anómalas:
 - Con ayuda de la máquina de soporte vectorial (SVM) se determina si el cambio de comportamiento captado sobre algún elemento del sistema operativo (archivo, registro, paquete de red, carpeta, etc.) que forma parte del sistema es una posibilidad de que sea indicio de un ataque.
 - Evaluando el crecimiento de archivos en un periodo determinado: si el crecimiento de algún archivo es exagerado (crece más del 50% de su tamaño), se emite la alerta de posibilidad de ataque.

En caso de detectar comportamiento anómalo por cualquiera de las dos vías, la novedad se guarda en el historial local, se notifica al servidor y también se guarda registro en él.

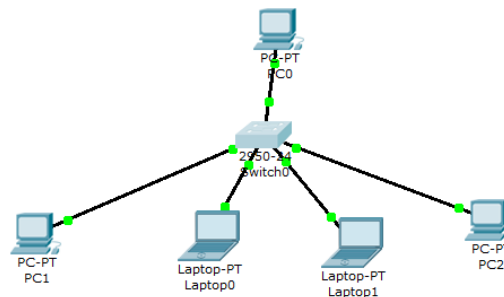


Ilustración 14. Topología diseñada como sistema para implementar el prototipo a construir

- Vista de Procesos: Toma en cuenta cómo las abstracciones principales obtenidas en la vista lógica se comunican entre sí en la arquitectura de la aplicación [60].

Para el caso de la herramienta, se especificarán los hilos que son creados por la aplicación para el envío y la recepción de alertas: Apenas se instale la herramienta en un equipo de la red LAN (cliente), el servidor establecerá un hilo que realice la comunicación con el cliente y quedará en estado "pendiente" escuchando si hay alguna alerta que se emita. Cuando el cliente haya emitido alguna alerta, enviará el mensaje al servidor que también desplegará en su máquina el evento anómalo.

- Vista Física: Tiene en cuenta los requerimientos no funcionales de primer tipo, es decir, disponibilidad, desempeño, confiabilidad y escalabilidad.

Para efectos del prototipo a construir, éste será diseñado como una aplicación que tiene alcance de una red LAN. Teniendo esto presente, se necesita que los computadores que forman parte de la red estén conectados vía alámbrica por cable UTP y posean una tarjeta Gigabit Ethernet 10/100/1000 con conector RJ45. Desde el punto de vista gráfico, se necesitaría tarjeta de mediana calidad (128 Mb) aunque la herramienta no requiere de un nivel alto en este aspecto.

- Vista de desarrollo: Describe los módulos principales de la aplicación mostrando las clases principales y sus asociaciones. Esta vista se representa a través de diagramas de componentes:

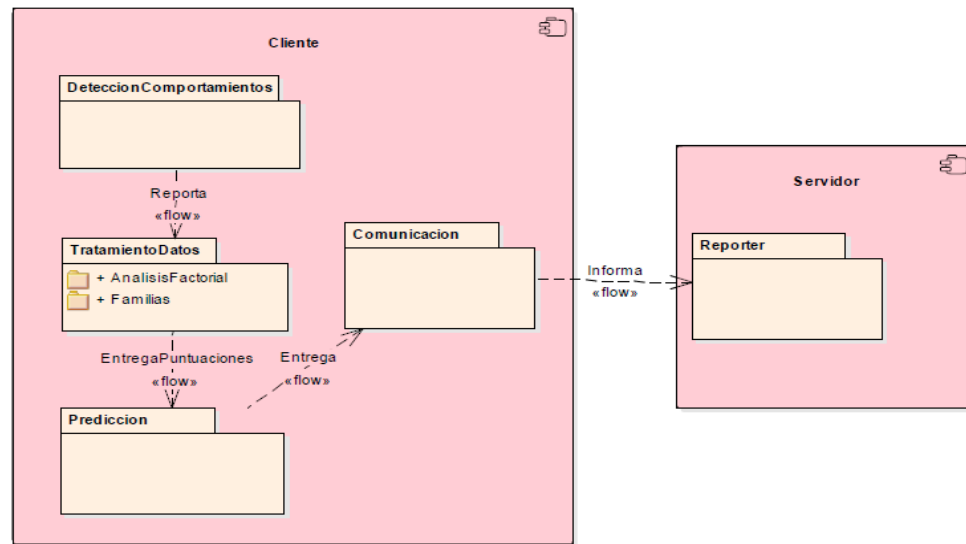


Ilustración 15. Diagrama de componentes prototipo BWCare

1.5 Fase de implementación

A continuación, se procede a implementar la herramienta. Se sigue el paradigma de programación Orientada a Objetos y el lenguaje de programación es Java debido a:

- El manejo del módulo de comunicación entre equipos tiene más variantes para programar (se puede utilizar sockets o RMI)
- Hay paquetes que ya tienen implementado el algoritmo de máquina de soporte vectorial e incluso se puede probar en ellos directamente.
- Posibilidad de extender a otros sistemas operativos como Linux o Unix e incluso, a Android.

El paquete seleccionado para hacer la ejecución del algoritmo en su fase de pronóstico es Weka el cual, como también está construido en Java, facilita la conexión con el prototipo.

La metodología propuesta para el desarrollo de la herramienta fue la espiral de Boehm, que consiste en la elaboración de las siguientes actividades [60]:

- Establecimiento de objetivos: Se definen las metas a cumplir para la fase, se identifican las restricciones, riesgos y alternativas ante estos riesgos.
- Evaluación de riesgos: Por cada riesgo se hace un análisis y se establece los pasos para establecer la acción requerida (mitigar, aceptar, transferir o eliminar)
- Desarrollo y validación: Se hace el desarrollo basado en las decisiones tomadas en el diseño de la herramienta y en los riesgos analizados.
- Planeación: Se toman decisiones sobre la realización del siguiente ciclo.

La metodología de la espiral de Boehm se eligió debido a que se necesita hacer una planeación de cuáles son los elementos que se requieren para adaptar la aplicación a los eventos que se desean detectar. Además, existe la posibilidad de situaciones donde se presenten dificultades en el diseño porque se requiere monitorear recursos sensibles del sistema operativo y posiblemente no se tenga mucha facilidad en esta labor (por tema de permisos); por tanto, resulta importante estimar el impacto y la probabilidad frente al desarrollo del prototipo para tomar las decisiones respectivas criterios más claros. Por cuestión de tiempo para la elaboración del proyecto, sólo se ejecutó un ciclo de esta espiral.

Es muy importante tener en cuenta que, para efectos del proyecto, se asume que el prototipo de la herramienta será instalado en una situación libre de ataques tipo APT.

1.6 Riesgos

En la ejecución de la metodología de desarrollo mencionada, se realizará una valoración de los riesgos que se pueden percibir sobre el funcionamiento de la aplicación. Durante las fases de obtención de requerimientos, diseño e implementación se lograron ver algunos que impedirían el buen funcionamiento del prototipo. Los riesgos más importantes que fueron identificados son:

Riesgo 1: Ingreso de formatos de bases de datos diferentes al solicitado

Riesgo 2: Carpeta o archivo que no pueda ser accedido por permisos de acceso insuficientes

Riesgo 3: Versión de Java obsoleta

Riesgo 4: No visualización del historial de cambios anómalos.

Para evaluar los riesgos inherentes, se realizará por impacto y por probabilidad de ocurrencia. La escala para cada categoría se organiza de la siguiente manera:

Impacto: insignificante (1), menor (2), intermedio (3), mayor (4), catastrófico (5).

Probabilidad: Raro (1), poco probable (2), moderado (3), probable (4), frecuente (5).

La calificación del riesgo inherente proveniente de relación impacto/probabilidad viene dado por el siguiente mapa de calor [61]:

| | | | | | | |
|---------|--------------------|---------------------|----------------------|-----------------|-----------------|------------------|
| Impacto | Catastrófico (5) | | | | | |
| | Mayor (4) | | | | | |
| | Intermedio (3) | | | | | |
| | Menor (2) | | | | | |
| | Insignificante (1) | | | | | |
| | | Raro (1) | Poco probable (2) | Moderado (3) | Probable (4) | Frecuente (5) |
| | | Probabilidad | | | | |

Ilustración 16. Mapa de calor utilizado para calificar riesgos del prototipo de la aplicación

Las valoraciones de los riesgos inherentes vienen dadas por los siguientes colores:

Verde claro: Riesgo muy bajo

Verde: Riesgo bajo

Amarillo: Riesgo medio

Naranja: Riesgo alto

Rojo: Riesgo grave

Basados en las escalas anteriores, las valoraciones de los riesgos obtenidos son:

| # riesgo | Impacto | Probabilidad | Riesgo inherente | Medida a tomar |
|----------|---------|--------------|------------------|--|
| 1 | 5 | 2 | Alto | Mitigar: Presentar una copia de la BD junto a la aplicación y el archivo donde están las entradas nuevas dejarlo con los campos necesarios |
| 2 | 2 | 3 | Bajo | Aceptar: La carpeta no se accede y se sigue analizando el resto de carpetas. |
| 3 | 4 | 2 | Medio | Mitigar: Se hace la instalación del JRE mínimo versión 1.7 |
| 4 | 5 | 2 | Alto | Mitigar: Crear nuevamente archivo con el historial de actividades. |

Tabla 6. Tabla estimación riesgos iniciales para la implementación del prototipo

1.7 Pasos para determinar si un dato que refleja un cambio de comportamiento es señal de un ataque o no

- El dato se compara con cada una de las entradas de la base de datos original (438.750 entradas): Si el dato es similar al que se está comparando, del dato asociado toma el valor de 1, caso contrario toma el valor de 0. Independientemente de si toma el valor 1 o 0 por cada familia, se hacen las 438.750 comparaciones. Salida: Vector de 403 casillas con 1s y 0s.
- Al vector resultante se le aplica regresión lineal en los 65 factores de la base de datos de factores. Salida: Vector de 65 posiciones con las puntuaciones factoriales.

- Se efectúa el cálculo de la máquina de soporte vectorial con las 65 puntuaciones factoriales. Salida: 0 no es ataque, 1 sí es ataque.

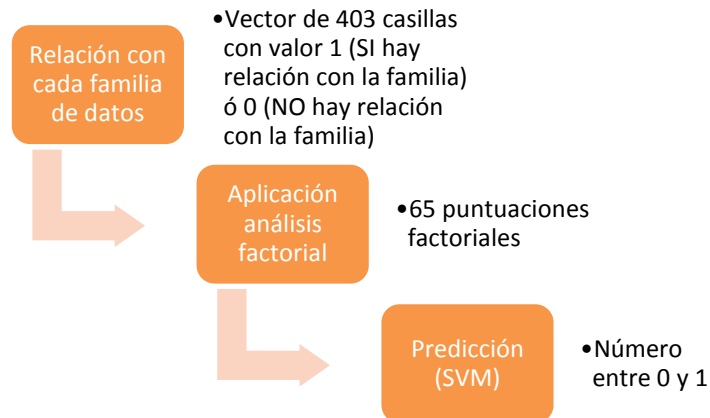


Ilustración 17. Pasos que se ejecutan en nuevo caso para determinar si es señal de ataque o no

Pruebas del prototipo

Los objetivos de las pruebas son:

- Analizar la usabilidad de la herramienta.
- Comprobar si la aplicación es congruente con los cambios de comportamiento dados por ataques tipo Amenazas Persistentes Avanzadas (APT) según los escenarios dados para este ejercicio.
- Conocer potenciales fortalezas y debilidades de la aplicación.

Para el desarrollo de las pruebas, se pueden ejecutar en un computador con las siguientes características:

- Computador 1:
 - Sistema operativo: Windows 7 Profesional
 - Memoria: 1 GB
 - Procesador: Intel Core i3
 - Java Runtime Environment (JRE) versión 1.7
- Computador 2:
 - Sistema operativo: Windows XP SP 2
 - Memoria: 1 GB
 - Procesador: Intel Core i3
 - Java Runtime Environment (JRE) versión 1.7

El desarrollo de las pruebas constará de tres actividades:

1. Pruebas básicas: Prueba en donde manualmente se harán las siguientes operaciones basado en la investigación de ataques tipo Amenaza Persistente Avanzada (APT), más específicamente, Stuxnet, Duqu y Operación Aurora:

- a. Adición de archivos
 - b. Eliminación de archivos
 - c. Inflado de archivos
 - d. Creación de carpetas.
2. Prueba de detección de cambios de comportamiento según un ataque tipo APT. Este procedimiento se hará con uno de los vectores del ataque Aurora implementado en el sistema Kali.
 3. Prueba de detección de cambios de comportamiento utilizando el programa McAfee Evader.

IV – RESULTADOS OBTENIDOS

A continuación, se presentan los resultados que se obtuvieron al terminar el prototipo de la herramienta propuesta y de seguir el proceso anteriormente mencionado.

1. Prototipo desarrollado

El prototipo de la herramienta tendrá como nombre “BWCare” y está desarrollada para funcionar en un ambiente cliente-servidor sobre los sistemas operativos Windows 7 y 10. La aplicación en ambas versiones se ejecuta con permisos de Administrador.

1.1 Prototipo modo cliente:

El archivo ejecutable del prototipo en su versión cliente tiene por nombre BWCare.exe desde el cual se ingresará a la aplicación.

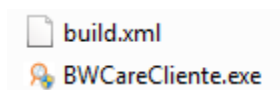


Ilustración 18. Ubicación archivo ejecutable herramienta BWCare

Al iniciarse, se presentará una ventana de diálogo solicitando la dirección IP del servidor. Luego de digitar este dato, se presentará la ventana principal:

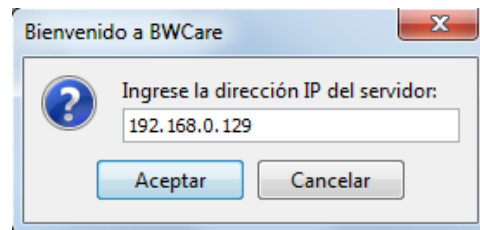
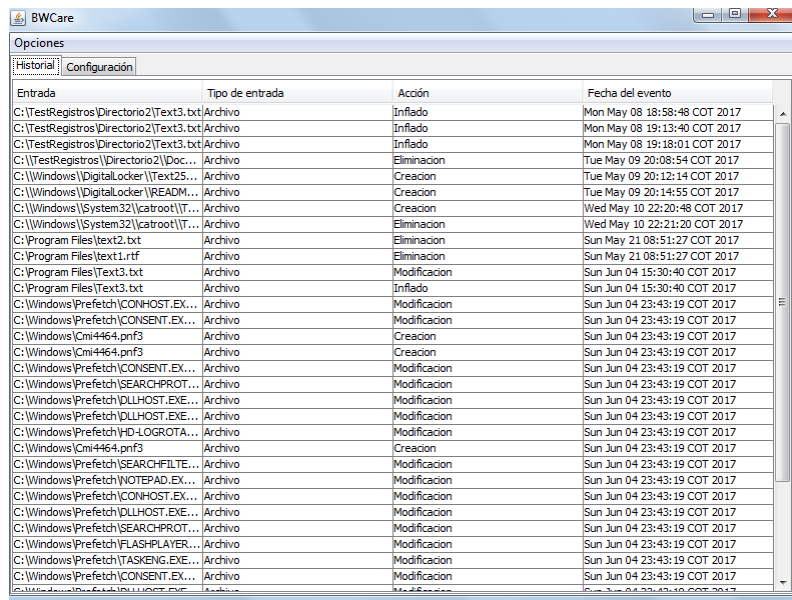


Ilustración 19. Conexión herramienta BWCare cliente con servidor

En la ventana principal se presentarán dos pestañas:



| Entrada | Tipo de entrada | Acción | Fecha del evento |
|--|-----------------|--------------|------------------------------|
| C:\TestRegistros\Directorio2\Text3.txt | Archivo | Inflado | Mon May 08 18:58:48 COT 2017 |
| C:\TestRegistros\Directorio2\Text3.txt | Archivo | Inflado | Mon May 08 19:13:40 COT 2017 |
| C:\TestRegistros\Directorio2\Text3.txt | Archivo | Inflado | Mon May 08 19:18:01 COT 2017 |
| C:\TestRegistros\Directorio2\Doc... | Archivo | Eliminacion | Tue May 09 20:08:54 COT 2017 |
| C:\Windows\DigitalLocker\Text25... | Archivo | Creacion | Tue May 09 20:12:14 COT 2017 |
| C:\Windows\DigitalLocker\READM... | Archivo | Creacion | Tue May 09 20:14:55 COT 2017 |
| C:\Windows\System32\catroot\T... | Archivo | Creacion | Wed May 10 22:20:48 COT 2017 |
| C:\Windows\System32\catroot\T... | Archivo | Eliminacion | Wed May 10 22:21:20 COT 2017 |
| C:\Program Files\text2.txt | Archivo | Eliminacion | Sun May 21 08:51:27 COT 2017 |
| C:\Program Files\text1.rtf | Archivo | Eliminacion | Sun May 21 08:51:27 COT 2017 |
| C:\Program Files\Text3.txt | Archivo | Modificacion | Sun Jun 04 15:30:40 COT 2017 |
| C:\Program Files\Text3.txt | Archivo | Inflado | Sun Jun 04 15:30:40 COT 2017 |
| C:\Windows\Prefetch\CONHOST.EX... | Archivo | Modificacion | Sun Jun 04 23:43:19 COT 2017 |
| C:\Windows\Prefetch\CONSENT.EX... | Archivo | Modificacion | Sun Jun 04 23:43:19 COT 2017 |
| C:\Windows\Cmi4464.pnf3 | Archivo | Creacion | Sun Jun 04 23:43:19 COT 2017 |
| C:\Windows\Cmi4464.pnf3 | Archivo | Creacion | Sun Jun 04 23:43:19 COT 2017 |
| C:\Windows\Prefetch\CONSENT.EX... | Archivo | Modificacion | Sun Jun 04 23:43:19 COT 2017 |
| C:\Windows\Prefetch\SEARCHPROT... | Archivo | Modificacion | Sun Jun 04 23:43:19 COT 2017 |
| C:\Windows\Prefetch\DLHOST.EXE... | Archivo | Modificacion | Sun Jun 04 23:43:19 COT 2017 |
| C:\Windows\Prefetch\DLHOST.EXE... | Archivo | Modificacion | Sun Jun 04 23:43:19 COT 2017 |
| C:\Windows\Prefetch\DLHOST.EXE... | Archivo | Modificacion | Sun Jun 04 23:43:19 COT 2017 |
| C:\Windows\Prefetch\HD-LOGROTA... | Archivo | Modificacion | Sun Jun 04 23:43:19 COT 2017 |
| C:\Windows\Cmi4464.pnf3 | Archivo | Creacion | Sun Jun 04 23:43:19 COT 2017 |
| C:\Windows\Prefetch\SEARCHFILTE... | Archivo | Modificacion | Sun Jun 04 23:43:19 COT 2017 |
| C:\Windows\Prefetch\NOTEPAD.EX... | Archivo | Modificacion | Sun Jun 04 23:43:19 COT 2017 |
| C:\Windows\Prefetch\CONHOST.EX... | Archivo | Modificacion | Sun Jun 04 23:43:19 COT 2017 |
| C:\Windows\Prefetch\DLHOST.EXE... | Archivo | Modificacion | Sun Jun 04 23:43:19 COT 2017 |
| C:\Windows\Prefetch\SEARCHPROT... | Archivo | Modificacion | Sun Jun 04 23:43:19 COT 2017 |
| C:\Windows\Prefetch\FLASHPLAYER... | Archivo | Modificacion | Sun Jun 04 23:43:19 COT 2017 |
| C:\Windows\Prefetch\TASKENG.EXE... | Archivo | Modificacion | Sun Jun 04 23:43:19 COT 2017 |
| C:\Windows\Prefetch\CONSENT.EX... | Archivo | Modificacion | Sun Jun 04 23:43:19 COT 2017 |

Ilustración 20. Pestaña principal prototipo de la herramienta versión cliente

- **Historial:** Muestra todas las actividades anómalas que ha registrado la herramienta y que han sido confirmadas por el Administrador de Sistema. Para actualizar, se debe dar clic sobre la pestaña. Los datos que se mostrarán sobre el evento son:
 - Entrada afectada: Nombre del recurso que fue afectado (Archivo, registro, IP, etc.)
 - Tipo de entrada: archivo, carpeta, clave de registro, valor de registro, dirección IP.
 - Acción: Evento que sucedió con la entrada. Ejemplo: Creación o eliminación de un archivo.
 - Fecha: Momento en que ocurrió el evento y fue aceptado por el Administrador de sistema
- **Configuración:** Hay un campo que indica la dirección IP del servidor a conectarse. Después de dar clic al botón “Aceptar”, se hace la conexión:

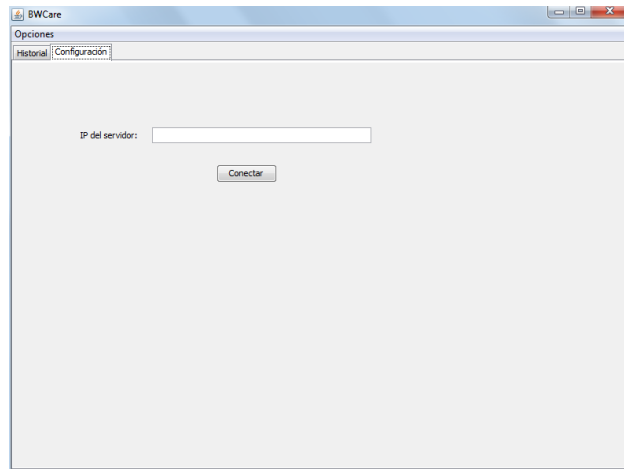


Ilustración 21. Pestaña "Configuración" prototipo de la aplicación versión cliente

Cuando ocurre una novedad o un evento sobre alguna de las carpetas que se están monitoreando, se presenta una ventana indicando el suceso:

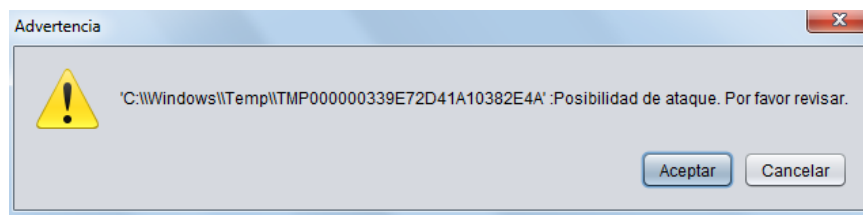


Ilustración 22. Ventana mostrando novedad en carpeta monitoreada

Como se puede observar en la ilustración anterior, hay dos botones: "Aceptar" y "Cancelar". Se deja esta disposición para que el Administrador pueda observar el evento ocurrido y pueda confirmar la novedad o que pudo haber sido resultado de una actividad realizada de manera consciente (Ejemplo, la instalación de un software). Si el evento fue anómalo, el Administrador debe dar clic en "Aceptar" y la entrada quedará registrada en el historial. Si se selecciona el botón "Cancelar", no queda ningún registro.

Otra de las funciones especiales del prototipo "BWCare" es la posibilidad de detectar archivos inflados. Cuando ocurre un cambio en un archivo, primero se recalculan los pesos, el peso anterior toma el valor del peso actual y este último, toma el valor del tamaño actual que tiene el archivo revisado. Si la diferencia entre el peso anterior y el peso actual es mayor al 50%, la alerta es mostrada.

1.2 Herramienta en modo servidor

La interfaz de la herramienta presenta las mismas funcionalidades respecto a la versión cliente con las siguientes excepciones:

- No se encuentra la pestaña de configuración
- En la pestaña “historial” se incluye un dato más: la dirección IP del computador que presentó el evento.

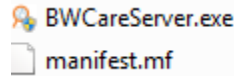


Ilustración 23. Ubicación archivo ejecutable herramienta BWCare Server

| Entrada | Tipo de entrada | Acción | Fecha del evento | IP |
|--|-----------------|--------------|------------------------------|---------------|
| C:\TestRegistros\Directorio2\Text3.bt | Archivo | Inflado | Mon May 08 18:58:48 COT 2017 | 192.168.0.106 |
| C:\Windows\Prefetch\CONSENT.EXE-40419367.pf | Archivo | Modificación | Sun Jun 04 21:34:45 COT 2017 | 192.168.0.6 |
| C:\Windows\Text3.bt | Archivo | Creación | Sun Jun 04 21:34:45 COT 2017 | 192.168.0.6 |
| C:\Windows\Prefetch\CONHOST.EXE-0C6456FB.pf | Archivo | Modificación | Sun Jun 04 23:43:19 COT 2017 | 192.168.0.6 |
| C:\Windows\Cmi4464.pnf3 | Archivo | Creación | Sun Jun 04 23:43:19 COT 2017 | 192.168.0.6 |
| C:\Windows\Cmi4464.pnf3 | Archivo | Creación | Sun Jun 04 23:43:19 COT 2017 | 192.168.0.6 |
| C:\Windows\Prefetch\CONSENT.EXE-40419367.pf | Archivo | Modificación | Sun Jun 04 23:43:19 COT 2017 | 192.168.0.6 |
| C:\Windows\Prefetch\CONSENT.EXE-40419367.pf | Archivo | Modificación | Sun Jun 04 23:43:19 COT 2017 | 192.168.0.6 |
| C:\Windows\Prefetch\SEARCHPROTOCOLHOST.EXE-69C456... | Archivo | Modificación | Sun Jun 04 23:43:19 COT 2017 | 192.168.0.6 |
| C:\Windows\Prefetch\DLLHOST.EXE-576CF6B2.pf | Archivo | Modificación | Sun Jun 04 23:43:19 COT 2017 | 192.168.0.6 |
| C:\Windows\Prefetch\DLLHOST.EXE-4B6CB38A.pf | Archivo | Modificación | Sun Jun 04 23:43:19 COT 2017 | 192.168.0.6 |
| C:\Windows\Cmi4464.pnf3 | Archivo | Creación | Sun Jun 04 23:43:19 COT 2017 | 192.168.0.6 |
| C:\Windows\Prefetch\HD-LOGROTATOR.EXE-EE83AC01.pf | Archivo | Modificación | Sun Jun 04 23:43:19 COT 2017 | 192.168.0.6 |
| C:\Windows\Prefetch\SEARCHFILTERHOST.EXE-44162447.pf | Archivo | Modificación | Sun Jun 04 23:43:19 COT 2017 | 192.168.0.6 |
| C:\Windows\Prefetch\NOTEPAD.EXE-C5670914.pf | Archivo | Modificación | Sun Jun 04 23:43:19 COT 2017 | 192.168.0.6 |
| C:\Windows\Prefetch\CONHOST.EXE-0C6456FB.pf | Archivo | Modificación | Sun Jun 04 23:43:19 COT 2017 | 192.168.0.6 |
| C:\Windows\Prefetch\DLLHOST.EXE-576CF6B2.pf | Archivo | Modificación | Sun Jun 04 23:43:19 COT 2017 | 192.168.0.6 |
| C:\Windows\Prefetch\SEARCHPROTOCOLHOST.EXE-69C456... | Archivo | Modificación | Sun Jun 04 23:43:19 COT 2017 | 192.168.0.6 |
| C:\Windows\Prefetch\FLASHPLAYERUPDATESERVICE.EXE-01... | Archivo | Modificación | Sun Jun 04 23:43:19 COT 2017 | 192.168.0.6 |
| C:\Windows\Prefetch\TASKENG.EXE-35FA9C06.pf | Archivo | Modificación | Sun Jun 04 23:43:19 COT 2017 | 192.168.0.6 |
| C:\Windows\Prefetch\CONSENT.EXE-40419367.pf | Archivo | Modificación | Sun Jun 04 23:43:19 COT 2017 | 192.168.0.6 |

Ilustración 24. Pantalla historial en versión servidor

Consideraciones adicionales

Basado en el resultado de la herramienta obtenida, vale la pena mencionar las diferencias del prototipo de la herramienta respecto a las demás herramientas de seguridad:

- Antivirus: Debido a que los ataques tipo APT están diseñados para evadir los antivirus, el prototipo lo puede complementar bien evaluando el cambio que presenten en el entorno.
- IPS de host: Estas herramientas trabajan sobre firmas, el prototipo no cuenta con firmas.
- Firewall: Los firewalls solo tienen un perímetro de defensa que si es vulnerado hace que esta herramienta no funcione, además, depende mucho de las reglas. El prototipo construido no tiene reglas principalmente.
- *Honeypots*: Los *honeypots* no pueden estar funcionando en la red de producción porque pierden su valor y terminan siendo un peligro para la infraestructura de red. Este prototipo si puede funcionar en una red de producción.

Paquetes incluidos:

- Weka: Es la aplicación que ayudará con la clasificación de un nuevo caso en “Ataque” o “No ataque” utilizando el algoritmo de Máquina de Soporte Vectorial.
- OpenCSV: Paquete que permite hacer la lectura y escritura sobre archivos CSV que es el formato en el cual se encuentran los datos de entrenamiento y prueba con que funciona la máquina de soporte vectorial.

Resultados de las pruebas

Las pruebas se realizaron en el siguiente ambiente:

- Computador 1: Sistema operativo Windows 7, 2 GB de memoria con JRE 1.7 instalado
- Computador 2: Sistema operativo Windows XP, 1 GB de memoria con JRE 1.7 instalado.
- Computador 3: Sistema operativo Linux. Sólo tendrá la herramienta McAfee Evader
- Computador 4: Sistema Kali Linux. 2 GB de memoria

Carpeta a revisar: C:\Windows\System32

Los resultados de las pruebas fueron:

- Prueba 1: Prueba edición de archivos y folders: Se espera que el prototipo pueda registrar todos los cambios que se registran. Estos cambios se hicieron sobre la carpeta C:\Windows\System32\drivers:
 - Creación de archivo: Se creó el archivo text4.txt. Resultado: Exitoso

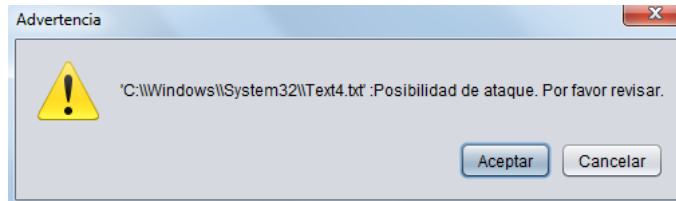


Ilustración 25. Notificación creación de archivo

- Eliminación de archivo: Se eliminó el archivo text4.txt. Resultado: Exitoso.

| Entrada | Tipo de entrada | Acción | Fecha |
|---------------------------------|-----------------|-------------|--------------------------|
| 'C:\Windows\System32\Text4.txt' | Archivo | Creación | Sat May 14 21:35:59 COT; |
| 'C:\Windows\System32\Text4.txt' | Archivo | Eliminación | Sat May 14 21:36:01 COT; |
| 'C:\Program Files\Common F... | Archivo | Creación | Sat May 14 21:36:07 COT; |
| 'C:\Program Files\Common F... | Archivo | Eliminación | Sat May 14 21:36:09 COT; |

Ilustración 26. Notificación eliminación de archivo

- Inflado de archivo: Se puede inflar el archivo en dicha carpeta con la condición de ser abierto desde una cuenta Administrador. De lo contrario, el archivo no se puede guardar y el prototipo no detecta el cambio.

- Prueba 2: Prueba con *malware* APT del sistema Kali Linux:

Para esta parte de la prueba, la intención es mostrar el funcionamiento del prototipo con un *malware* diferente a los analizados previamente. El *malware* elegido fue Aurora ya que la idea es demostrar que el cambio de comportamiento también aplica para ataques diferentes a los estudiados:

Para la ejecución del ataque Operación Aurora, se tendrá el apoyo de la herramienta Metasploit que se encuentra dentro de la aplicación Kali Linux. Dicha herramienta tiene el exploit que aprovecha la vulnerabilidad enunciada en el boletín número MS10-002, en donde se podía permitir la ejecución remota de código si un usuario de Internet Explorer visita una página web diseñada de manera especial por un atacante [36].

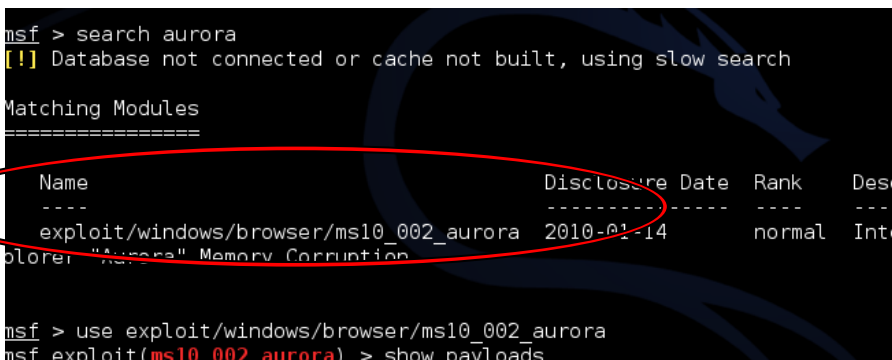
A screenshot of a Metasploit terminal window. The user enters 'search aurora'. The terminal shows a message: '[!] Database not connected or cache not built, using slow search'. Below that, it says 'Matching Modules' followed by a table. The table has columns: Name, Disclosure Date, Rank, and Description. One entry is circled in red: 'exploit/windows/browser/ms10_002_aurora' with a disclosure date of '2010-01-14', rank 'normal', and description 'Internet Explorer "Aurora" Memory Corruption'. Below the table, the user enters 'use exploit/windows/browser/ms10_002_aurora' and 'show payloads'.

Ilustración 27. Exploit que posee la herramienta Metasploit respecto al Ataque Aurora

Posteriormente, se elige el payload a utilizar para explotar la vulnerabilidad. El elegido será el “reverse TCP”:

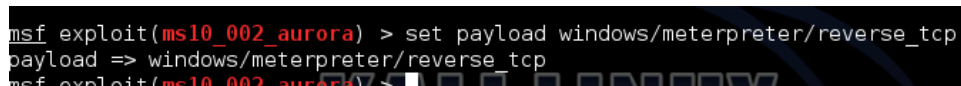
A screenshot of a Metasploit terminal window. The user enters 'set payload windows/meterpreter/reverse_tcp'. The terminal shows 'payload => windows/meterpreter/reverse_tcp'. The user then enters 'show payloads'.

Ilustración 28. Exploit elegido para atacar la vulnerabilidad ms10_002

Luego, se configuran los parámetros para indicar el host y el puerto del atacante que estará pendiente de la víctima que intentará ingresar a la página maliciosa. Para este caso, la dirección IP del host es la 192.168.0.135 y el puerto es el 443.

```
msf exploit(ms10_002_aurora) > set SRVPORT 80
SRVPORT => 80
msf exploit(ms10_002_aurora) > set URIPATH /
URIPATH => /
msf exploit(ms10_002_aurora) > set LHOST 192.168.0.135
LHOST => 192.168.0.135
msf exploit(ms10_002_aurora) > set LPORT 443
LPORT => 443
msf exploit(ms10_002_aurora) >
```

Ilustración 29. Configuración del atacante para ejecutar exploit de Aurora

Cuando la víctima ingresa a la página maliciosa, el atacante recibirá una notificación de la dirección IP que tuvo el ingreso y creará el backdoor para ingresar al computador de la víctima. Este último tiene como dirección IP la 192.168.0.160:

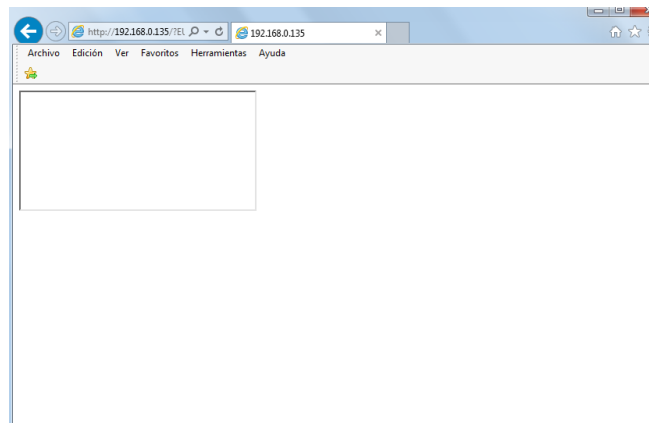


Ilustración 30. La víctima ingresa a página maliciosa creada por exploit de Operación Aurora

```
msf exploit(ms10_002_aurora) > exploit -z
[*] Exploit running as background job.

[*] Started reverse handler on 192.168.0.135:443
[*] Using URL: http://0.0.0.0:80/
[*] Local IP: http://192.168.0.135:80/
[*] Server started.
msf exploit(ms10_002_aurora) > [*] 192.168.0.160 ms10_002_aurora - Sending Internet Explorer "Aurora" Memory Corruption
[*] 192.168.0.160 ms10_002_aurora - Sending Internet Explorer "Aurora" Memory Corruption
```

Ilustración 31. Desde la máquina del atacante, se envía el backdoor para ingresar a la máquina de la víctima.

Se revisa que la sesión desde el servidor haya quedado abierta:

```
sessions -l

Active sessions
=====

Id  Type                Information                                     Connection
--  -
1   meterpreter x86/win32  0AMB-D8C12407C4\oamb @ 0AMB-D8C12407C4  192.168.0
35:443 -> 192.168.0.160:1090 (192.168.0.160)
```

Ilustración 32. Sesión de la víctima abierta desde el servidor

Se decide ingresar remotamente al equipo de la víctima desde el equipo del atacante, el cual ya tiene todos los permisos para acceder, borrar o modificar los archivos a su preferencia. Se dejó el prototipo BWCare abierto para notificar los cambios que se pueden hacer. Para esta ocasión, se creó el archivo Dummy.dll en la carpeta C:\Windows\Media:

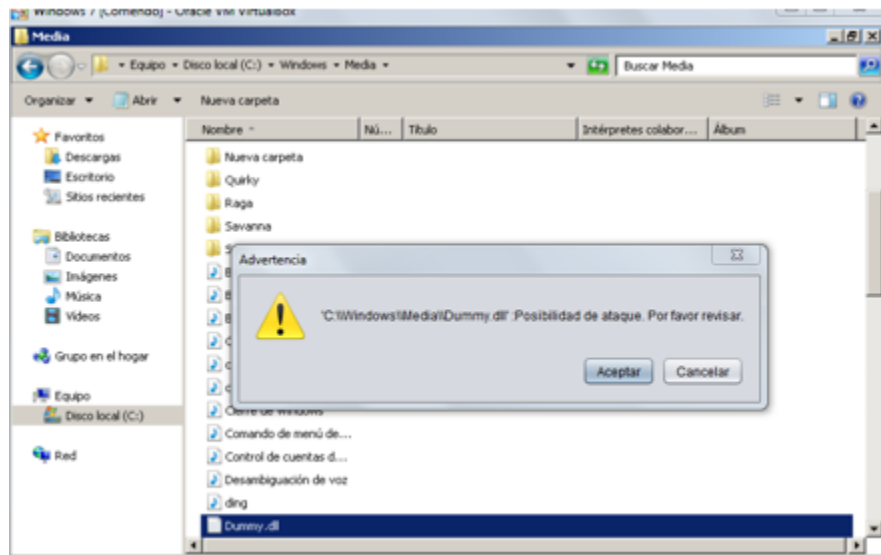


Ilustración 33. Notificación de novedad en máquina de la víctima

Prueba 3: Ejecución con herramienta McAfee Evader:

Al ejecutar la herramienta Evader de McAfee, se utilizó como vector de ataque la vulnerabilidad de un ataque informático llamado Conficker para ejecutar un archivo determinado (calculadora – calc.exe). Al realizar esta prueba, el ataque no fue exitoso sobre la máquina Windows 7, por tal motivo, el sistema no registró ningún cambio de comportamiento.

V – CONCLUSIONES Y TRABAJOS FUTUROS

1. Conclusiones

- Si existen características que permiten asociar el comportamiento de los elementos del sistema operativo y la ocurrencia de los ataques tipo APT en el sentido de:
- Las carpetas más afectadas son las de sistema, en especial, la ruta C:\Windows y C:\Archivos de programa con profundidad entre 4 y 7 carpetas.
- La clave de registro más comprometida fue la HKEY_Local_Machine también a una profundidad de 4 a 7 niveles.
- El modelo de máquina de soporte vectorial fue utilizado por encima de otros métodos por su eficiencia para clasificar datos nuevos teniendo en cuenta la estructura de datos (numéricos como variables independientes y categóricos nominales en la variable dependiente) y la menor tasa de error.
- El hecho de establecer una metodología de desarrollo que incluya los riesgos permite tener un nivel de conciencia más alto sobre las decisiones que se toman y permite identificar otros requerimientos para que la herramienta a desarrollar cumpla su función. Además, este tipo de metodologías son útiles cuando se desarrollan herramientas de seguridad porque pueden evitar huecos que hagan perder la utilidad de la aplicación.
- Sin embargo, con todas las herramientas de seguridad no es suficiente para tener una protección apropiada ante este tipo de ataques: Hay que fortalecer eslabones débiles en la infraestructura de seguridad, es especial, las personas: ayudar a tomar conciencia de los peligros que hay en la red, no confiar y comunicar al personal idóneo permiten que estas amenazas tengan un impacto menor.
- Es importante tener en cuenta tener claro el modelo de datos que se tiene antes de hacer cualquier labor de minería. Con la teoría no es suficiente para identificar los mejores algoritmos para solucionar un problema.

2. Trabajos futuros

Después del desarrollo de este prototipo, se puede extender a otros contextos como por ejemplo otros sistemas operativos como Linux, Unix o Android. En parte, la elección de Java como lenguaje de desarrollo del prototipo obedece a la facilidad que éste tiene para extender sus desarrollos a estos sistemas operativos. Asimismo, se pueden incluir más técnicas para hacer de este prototipo un producto más robusto. Por ejemplo, se puede incluir minería de datos para analizar con un mayor nivel de detalle la estructura de registros y archivos que pueden sufrir mayor nivel de riesgo.

Por otro lado, se puede incluir funcionalidades para que el prototipo sea más didáctico: se puede incluir opciones para ver qué respuesta tomaría el Sistema Operativo si se simula la eliminación o la adición de un archivo. Queda claro que para este tipo de desarrollos se debe tener cuidado del dato y de la tarea que se va a modificar porque pueden causar daño al Sistema Operativo.

VI – REFERENCIAS Y BIBLIOGRAFÍA

- [1] E. Messmer, “What is an ‘Advanced Persistent Threat,’ anyway?”, *Network World*, vol. 28, núm. 3, 02-jul-2011.
- [2] McAfee Labs, “McAfee Labs Threats Report”, McAfee Labs, ago. 2015.
- [3] M. Hopkins y A. Dehghantanha, “Exploit Kits: The production line of the Cyber-crime Economy?”, presentado en University of Salford, Manchester, United Kingdom, 2015.
- [4] Information Systems Audit and Control Association (ISACA), “Advanced Persistent Threat awareness 2015”, Information Systems Audit and Control Association (ISACA), 2015.
- [5] S. Siddiqui, M. S. Khan, K. Ferens, y W. Kinsner, “Detecting Advanced Persistent Threats using Fractal Dimension based Machine Learning Classification”, 2016, pp. 64–69.
- [6] Y. Wang, Y. Wang, J. Liu, y Z. Huang, “A Network Gene-Based Framework for Detecting Advanced Persistent Threats”, 2014, pp. 97–102.
- [7] G. Zhao, K. Xu, L. Xu, y B. Wu, “Detecting APT Malware Infections Based on Malicious DNS and Traffic Analysis”, *IEEE Access*, vol. 3, pp. 1132–1142, 2015.
- [8] N. Virvilis y D. Gritzalis, “The Big Four - What We Did Wrong in Advanced Persistent Threat Detection?”, en *Information Security and Critical Infrastructure Protection Research Laboratory*, Athens University of Economics and Business, Athens, Greece, 2013, pp. 248–254.
- [9] McAfee Labs y McAfee Foundstone Professional Services, “Protecting your critical assets”, 2010.
- [10] L. G. Harbaugh, “Trend Micro Deep Security 8.0”, *EdTech*, 31-jul-2012. [En línea]. Disponible en: <http://www.edtechmagazine.com/higher/article/2012/07/trend-micro-deep-security-80>.
- [11] P. Chapman *et al.*, “CRISP-DM 1.0”, 1999.
- [12] J. A. Gallardo Arancibia, “Metodología para la definición de requisitos en proyectos de data mining”, *Informatica*, 2009.
- [13] M. J. A. Berry, G. Linoff, y M. J. A. Berry, *Mastering data mining: the art and science of customer relationship management*. New York: Wiley Computer Pub, 2000.
- [14] Kaspersky Labs, “Targeted Cyberattacks Logbook”. [En línea]. Disponible en: <https://apt.securelist.com/#firstPage>.

- [15] McAfee Labs, “Evader: Ready made evasion test lab from McAfee”.
- [16] Offensive Security, “Metasploit Unleashed - Free Online Security Training”. [En línea]. Disponible en: <https://www.offensive-security.com/metasploit-unleashed/>.
- [17] J. M. Holguín, M. Moreno, y B. Merino, “Detección de APTs”. Centre Seguretat de la Comunitat Valenciana, may-2013.
- [18] N. Moran, “Understanding Advanced Persistent Threats: A case study”, *Login*, vol. 36, núm. 4, p. 6, ago-2011.
- [19] E. Cole, “The Changing Threat”, en *Advanced Persistent Threat*, Elsevier, 2013, pp. 3–26.
- [20] P. Giura y W. Wang, “A Context-Based Detection Framework for Advanced Persistent Threats”, 2012, pp. 69–74.
- [21] M. A. Dye, R. McDonald, y A. Rufi, *Guía de estudio de CCNA Exploration: Aspectos básicos de Networking*. Cisco Networking Academy: Pearson Education, 2008.
- [22] A. S. Tanenbaum y E. Núñez Ramos, *Redes de computadoras*. México: Pearson Educación, 2003.
- [23] J. R. Vacca, Ed., *Computer and information security handbook*. Amsterdam ; Boston : Burlington, MA: Elsevier ; Morgan Kaufmann, 2009.
- [24] E. J. Mira Alfaro, “implantación de un sistema de detección de intrusos en la universidad de valencia”, Universidad de Valencia, Valencia, 2002.
- [25] J. Chee, “Host Intrusion Detection Systems and beyond”, SANS Institute Infosec Reading Room, jun. 2008.
- [26] A. S. Tanenbaum y D. J. Wetherall, *Redes de computadoras*. México: Pearson Educación, 2012.
- [27] S. Erdheim, “Deployment and management with next-generation firewalls”, *Netw. Secur.*, vol. 2013, núm. 10, pp. 8–12, 2013.
- [28] P. Szor, *The Art of Computer Virus Research and Defense*. Addison Wesley Professional, 2005.
- [29] S. Chandran, P. Hrudya, y P. Poornachandran, “An efficient classification model for detecting advanced persistent threat”, en *Advances in Computing, Communications and Informatics (ICACCI), 2015 International Conference on*, 2015, pp. 2001–2009.

- [30] Z. Saud y M. H. Islam, "Towards proactive detection of advanced persistent threat (APT) attacks using honeypots", 2015, pp. 154–157.
- [31] P. Shakarian, J. Shakarian, y A. Ruef, *Introduction to cyber-warfare: a multidisciplinary approach*. Amsterdam ; Boston: Syngress, 2013.
- [32] NSS Labs, "Vulnerability-based protection and the Google 'Operation Aurora' attack", 08-mar-2010. [En línea]. Disponible en: https://www.nsslabs.com/sites/default/files/public-report/files/NSSLabs_Vulnerability-based%20Protection-Google-EPPv14201003.pdf.
- [33] R. Genes, "Operation aurora and Beyond: How to avoid that this happens to your organization", presentado en 11th Annual Computerworld: 100 premier IT leaders conference, Phoenix, Arizona, 09-mar-2010.
- [34] D. Drummond, "A new approach to China", *Official Google Blog*, 12-ene-2010.
- [35] McAfee Labs Threat Center, "Exploit-Comele - Malware", *Exploit Comele*, 13-ene-2010. [En línea]. Disponible en: <http://www.mcafee.com/threat-intelligence/malware/default.aspx?id=253210>.
- [36] McAfee Labs Threat Center, "Roarur.dr - Malware", *Roarur.dr*, 14-ene-2010. [En línea]. Disponible en: <http://www.mcafee.com/threat-intelligence/malware/default.aspx?id=253415>
- [37] McAfee Labs Threat Center, "Roarur.dll | Virus Profile & Definition | McAfee Inc.", *Roarur.dll*, 19-ene-2010. [En línea]. Disponible en: <http://home.mcafee.com/virusinfo/virusprofile.aspx?key=253416#none>.
- [38] T. de seguridad Microsoft, "Boletín de seguridad de Microsoft MS10-002 - Crítica", 10-feb-2010. [En línea]. Disponible en: <https://technet.microsoft.com/library/security/ms10-002>.
- [39] N. Falliere, L. O. Murchu, y E. Chien, "W32. stuxnet dossier", *White Pap. Symantec Corp Secur. Response*, vol. 5, 2011.
- [40] A. Matrosov, E. Rodionov, D. Harley, y J. Malcho, "Stuxnet under the microscope", *ESET LLC Sept. 2010*, 2010.
- [41] T. de seguridad Microsoft, "Boletín de seguridad de Microsoft MS10-046 - Crítica", 02-ago-2010. [En línea]. Disponible en: <https://technet.microsoft.com/library/security/ms10-046>.
- [42] "Boletín de seguridad de Microsoft MS08-067 - Crítica". [En línea]. Disponible en: <https://technet.microsoft.com/library/security/ms08-067>

- [43] T. de seguridad Microsoft, “Boletín de seguridad de Microsoft MS10-092 - Importante”, 02-mar-2011. [En línea]. Disponible en: <https://technet.microsoft.com/library/security/ms10-092>.
- [44] Symantec Corporation, “OS Attack: MSRPC Server Service RPC CVE-2008-4250: Attack Signature - Symantec Corp.” [En línea]. Disponible en: https://www.symantec.com/security_response/attacksignatures/detail.jsp?asid=23179.
- [45] T. de seguridad Microsoft, “Boletín de seguridad de Microsoft MS10-061 - Crítica”, 29-sep-2010. [En línea]. Disponible en: <https://technet.microsoft.com/library/security/ms10-061>.
- [46] B. Bencsáth, G. Pék, L. Buttyán, y M. Félegyházi, “Duqu: A Stuxnet-like malware found in the wild”, *CrySyS Lab Tech. Rep.*, vol. 14, 2011.
- [47] Symantec Security Response, “W32.Duqu: The Precursor to the Next Stuxnet”, oct. 2011.
- [48] Emergency Response Team, “Duqu: son of Stuxnet?”, *Comput. Fraud Secur.*, vol. 2011, núm. 11, p. 3, nov. 2011.
- [49] O. Montoya Suárez, “Aplicación del análisis factorial a la investigación de mercados. Caso de estudio”, *Sci. Tech.*, vol. 1, núm. 35, 2007.
- [50] J. Hernández Orallo, M. J. Ramírez, y C. Ferri, *Introducción a la minería de datos*. Prentice Hall, 2004.
- [51] D. Peña, *Análisis de datos multivariantes*, vol. 24. McGraw-Hill Madrid, 2002.
- [52] S. De la Fuente Fernández, “Análisis Factorial”, Universidad Autónoma de Madrid, Facultad de Ciencias Económicas y Empresariales, 2011.
- [53] J. L. V. Villardón, “Análisis de componentes principales”, *Cataluña UOC Dep. Estad.*, vol. 32, 2002.
- [54] C. M. Cuadras, *Nuevos métodos de análisis multivariante*. Barcelona, España: CMC Editions, 2014.
- [55] J. M. Bande Serrano y J. Hernández Palancar, “Técnicas y algoritmos de minería de datos empleados en sistemas de detección de intrusiones”, Centro de Aplicaciones de Tecnologías de Avanzada CENATAV, La Habana, Cuba, Reporte técnico, jun. 2014.
- [56] R. O. Duda, P. E. Hart, y D. G. Stork, *Pattern classification*, Second edition. 1996.
- [57] C. Cortes y V. Vapnik, “Support-vector networks”, *Mach. Learn.*, vol. 20, núm. 3, pp. 273–297, 1995.

- [58] University of Waikato, “Weka 3 - Data Mining with Open Source Machine Learning Software in Java”, 2015. [En línea]. Disponible en: <http://www.cs.waikato.ac.nz/ml/weka/>.
- [59] P. Kruchten, “Architectural Blueprints—The ‘4+ 1’ View Model of Software Architecture”, *Tutor. Proc. Tri-Ada*, vol. 95, pp. 540–555, 1995.
- [60] I. Sommerville, *Software engineering*, 9th ed. Boston: Pearson, 2011.
- [61] L. Méndez Morales, “Gestión de riesgos seguridad de la información”, presentado en Presentación para la materia de Seguridad de la información de la Maestría de Ingeniería de Sistemas y Computación de la Pontificia Universidad Javeriana, Pontificia Universidad Javeriana, jul-2014.
- [62] W. Stallings, *Data and computer communications*, 8th ed. Upper Saddle River, N.J: Pearson/Prentice Hall, 2007.

VII - ANEXOS

Glosario

- Amenaza: Cualquier circunstancia o evento con el potencial de impactar adversamente los activos de la organización [61]
- Ataque “Hombre en el medio” (*Man in the middle attack*): Ataque en el que el atacante puede leer, insertar y modificar a su voluntad los mensajes que se envían entre dos puntos finales sin que ninguna de las partes sea consciente de que la ruta de los datos se ha visto comprometida [21].
- Backdoor: Es un punto de entrada secreto dentro de un programa que permite a alguien obtener acceso sin tener que ir a los procesos de acceso usuales [62].
- Denegación de servicios: Es un ataque diseñado para saturar los enlaces de una red con datos ilegítimos. También pueden saturar el enlace de Internet provocando que este tráfico sea descartado [21].
- Entorno: Estado del sistema operativo del computador del sistema en determinado momento.
- Entorno seguro: El panorama inicial del computador perteneciente al sistema cuando se instala el prototipo de la herramienta. Para efectos del proyecto, se asume que el prototipo de la herramienta será instalado en una situación libre de ataques tipo APT.
- Entorno inseguro: Estado del sistema que ya ha sufrido variaciones.
- Exploit del día cero: Exploit desconocido proveniente de una vulnerabilidad que no había sido identificada previamente. El peligro de este tipo de exploits está en función de qué tan amplio es el software y qué nivel de acceso tiene el atacante. Su ciclo de vida dura el tiempo que se descubrió la vulnerabilidad y la publicación del fabricante o de investigadores. [23].
- Ingeniería social: Es el grupo de prácticas que utiliza el atacante para engañar a la víctima con el fin de conseguir su objetivo, que bien puede ser, acceder y/o tomar información privilegiada que se puede dar visitando un enlace, abriendo un documento que le fue enviado por correo electrónico o accedido desde una memoria USB, permitir el paso a un desconocido a las instalaciones físicas de la empresa de la víctima o incluso, acceder al perfil del usuario en una determinada red social [61].

- Intranet: Redes privadas utilizadas solo por una empresa. Les permite comunicarse y realizar transacciones entre empleados y sucursales globales [21].
- Malware: Es el conjunto de instrucciones que se ejecutan en un computador y hace que éste haga algo que el atacante le indique. Es muy utilizado por los intrusos para obtener accesos, buscar datos como contraseñas, monitorear comunicaciones en tiempo real, obtener control y acceso remoto y atacar otros sistemas [23].
- *Phishing*: Son ataques donde comúnmente se utilizan correos falsificados y páginas web fraudulentas para engañar a las personas que van a recibir estos comunicados con el fin de obtener de ellos información personal [28].
- *Rootkit*: Conjunto especial de herramientas de hackeo que son usados después de que el atacante haya roto la seguridad de un sistema y haya obtenido accesos de nivel de súper administrador. Usualmente, hay tipos de rootkits especiales llamados *rootkits de modo usuario* en donde los hackers rompen en un sistema con *exploits* e instalan versiones modificadas de estas herramientas [28].
- Vulnerabilidad: Debilidad en un sistema de información, procedimientos de seguridad, controles internos o implementación que pueden ser explotadas por la amenaza [61].