ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

Facultad de Ingeniería en Electricidad y Computación

“DISEÑAR E IMPLEMENTAR EL PROTOTIPO DE UNA ARQUITECTURA DE SEGURIDAD APLICADA A UNA INSTITUCIÓN FINANCIERA PARA MITIGAR LOS ATAQUES DE MALWARE "

TRABAJO DE TITULACIÓN

PREVIO A LA OBTENCIÓN DEL TITULO DE:

MAGISTER EN SEGURIDAD INFORMÁTICA APLICADA

AUTOR

LUIS PAÚL GARCÍA ANGUISACA

Guayaquil – Ecuador

2017

AGRADECIMIENTO

A mi familia de la cual siempre he tenido su ayuda y comprensión. A mi Ratona, ya que sin tu apoyo y compañía esto no hubiera sido posible.

.

DEDICATORIA

A mi hijo Matías, a la Damita que me dio ánimos para continuar y a todas las personas que me inspiraron para darle este rumbo profesional a mi vida.

TRIBUNAL DE SUSTENTACIÓN

_____

Ing. …

DIRECTOR MSIA

_____

Ing. …

DIRECTOR DEL PROYECTO DE GRADUACIÓN

_____

Mgs. …

MIEMBRO DEL TRIBUNAL

DECLARACIÓN EXPRESA

"La responsabilidad del contenido de esta Tesis de Grado, me corresponde exclusivamente; y el patrimonio intelectual de la misma a la Escuela Superior Politécnica del Litoral".

(Reglamento de exámenes y títulos profesionales de la ESPOL)

RESUMEN

El presente proyecto de tesis consta de seis capítulos, los cuales inician con una descripción de los diversos tipos de malware y su entorno, seguidamente se realiza un análisis estático y dinámico de las funcionalidades que la convierten en una amenaza persistente avanzada.

En los capítulos siguientes se propone un modelo de seguridad basado en un conjunto de componentes que usan técnicas automatizadas de análisis estático y dinámico para la detección y control de las amenazas de seguridad, se genera un prototipo del modelo que se aplica sobre un entorno virtual que simula una entidad financiera y sobre el cual se corren ciertas muestras de malware para probar la efectividad del modelo de seguridad.

ÍNDICE GENERAL

ABREVIATURAS

APT	**:**	Advanced Persistent Threat
ATM	**:**	Automated Teller Machine
DNS	**:**	Domain Name System
HTML	**:**	Hypertext Markup Language
IP	**:**	Internet Protocol
LAN	**:**	Local Area Network
RAM	**:**	Random Access Memory
SQL	**:**	Structured Query Language
URL	**:**	Uniform Resource Locator
USB	**:**	Universal Serial Bus
VPN	**:**	Virtual Private Network
WEB	**:**	World Wide Web

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

INTRODUCCIÓN

La seguridad de la información y la seguridad informática hoy en día se han convertido en un pilar fundamental para las organizaciones. Las tecnologías de la información y comunicación deben estar respaldadas por una política de seguridad, así como de los controles informáticos necesarios que mitiguen las amenazas informáticas que se afrontan actualmente.

El Malware es una de las amenazas de seguridad informática más antiguas en el mundo informático y a pesar de eso hoy en día es una de las más sofisticadas y más utilizadas para realizar ataques informáticos ya sean con fines lucrativos políticos de espionaje entre otros. Uno de los segmentos más afectado es el financiero. De aquí que el presente trabajo está orientado en el análisis de este tipo de amenazas con la finalidad de estructurar una arquitectura de seguridad para detectar bloquear y mitigar los ataques de malware en una institución financiera.

CAPÍTULO 1

GENERALIDADES

1.1 Antecedente

Un informe publicado en el año 2013 por el The Wall Street Journal calculaba que solamente en los Estados Unidos de América el costo del Cyber Crimen llegaba a los 100 Billones de dólares anuales [1].

Según un estudio realizado en enero del 2016 por la revista Forbes se estima que el costo de Cyber Crimen se cuadruplico entre el año 2013 y 2015, pasando

de 100 billones a 400 billones al año y alcanzará los 2 Trillones de dólares en el año 2019 [2].

Uno de los primeros ataques considerados como una amenaza persistente fue la Operación Aurora, llevada a cabo en contra de Google y atribuida presuntamente al Gobierno de China, ya que se encontraron rastros de algoritmos de verificación de redundancia cíclica (CRC) publicados únicamente en el idioma Chino mandarín. En el ataque se logró infiltrar malware en los computadores de las oficinas de Google, por medio de un zero-day que explotaba una vulnerabilidad del navegador Internet Explorer, el propósito fue el espionaje tecnológico y el robo de información [3].

El sector financiero ha sido muy afectado en la última década por los cibercriminales y de hecho ha pasado a ser el principal riesgo para una entidad bancaria, según el estudio de la Federación Latinoamericana de Bancos (Felaban) publicado en 2015 [4] el 98.5% de los riesgos de la banca son informáticos, representando solamente el 1.5% los asaltos a manos armada en oficinas.

Uno de los ejemplos más recientes es el sabotaje y robo cibernético sufrido en febrero de 2016 por parte del Banco Central de Bangladesh. En el ataque se pudo realizar una transferencia de 81 millones de dólares desde una cuenta que tenía el Banco en la Reserva Federal en New York, hacia cuentas en Filipinas y Sri Lanka. El malware utilizado para el ataque permitía la interacción con el sistema SWIFT (Society for Worldwide Interbank Financial Telecomunications) [5]. Se cree que los atacantes deben tener profundo conocimiento sobre el

sistema SWIFT así como en el desarrollo de malware. El portal de noticias The Hacker News [5] inclusive publicó que El Banco del Austro fue afectado por un monto de 12 millones de dólares en el año 2015.

Entre los ataques más significativos al sector financiero esta Carbanak que se ejecutó en contra de las entidades bancarias de Europa y América, se calcula que ocasionó pérdidas en alrededor de 1000 millones de dólares según los reportes de Kaspersky Lab [6].

1.2 Descripción del Problema

Una amenaza persistente avanzada (APT) posee dos características especiales: Se enfoca en un objetivo específico y trata de persistir en el entorno informático de su víctima la mayor cantidad de tiempo. De esta forma se obtiene un mayor provecho del ataque, siendo en este caso el malware una de las principales herramientas usadas en los procesos de una APT.

Muchos fabricantes han evolucionado sus soluciones de seguridad pero la complejidad de los nuevos ataques abre una brecha que no está cubierta.

Entre las vulnerabilidades que afronta una organización podemos mencionar que en un ataque de APT dirigido, el malware ha sido desarrollado para un objetivo particular, y por ende no existe una firma en la base de datos de ninguna herramienta de seguridad que pueda detectarlo. Los sistemas de detección basados en heurística o comportamiento generalmente requieren agrupar diferentes acciones como capturas del teclado y creación de fichero, ejecutadas desde un mismo proceso. Para evadir esto, el malware usa técnicas de

programación polimórfica para mutar en tiempo de ejecución y generar diferentes firmas para no ser detectado [7], además de mantener una arquitectura modular que le permite distribuir sus acciones en diferentes componentes de software [7].

Otra característica que posee es la capacidad de detectar que está siendo ejecutado en un entorno de Sandbox e interrumpir su actividad por un periodo de tiempo o cambiar su comportamiento a “no malicioso” [7] para evadir el análisis dentro de la Caja de arena.

El uso de anti-disassembling, anti-debuggin y métodos de ofuscación de código son también técnicas utilizadas que dificultan el trabajo de los investigadores forenses y las herramientas de seguridad [7].

Con estos antecedentes podemos asumir que los controles de seguridad tradicionales como antivirus, firewall, sistemas de prevención de intrusos (IPS), etc. no han sido eficientes ante malware elaborado para una APT.

Para el presente proyecto se ha considerado como patrón objetivo del ataque una organización del sector financiero, ya que representa el sector actualmente más afectado y sobre el cual se han elaborado y aplicado más normativas de seguridad. Entre las cuales podemos mencionar: a nivel nacional la Resolución de la Junta Bancaria 2148 [8], y a nivel internacional la normativa PCI DSS [9] respaldada por las más grandes emisoras de tarjetas de crédito como VISA y MasterCard.

La estructura de una institución financiera es el modelo de organización que reúne la mayor cantidad de canales electrónicos de pago como son: Pagina Web Transaccional, Aplicaciones para Móviles, Tarjetas de Pago (Débito y Crédito), Puntos de Venta (POS), Cajeros Automáticos (ATM), etc. Y por tanto brinda una superficie de ataque amplia.

Sin duda el activo más importante de las entidades financieras se encuentra en los sistemas informáticos donde se gestiona el dinero en su mayor parte de forma electrónica. Al estar estas instituciones conectadas al mundo por medio de sus canales electrónicos se exponen a organizaciones criminales estructuradas y capacitadas para elaborar ataques de APT.

El riesgo para las entidades financieras radica no únicamente en el robo de dinero sino también en el desprestigio al ser víctimas de un ataque de este tipo, esto es algo muy importante, puesto que una entidad financiera se fundamenta y crece por la confianza que tienen sus clientes para consignarle su patrimonio.

1.3 Solución Propuesta

Realizar un estudio sistemático de las metodologías usadas en el análisis de malware, que permitan identificar su comportamiento y comprender las técnicas de infección, evasión, propagación, etc. que vulneran los controles de seguridad tradicionales, con la finalidad de establecer una arquitectura de seguridad basada en las mejores prácticas.

Análisis

A diferencia de la detección y tratamiento del malware tradicional donde el proceso es detectar, analizar, limpiar y actualizar, en una APT se analiza malware avanzado y el proceso incluye además: un análisis de indicadores de compromiso, análisis de tráfico de red, ingeniería inversa, análisis estático de cadenas de texto, dependencia de librerías, etc. Análisis dinámico de comportamiento y heurística, emulación de objetivos de ataque por medio de virtualización de redes y de sistemas operativos entre otras.

Las técnicas de análisis siempre estarán orientadas en responder preguntas como ¿Que ataca?, ¿Cómo lo hace?, ¿Cómo se propaga? y ¿Quién está detrás del ataque?.

Es por eso que en el presente proyecto de tesis se pretende ir más allá de un análisis sistemático solamente y dar las pautas para realizar un análisis de tendencias y peculiaridades de codificación que también es fundamental a la hora de identificar la procedencia de un malware. Como ejemplo podemos mencionar que un desarrollador latinoamericano no codifica un programa de computador igual que un desarrollador asiático.

También se debe considerar que en muchos casos los desarrolladores reutilizan los mismos componentes de código, que denotan las mismas peculiaridades, en diferentes muestras de malware. La finalidad de este tipo de análisis es que nos ayuda a determinar el autor del malware.

Para el estudio se utilizarán muestras reales ejecutadas sobre entornos controlados. La selección de muestras se realizará de acuerdo al tipo empleado

dentro de una APT enfocada en entidades financieras. Entre los cuales se nombra, pero no se limita a: backdoor, keylogger, ramsomware, rootkits, bootkits, cyber worm, spyware, packers, etc.

Se examinarán muestras reales de la base de conocimientos de Cisco ThreatGrid, Kaspersky Lab, Cuckoo SandBox, Malware Open Source de diferentes proyectos de investigación.

Arquitectura de seguridad

En base al conocimiento, las destrezas y la experiencia adquirida a partir del estudio del malware se establecerá el diseño de una arquitectura de seguridad aplicada a una institución financiera.

Trabajar sobre una organización que tiene un nivel de madurez en seguridad informática como una entidad financiera regida por normativas, exigirá que el diseño de la arquitectura cubra las brechas no consideradas por los controles convencionales de seguridad.

Se tomará en consideración, para el diseño, los controles requeridos a nivel de red que permitan gestionar los posibles vectores de ataque como: web, email, file servers, entre otros, que son usados por el malware como medio de transporte y propagación. A nivel de ordenador serán considerados los motores de análisis necesarios para vigilar el ambiente informático donde una amenaza puede coexistir.

El diseño se enfocará en un modelo de seguridad retrospectiva que permita rastrear el malware, que inicialmente no fue detectado como malicioso, para poder trazar su ruta de acción previo a ser detectado.

La correlación de eventos será considerada dentro del diseño, tanto el colector centralizado donde se agrupen los incidentes de seguridad de las diferentes fuentes como firewalls, antivirus, file integrity, IPS, etc. Así como el motor de correlación de eventos para identificar y reconstruir la estructura de un ataque en sus diferentes etapas.

Entre los beneficios de la solución se señala que el diseño de la arquitectura parte de un análisis previo de las nuevas técnicas y mecanismo usados en el malware, por lo que el diseño de la solución contemplará los controles de seguridad necesarios para cubrir los agujeros de seguridad no considerados en las arquitecturas convencionales que han venido siendo vulneradas.

1.4 Objetivo General

Establecer una arquitectura de seguridad que incluya los mecanismos y herramientas necesarias, por medio del análisis estático y dinámico, para mitigar las técnicas de malware avanzado.

1.5 Objetivo Específicos

Los objetivos específicos para este trabajo son:

1. Identificar la problemática, antecedentes relevantes, los objetivos y la metodología adecuada para desarrollar el análisis de malware y la arquitectura de seguridad.

2. Definir el entorno de ejecución y los componentes de una Amenaza Persistente Avanzada.

3. Identificar las propiedades y patrones de comportamiento que definen el malware utilizado en Amenazas Persistentes Avanzadas.

4. Describir el diseño de una arquitectura de seguridad, basada en los resultados del análisis estático y dinámico de malware.

5. Implementar el prototipo de la arquitectura, y evaluar el desempeño simulando ataques en el entorno de una institución del sector financiero.

6. Determinar el nivel de efectividad del prototipo de la arquitectura de seguridad en la detección y contención de ataques avanzados de malware.

1.6 Metodología

El presente proyecto contempla el análisis de varios componentes de software malicioso, mucho de ellos independientes unos de otros, incluyendo para cada componente el diseño e implementación del prototipo de la contramedida defensiva. A partir de esto se ha considerado que la metodología adecuada sería el “Modelo de Prototipos” que consta de las siguientes etapas.

Toma de Datos

Esta etapa es requerida ya que aquí está considerado el análisis de muestras de malware para entender su funcionalidad, las técnicas de programación y su comportamiento en los diferentes entornos.

Planificación

Basado en el análisis y los requerimientos adquiridos se planificará el mejor escenario que permita visualizar la efectividad de contramedidas aplicadas a un tipo de malware.

Modelado y Diseño

Diseño de los controles de seguridad que detecten, mitiguen y en lo posible permitan brindar remediación de los daños causados por un malware particular.

Construcción del prototipo

Desarrollo e implementación de las contramedidas de seguridad consideradas como necesarias a nivel de host y de red.

Implementación de los escenarios de prueba, ambientes virtuales requeridos a nivel de sistemas operativos y redes de datos simuladas.

Desarrollo entrega y retroalimentación

Simulación de las infecciones de malware dentro de los escenarios de prueba y obtención de resultados.

Este proceso se realiza de manera iterativa con los diferentes tipos de amenazas de malware que enfrenta el entorno de una institución financiera.

En la siguiente figura se describe la metodología y las etapas empleadas:

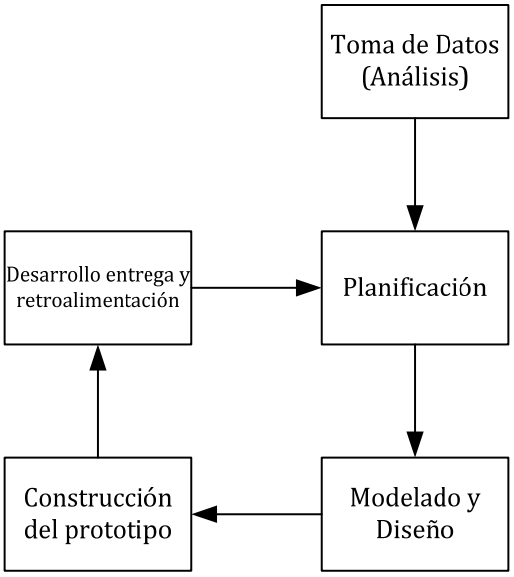

Figura 0.1 Metodología

Nota: Las etapas de “Comunicación” y “Entrega del Desarrollo Final” que constan en la metodología no han sido consideradas dentro de desarrollo del presente proyecto.

CAPÍTULO 2

MARCO TEÓRICO

2.1 Ecosistema del cybercrimen

Se especifica que las amenazas de malware de nueva generación no son utilizadas únicamente en APT sino en el cybercrimen común a nivel mundial.

Es por esta razón que se detalla en la presente sección el Ecosistema del malware que contempla todos los elementos involucrados y cuya finalidad es la generación de lucro por medio de software malicioso. En la figura se identifican los actores involucrados en este ecosistema:

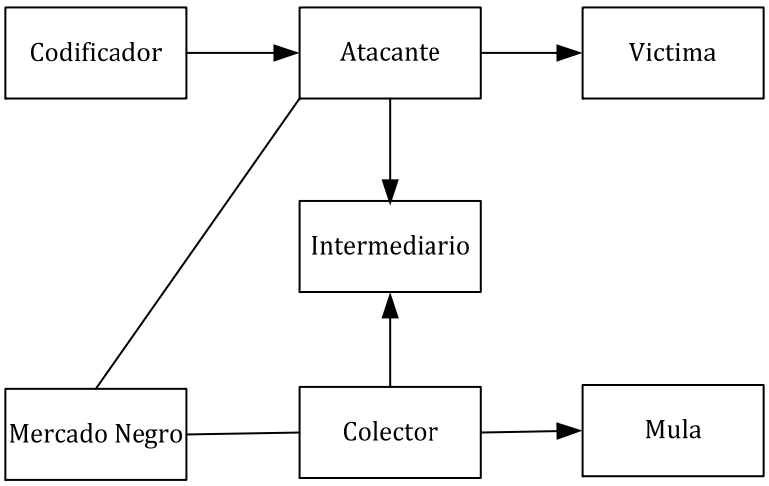

Figura 0.1 Ecosistema del Cybercrimen [10]

2.1.1 Codificadores

Son los programadores que desarrollan el malware y lo ponen a la venta, utilizan contratos de descargo para evadir su responsabilidad en los ataques donde se haga uso de sus productos. Este software malicioso se vende inclusive con contratos de soporte y garantía de uso.

2.1.2 Atacantes

Los atacantes, a veces mal llamados “hackers”, son las criminales que usan el malware para atacar a una institución o a una persona y conseguir información que le permita obtener créditos económicos.

2.1.3 Víctima

Dentro de este rol podemos mencionar que cualquier organización o persona, ya sea que tenga conocimientos de seguridad informática o no, que puede ser víctima del cyber crimen.

2.1.4 Mercado negro

También conocido como la Deep Web, es el espacio cibernético clandestino en internet donde se oferta la información obtenida por el atacante. Es aquí donde los atacantes venden al mejor postor la información obtenida de la víctima.

2.1.5 Colector

Este rol lo cumplen los delincuentes que compran la información ofertada por los atacantes, esto puede ser números de tarjetas de crédito o cualquier información que les permita generar créditos económicos.

2.1.6 Intermediarios

Validan la transacción electrónica monetaria entre el colector y el atacante, cobran de 2% a 3% de transacción.

Surgieron por la necesidad de que los colectores validaran la autenticidad de la información ofertada por los atacantes.

2.1.7 Mulas de Dinero

Una de los inconvenientes del cyber-crimen es como sacar el dinero robado y no dejar huellas para ser rastreado.

Para eso los colectores usan las llamadas “Mulas de Dinero”, son personas incautas que prestan sus cuentas bancarias para el depósito de los fondos mal habidos con el pago de un 10% 0 20% del robo, para después reenviarlo al colector por canales no rastreables como bitcoin, Western Union, etc. Por supuesto ellos no saben de la fuente ilícita del dinero.

Estas personas son generalmente quienes pagan por el crimen.

2.2 Definición de una APT

Las siglas en ingles de una APT corresponden a Advanced Persistent Threat, que traducido significa Amenaza Persistente Avanzada. Se puede definir como un procesos planificado y estructurado por un equipo de personas multidisciplinarias para realizar un ataque dirigido con un objetivo específico, donde se utilizan sofisticadas herramientas de software malicioso [11].

El objetivo de una APT puede ser una entidad o una persona especifica.

La meta detrás de un ataque de APT puede variar según la necesidad, y dependiendo también del origen o de su autoría. Dicho origen o autor puede ser categorizado dentro de cuatro grupos principales: Los Gobiernos cuya finalidad puede ser el espionaje, el sabotaje; Los Criminales cibernéticos que buscan el

lucro; Los Hacktivistas cuyo objetivo básicamente es el reconocimiento de sus ideales; y finalmente los Mercenarios o Paramilitares cuyo propósito depende de quién los contrate. En la siguiente tabla se detallan los grupos, su objetivo y lo que debemos defender ante estos grupos:

Tabla 1 Grupos de origen de un ataque informático [12]

Grupos de origen.	**Objetivo**	**Defender**
Hacktivistas	Buscan Reconocimiento.	Prestigio
Gobiernos	Espectro amplio: Espionaje, sabotaje, etc.	Información confidencial, privacidad personal, prestigio, etc.
Criminales Cybernéticos	Lucro	Dinero
Mercenario/ Paramilitares	Depende quien de los tres anteriores le contrato	Ya que depende de quién los contrate se tiene un espectro amplio.

2.3 Definición de malware

El malware es la palabra compuesta por la combinación de malicious software [12] [13], por definición hace referencia a software desarrollado con un fin malintencionado y engloba todos los tipos conocidos, entre ellos: Virus,

Gusanos, Troyanos, Ransomware, Rootkit, Backdoor, Downloader, Spyware, etc.

2.3.1 Categorización del Malware

Dentro de las categorías se describen a continuación las que engloban tipos de software malicioso considerado como malware:

Virus

Es un código que se puede replicar y requiere de intervención para trasmitirse a otro huésped o sistema informático [14] [13], la infección de un sistema se puede dar por la interacción con el usuario que lo ejecuta e infecta archivos existentes en el sistema. Un virus presenta tres componentes [15] que lo definen:

Ocultador.- Le permite evadir a los antivirus y no ser detectado, ya sea ocultando su actividad o aparentando ser una aplicación no maliciosa.

Payload.- Conocido como carga útil, contiene los módulos de software o instrucciones específicas que serán ejecutadas en el sistema víctima.

Replicador.- le permite al virus realizar varias copias de sí mismo dentro del sistema.

Gusanos

Se pueden auto-replicar, a diferencia de los virus estos no infecta archivos existentes sino que simplemente se instalan o se ejecutan en

la memoria ram [14] [13]. Su finalidad principal es propagarse por una red de sistemas informáticos. Los componentes que lo definen [15] son:

Escáner.- analiza el ordenador huésped en busca de vulnerabilidades.

Herramienta de Pentest.- explota la vulnerabilidad.

Instalador.- evade los mecanismos de protección para instalar el código malicioso.

Payload.- representa el código malicioso o carga útil que lleva el gusano.

Herramientas de descubrimiento.- descubre los sistemas conectados con la máquina víctima y busca un mecanismo de retransmisión.

Entre los tipos más comunes de gusanos podemos listar:

Email.- se propaga mediante adjuntos en correos electrónicos.

IRC (Internet Relay Chat).- se trasmite mediante link en los chats como adjunto.

IM (Instant Messaging).- se aprovecha de las aplicaciones de IM instaladas para acceder al directorio de contactos y retrasmitirse.

File Sharing.- utiliza los protocolos de compartición de archivos para trasmitirse.

Troyanos

Su cualidad es que aparentan ser programas legítimos, se camuflan dentro de software no malicioso, su funcionalidad se asemeja a la de un parásito, se difunden a través de internet [14] [13]. Entre sus principales componentes [15] se listan:

Payload.- es el código malicioso o carga útil de instrucciones ejecutadas en el sistema informático víctima.

Técnicas de Ocultación.- le ayudan a ocultarse y no ser identificado por programas de antimalware.

Renombrador.- le ayuda a cambiar el tipo de archivo por uno no malicioso.

Corruptor.- este componente se encarga de corromper el software de protección de un sistema para evitar la detección.

Código polimórfico.- es el componente encargado de cambiarse el tipo de archivo así mismo cada vez que es ejecutado.

Wrapper o Envoltura.- es la aplicación benigna que cubre el troyano, y lo ayuda a ejecutarse en segundo plano.

Ransomware

Es un software desarrollado para extorsionar a sus víctimas, a las cuales se les muestra un mensaje informándoles sobre una falta cometida (Ver pornografía, usar software pirata, etc.) Pidiendo dinero a cambio de su Exoneración [16] [13].

Rootkit

Su principal función es evadir la detección por parte del usuario o de herramientas automatizadas de seguridad [14] [16]. Cuando arranca antes del sistema operático se lo conoce como bootkit.

Backdoor

También son llamados RAT del inglés “Remote administration Tool” su principal función es brindar acceso a un sistema victima sin autorización [16].

Cuando el atacante trasmite un backdoor a la víctima, el canal de trasmisión se denomina overt channel, mientras que la comunicación entre la victima infectada con un backdoor y el atacante se conoce como covert cannel [15].

Downloader

Son piezas de software utilizados por los atacantes como su nombre lo indica para descargar y ejecutar otro tipo de software malicioso como keylogger, rootkits, etc. [15].

Spyware

Este tipo de malware es desarrollado exclusivamente para colectar información generado por los usuarios en los sistemas informáticos [16].

2.3.2 Clasificación del malware

El malware se puede clasificar como Kernel Mode Malware y User Mode Malware [15], esta clasificación está dada por la protección jerárquica de dominios también conocida como protection rings, la siguiente figura describe la implementación que consta de 4 anillos que brinda los mayores privilegios en el ring0 y menores privilegios en el ring4.

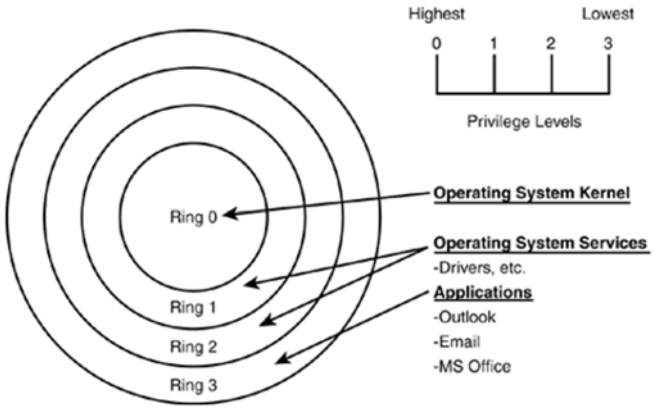

Figura 0.2 Ring Protection [17]

Kernel Mode Malware

Este tipo de malware trabaja a nivel de Ring 0, puede provocar que el enlace directo entre las aplicaciones y el Sistema operativos se vea afectado [15].

User Mode Malware

Este tipo de malware trabaja a nivel de Ring 3, y afecta a archivos específicos sobre los cuales trabajan los usuarios y que han sido infectados de manera particular [15].

2.4 Infección y Propagación

La siguiente figura muestra la distribución por categoría de las aplicaciones más atacadas por exploits que han aprovechado alguna de las vulnerabilidades descritas con anterioridad para la infección y distribución del malware.

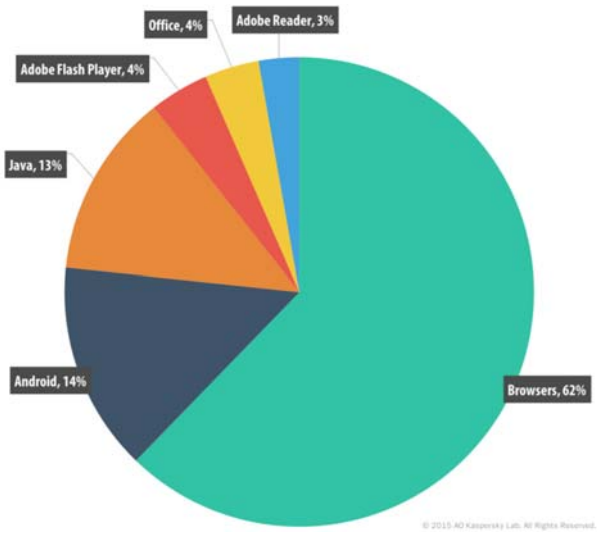

Figura 0.3 Distribution of exploits used in cyberattacks, by type of application attacked [18]

2.4.1 Técnicas de infección

Ingeniería Social

La ingeniería social es una de las técnicas más efectivas utilizadas para realizar una infección. Su efectividad se debe a la falta de cultura en seguridad informática que tienen las personas.

La ingeniería social manipula a las personas ya sea con el fin de obtener información confidencial o hacerlas participes activos del ataque, provocando por ejemplo que ellas mismo infecten sus equipos informáticos [16] [19].

Los vectores de ataque pueden ser muchos, como llamadas telefónicas, chat, email, etc. En estos casos resulta un tanto complejo la aplicación de controles de seguridad.

Adjunto en el Mail

En muchas ocasiones el malware se vale de email con adjuntos, y de software de terceros como medio para infectar un equipo víctima.

En el caso del siguiente ejemplo el software elegido es el de uso común por parte de los usuarios, software de ofimática.

En particular los paquetes de ofimática que ofrecen el uso de Macros, como Excel que incorpora el uso de código Visual Basic, y la posibilidad de programar scripts abre un abanico de posibilidades para la infección de malware.

En la siguiente figura se muestra un ejemplo de un script que se auto ejecuta cuando el usuario abre el documento de Excel y muestra un mensaje de “You are infected..!”

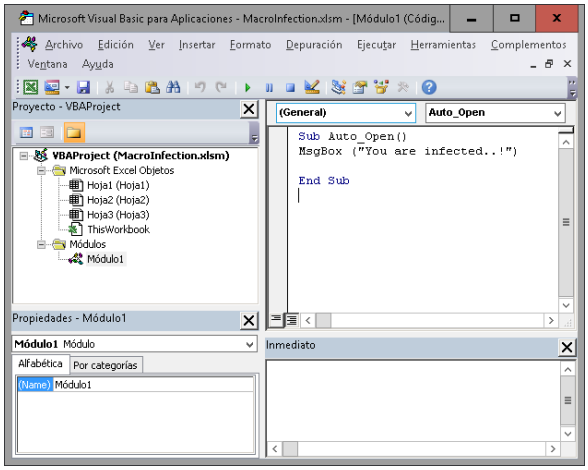

Figura 0.4 Código Auto_Open - Macro Excel

El resultado de abrir el documento Excel se muestra en la siguiente figura:

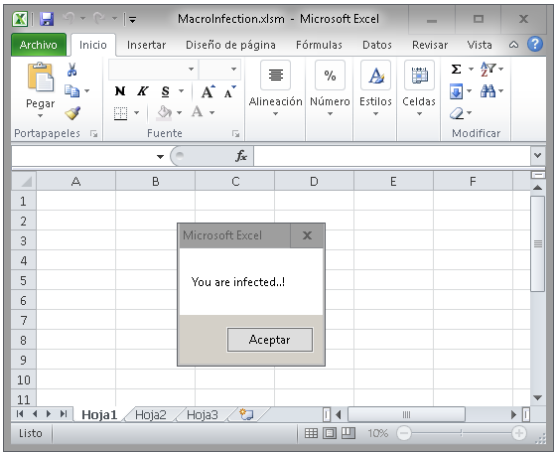

Figura 0.5 Ejecución de una Macro - ejemplo de infección

Un documento de este tipo podría usarse por ejemplo para programar una Macro que descargue y ejecute un script malicioso. La Ingeniería social se utiliza en este caso para que el usuario victima confíe en el documento y lo ejecute en el equipo a ser infectado.

URL maliciosa

Como revisamos el 62% de las aplicaciones atacadas son los navegadores web. Si un atacante cuenta con un exploit podría infectar una máquina con Html, CSS, and Javascript solo con acceder a un sitio web.

Un ejemplo es el exploit Aurora [20] con CVE: CVE-2010-0249 que afecta al navegador web Internet Explorer versión 6, 7 y 8. El mismo le permite al atacante la ejecución de código dentro de la máquina para generar por ejemplo sesiones de conexión reversa contra máquinas de los atacantes.

Memoria Flash USB

Uno de los medios más comunes de distribución de archivos son los Drive USB, existen opciones a nivel de hardware que permiten la infección de una máquina sin importar el nivel de actualización y de parches del sistema Operativo.

Un ejemplo es Rubber ducky [21], este dispositivo usa la especificación en el estándar USB HID (Human Interface Device) o Dispositivo de Interfaz Humana. Es decir, el dispositivo será reconocido como un teclado, por lo que hará el dispositivo USB es teclear comandos, como si lo estuviera haciendo un humano.

Podemos encontrar la tienda en línea HakShop [22] donde se pueden adquirir estos dispositivos. Además, en el repositorio de GitHub [23], se encuentran ejemplos de scripts que se pueden usar con la herramienta.

Aparte de las herramientas comerciales se pueden usar dispositivos que prestan versatilidad y flexibilidad como Arduino para la emulación de teclados y que permiten “ingresar” pulsaciones del teclado.

De esta manera se burlan los controles de seguridad a nivel de host. Estos controles se implementan con herramientas como Active Directory y Consolas centralizadas de antivirus que bloquean la conexión de Drives en las entradas USB, pero siempre deben permitir conectar teclados y ratones USB.

2.4.2 Explotación de vulnerabilidades

Para entender cómo se realiza una infección dentro de un sistema informático es importante primero revisar los conceptos teóricos de los archivos ejecutables, vulnerabilidades y exploits.

En el caso del presenta proyecto de titulación enfocado en entidades bancarias se realizará el análisis sobre el sistema operativo Windows de Microsoft ya que es el de mayor uso y difusión en el sector financiero [24].

Ejecución de un programa

Para entender la forma en que se aprovechan las vulnerabilidades de los programas es primero necesario entender cómo se ejecuta un programa.

En la ejecución de un programa están involucrados varios actores, entre los que destacan la memoria RAM y el procesador, y la ejecución depende mucho de la arquitectura del procesador sobre el cual está trabajando, entre las arquitecturas de procesadores las más importantes están: X86 con sus derivados y ARM. En general todos tienen un enfoque similar que se describe en este apartado.

Para realizar la ejecución debemos repasar tres conceptos fundamentales.

Registros

Los registros son un espacio de memoria integrado al procesador, de ahí su alta velocidad pero poca capacidad [25].

Hay dos registros importantes a tomar en consideración en el desarrollo de exploits: EIP que apunta a la dirección en memoria donde se encuentra la siguiente instrucción a ejecutar y ESP apunta a la siguiente dirección de memoria en la pila.

Pila

Es una estructura de datos que permite el almacenamiento de espacios de memoria [26], la pila tiene una implementación LIFO, el dato importante dentro del análisis de malware es considerar que en la misma se almacena información relativa al registro ESP o EBP.

Offset

Es un desplazamiento en memoria [27].

En la ejecución de un programa se cumplen los siguientes pasos:

1. Paso: Cuando un usuario da doble clic para abrir un programa, este carga sus instrucciones a la memoria RAM. Dentro de la memoria RAM un programa se organiza en diferentes secciones, entre las más importantes listamos: .BSS, .DATA, .TEXT, .CODE.

   Estas secciones están organizadas dentro de un espacio de memoria de manera secuencial. En la siguiente tabla se muestran la forma en que los segmentos de un programa se cargan en memoria.

Tabla 2 Composición de la Pila o Stack

Stack Frame Previo
Argumentos
Dirección de retorno EIP o RET
Puntero Base Guardado EBP
char nombre(32)
Espacio no reservado

En la tabla anterior se ilustra un programa que recibe una variable llamada nombre con tamaño de 32 bits y que recibe una entrada de usuario no controlada. El código correspondiente al programa es el siguiente:

```
#include <string.h>
#include <stdio.h>
void func (char *arg)
{
char nombre 32[]
strcpy (nombre *arg);
printf("\nBienvenido%s\n", nombre);
```

```
}
int main  (int argc, char *argv[])
{
if ( argc !=2){
printf("Uso: %s NOMBRE\n", argv[0]);
exit(0);
}
func(argv[1]);
printf("Fin del programa\n\n");
return 0;
}
```

2. Paso: Leer la tabla de Import, identificar las DLLs requeridas y mapear en el espacio de memoria correspondiente, en la siguiente figura se muestra como se cargó la DLL Kernel32.dll en memoria:

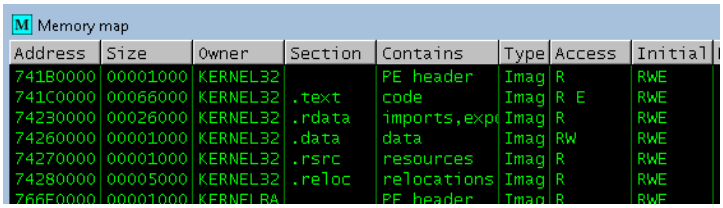

Figura 0.6 Mapeo de memoria – DLL Library

3. Paso: El procesador utiliza la unidad de Control para traer el punto de entrada de las instrucciones que están en memoria RAM para ejecutarlas.

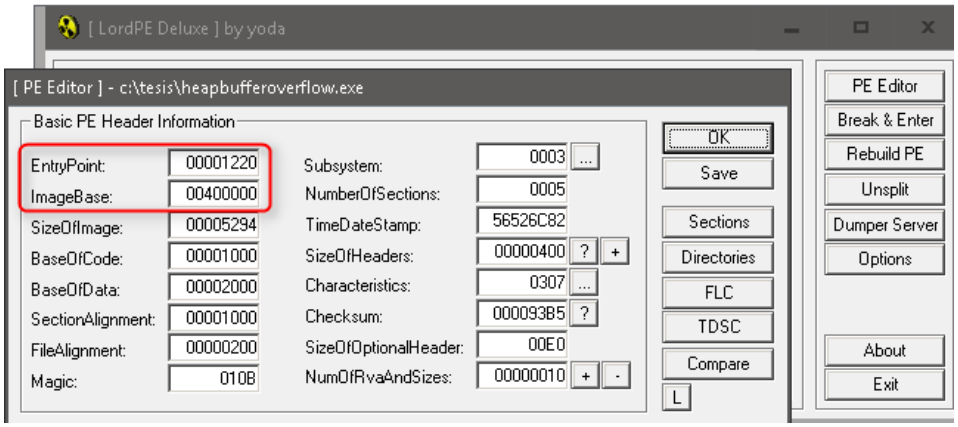

Figura 0.7 Dirección de memoria de la primera instrucción

En el caso del ejemplo los valores de EntryPoint más ImageBase definen la dirección de memoria donde se ubica la primera instrucción.

00001220 + 00400000 = 0x401220 (Valores en Hexadecimal).

En la siguiente figura cargada de Immunity Debbuger podemos ver cuál es la primera instrucción que será ejecutada.

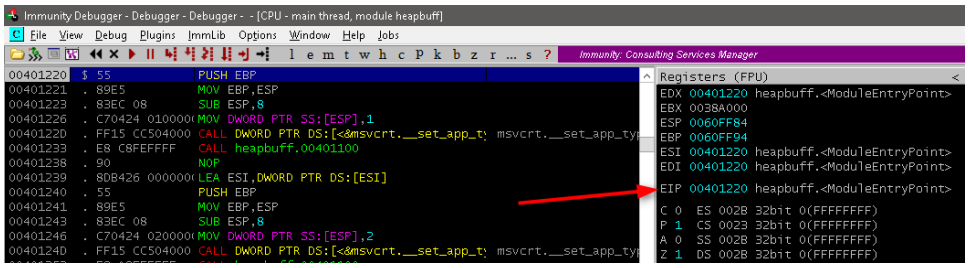

Figura 0.8 Registro EIP

Las instrucciones tienen un orden de procesamiento determinado por el registro EIP del procesador.

4. Paso: La unidad de control decodifica la tarea de la instrucción y las procesa en un orden secuencial.

Vulnerabilidades de software

El software es diseñado y desarrollado por personas y por ende implícitamente va a tener errores, tanto de programación como de diseño.

A pesar de las buenas prácticas y guías de desarrollo seguro como la "OWASP Developer Guide" los bugs de programación y en consecuencia los posibles huecos de seguridad siempre están presentes, entre los más comunes detallamos:

Buffer overflow

Cuando un usuario ingresa una entrada mayor a 32 bits (XXXX XXXX XXXX), se sobrescriben en memoria a las direcciones más altas de memoria como se muestra en la siguiente figura:

Tabla 3 Buffer overflow

Stack Frame Previo
Argumentos.
XXXXXX

XXXXXX
XXXXXX
Espacio no reservado

Para aprovechar este mecanismo se utiliza un string (XXXX XXXX 010203) que sobrescriba en el segmento de memoria de “EIP o RET” una dirección de memoria “010203” que apunte a un segmento especifico de código:

Tabla 4 Buffer overflow – redirección

Stack Frame Previo
Argumentos.
010203
XXXXXX
XXXXXX
Espacio no reservado

De esta manera se puede controlar el flujo del programa, de ahí que la dirección 010203 debe apuntar a un segmento de código también proporcionado por el atacante.

Stack Buffer Overflow

Este tipo de vulnerabilidades afecta cuando el stack crece fuera del segmento de memoria asignado sobrescribiendo los otros segmentos.

```
#include <string.h>
void functionFunction(char *param, int p2, float p3)
{
  char local[10];
  strcpy(local, param);
  printf("You entered: ");
  printf(local);
  printf("\n\n");
  system("PAUSE");
}
int main(int argc, char** argv)
{
  functionFunction(argv[1], 42, 3.14);
}
```

La ejecución del programa:

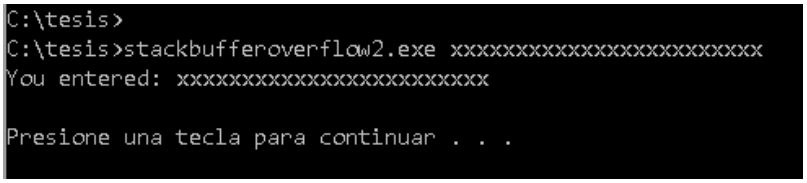

Figura 0.9Stack Buffer Overflow – Ejecución

El resultado:

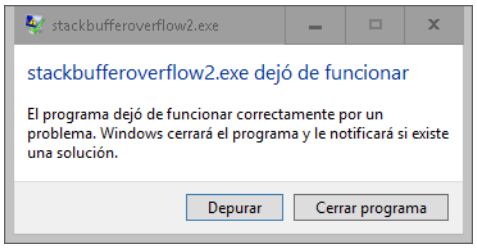

Figura 0.10 Error de ejecución Stack Buffer Overflow

Heap Buffer Overflow

Este tipo de vulnerabilidades ocurre cuando el segmento de data crece fuera del segmento de memoria asignado sobrescribiendo los otros segmentos.

Código fuente de ejemplo:

```
#include <string.h>
#include <stdio.h>
void main(int argc, char** argv)
{
  char* buff=malloc(20);
  strcpy(buff, argv[1]);
  free(buff);
```

```
}
```

Ejecución:

```
C:\tesis>heapbufferoverflow.exe xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
```

Figura 0.11 Heap buffer Overflow

El resultado:

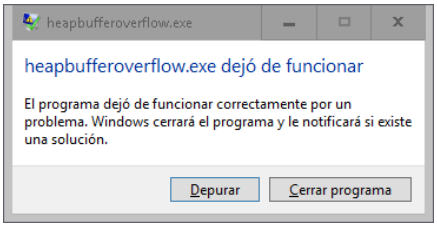

Figura 0.12 Error de ejecucion Heap buffer Overflow

Existen otros tipos de vulnerabilidades como Format Strings y Section overflow que serán analizadas con más detalle en la sección de análisis de malware.

Exploit

Un exploit es una parte o modulo importante dentro del malware. Se define como un módulo de software que se ejecuta contra un sistema o aplicación que contiene una carga útil o payload [13].

Zero Day

Existe un mercado muy activo en la deepweb donde la oferta y demanda de exploits así como el valor monetario depende de algunos factores

como el nivel de privilegios obtenido dentro de un sistema al ejecutar el exploit, el modo de ejecución y el sistema afectado.

En la siguiente figura se puede ver una tabla obtenida del sitio web de ZERODIUM con el rango de pagos donde se clasifican los exploits y el precio correspondiente que la organización está dispuesta a pagar a los desarrolladores.

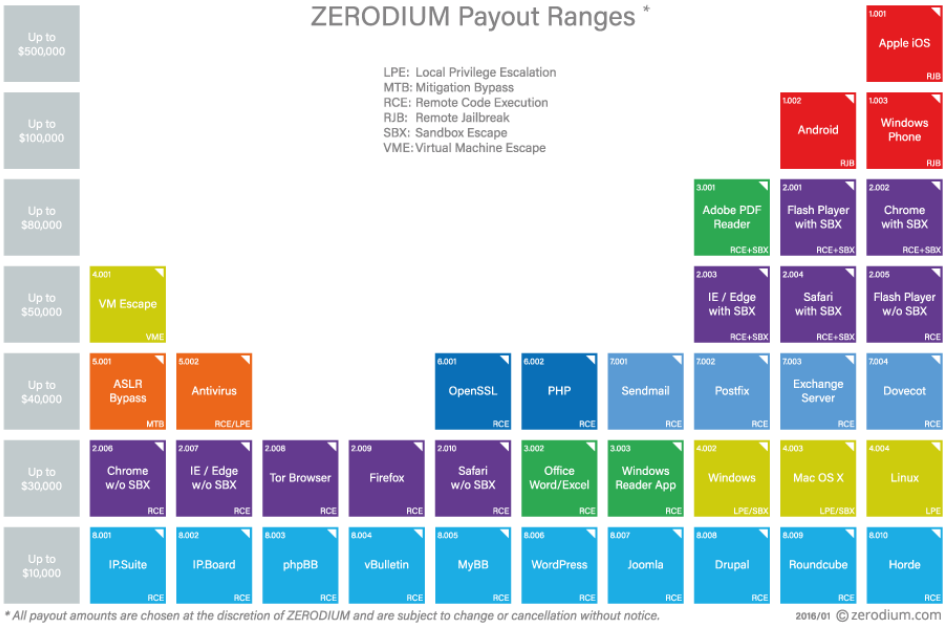

Figura 0.13 Rango de pagos, tomado de ZERODIUM [28], Recuperado el 15 junio 2016

En el blog Spiderlabs [29] se detalla el anuncio de un desarrollador que ofrece un Zero Day para escalar privilegios de manera local dentro de un Sistema Operativo Windows, que afecta a todas las versiones desde Windows 2000 hasta Windows 10. Para el cual acepta ofertas a partir de los 97.000 USD.

Después de los zero day para explotación remota, los de elevación local de privilegios son los más buscados.

Dispositivos externos

Los dispositivos móviles permiten varias formas de infección, algunas similares al caso de los computadores de escritorio o laptops, sin embargo, un método propio de este tipo de dispositivos es la infección por medio del cable de datos que utiliza el puerto USB para la transmisión de datos y la carga de energía.

Kaspersky [30] muestra en detalle el análisis realizado respecto de estos métodos de infección.

2.4.3 Metodologías de propagación

Una vez que ha sido posible infectar el primer equipo dentro de una organización, también conocido como paciente cero, se utilizan el entorno de red de área local para la propagación de malware en los otros equipos presentes en la red.

Reconocimiento

En esta primera etapa ya sea bajó supervisión del atacante o por la propia programación del malware, se realiza un reconocimiento de los protocolos a nivel de red que se pueden utilizar.

En este caso los protocolos más utilizados son:

- NetBIOS
- Samba para archivos compartidos
- FTP Servidores de archivos

Estos protocolos permiten la distribución de archivos infectados entre dispositivos de la red interna.

Contagio

Este mecanismo depende del reconocimiento previo y de cómo haya sido implementada cada amenaza.

La ingeniería social es una de las técnicas, por ejemplo, un archivo con el nombre nomina.xlsx con un macro que instale el malware para que sea el propio usuario el que realice la copia e instalación.

Otra forma de contagio es el abuso de los mecanismos de actualización automática de los sistemas operativos. Un ejemplo es el malware conocido como flame, donde el paciente cero realizaba un ataque de hombre en el medio para interceptar las solicitudes de Windows Update y emulando ser un proxy, mediante el protocolo WPAD (Web Proxy Autodiscovery Protocol), distribuía paquetes de actualización falsos a los demás equipos de la red que lo instalaban en el sistema de manera automática. Esto era posible ya que flame firmaba los paquetes de actualización con un certificado digital robado.

2.5 Comando y Control

Comando y Control, también Llamado por su abreviatura C2 o C&C es la consola de administración que utilizan los atacantes para controlar las actividades del malware presente en máquinas infectadas [31].

Arquitectura de C&C

Una arquitectura para C&C se puede definir de dos maneras. Centralizada en donde se tiene un único servidor de administración centralizada. La otra arquitectura utilizada para C&C es la Distribuida, donde todos los componentes actúan como cliente y como servidor a la vez.

Técnicas de evasión

Dado que las consolas de comando y control deben estar “publicadas” en internet y disponibles, en espera de las conexiones provenientes de las máquinas infectadas, los administradores de C2 utilizan técnicas de anti trazado para evitar su detección en el análisis de malware.

Fast Flux DNS.

Esta técnica previene la identificación de las direcciones IP pertenecientes a los servidores donde está instalado el comando y control.

El proceso consiste en aprovechar el mecanismo de balanceo de carga del sistema DNS. Este mecanismo permite el uso de varias direcciones IP para un mismo nombre de domino, al realizar registros y des registros continuos de

varias direcciones IP se logra evadir los controles de ACL (Listas de control de acceso) basados en IP.

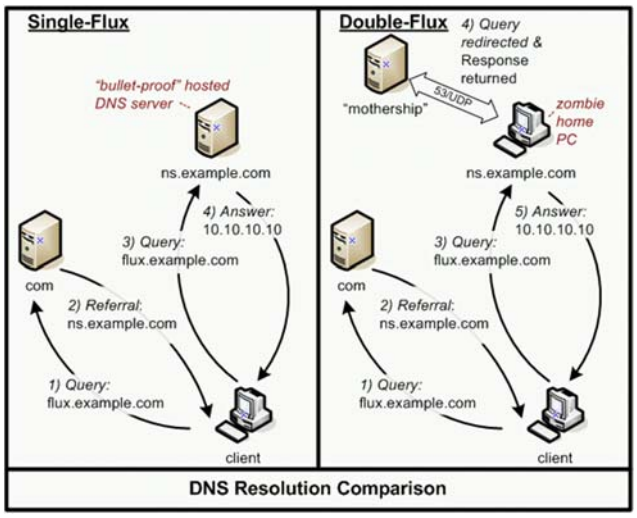

Figura 0.14 Fast Flux DNS [31]

La forma de mitigar esta técnica es el bloqueo no de las direcciones IP, sino del nombre de dominio usado por el servidor C2.

Proyecto Tor

El proyecto Tor conocido como un anonymizer impide el análisis del tráfico de red, con el uso de nodos que ocultan el origen del cual proviene una comunicación.

La red Tor permite publicar servicios llamados hidden services (HSs) usando Hidden Service Directories (HSDs). Este se puede resumir en la siguiente figura:

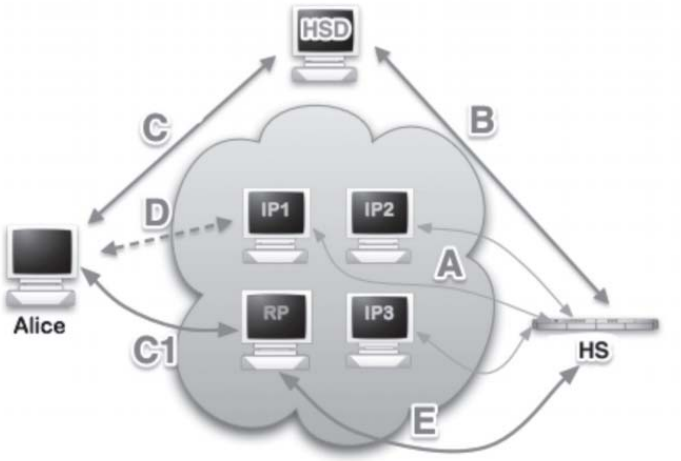

Figura 0.15 Tor Hidden Service Protocol [32]

El HS selecciona un conjunto de nodos como sus puntos de entrada (P1, P2, P3) [A]. HS crea su hidden service que contiene una llave pública (PK) y una dirección del tipo XYZ.onion asociada a sus puntos de entrada. Esta dirección se registra en la base de datos HSD [B].

La dirección .onion es usada por el cliente para contactar el HS [C]. El cliente crea un nuevo circuito virtual y prepara un punto de encuentro RP (rendezvous point)[C1]. El cliente usa los puntos de entrada (P1, P2, P3) para informar al HS sobre el RP [D]. Finalmente, el HS crea un circuito virtual con el cliente a través del RP para iniciar la comunicación.

En la página de Tor [33] se tiene una guía para la implementación de Hidden Services sobre la red Tor.

Técnicas de comunicación

HTTP

El protocolo HTTP, permite el uso de varias técnicas de comunicación como los métodos POST/GET, y el uso de servicios web

Así también http tiene mecanismos de Cifrado de datos como SSL y TLS que permite cifrar la comunicación con el servidor C&C.

Internet Relay Chat

IRC es un protocolo de comunicación, de los primeros en ser usados por su sencillez, muy difundido y utilizado en redes de botnet o redes zombies, este permite la comunicación entre el malware de los equipos infectados y los servidores de C2.

Peer-to-Peer

Este tipo de comunicación es utilizada en las arquitecturas de C&C distribuidas. Se usa para evitar la pérdida de comunicación con los miles de equipos infectados solo por el bloqueo de un dominio o dirección IP del servidor central C&C.

La comunicación peer to peer introdujo el concepto de Botnet resilience, ya que resulta difícil bloquear la comunicación cuando se mantiene enlaces múltiples con varios nodos donde cada uno de ellos puede funcionar como una consola de administración C2.

CAPÍTULO 3

LEVANTAMIENTO DE INFORMACIÓN

3.1 Antecedente de ataques a entidades financieras

Carbanak [6] es una herramienta de administración remota con propiedades de espionaje, ex filtración de datos a nivel de host. Fue uno de los ataques con más relevancia que se ha presentado en los últimos años. En el 2013 varios bancos fueron atacados. Se llegó a determinar que Carbanak fue elaborado a partir de Carberp [6], considerado como un troyano modular que roba información y afectó principalmente a entidades financieras de Rusia, en los últimos análisis

realizados a Carbanak ya no se encontraron muestras de código perteneciente a Carberp siendo un desarrollo totalmente re editado.

La infección inicial se realizó mediante phishing con email de comunicados bancarios aparentemente legítimos. Después de explotar vulnerabilidades conocidas presentes en Microsoft Office Word el malware instalaba un backdoor con el nombre de Carbanak.

Como parte del reconocimiento carbanak grababa videos de la pantalla de las actividades que realizaban los empleados del banco y los enviaba a los servidores C2.

A nivel de red, el malware utiliza movimientos laterales con reconocimiento de red con la finalidad de identificar los sistemas críticos en la infraestructura del cliente.

El ataque se extendía a los dispositivos ATM (Automated Teller Machines) para dispensar dinero en efectivo, usando mulas para captar el dinero.

También se utilizó la red del sistema SWIFT (Society for Worldwide Interbank Financial Telecommunication) para realizar transferencias a las cuentas del grupo atacante.

Las Bases de Datos Oracle utilizada como estándar de facto en las instituciones financieras se atacaban para crear cuentas ficticias de clientes y realizar transferencias y el retiro de dinero también con la ayuda de mulas.

3.2 Estructura de una entidad financiera

La estructura de una entidad financiera consta de dos áreas que sostienen la infraestructura tecnológica. El área de Tecnología de la Información, y el Área de Seguridad de la Información que engloba a su vez varios departamentos que forman el Gobierno de Seguridad de la Información.

3.2.1 Gobierno de Seguridad de la información

El gobierno de seguridad de la información está conformado por un comité que costa de tres miembros principales:

Tabla 5 Gobierno de Seguridad de la información

Integrante	**Rol**
Delegado del Gerente de la Institución	Presidente
Representante Legal de la Institución	Miembro
Representante de Seguridad de la Información	Secretario

Este comité es responsable de crear un reglamento y estatutos que conforman la política de seguridad de la información.

3.3 Canales electrónicos bancarios

3.3.1 Banca Web

Consta de un portal web al que se puede acceder por medio de un navegador web, su uso está orientado principalmente a las computadoras de escritorio

El acceso generalmente se maneja con doble factor de autenticación, las credenciales de usuario y un segundo factor que puede ser un token, un SMS o un correo electrónico de donde se obtiene un código de verificación que se debe ingresar en el portal previo al acceso.

Utiliza comunicación cifrada HTTPS mediante SSL o TLS y el uso de certificados digitales firmados por una tercera entidad de confianza.

Los certificados deben ser SSL de validación extendida conocido como SSL EV, esto permite que la barra de URL en el navegador se ponga del color verde y se muestre el nombre de la entidad, como se muestra en la siguiente figura:

Figura 0.1SSL EV Extención validada

Otra característica que se está haciendo muy común es el uso de navegadores seguros, un ejemplo es el servicio Safe Money que ofrece la empresa kasperky y ayuda a mejorar la seguridad cuando se realizan compras y transacciones en línea, el servicio básicamente impide que se realicen capturas del teclado, captura de pantalla ya sean imágenes o video.

3.3.2 Banca Celular

La Banca celular le permite al cliente final usar los servicios electrónicos por medio de SMS y llamadas telefónicas.

Esta modalidad se usa generalmente para servicios de consulta y se integra con un único número celular.

3.3.3 Aplicaciones móviles

Este aplicativo móvil para teléfono inteligente o Tablet generalmente está disponible para las plataformas de Android e iPhone.

Al momento de la instalación en el dispositivo personal del cliente se realiza un proceso de verificación de datos personales para asegurar la valides de la fuente. Dependiendo de las funcionalidades implementadas permite realizar consultas y transacciones financieras.

3.3.4 ATM

El ATM (Automated Teller Machine) o cajeros electrónicos que en sus inicios únicamente se utilizaba para expender el dinero de las cuentas

de los clientes y realizar consultas, constantemente ha ido evolucionando hasta los equipos de hoy en día donde se puede realizar depósitos de dinero.

3.3.5 Banca telefónica

Esta es una de las primeras modalidades de canales electrónicos utilizadas por las entidades financieras, el uso de centrales telefónicas más sofisticadas a permitido a las entidades financieras la elaboración de Respuesta de Voz Interactiva (IVR) con menús muy completos para la atención a los clientes.

3.4 Soluciones antimalware existentes

3.4.1 File Integrity Monitoring

Un FIM es un herramienta que permite llevar un control de los cambios que se realizan en los archivos que residen dentro de un sistema informático, ya sean estos archivos del sistema operativo o de aplicaciones instaladas, esto permite bloquear los intentos de instalar software malicioso o la modificación mal intencionada en propiedades del sistema [34].

Antes de instalar una herramienta de este tipo se aconseja realizar un análisis con la finalidad de depurar cualquier anomalía y realizar el control sobre una imagen limpia del sistema, caso contrario se corre el

riesgo de ocultar una amenaza latente que previamente ya se encontraba en el sistema.

3.4.2 Antimalware

Dentro de esta sección se consideran las soluciones de software como anti-virus, anti-ransomware, anti-exploits, etc.

Este tipo de software básicamente contiene una base de datos con entradas que le ayudan a identificar malware. Estas entradas pueden ser de diferente tipo como las clásicas firmas o métodos sofisticados de identificación.

Una entrada de tipo firma es una secuencia continua de bytes identificados en una muestra de malware, en la siguiente figura se muestra un ejemplo del mismo:

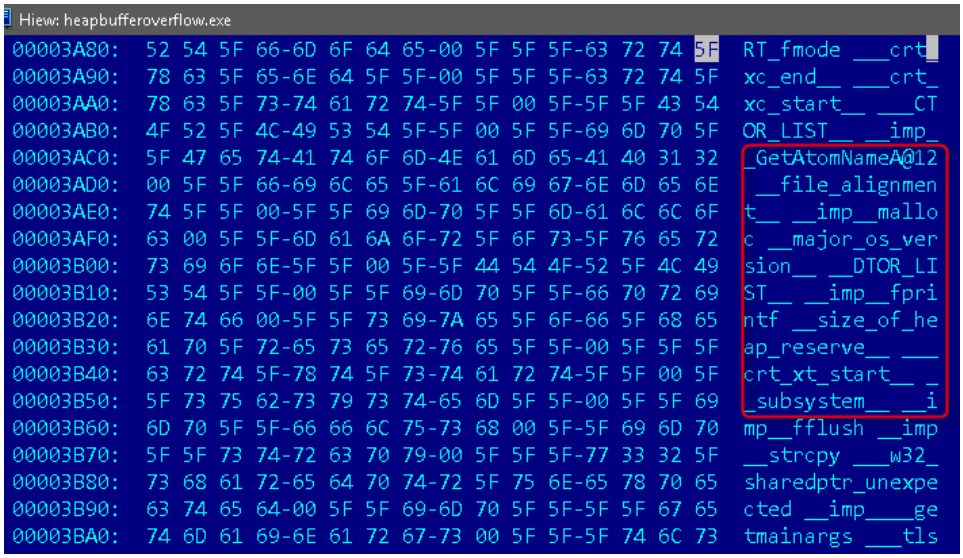

Figura 0.2 Firma de malware

Sin embargo, la identificación por medio de firmas se puede evadir con el uso de métodos de ofuscación.

Entre las técnicas sofisticadas podemos identificar dos categorías principales:

Heurística

La Heurística se basa en la identificación de comportamiento anómalo [15], lo que hace esta técnica es leer las instrucciones del programa para identificar rutinas de instrucciones sospechosas, después se realiza un monitoreo de las actividades de la muestra en un entorno virtual emulado.

Sandboxing

Sandboxing es puramente un análisis dinámico [35] en un entorno controlado. Sandbox contempla varias técnicas como el análisis de tráfico de red, la emulación de sistemas operativos y la emulación de redes. Existen diferentes implementaciones de Sandbox como:

- Enjaular procesos dentro de un espacio de directorios específico.
- Aislamiento basado en entornos de máquinas virtuales independientes del sistema operativo hospedero.

Diferentes plataformas han implementado tecnologías de Sandbox particulares, en el caso del Kernel de Linux, se cuenta con frameworks como AppArmor y SELinux que manejan un modelo de ejecución basada en reglas.

Recientemente los lenguajes de nueva generación como HTML5 han implementado también soluciones como el atributo sandbox que permite la ejecución de código de manera aislada dentro de un iframe.

3.5 Selección de muestras de malware

La selección de malware para analizar se dio en base al tipo de malware utilizado en el sector financiero, se listan a continuación:

Tabla 6 Repositorios de Malware

Nombre	**URL**	**Detalle**
Virus Share	https://virusshare.com/	El acceso se permite mediante invitación previa solicitud dirigida al administrador
Avcaesar	https://avcaesar.malware.lu	Acceso autorizado previo registro de cuenta y solicitud de acceso
ThreatGRID CISCO	https://console.amp.sourcefire.com	Acceso abierto al repositorio para los clientes de con

		licencias de FireAMP Antimalware Protection
Mal Code	http://malc0de.com/database/	Acceso abierto al público en general

En la siguiente tabla se describe cada una de las muestras seleccionadas:

Tabla 7 Muestras específicas de Malware

Nombre (Kaspersky)	**Categoría**	**Archivo**	**Descripción**
Backdoor.Win32.IRCBot.gen	Backdoor	61780002.ex	Este ejecutable permite la instalación de una puerta trasera para el acceso no autorizado al sistema.
Trojan.Win32.Agent.hwwe	Clicker	9C4DB7EE	Captura printscreen de los clic realizados por el usuario

HEUR:Trojan.Win32.Generic	DDOS	06377098cd6228696598e66db000f1b0	Afecta el rendimiento del equipo y ejecuta ataques coordinados de denegación de servicios distribuida
Type_VBS_Autorun	Downloader	4c8208ae.vbs_	Este ejecutable descargar, descomprime, instala y ejecuta nuevos módulos de malware
Exploit.VBS.Agent.ae	Exploits	Taskxpl.wsf	Es un script programado en visual Basic Utilizado para escalar privilegios en Microsoft Windows
Trojan.Win32.Filecoder.w	Ransomware	cd3bc0bedb3900ff3c64.exe_	Este ejecutable cifra los archivos de un directorio específico y se comunica con su C2 para notificar sus actividades
Trojan-Spy.Win32.KeyLogger.ahrn	Spyware	Igfxper.exe_	Tiene funcionalidades de captura de pulsaciones y navegación web.

Trojan.Win32.Vilsel.bjru	Troyano	63780001.9287.dbe	Ejecutable con funcionalidades para ocultar procesos y archivos

CAPÍTULO 4

ANÁLISIS Y DISEÑO

4.1 Análisis estático y dinámico de malware

4.1.1 Análisis Estático

En el análisis estático de malware la muestra de malware no se ejecuta [19] y el análisis se realiza sobre la muestra con diversas técnicas como ingeniería inversa y lectura de código ensamblador, acompañado del uso de herramientas que permiten ver las propiedades y los parámetros que describen el comportamiento de la muestra.

Para realizar un análisis estático como recomendación se sugiere modificar la extensión de la muestra, en el caso de un archivo ejecutable

“.exe” modificarlo por “.ex”. Se detallan a continuación las técnicas de análisis estático de malware:

Análisis de la estructura de un Ejecutable.

En la siguiente tabla se detalla la estructura de un ejecutable de Microsoft Windows, información necesaria sobre la cual se realiza un análisis de malware:

Tabla 8 Estructura Base de un ejecutable para el Sistema Operativo Windows

DOS MZ Header
DOS Stub
PE Header
Section Table
Section 1
Section 2
Section …

Section n

La estructura consta de varias secciones y cada una consta de un conjunto de componentes, que se detallan a continuación:

DOS MZ Header

Este segmento que nos permite identificar un ejecutable como tal, inicia con el numero: “0x4D 0x5A” (Hexadecimal), “77 90” (ASCII), que corresponden a MZ las siglas de Mark Zbikowsky arquitecto del formato de ejecutables para MS-DOS.

La siguiente figura es una captura por medio del editor hiew donde se evidencia la Marca MZ:

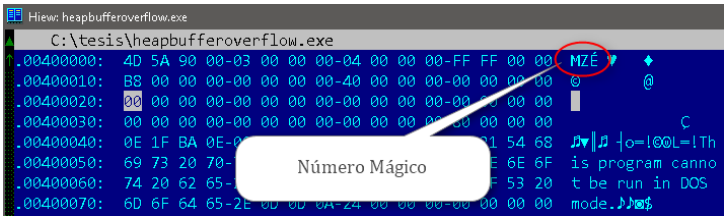

Figura 0.1 MZ_Header

DOS Stub:

Este campo se ha conservado por un tema de compatibilidad con el sistema DOS por propósitos legales.

Su única funcionalidad es mostrar el texto “This program cannot be run in MS-DOS mode.”

PE Header

Esta sección está compuesta por varios elementos, entre los que destacan:

Dos Header: tiene un tamaño de 64 bytes, el tamaño de sus elementos se detalla en el siguiente listado:

```
struct DOS_Header
{
// short is 2 bytes, long is 4 bytes
 char signature[2] = "MZ";
 short lastsize;
 short nblocks;
 short nreloc;
 short hdrsize;
 short minalloc;
 short maxalloc;
 void *ss;
 void *sp;
 short checksum;
 void *ip;
 void *cs;
 short relocpos;
 short noverlay;
 short reserved1[4];
 short oem_id;
 short oem_info;
```

```
short reserved2[10];
long  e_lfanew;
}
```

El componente e_lfanew apunta a la siguiente instrucción a ser ejecutada que representa el inicio de PE-Header. En la siguiente figura podemos ver que e_lfenew apunta a la dirección 0 x 00 00 00 80 en formato Little Endian.

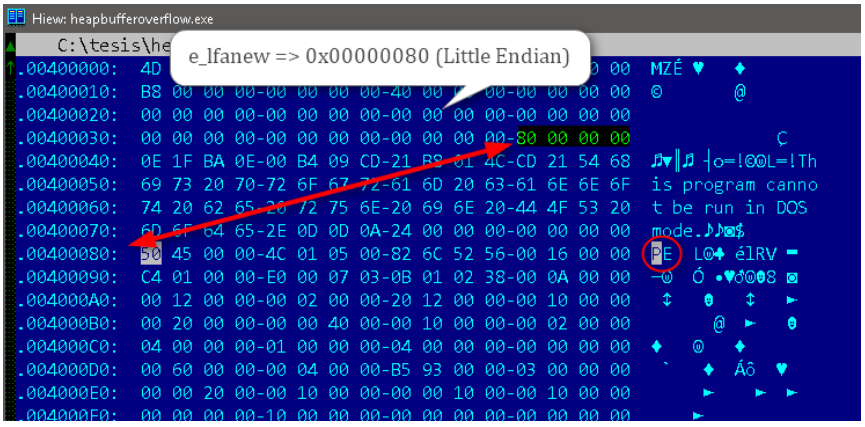

Figura 0.2 Componente e_lfanew

Nota: Little Endian: es una forma de trabajar en memoria donde todo se almacena al inverso de su orden original.

Nt Header

Está compuesta por los siguientes componentes:

```
IMAGE_NT_HEADERS {
DWORD                   Signature;
IMAGE_FILE_HEADER       FileHeader;
```

```
IMAGE_OPTIONAL_HEADER OptionalHeader;
} IMAGE_NT_HEADERS, *PIMAGE_NT_HEADERS;
```

El componente Signature representa los caracteres 0x50 0x45 correspondientes a PE y los identifica como un archivo de tipo ejecutable.

File Header

Los componentes de esta cabecera listan a continuación:

```
IMAGE_FILE_HEADER {
WORD  Machine;
WORD  NumberOfSections;
DWORD TimeDateStamp;
DWORD PointerToSymbolTable;
DWORD NumberOfSymbols;
WORD  SizeOfOptionalHeader;
WORD  Characteristics;
} IMAGE_FILE_HEADER, *PIMAGE_FILE_HEADER;
```

Los componentes de *File Header* describen el esquema físico y las propiedades del archivo como TimeDateStamp que indica la fecha y hora en la que fue compilado.

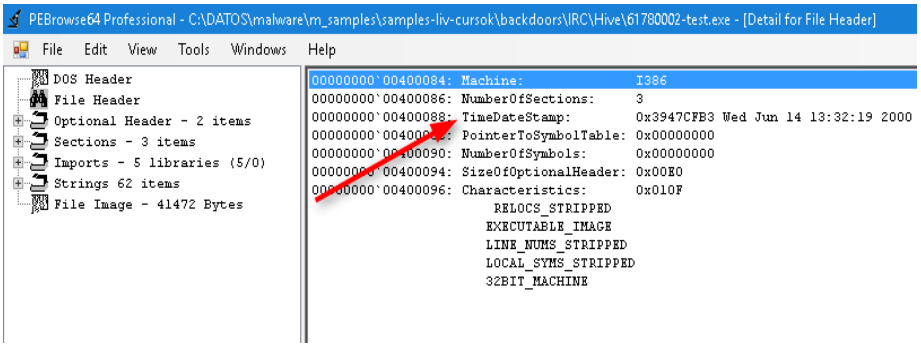

Figura 0.3 File Header TimeStamp

Image Optional Header

```
IMAGE_OPTIONAL_HEADER {

  WORD                  Magic;
  BYTE                  MajorLinkerVersion;
  BYTE                  MinorLinkerVersion;
  DWORD                 SizeOfCode;
  DWORD                 SizeOfInitializedData;
  DWORD                 SizeOfUninitializedData;
  DWORD                 AddressOfEntryPoint;
  DWORD                 BaseOfCode;
  DWORD                 BaseOfData;
  DWORD                 ImageBase;
  DWORD                 SectionAlignment;
  DWORD                 FileAlignment;
  WORD                  MajorOperatingSystemVersion;
  WORD                  MinorOperatingSystemVersion;
  WORD                  MajorImageVersion;
  WORD                  MinorImageVersion;
  WORD                  MajorSubsystemVersion;
  WORD                  MinorSubsystemVersion;
  DWORD                 Win32VersionValue;
```

```
 DWORD                   SizeOfImage;
 DWORD                   SizeOfHeaders;
 DWORD                   CheckSum;
 WORD                    Subsystem;
 WORD                    DllCharacteristics;
 DWORD                   SizeOfStackReserve;
 DWORD                   SizeOfStackCommit;
 DWORD                   SizeOfHeapReserve;
 DWORD                   SizeOfHeapCommit;
 DWORD                   LoaderFlags;
 DWORD                   NumberOfRvaAndSizes;
IMAGE_DATA_DIRECTORY
DataDirectory[IMAGE_NUMBEROF_DIRECTORY_ENTRIES];
} IMAGE_OPTIONAL_HEADER, *PIMAGE_OPTIONAL_HEADER;
```

Uno de los componentes más importantes que podemos encontrar es AddressOfEntryPoint que representa la dirección donde se encuentra la primera instrucción que debe ejecutar.

Los desarrolladores de malware suelen cambiar la dirección de esta instrucción de manera que al ser cargado en memoria por el sistema operativo salte a un espacio de direcciones donde se ejecutan acciones maliciosas.

Image Data Directory

```
IMAGE_DIRECTORY_ENTRY_EXPORT,
IMAGE_DIRECTORY_ENTRY_IMPORT,
IMAGE_DIRECTORY_ENTRY_RESOURCE,
```

```
IMAGE_DIRECTORY_ENTRY_EXCEPTION,
IMAGE_DIRECTORY_ENTRY_SECURITY,
IMAGE_DIRECTORY_ENTRY_BASERELOC,
IMAGE_DIRECTORY_ENTRY_DEBUG,
IMAGE_DIRECTORY_ENTRY_COPYRIGHT,
IMAGE_DIRECTORY_ENTRY_GLOBALPTR,
IMAGE_DIRECTORY_ENTRY_TLS,
IMAGE_DIRECTORY_ENTRY_LOAD_CONFIG,
IMAGE_DIRECTORY_ENTRY_BOUND_IMPORT,
IMAGE_DIRECTORY_ENTRY_IAT,
IMAGE_DIRECTORY_ENTRY_DELAY_IMPORT,
IMAGE_DIRECTORY_ENTRY_COM_DESCRIPTOR,
IMAGE_NUMBEROF_DIRECTORY_ENTRIES
```

Entre los componentes más importantes tenemos *Resource* que detalla los recursos del sistema que serán demandados por el ejecutable.

Security es el componente que apunta a la ubicación de los certificados en el caso de que la aplicación este firmada por un fabricante valido.

TLS (Thread-local storage) es un Second Entry Point, que es utilizado como un segundo punto de entrada para la ejecución inicial de un programa.

La *Export Table* y la *Import Table* que nos indican cuales son las librerías del sistema utilizadas por el programa.

La siguiente figura muestra una captura del programa LordPE donde se listan las DLLs importadas por un programa ejecutable:

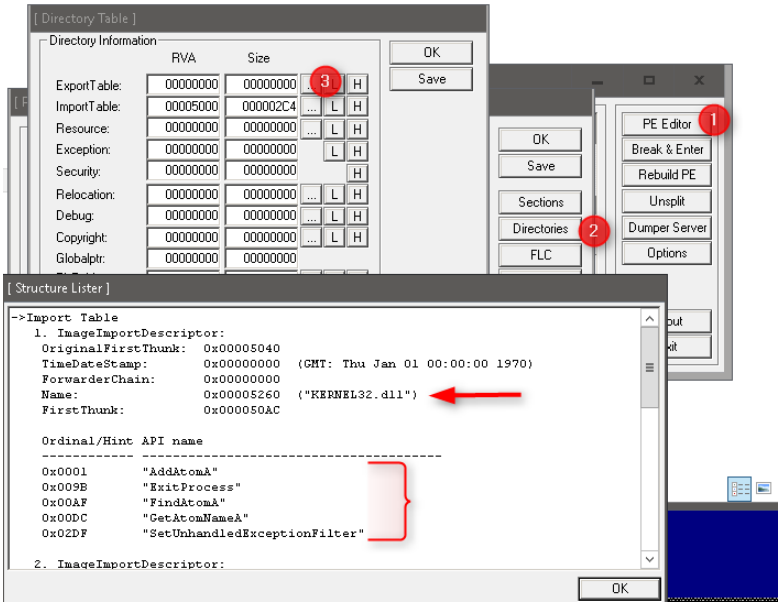

Figura 0.4 Lord PE - Import Table

Para ver información en la tabla de export podemos abrir cualquier DLL del sistema dentro del directorio C:\Windows\System32, la siguiente figura muestra un ejemplo con la ActionCenter.dll, que tiene a disposición dos funciones: DllCanUploadNow y DllGetClassObject.

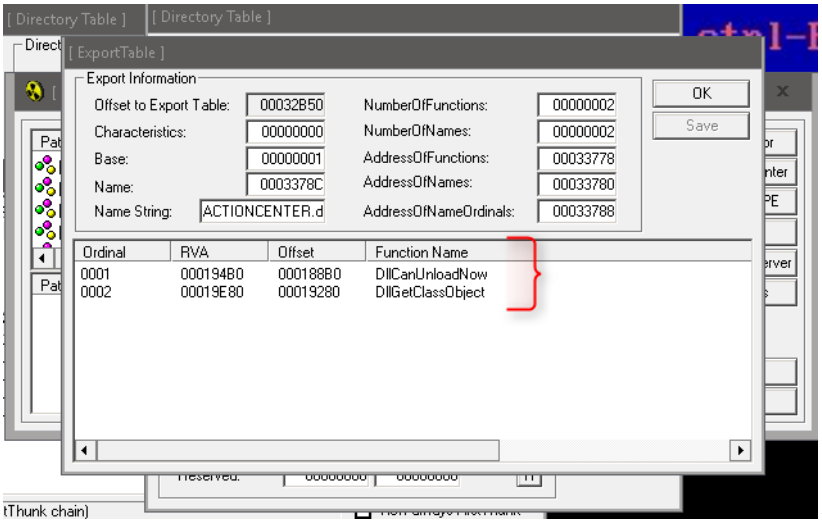

Figura 0.5 Lord Pe - Export Table

Section Table

Esta sección es especialmente importante ya que contiene los valores de las direcciones en la memoria virtual, tamaños virtuales y virtual offsets, etc. Referentes a la memoria Virtual con la que trabaja el computador. En la siguiente figura se muestra la estructura de Section Table.

```
IMAGE_SECTION_HEADER STRUCT
    Name1                   BYTE
    union Misc
        PhysicalAddress     DWORD
        VirtualSize         DWORD
    ends
    VirtualAddress          DWORD
    SizeOfRawData           DWORD
    PointerToRawData        DWORD
    PointerToRelocations    DWORD
    PointerToLinenumbers    DWORD
    NumberOfRelocations     WORD
    NumberOfLinenumbers     WORD
    Characteristics         DWORD
IMAGE_SECTION_HEADER ENDS

IMAGE_SIZEOF_SHORT_NAME equ 8
```

Figura 0.6 Section Table

Entre los principales componentes están: PointerToRawData que me indica la dirección donde apunta la sección de datos. En la siguiente tabla se detallan los *section type* que son cargados en memoria al correr un ejecutable en el sistema:

Tabla 9 Section Type

Section Type	**Detalle**
.text	Contiene el código ejecutable, sus permisos son de ejecución y de lectura únicamente pero los desarrolladores de malware pueden cambiar los permisos para escribir o modificar el código en tiempo de ejecución.

.data	Contiene las variables del programa, también conocida como .rdata y .bss
.rsrc	Contiene los recursos del programa, como iconos, cursores y bitmaps. En esta sección es donde suele ocultar partes de código una aplicación de malware.
.edata	Este es el principal directorio de export para las dlls.
.idata	Tabla de direcciones de import.

Memoria Virtual

Para un análisis correcto hay que tomar en consideración que un programa es mapeado desde el disco duro a un espacio de memoria con direcciones virtuales, esto se consigue por medio del mapeo que traduce la dirección virtual a una dirección física.

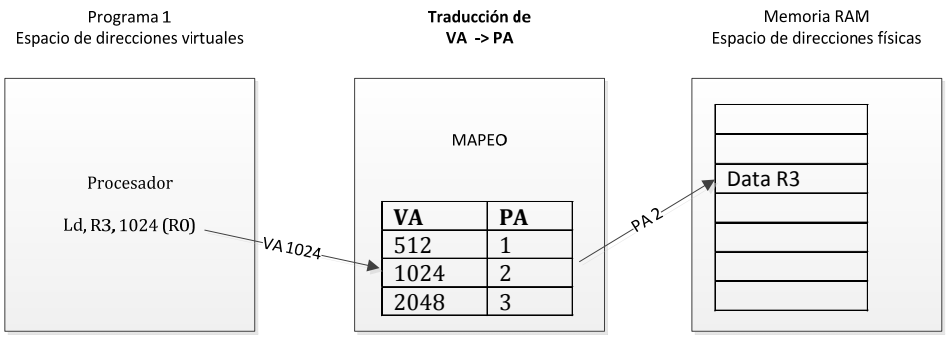

Figura 0.7 Memoria RAM - traducción de direcciones: Virtual Address (VA) a Physical Address (PA)

Los espacios de memoria virtual asignados a diferentes programas están aislados en la memoria RAM física, la siguiente figura ilustra este concepto:

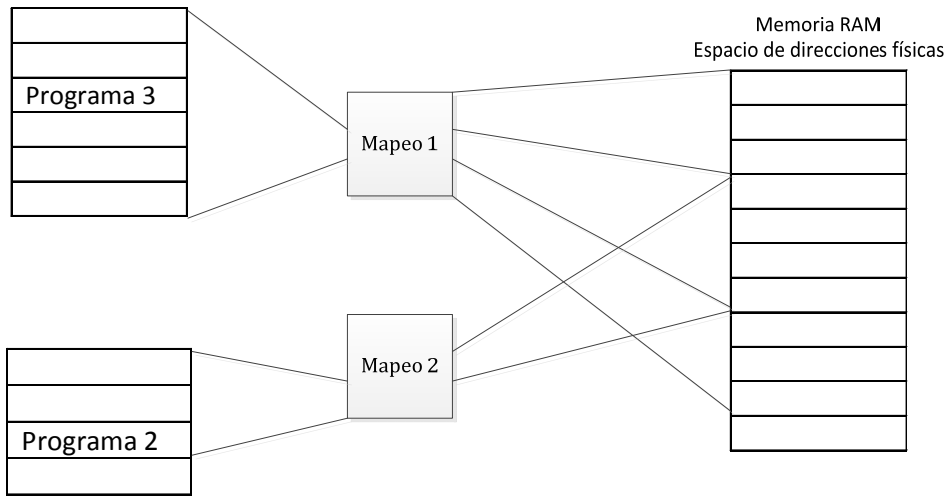

Figura 0.8 Memoria RAM - Secciones aisladas

La dirección virtual permite que se pueda cargar varios programas que accedan a la “misma” dirección virtual pero que estarán en diferentes regiones aisladas de la memoria RAM física.

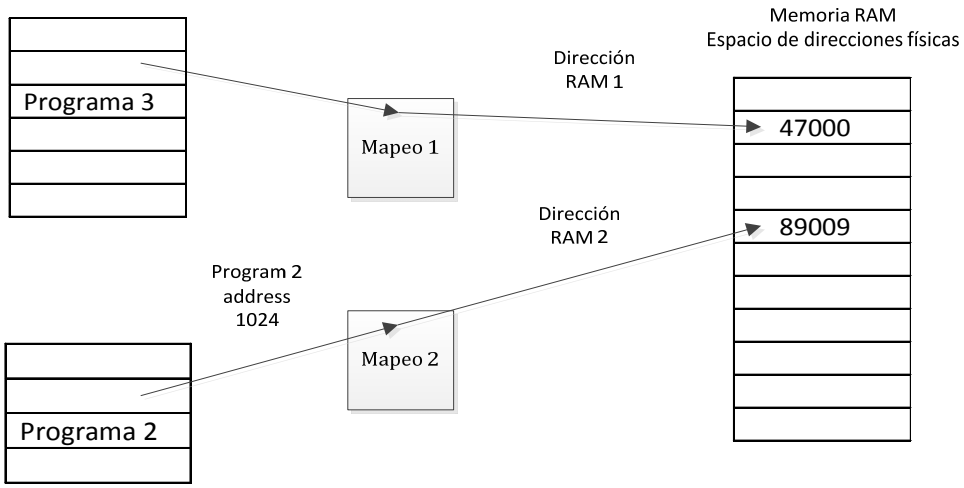

Figura 0.9 Memoria RAM - Direcciones de memoria en programas aislados

Conocer la forma en que se organiza la memoria RAM nos ayuda a interpretar como se cargaron las secciones de un programa en RAM.

En la siguiente figura de la herramienta LordPE podemos ver como se organizan las secciones del programa y los valores correspondientes de Virtual Size, VOffset y ROffset.

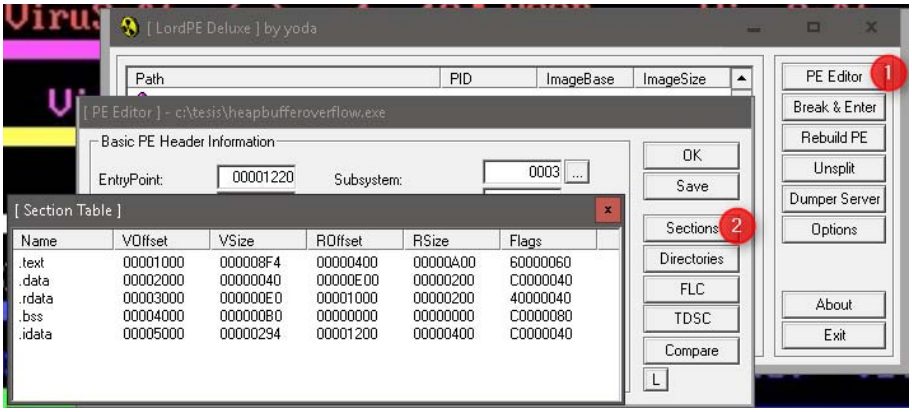

Figura 0.10 Secciones de Memoria de un programa – Lord PE

La herramienta ImmunityDebbuger nos permite ver la organización de las secciones cargadas en memoria RAM, como se muestra en la siguiente figura:

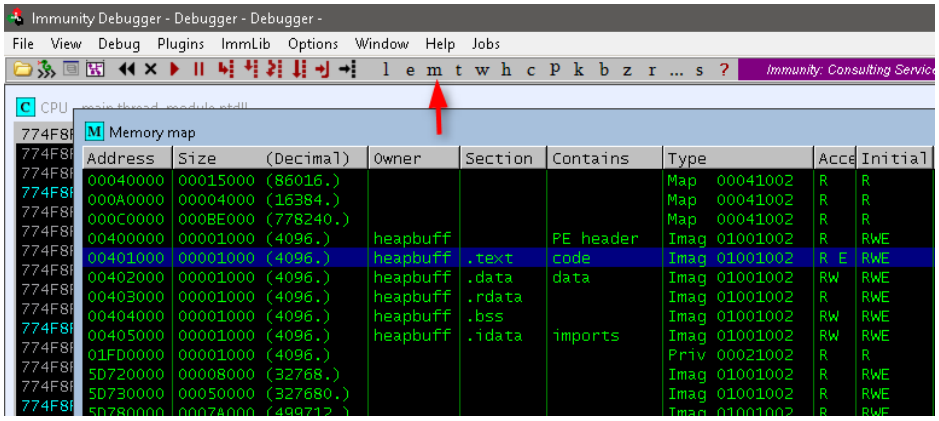

Figura 0.11 Secciones de memoria - Inmunity Debbuger

Lectura de Strings

Dentro de los ejecutables existen cadenas de texto que pueden describir propiedades del ejecutable. Como URLs de C2.

Esta técnica de análisis se puede ejecutar solo en muestras de malware que no están empaquetadas.

En la siguiente figura se puede apreciar el análisis de string sobre una muestra donde se puede identificar las librerías que la muestra está utilizando.

En este caso se analiza el backdoor 61780002.ex. En el ejemplo se puede ver el listado de DLLs que utiliza el aplicativo.

Figura 0.12 Strings - Filtro para DLLs

Este filtro nos ayuda a identificar las DLLs que contiene el ejecutable y nos da una idea de las funciones que realiza, el siguiente filtro podemos identificar las direcciones URL que contiene el ejecutable, dichas direcciones generalmente hacen referencia a los C&C.

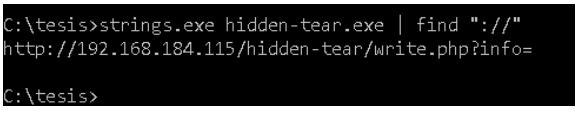

Figura 0.13 Strings - Filtro URL

Esta información nos servirá en el análisis dinámico como filtro para capturar el trafico interesante entre la máquina víctima y el C&C.

Identificación de empaquetados o wrappers

Un empaquetado o packer es utilizado por los desarrolladores para ocultar el código de un ejecutable. La siguiente figura ilustra el mecanismo utilizado:

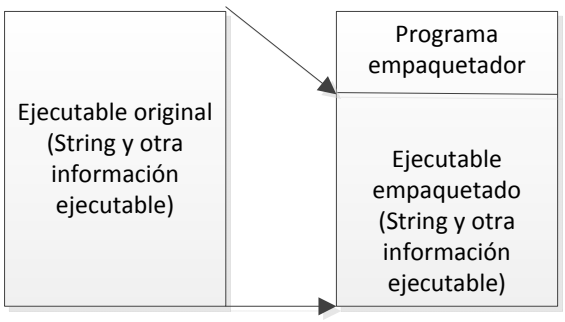

Figura 0.14 Mecanismo de empaquetado o wrapper

Para ilustrar el ejemplo utilizamos UPX que es un EXE Packer Compression, de los más utilizados. (No sirve para código de .Net, C#). Para esto utilizamos un ejecutable de ejemplo “ConsoleApp1Cplus.exe”, escrito en C++ y que únicamente muestra el mensaje “Hello World”. En el paso 2 se ve el resultado de la ejecución del programa de manera convencional. En el paso 3 se empaqueta el aplicativo, en el paso 4 se ve el resultado donde se identifica un tamaño menor al original, pero con la misma extensión .exe, adicional cambia la firma o hash que identifica al archivo y que se encuentra en la base de datos de muchos programas de antimalware.

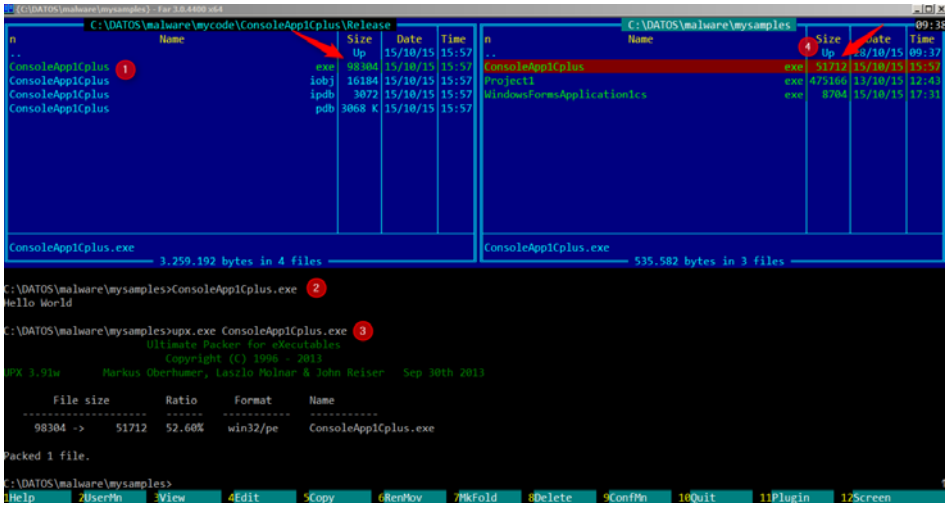

Figura 0.15 Ejemplo de un Packer

El uso de PEiD permite identificar los empaquetadores utilizados, en la siguiente figura se ve el resultado de la ejecución, donde se identifica el uso de UPX.

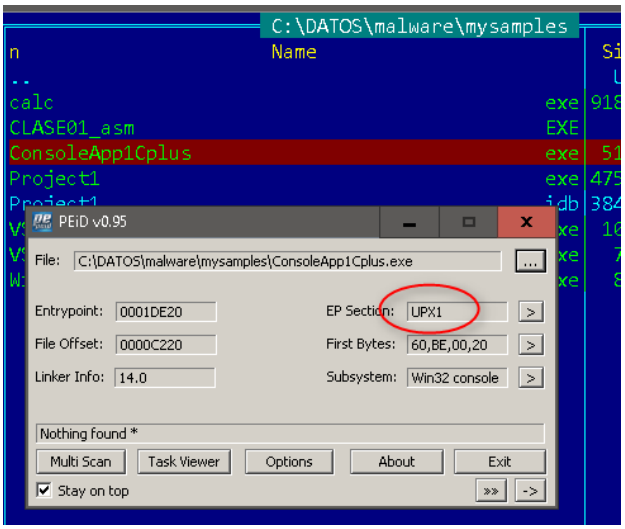

Figura 0.16 PEiD - identificación de packer

En la página *http://compression.ca/act/act-exepack.html* se tiene un listado de los mejores EXE Packer Compression.

Estas muestras tratadas con packers bien conocidos son identificadas igualmente por los programas de antimalware y antivirus, por tal razón en el desarrollo de malware avanzado se utilizan algoritmos de cifrado como RC4 pero que son modificados y desarrollados de manera específica para ser aplicados sobre las muestras dirigidas.

Segundo punto de entrada - análisis estático.

En ocasiones los desarrolladores de malware utilizan un segundo punto de entrada en la ejecución de una aplicación, esto se utiliza para realizar una acción maliciosa corriendo inicialmente un segundo punto de entrada con el uso de TLS (Thread-local storage) antes de que se ejecute el Entry Point principal de la “aplicación benigna” que abre el usuario.

Ejemplo de una aplicación con dos puntos de entrada usando TLS [36]:

```
#include "stdafx.h"
#include <windows.h>
#include "iostream"
void NTAPI tls_callback(PVOID DllHandle, DWORD dwReason,
PVOID)
{
 if (dwReason == DLL_THREAD_ATTACH)
```

```
 {
MessageBox(0, L"DLL_THREAD_ATTACH", L"DLL_THREAD_ATTACH",
0);
}
 if (dwReason == DLL_PROCESS_ATTACH)
{
MessageBox(0, L"DLL_PROCESS_ATTACH",
L"DLL_PROCESS_ATTACH", 0);
}
}

#ifdef _WIN64
#pragma comment (linker, "/INCLUDE:_tls_used")  // See p.
1 below
#pragma comment (linker, "/INCLUDE:tls_callback_func")
// See p. 3 below
#else
#pragma comment (linker, "/INCLUDE:__tls_used")  // See
p. 1 below
#pragma comment (linker, "/INCLUDE:_tls_callback_func")
// See p. 3 below
#endif
#ifdef _WIN64
#pragma const_seg(".CRT$XLF")
EXTERN_C const
#else
#pragma data_seg(".CRT$XLF")
EXTERN_C
#endif
PIMAGE_TLS_CALLBACK tls_callback_func = tls_callback;
#ifdef _WIN64
#pragma const_seg()
```

```
#else
#pragma data_seg()
#endif

DWORD WINAPI ThreadProc(CONST LPVOID lpParam)
{
 ExitThread(0);
}
int main(void)
{
 MessageBox(0, L"Hola desde main", L"main", 0);
 CreateThread(NULL, 0, &ThreadProc, 0, 0, NULL);
 return 0;

}
```

Como se detalla, en el proceso main la primera línea de código ejecuta en MessageBox con el contenido “Hola desde main”.

```
}
int main(void)
{
    MessageBox(0, L"Hola desde main", L"main", 0);
    CreateThread(NULL, 0, &ThreadProc, 0, 0, NULL);
    return 0;
}
```

Figura 0.17 TLS ejemplo - código main

Pero al ejecutar la aplicación lo primero que se visualiza es:

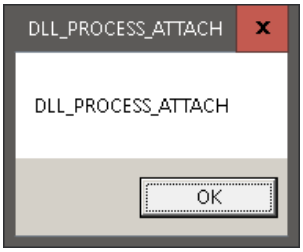

Figura 0.18 TLS ejemplo – mensaje DLL

Este mensaje está presente en la definición de TLS.

```
void NTAPI tls_callback(PVOID DllHandle, DWORD dwReason, PVOID)
{
    if (dwReason == DLL_THREAD_ATTACH)
    {
        MessageBox(0, L"DLL_THREAD_ATTACH", L"DLL_THREAD_ATTACH", 0);
    }

    if (dwReason == DLL_PROCESS_ATTACH)
    {
        MessageBox(0, L"DLL_PROCESS_ATTACH", L"DLL_PROCESS_ATTACH", 0);
    }
}
```

Figura 0.19 TLS ejemplo - código de mensaje

Posteriormente se muestra el mensaje de la función main.

Figura 0.20 TLS ejemplo - mensaje de main

En el análisis de la aplicación podemos ver el valor de TLS table está presente en el ejecutable.

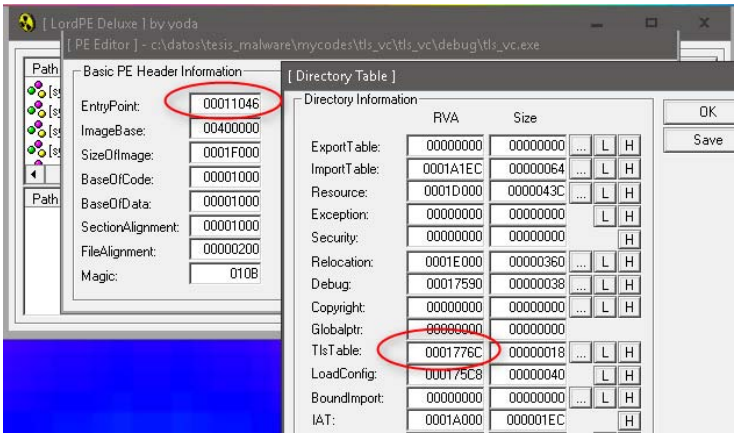

Figura 0.21 Valor TLS - Lord PE

Análisis de funciones y librerías

Esta técnica de análisis permite conocer las funciones de las librerías enlazadas al ejecutable del malware.

En la siguiente figura se muestra el resultado de analizar el malware con la herramienta DependencyWalker:

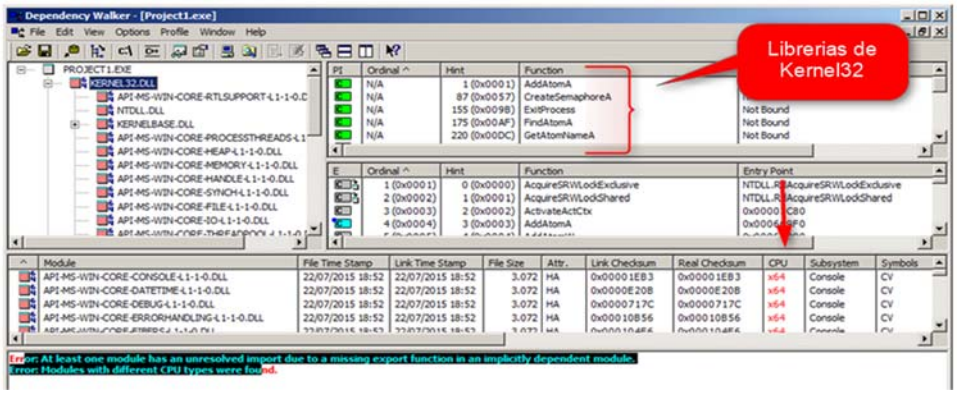

Figura 0.22 Dependency Walker

En la figura podemos ver, entre otras, las funciones de Kerlnel32.dll que utiliza el archivo ejecutable del ejemplo.

Dependiendo de los propósitos del archivo este puede importar diferentes librerías:

User32.dll.- esta librería es importada cuando el programa incorpora una interfaz gráfica o cuando va a manipular la interfaz de otros programas.

SetWindowsHookEx.- esta función es para capturar las entradas de los teclados y es la elección de los desarrolladores al programar keyloggers.

En *Practical Malware Analysis* se encuentra un listado de las principales funciones utilizadas por los desarrolladores de malware [37].

Ingeniería inversa

Cuando se analiza malware, generalmente se cuenta con un archivo en formato binario o el ejecutable a nivel 2 o de código máquina.

El proceso de desensamblar una muestra consiste en el proceso de obtener código ensamblador a partir del ejecutable del malware en formato binario, La herramienta IDA Pro es uno de los desensambladores más populares.

Para realizar el análisis del desensamblado de una muestra de malware se debe tener conocimiento de lenguaje ensamblador y de las particularidades que este presenta en las diferentes arquitecturas de procesadores, como X86, ARM, etc.

Una consideración importante a tomar en cuenta al realizar un análisis de ingeniería inversa son los conceptos de Little Endian y Big Endian, estos formatos que representan la forma que se visualizan los valores en hexadecimal. En el caso de la arquitectura X86 utiliza Little Endian, mientras que los protocolos de red de datos utilizan Big Endian. En tal caso la dirección IP 127.0.0.1 se va representar como 0x7F000001 en la red, y como 0x0100007F cuando este en memoria RAM.

Dado el grado de habilidades y conocimientos requeridos para este tipo particular de análisis es requerido personal especializado en la materia por lo que generalmente este tipo de servicios se contratan bajo demanda.

4.1.2 Análisis Dinámico

En el análisis dinámico se ejecuta la muestra sobre un entorno controlado para visualizar su comportamiento, entre las técnicas utilizadas está el análisis de tráfico de red, revisión de procesos que corren en el sistema operativo.

Análisis en Máquinas Virtuales

El uso de máquinas virtuales se debe realizar con las medidas necesarias que permitan una investigación exitosa ya que el malware avanzado puede identificar que está siendo ejecutado en un entorno virtual y que posiblemente está siendo analizado.

El malware se vale de las características propias de una máquina virtual, por ejemplo, cuando identifica un disco duro de 20GB es casi seguro que se trate de una máquina virtual, o si el huésped tiene instalado complementos como Vmware Tools, el nombre de las interfaces de red, son algunos de las particularidades que presenta una VM y la hacen fácil de identificar.

Una de las acciones que toma el malware en este caso es entrar en un modo sigiloso o sleep y no realizar actividades maliciosas por un periodo de al menos 72 horas. En tal caso el analista generalmente no cuenta con ventanas de tiempo extensas para esperar un comportamiento inusual.

Dadas estas circunstancias se debe preparar un entorno virtual a fin de evitar ser detectado como tal, entre las principales recomendaciones se deben tomar en cuenta:

1. No instalar los Guest addons.

2. Realizar cambios en el registro de Windows.

   Nota: para la modificación de los registros es aconsejable crear un archivo .bat

3. En el caso de Virtual Box usar la herramienta VBoxManage para cambiar los parámetros de la máquina virtual como el BIOS, Drivers de Disco Duro, entre otros.

4. Otra consideración necesaria es el aislamiento a nivel de red, se debe utilizar una red virtual aislada del entorno de trabajo.

   Una vez creado nuestro entorno lo podemos probar con herramientas como pafish.

   Pafish

   Pafish es una herramienta que emula lo que hace el malware que detecta entornos virtuales de análisis y se utiliza para probar la valides de un entorno de análisis.

   Una máquina virtual con Sistema Operativo Windows XP SP2 sin la instalación de addons resulta fácilmente detectable con la configuración por defecto de Vmware. En la siguiente figura se ve el resultado de la ejecución de Pafish:

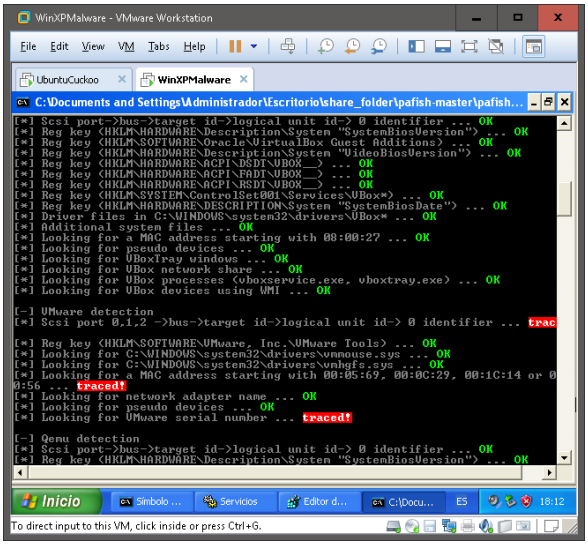

Figura 0.23 Ejecución de Pafish – Vmware

Una configuración simple es la MAC address, Vmware como fabricante tiene asignado el rango 00:0c:29, al modificar este valor y correr nuevamente el software obtenemos el resultado esperado para los parámetros específicos:

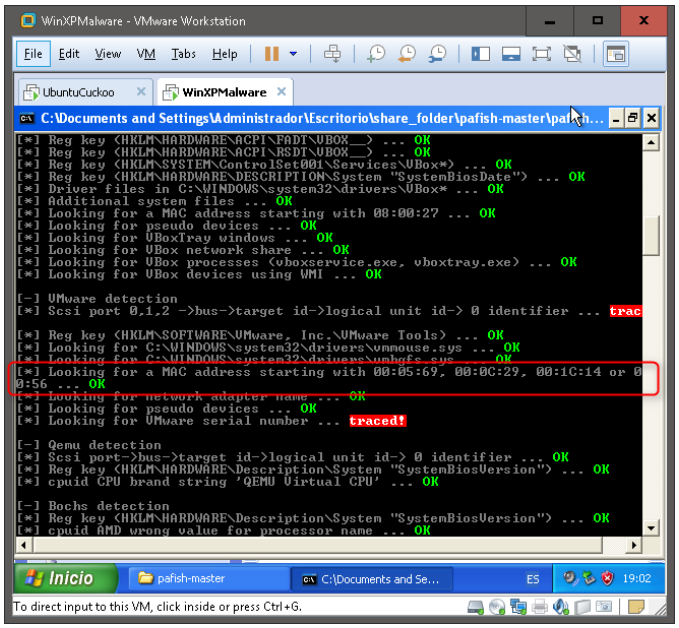

Figura 0.24 Ejecución de Pafish – Vmware - MAC Address

De esta manera debemos personalizar nuestro entorno de ejecución tomando para cada parámetro la medida necesaria.

Volcado de memoria

Al ejecutar un malware empaquetado este descomprime su código para ser ejecutado y lo carga en memoria. En este punto se pueden usar herramientas para realizar un volcado de memoria sobre el proceso del

ejecutable y realizar un posterior análisis estático sobre el volcado de memoria.

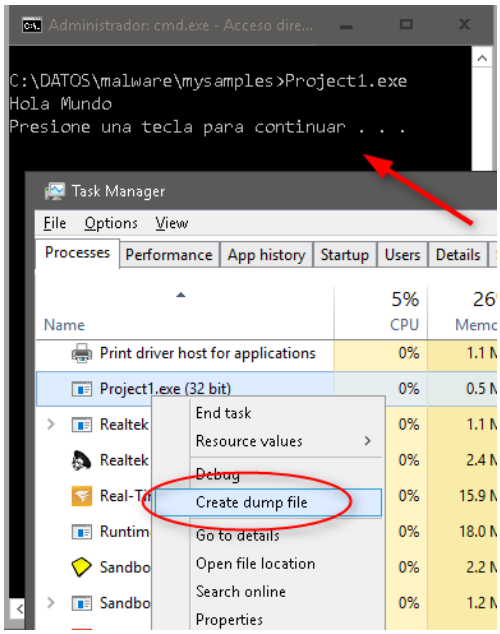

Figura 0.25 Volcado de memoria en Windows

Lo recomendable es correr el malware y pescarlo cuando está corriendo en memoria RAM. Al realizar el volcado de memoria lamentablemente se pierde la integridad o estructura del formato PE y en muchos casos hay que realizar una reconstrucción para obtener la tabla de imports.

Análisis cloud

Existen servicios cloud de acceso público que permiten cargar muestra de malware que realizan el análisis y almacenan los resultados para que sean de acceso público, lo que permite que otros usuarios se vean beneficiados de esta información.

El portal *Malwr* de Cuckoo SandBox [38] es un ejemplo de este tipo de servicios:

Si se está realizando una investigación donde se sospecha un ataque dirigido, no es recomendable utilizar los servicios de análisis cloud que son de acceso público ya que esto puede alertar a los atacantes y la investigación se verá afectada.

Revisión del estado de los procesos

Existe malware que no tiene la modalidad de rootkit o bootkit por lo que no se puede esconder la ejecución de sus procesos, en esa situación se puede observar de manera explícita el listado de los procesos y los recursos que se están consumiendo.

La herramienta de CLI Tasklist, forma parte del paquete de herramientas sysinternals de Windows y permite visualizar el listado de procesos que están corriendo en el sistema, en la siguiente figura se puede ver un ejemplo.

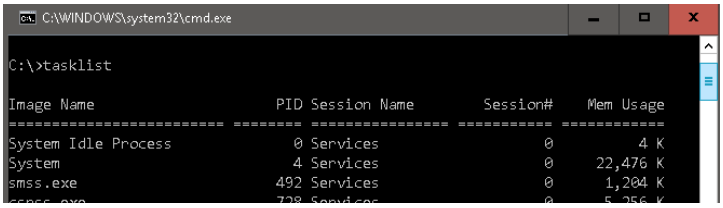

Figura 0.26 Tasklist Ejecución

La herramienta permite aplicar filtrados basados en cadenas de string:

Figura 0.27 Tasklist - Filtro De Búsqueda

Análisis de red

El análisis de red se debe realizar con el objetivo de identificar las conexiones generadas por una muestra y capturar el tráfico generado por el malware.

Netstat y tasklist

Debido a que existen muestras que se conectan hacia su C2, en el siguiente ejemplo se simula una conexión a google, veremos que con ayuda de netstat y tasklist esta conexión puede ser identificada.

Figura 0.28 Telnet conexión

Obtenemos el process ID en base al nombre del proceso:

Figura 0.29 Tasklist - Process ID

Revisamos si existen conexiones salientes y filtramos el process ID:

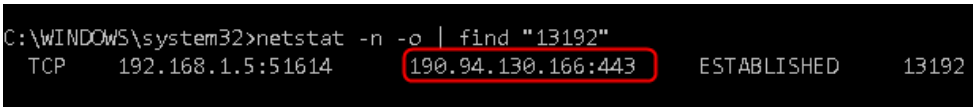

Figura 0.30 Filtros - Process ID

A partir de aquí se pueden hacer uso de herramientas como wireshark y tcpdump para un análisis más profundo, este análisis se debe realizar a nivel de red y a nivel de sistema operativo.

Sistema de detección de intrusos

En el caso del análisis de malware se debe considerar la capacidad de analizar una cantidad de datos en un periodo de tiempo determinado, también conocido con el término de throughput.

Dentro del segmento de IDS, la herramienta orientada al análisis de malware es BRO Network Security Monitor, por su flexibilidad, ya que permite el uso de scripts para definir las políticas de análisis de tráfico, a diferencia de las firmas o reglas usadas por los IDS tradicionales.

El siguiente ejemplo de un script muestra la sintaxis para la identificación de tráfico relacionado al protocolo SMB y movimientos laterales realizados por el malware [39]. Captura archivos de la red y los analiza contra la nube de VirusTotal.

```
@load base/frameworks/files
@load base/frameworks/notice
@load frameworks/files/hash-all-files
export {
 redef enum Notice::Type += {
 SMB
 };
global trustedIPs: set[addr] = {192.168.1.22,192.168.1.20}
&redef;
# url needed to use VirusTotal API
 const                        vt_url                        =
"https://www.virustotal.com/vtapi/v2/file/report" &redef;
 # VirusTotal API key
 const vt_apikey = "<---- Enter your Virus Total API key
here ---->" &redef;
 # threshold of Anti-Virus hits that must be met to trigger
an alert
 const notice_threshold = 2 &redef;
event file_hash(f: fa_file, kind: string, hash: string)
{
# If the file "f" for the event has a source type, and if
the source type equals
SMB, check file hash against VirusTotal
if ( f?$source && f$source == "SMB")
{
local data = fmt("resource=%s", hash);
```

```
 local key = fmt("-d apikey=%s",vt_apikey);
# HTTP request out to VirusTotal via API
 local          req:          ActiveHTTP::Request          =
ActiveHTTP::Request($url=vt_url,
$method="POST",$client_data=data, $addl_curl_args=key);
 when (local res = ActiveHTTP::request(req))
{
if ( |res| > 0)
{
if ( res?$body )
{
local body = res$body;
 local tmp = split_string(res$body,/\}\},/);
if ( |tmp| != 0 )
{
local stuff = split_string( tmp[1], /\,/ );
# splitting the string that contains the amount of
positive anti-virus hits on ":" "positives:23"
local pos = split_string(stuff[9],/\:/);
# converting the string from variable pos into a
integer
local notic = to_int(pos[1]);
# If the number of positives (number stored in
variable notic) equals or exceeds the threshold, generate
a notice
if (notic >= notice_threshold )
{
local msg = fmt("%s,%s,%s","Potentially
Malicious File Transfered via SMB",stuff[9],stuff[4]);
local n: Notice::Info = Notice::Info($note=SMB,
$msg=msg, $sub=stuff[5]);
```

```
Notice::populate_file_info2(Notice::create_file_info(f),
n);
 if (n$id$orig_h !in trustedIPs){
NOTICE(n);
}
}
}
}
}
}
}
}
```

Emular entornos de red

Este tipo de emuladores le permite al analizador conocer el comportamiento a nivel de red que genera una muestra activa pero dentro de un entorno seguro.

Dentro de este segmento podemos nombrar a Fakenet, es una herramienta que simula ser internet y disponer de los servicios de red solicitados por el malware [40]. Una versión evolucionada de fakenet es fakenet-NG patrocinada y mantenida por la empresa FireEye. En la siguiente figura se puede ver la ejecución de la herramienta.

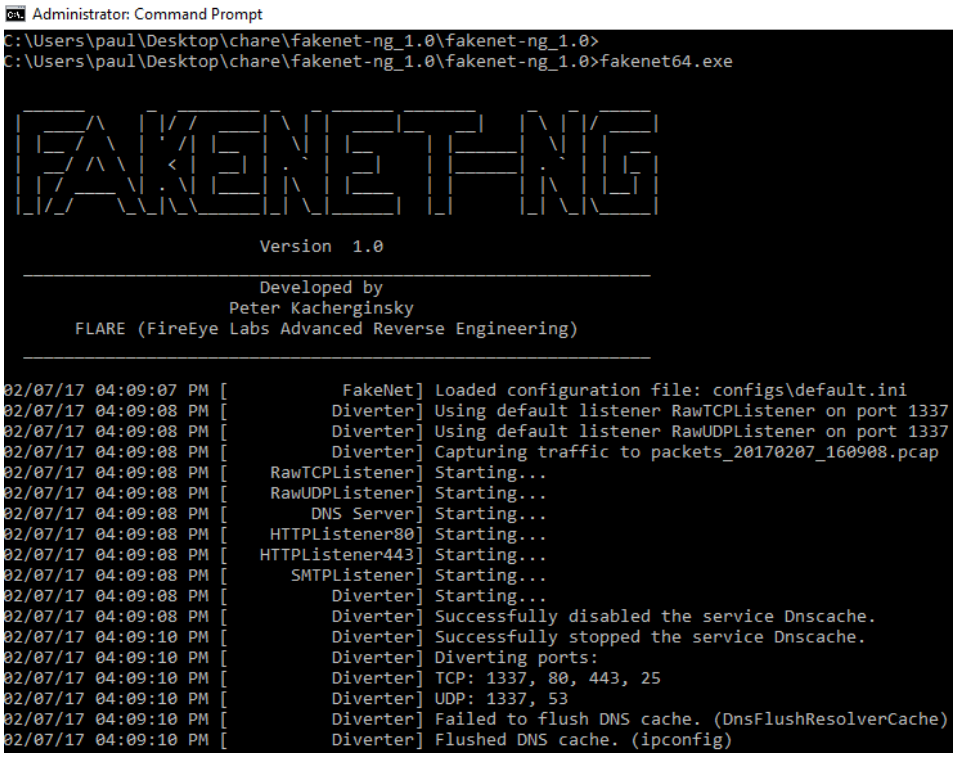

Figura 0.31 FakeNet – Ejecución

Con fakenet simulando ser internet y respondiendo a las peticiones, podemos obtener la información generada por el malware para saber cuáles son las conexiones externas que provoca, como ejemplo en la siguiente figura vemos la respuesta de ping por parte de un dominio que no existe:

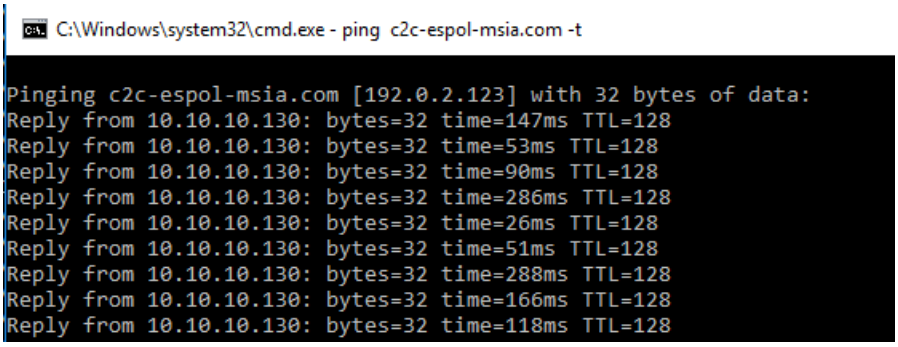

Figura 0.32 FakeNet - Respuesta dominio falso

Fakenet modifica este tráfico y nos visualiza la información como se muestra en la siguiente figura:

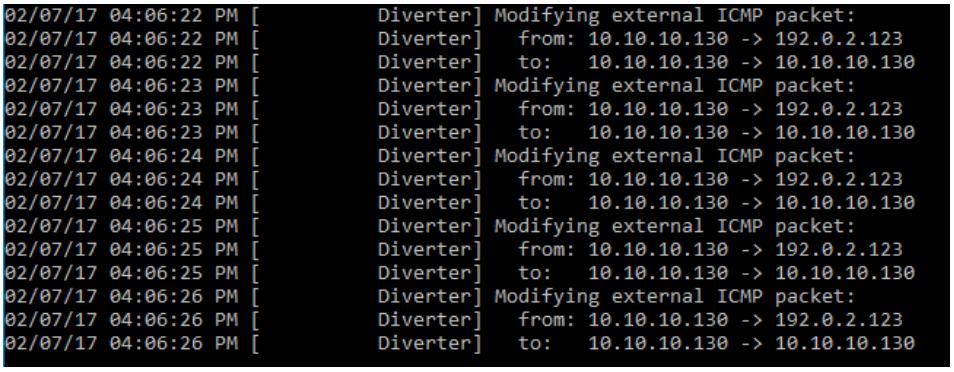

Figura 0.33 FakeNet – Información trafico modificado

Del monitoreo realizado por FakeNet se puede obtener un archivo .pcap que puede ser analizado con sniffers de red como Wireshark o TCPdump.

Debugger de Malware

Para realizar debugging se utilizan herramientas como Ollydbg o Inmunity Debugger, Windbg que permiten realizar la ejecución paso a

paso de las instrucciones programadas en un ejecutable, visualizar el estado de la memoria, así como de los registros que mantiene el sistema. En las siguientes secciones se utiliza estas herramientas y se hace mención de las precauciones sobre su uso.

Es importante mencionar que herramientas como Inmunity debugger corre únicamente en User mode debugger, mientras que Windbg corre en modo usuario y en modo Kernel, WinDbg es desarrollado y mantenido por Microsoft.

Segundo punto de entrada – análisis dinámico

Cuando se analiza malware hay que considerar que puede tener múltiples puntos de entrada. Por defecto los debugger inician su tarea a partir de EntryPoint, si el malware tiene configurado un segundo punto de entrada con TLS este no será considerado y se ejecutará sin el análisis, en la siguiente figura se muestra el análisis del programa TLS_test, detallado en la sección “Segundo punto de entrada- análisis estático” con el uso del debugger OllyDBG.

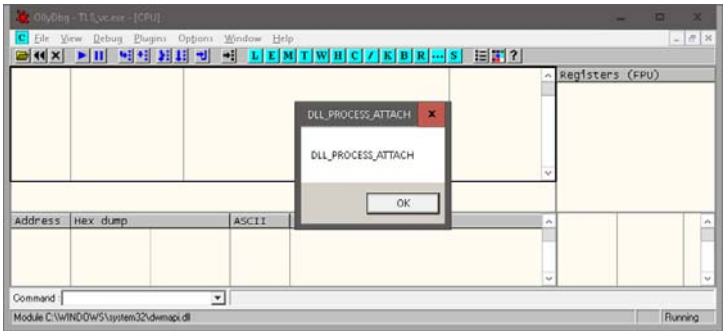

Figura 0.34 TLS Ejecución en Debugger

Como se puede ver en la figura anterior, al abrir el ejecutable con Ollydbg aún no realiza mapeo completo de las instrucciones por parte del debugger pero el Segundo punto de entrada ya se ejecutó, en este momento ya es posible realizar una infección. Apenas cuando damos clic en ok, empieza el análisis del punto de entrada principal por parte del debugger.

Si damos clic en la opción play para correr la aplicación vamos a ver la ejecución del código dentro de la función main que si es analizada.

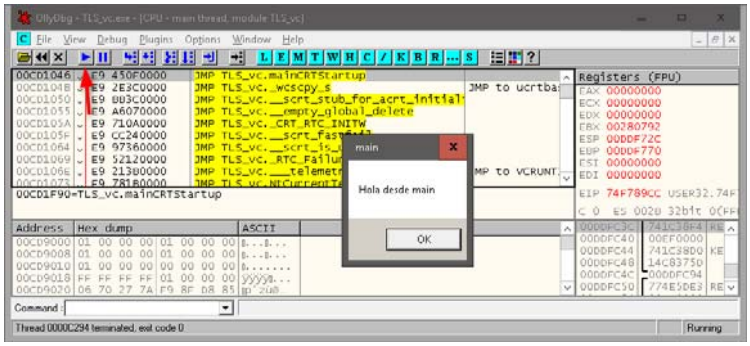

Figura 0.35 TLS - Ejecución punto de entrada principal

Inicialmente el Debugger nos muestra una alerta donde detalla la existencia de un EntryPoint fuera del código.

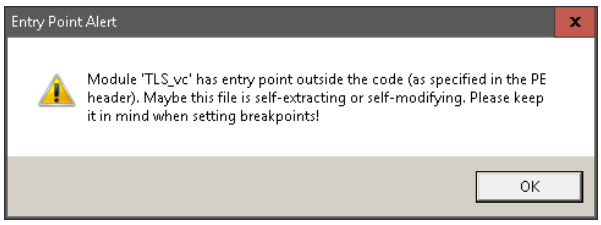

Figura 0.36 Mensaje de alerta de TLS

Remediación: En el caso de inmmunity debugger podemos configurar el debugger para que analice desde el primer “system breakpoint”.

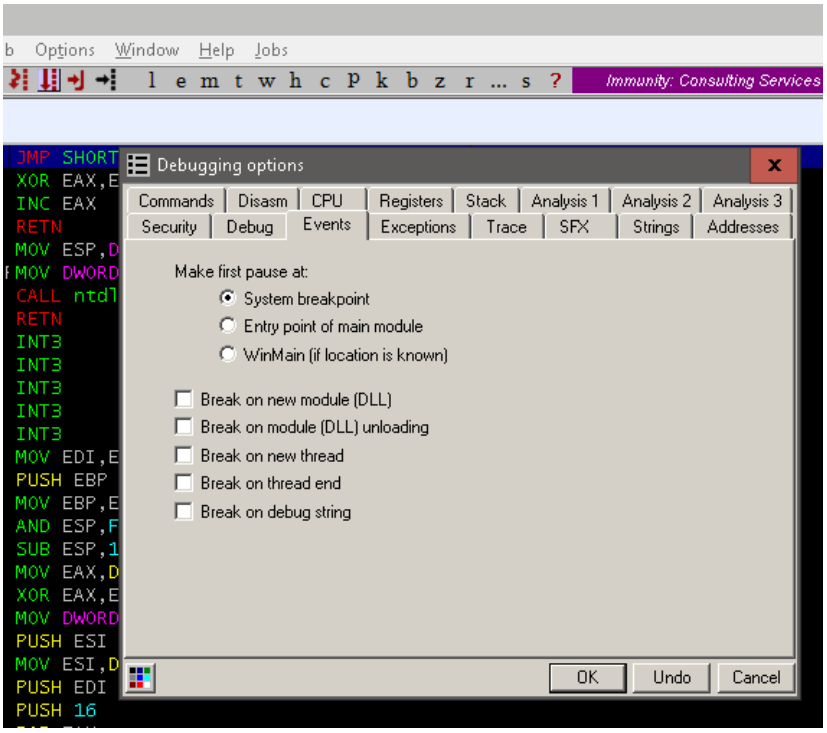

Figura 0.37 TLS configuración de Debugger

Con esto podremos hacer debugging antes de que se ejecute el TLS.

Niveles de abstracción

En la arquitectura de computadores hay definidos 6 niveles de abstracción definidos en el orden que se muestra en la siguiente figura:

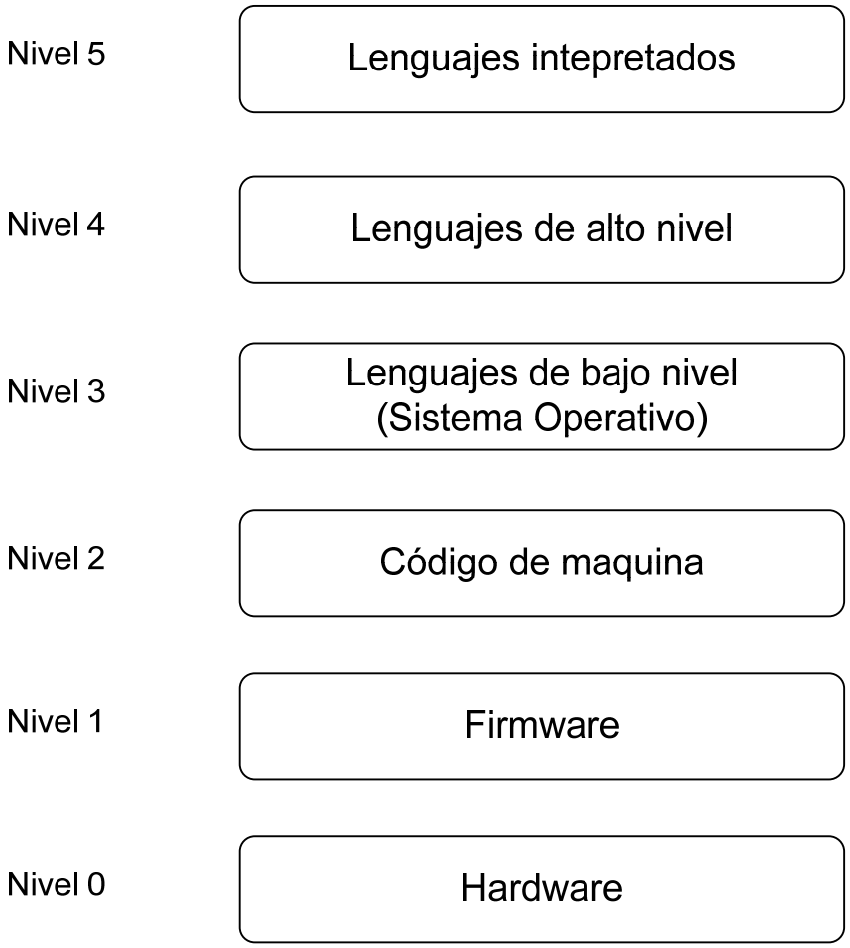

Figura 0.38 Niveles de abstracción

En el nivel 3 se encuentra el lenguaje de bajo nivel o ensamblador, este es nivel más bajo de abstracción al que se puede acceder cuando se realiza el análisis de una muestra.

4.2 Técnicas de anti detección

Una vez que se obtiene la muestra en binario de un malware, el analista de malware es consciente que los desarrolladores utilizaron varias técnicas para evitar ser detectados.

Ofuscar binarios

La Inyección de código permite insertar en binarios no maliciosos un shellcode para realizar tareas específicas. Herramientas como shellter y veil evasion nos permiten realizar este tipo de tareas. Lo que hacen estas herramientas es crear un nuevo hilo de ejecución sobre el binario para ejecutar el código del payload, en la siguiente figura de la herramienta shellter se describe la creación del nuevo hilo.

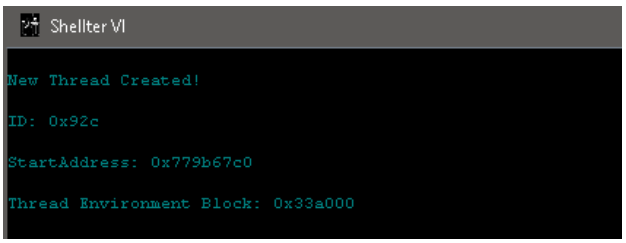

Figura 0.39 Ofuscar binarios 1

Una vez creado el nuevo punto de ejecución permite la inserción de varios payload predefinidos o un payload programado de manera personalizada.

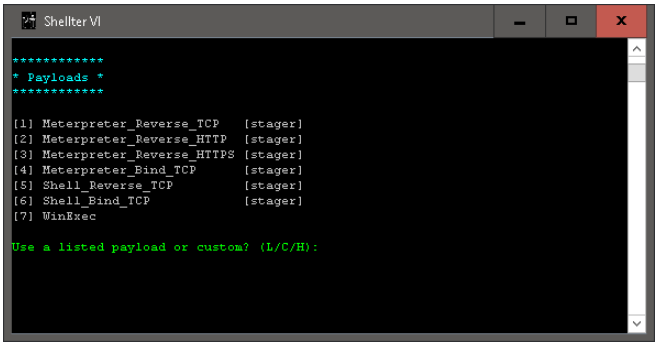

Figura 0.40 Ofuscar binario 2

Técnicas de programación

La programación de un rootkit se puede hacer a nivel de usuario o a nivel de kernel, siendo la primera la más compleja pero la que brinda los mejores resultados.

Hooking.- La modificación o alteración de funciones del sistema conocida como Hooking es una de las técnicas que permiten el ocultamiento de procesos o archivos e incluso de registros del sistema, mientras se realice a un nivel más bajo se obtendrá mayor sigilo.

La inyección de DLLs es una de las técnicas de Hooking más utilizadas.

Import Address Table - IAT Hooking.- consiste en modificar la llamada de la tabla IAT dentro del espacio de memoria de un proceso.

Export Address Table - EAT Hooking.- consiste en modificar las llamadas a la tabla EAT de una DLL específica en el espacio de memoria de un proceso.

Modificación de código.- consiste en la modificación de una función como tal, siendo la técnica más complicada de implementar.

Modificación del Boot.- esta técnica pretende ejecutar código antes de que inicie el MBR (Master Boot Record) del sistema operativo infectado. A nivel de Fabricante, Microsoft por ejemplo implementa controles como UEFI y Secure Boot con la finalidad de mitigar este tipo de amenazas, pero vulnerabilidades presentes en el firmware del hardware permite el uso de bootkits.

Symbol Table

Limpiar la “Symbol Table” dificulta la tarea del analizador de malware, esta tarea se puede realizar en el paso de compilación de un programa.

En la siguiente figura se puede ver la symbol table que resulta de un segmento de código:

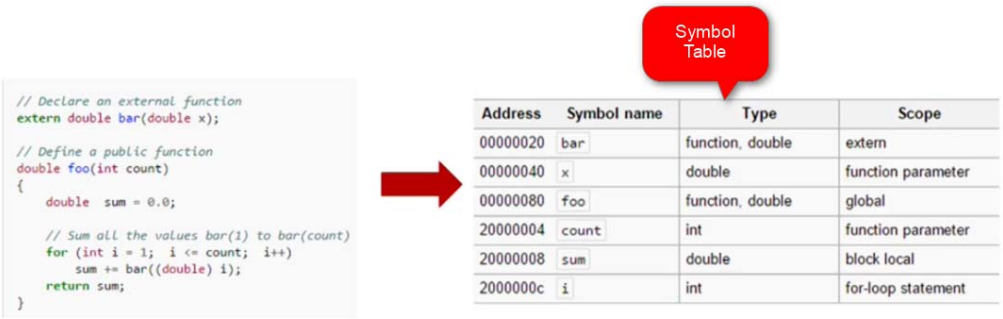

Figura 0.41 Symbol Table

4.3 Protección de endpoint y red

4.3.1 Protección de Endpoint

Reglas de Yara

Yara es un programa multiplataforma que ejecuta reglas y sirven para identificar patrones que ayudan a identificar propiedades o comportamientos maliciosos presentes en Archivos, Tráfico de Red y Memoria RAM.

En el siguiente código podemos ver una regla básica de Yara, en la misma se identifica a un ejecutable que contenga una propiedad de string con el contenido “Hola mundo”.

```
rule ExampleRuleName
{
 meta:
       source =
“http://yara.readthedocs.org/en/v3.4.0/writingrules.html”
       description = "This is a very basic example rule."
 strings:
       $my_text_string = "Hola Mundo"
       $my_hex_string = { E2 34 A1 C8 23 FB }
       $my_regex = /[0-9a-zA-Z]{32}/
 condition:
       $my_text_string or $my_hex_string or $my_regex
}
```

Ejecución de un ejecutable de ejemplo:

Figura 0.42 Ejemplo de ejecución de ejecutable

Ejecución de una regla YARA disponible:

Figura 0.43 Ejecución regla Yara

El tipo de firmas que podemos implementar en las reglas de YARA se muestran en la tabla a continuación:

Tabla 10 Reglas Yara – Subtipos Reglas

Cadenas de texto	**Referente a la implementación**	**Funciones personalizadas**
String Constants	Memory alocation habits	Obfuscation techniques
API Names	Use of global variables	Stealth and evasion techniques
Error messages	Multi-threading model	Encryption and compression algorithms

String formatting style	Software architecture and design	Cryptographic Keys & Constant
Grammar mistakes	Constructor design	Re-used source code
C&C commands	Dynamic API loading technique	Malware specific features
Timestamp formatting	Exception handling	System infiltration
Unique Sequences	Usage of public source code	Propagation mechanisms
Regular Expressions	Programming language and compiler	Artifact names schemas / Algorithms
	Compilation time stamp and time zones	Data exfiltration techniques
		System OS version determination techniques
		C&C Command Parsing Implementation.

Endpoint Security Software Architecture

La solución de seguridad de endpoint debe estar compuesta al menos por tres módulos:

- Base de datos de firmas.- Base de datos de firmas que identifique al malware bien conocido.

- Módulo de Heurística.- este módulo debe permitir identificar muestras de malware no presentes en la Base de datos de firmas mediante el uso de firmas genéricas que detectan segmentos de código bien conocidos presente en las muestras.

- Módulo de Comportamiento.- este módulo debe permitir identificar y valorar las acciones realizadas por una aplicación mientras se ejecuta en el sistema, con la finalidad de determinar si dichas acciones corresponden al comportamiento de un malware.

El diseño del modelo que incluye los módulos descritos se muestra a continuación:

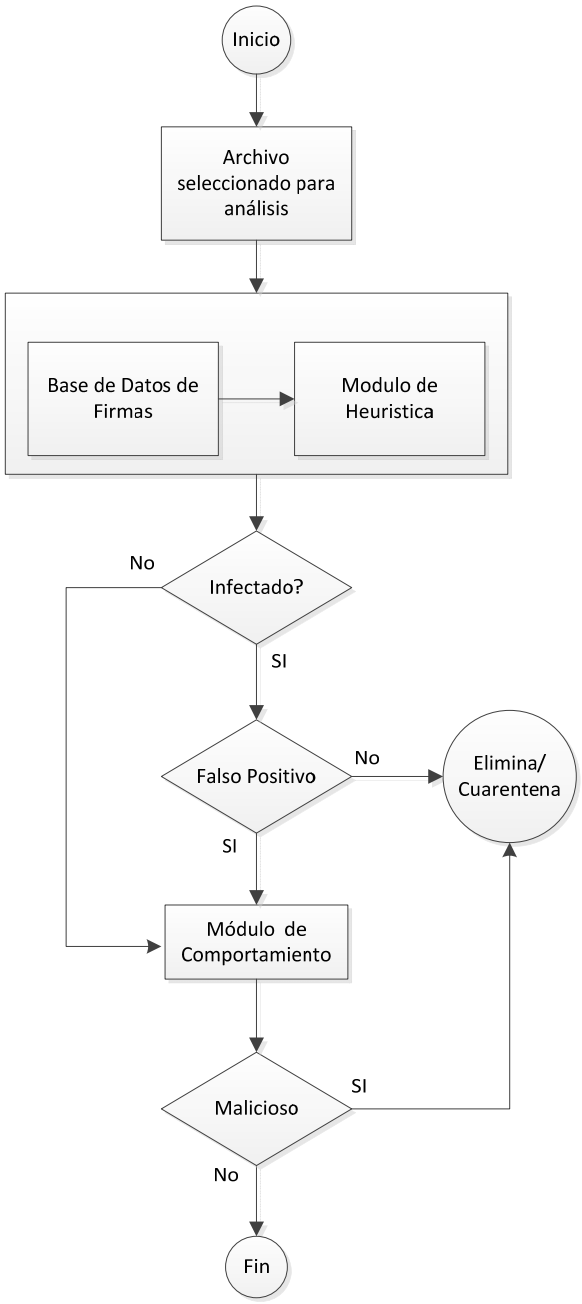

Figura 0.44 Módulos para seguridad de endpoint

La decisión de si el archivo está infectado o no es el resultado del análisis llevado a cabo por medio de la base de datos de firmas y el módulo de heurística.

La decisión de si se trata de un Falso positivo o no, se lleva a cabo mediante una consulta al usuario.

La decisión de si se trata de una aplicación maliciosa o no de determina en base al análisis del módulo de comportamiento.

4.3.2 Protección de la Red

Breach Detection System

BDS es una solución desplegada a nivel de red, está diseñada para detectar la actividad de malware cuando una amenaza se encuentra latente en la red, tienen tres características principales:

- Se despliega generalmente fuera de banda, es decir no es intrusivo. Los sensores para el monitoreo de red deben colectar información directamente de los equipos core de Data Center y LAN y monitorear el tráfico de red por medio de puertos SPAN o mirror.

- Define las características técnicas de una amenaza o Identifica indicadores de compromiso a nivel de red tales como conexiones con dominios de C2 bien conocidos, levantar puertos de escucha no habituales en workstation de trabajo, etc.

- Ejecución de técnicas de detección dinámica como la captura de archivos en la red para el análisis mediante cajas de arena, heurística, entre otras.

Es importante para una BDS conocer el entorno donde esta implementado y tener una lista de aplicaciones aprobadas, y aplicaciones que tiene permitido conectarse a internet, así como conocer los activos de información como servidores de base de datos, fileserver para realizar un monitoreo más eficiente.

4.4 Correlación de eventos

La correlación es un término muy asociado a los sistemas SIEM, y por ende es necesario entender las tecnologías de la información que sirvieron de base para su evolución. Es así que debemos tener en cuenta los siguientes conceptos y herramientas:

Fuentes de datos

Los mecanismos de obtención de datos pueden ser:

- Push.- la fuente que genera los datos es quien los envía al SIEM.
- Pull.- es el SIEM quien va a la fuente para recuperar los datos. Se utiliza cuando la fuente provee el mecanismo para acceder.

Estos datos provienen de diferentes fuentes, en formatos como:

- Archivos Raw
- Logs
- Hojas de Calculo

De igual manera se debe soportar la recolección de datos mediante el uso de varios protocolos: SNMP, Syslog, FTP, SCP entre otros.

Normalización de datos

Diferentes fabricantes pueden tener funciones similares como por ejemplo el login exitoso de un usuario, este evento tiene firmas diferentes en cada fabricante por lo que es recomendable asignar una Taxonomia común para mejorar el análisis y facilitar la interpretación. En la siguiente tabla se especifica un ejemplo del SIEM de Trustwave, donde se especifica la taxonomía para los identificadores de eventos:

Tabla 11 Taxonomía SIEM Trustwave

Dispositivo	**Identificador de evento**	**Taxonomía**
SonicOC	725	auth.wifi.access.grant
CiscoVPN	HTTP47	auth.web.login_admin.grant
Windows	528 success audit	auth.os.login.grant
Oracle	Connect success	auth.db.login_admin.grant
MS SQL	Audit-14	auth.db.login.grant
Imapd	Login user	auth.email.login.grant

Webmin	Successful login	auth.web.login.grant
PIX	109005	auth.fw.login.grant

Análisis Gramatical

Los datos que provienen de los equipos en forma de logs generalmente son recibidos como una cadena de texto y esta debe ser analizada gramaticalmente a fin de obtener los datos de utilidad y generar la información que debe ser almacenada en la base de datos del SIEM.

De manera general se debe utilizar un modelo de par “llave y valor”, esto permite obtener un dato específico y asignarle una etiqueta o llave que lo describa. En la siguiente figura se observa el procedimiento para generar los pares llave valor a partir de un log.

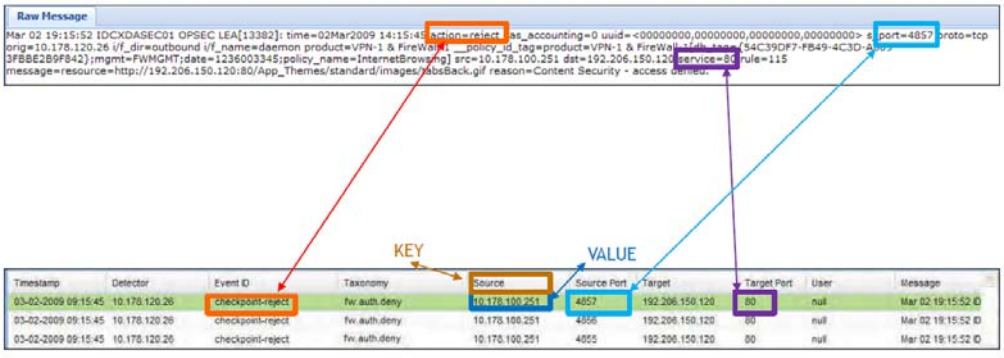

Figura 0.45 Análisis de Datos

SIM (Security Information Management)

Es un sistema dedicado para administrar de manera centralizada los logs, analiza logs y les da el tratamiento correspondiente para poder indexarlos, su propósito es ser una herramienta operativa para la gestión de reportes que ayuda a cubrir los requerimientos de cumplimiento.

SEM (Security Event Management)

Este sistema permite realizar un monitoreo en tiempo real de los eventos registrados a nivel de red y a nivel de aplicaciones. Está enfocada especialmente en las fuentes de datos que son controles de seguridad, entre las cuales podemos incluir los Firewall, IPS, DLP, Endpoint Security, etc. Este sistema incluye entre sus funcionalidades la correlación básica de eventos.

SIEM (Security Information and Event Management)

Es la suma de características de un SIM y un SEM, adicional a esto se le suman características adicionales como la detección de anomalías y amenazas. También realiza una correlación avanzada y profunda de los eventos.

4.4.1 Gestión de Logs

Dentro de la gestión de logs, el protocolo syslogs definido en la RFC 3164 [41] específica ocho posibles grados de severidad detallados en la siguiente tabla:

Tabla 12 Gestión de Logs

Códigos de severidad	**Descripción**
0	Emergencia: el sistema está inutilizable
1	Alerta: se debe actuar inmediatamente
2	Crítico: condiciones críticas
3	Error: condiciones de error
4	Peligro: condiciones de peligro
5	Aviso: normal, pero condiciones notables
6	Información: mensajes informativos
7	Depuración: mensajes de bajo nivel

Cuando se considera una correlación de eventos de seguridad informática se deben tomar en cuenta logs con códigos de severidad cinco (5).

Es importante considerar que la comunicación entre las fuentes de datos y el gestor central debe ser cifrada para proteger la integridad de los datos enviados, por lo que existen varios mecanismos que serán analizados en el capítulo de implementación.

4.4.2 La correlación de eventos y sus técnicas

En estadística existen varias técnicas utilizadas para realizar los cálculos de correlación: Coeficiente de correlación de Pearson, Coeficiente de correlación de Spearman, Coeficiente de correlación de Kendall, Distancia euclídea, Euclídea al cuadrado, Distancia de Minkovsky, entre otras. Cada uno se utiliza dependiendo del tipo de variable estadística ya sea esta cuantitativa o cualitativa, así como el número de variables.

Servicios Gestionados de SIEM

Se recomienda que el servicio de SIEM sea tercearizado, ya que requiere de un equipo de personas especializadas y enfocadas de manera periódica para realizar el análisis más detallado dependiendo la interpretación que se le dé a los eventos de seguridad y las alertas de correlación que la herramienta SIEM tenga a disposición.

4.5 Esquema de la arquitectura

Componentes

Dentro de la arquitectura se han considerado los siguientes componentes de seguridad preventiva y correctiva:

- Servicio de directorio.- El despliegue de un servicio de directorio es la base sobre la cual las herramientas de seguridad se apoyan para la gestión y control de políticas de seguridad basadas en usuarios.

- Seguridad de Endpoint: este componente está basado en varias soluciones de seguridad implementadas a nivel del sistema operativo ya sea una terminal de trabajo o un servidor deben incluir al menos antimalware y FIM.

- Control de acceso a red.- este componente de seguridad preventiva está pesado para realizar una identificación y mapeo de las estaciones de trabajo de usuario final conectadas a la red, entre sus funcionalidades principales está, solicitar autenticación basada en el directorio activo y el cumplimiento de normas de seguridad establecidas. En la siguiente figura se ilustran sus componentes:

Figura 0.46 Fases del Componente de control de acceso a la red

El Proceso de control de acceso a red consta de tres etapas definidas.

Pre-admisión: esta etapa realiza el *descubrimiento* que involucra la detección de todos los dispositivos conectados a la red alámbrica e inalámbrica.

Admisión: esta etapa consta de dos procesos definidos autenticación y cumplimiento, sin embargo, no está limitado a ellos. En el proceso de *Autenticación* los usuarios deben presentar sus credenciales personales las cuales son validadas contra la base de usuarios generalmente un directorio activo o una solución radius. El proceso de *Cumplimiento* involucra la revisión de las condiciones mínimas requeridas que una estación de trabajo conectada a la red debe cumplir como: antivirus, actualización de parches de seguridad, antispyware, firewall de host, entre otras. Durante la ejecución de los procesos de *autenticación* y *cumplimiento* los usuarios interactúan con el proceso de remediación

para ser informados del estado actual de cada proceso y tomar las medidas necesarias a fin de estar en cumplimiento.

Post Admisión: En la etapa de Post Admisión se realizan los procesos de *Autorización* donde se le brindan a cada usuario los privilegios y permisos necesarios de acuerdo a su rol. Finalmente el proceso de *Accounting* permite llevar un registro de todas las actividades realizadas por los usuarios.

- Firewall Perimetral y concentrador de VPN.- El primer nivel de protección que separa la red conocida de lo desconocido es el firewall, establece un perímetro entre la red LAN o interna con las redes externas e internet. En este control de seguridad se deben establecer las políticas de seguridad y reglas para establecer los permisos de salida y entrada hacia y desde nuestra red interna. Otra funcionalidad importante de este componente es la concentración de VPN tanto de Sitio a Sitio como de Cliente a Sitio para establecer una comunicación segura con elementos externos que requieren acceso a los recursos internos.

- Firewall con IPS para Data Center.- Debido a que la mayoría de activos de la información están en el Data Center se requiere de un control de permisos proporcionado por un firewall pero también de un análisis del contenido por lo que resulta necesario contar con un Sistema de Prevención de Intrusos que analice (IPS).

- Firewall de Base de Datos.- el firewall de base de datos está pensado para ser el nivel de protección contra accesos y actividades inusuales de usuarios que tienen permisos validos de acceso. Permite la prevención o alerta de transacciones provenientes de estaciones de trabajo autorizadas con usuarios válidos y con privilegios pero que están siendo realizadas en realidad por malware que está presente en máquinas infectadas.

- Firewall de Aplicaciones Web.- un firewall de aplicaciones web permite la protección de los recursos publicados a internet a nivel de aplicación. Este control de seguridad analiza aparte de las consultas web el contenido de la paginas hospedadas en el servidor de aplicaciones previniendo la instalación y distribución de malware por medio de acceso a páginas web

- Proxy de correo.- un proxy de correo electrónico permite la inspección de un email, su domino de origen así como su contenido y sus adjuntos, siendo el correo electrónico uno de los principales medios para la distribución de malware el proxy de correo electrónico debe contar con el apoyo de un módulo de análisis de malware y sandboxing.

- Sistema Data Lost Prevention.- cuando ya se ha realizado la infección de una estación de trabajo, una de las tareas que ejecuta la amenaza es la ex filtración de la información, esto con el uso de los medios convencionales como email, tráfico de internet, por lo que se requiere un módulo de DLP para evitar que información sensible sea extraída. DLP

debe estar compuesto por módulos distribuidos que cubran la totalidad de brechas que permitan la extracción de información. En este sentido, DLP debe estar presente a nivel de host y red, incluida como un addon en los controles de seguridad de red como el firewall perimetral y el proxy de correo electrónico.

- Sistema de detección de brechas.- un BDS es un control correctivo y se implementa como una herramienta que permite la identificación de amenazas ya existentes por lo que se debe integrar con los controles de seguridad para compartir información con la finalidad de identificar y bloquear en medida de lo posible una amenaza de malware ya presente en el entorno informático.

- Correlacionador de eventos.- este elemento debe aglutinar los eventos o logs de seguridad de todos los controles de seguridad con la finalidad de tener trazabilidad de una cadena de eventos en un periodo de tiempo y determinar la presencia de una posible amenaza.

Diagrama de segmentación de la red basada en cortafuegos

En la siguiente figura se observa los puntos de control por los cuales va a fluir la información entre las principales áreas, Centro de Datos, LAN, WAN e Internet.

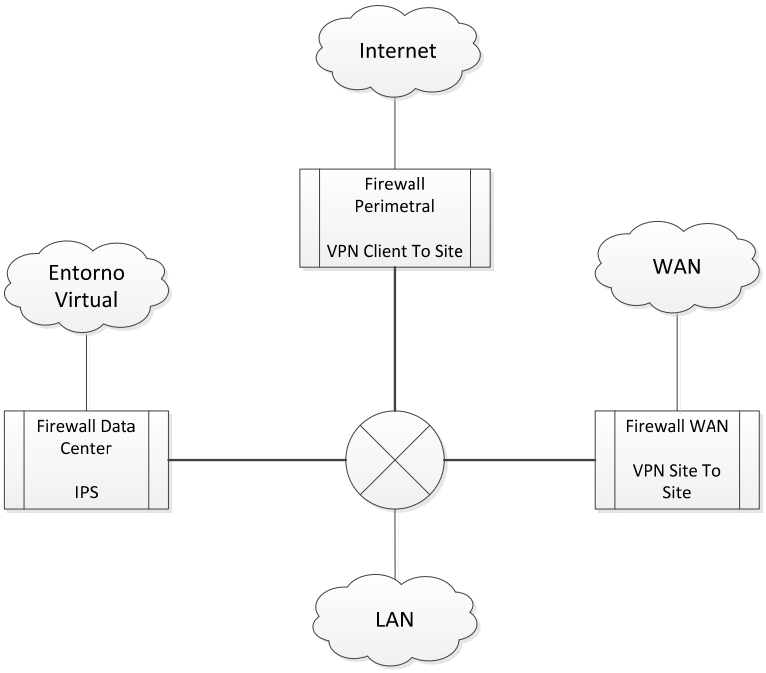

Figura 0.47 Distribución de firewall en la red

Esta división por medio de cortafuegos permite realizar un control entre los principales segmentos de red y monitoreo de las conexiones establecidas así como de los intentos de conexión fallidos ya sean estos realizados por usuarios o máquinas infectadas.

La política de seguridad en cada firewall consta de n reglas, estas están implementadas dependiendo del segmento de red donde se encuentran.

Política en el firewall Perimetral: control de navegación interna hacia internet basada en usuario, URL, IP, puerto y aplicación, conectividad VPN client-to-site, así como, el acceso de usuarios externos hacia la red.

Política en el firewall WAN: controlan el acceso desde las agencias hacia la matriz basadas en usuario, aplicación y servicio, conectividad de VPN site-to-site.

Política en el firewall de DataCenter: acceso a recursos internos del Data Center basado en usuario, segmento de red, protocolo y servicio.

CAPÍTULO 5

IMPLEMENTACIÓN Y PRUEBAS

En el presente proyecto se han considerado tecnologías estándar así como protocolos de acceso libre y herramientas de código abierto, pero también los productos comerciales de vanguardia disponibles en el mercado por sus bondades. Se hará uso de algunas tecnologías y funcionalidades propias a una marca así como los productos tecnológicos ya presentes en las entidades financieras con la finalidad de acercar la arquitectura de seguridad lo más posible a la realidad de una entidad financiera.

5.1 Simulación del entorno de una entidad financiera.

El entorno simulado cuenta con un switch cisco 3850 en el core, un switch cisco 2960x para la parte LAN de acceso, firewall perimetral checkpoint del lado de

internet, un firewall interno cisco que protege el acceso hacia el data center, un switch cisco 2960XR de data center al que se conectan los servidores de la entidad, existe un firewall dedicado checkpoint para el segmento de los servidores para protección de la información de los servicios transaccionales, para las conexiones WAN del lado de la matriz se coloca un router ISR cisco 880 que cuenta con firewall y VPN site-to-site hacia las agencias y cajeros, en donde se colocan routers cisco 800.

El siguiente diagrama detalla la conectividad entre los diferentes componentes presentes en la infraestructura del entorno utilizado.

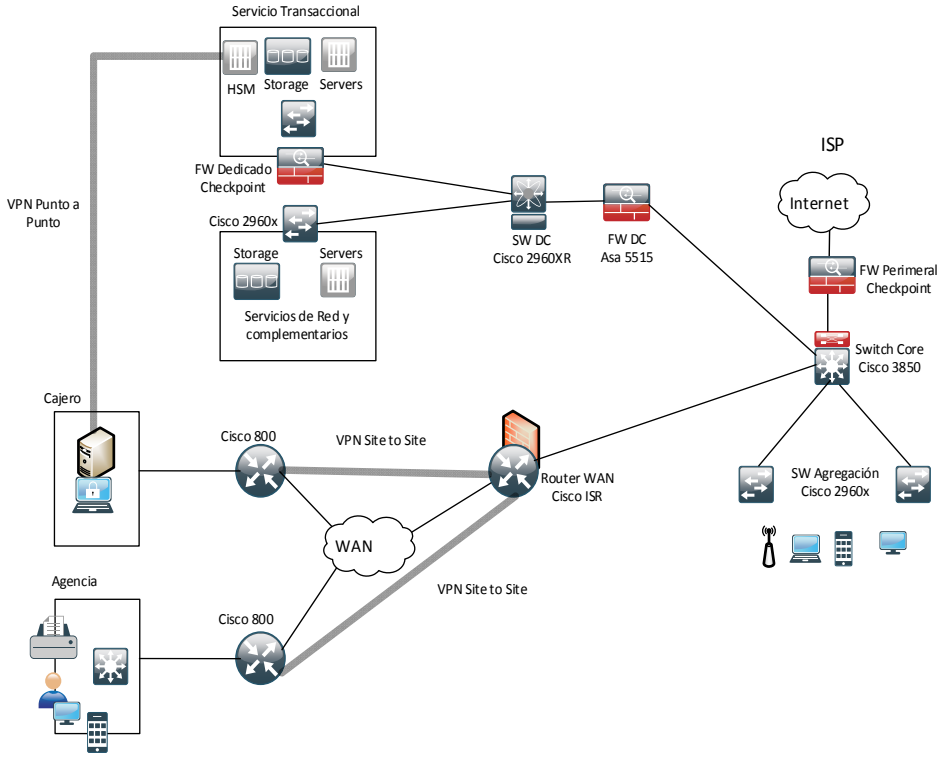

Figura 0.1 Entorno de entidad financiera

Se utiliza GNS3 y Vmware Workstation para implementar equipos de comunicaciones como router, switch, virtualización de servidores y máquinas de usuarios finales. En la siguiente tabla se detalla los IOS utilizados en cada uno de los dispositivos:

Tabla 13 IOS de dispositivos para el entorno

Dispositivo	**Herramienta**	**IOS/ Sistema Operativo**
switch cisco 3850	GNS3	c3725-adventerprisek9-mz.124-12.bin
switch cisco 2960x	GNS3	Propio de GNS3
firewall checkpoint	Vmware	open server
firewall interno cisco	Vmware	Virtual FTD
switch cisco 2960XR	GNS3	Propio de GNS3
router ISR cisco 880	GNS3	c7200-adventerprisek9-mz.152-4.S7.bin
routers cisco 800	GNS3	c7200-adventerprisek9-mz.152-4.S7.bin
Servidor DNS y AD	Vmware	Windows server 2012
Servidor BD	Vmware	Ubuntu 16.04

Máquina usuario	Vmware	Windows 7

La interconexión de los dispositivos va como la indica la figura a continuación:

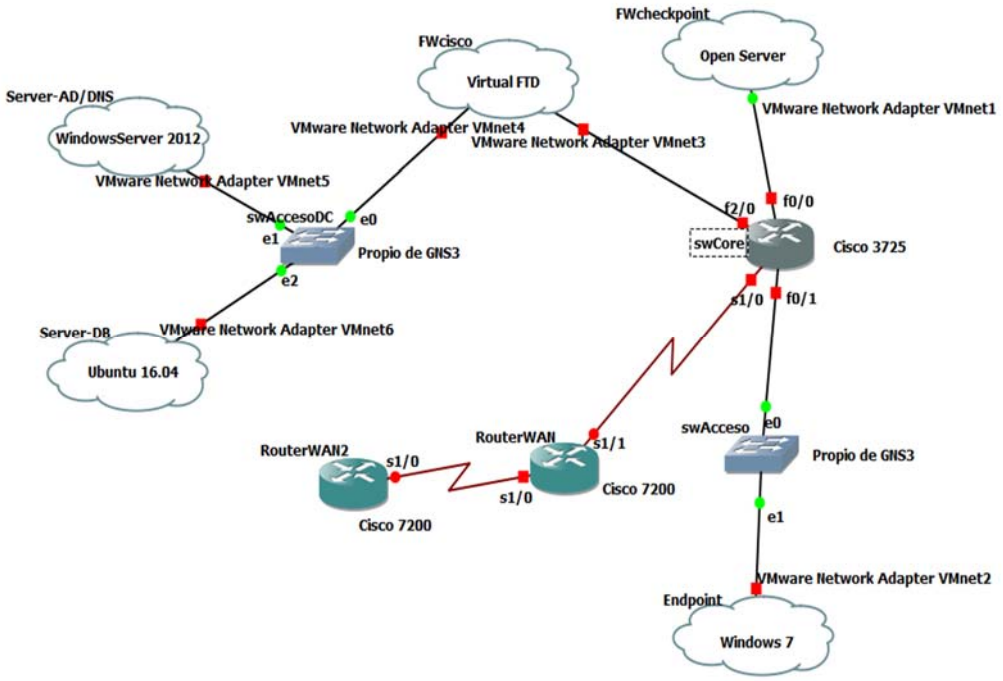

Figura 0.2 Interconexión de Dispositivos GNS3-VMware

Se configuran las interfaces del lado de GNS3 para la comunicación con los dispositivos virtuales:

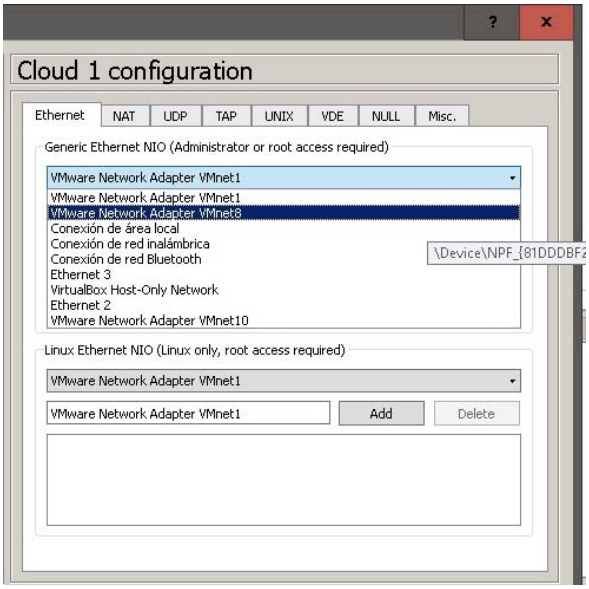

Figura 0.3 Configuración de interface en GNS3

De igual forma en Vmware, a nivel de cada máquina instalada se configura la interface:

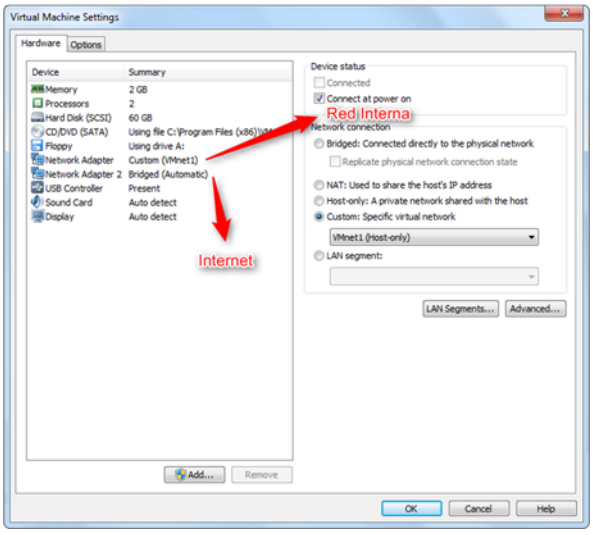

Figura 0.4 Configuración de interface en VMware

Las medidas de seguridad del entorno utilizado son reglas en los firewalls checkpoint y cisco ASA sistema de antivirus kaspersky para endpoints, se protege la comunicación con una VPN a nivel de aplicación entre el cajero automático y el servicio transaccional de la entidad. El entorno cuenta con el sistema de directorio activo para administración de usuarios.

5.2 Instalación e integración de componentes del prototipo

Dentro del entorno simulado se consideran los componentes descritos en el apartado 4.5 del capítulo 4, en la siguiente figura se designa a cada uno de los componentes un número para identificarlos dentro del escenario.

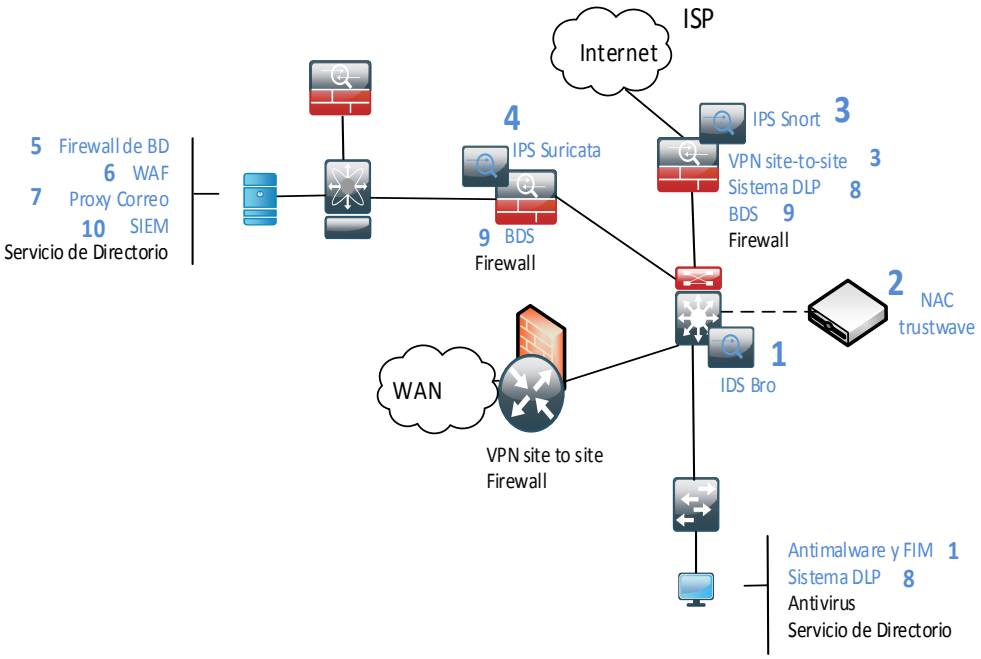

Figura 0.5 Integración de componentes

5.2.1 Componentes Base

Los componentes que se asumen como base en una entidad financiera y de los cuales se parte son el firewall perimetral, en el caso de la simulación se usó checkpoint virtualizado en workstation con open server GAIA versión R77.30.

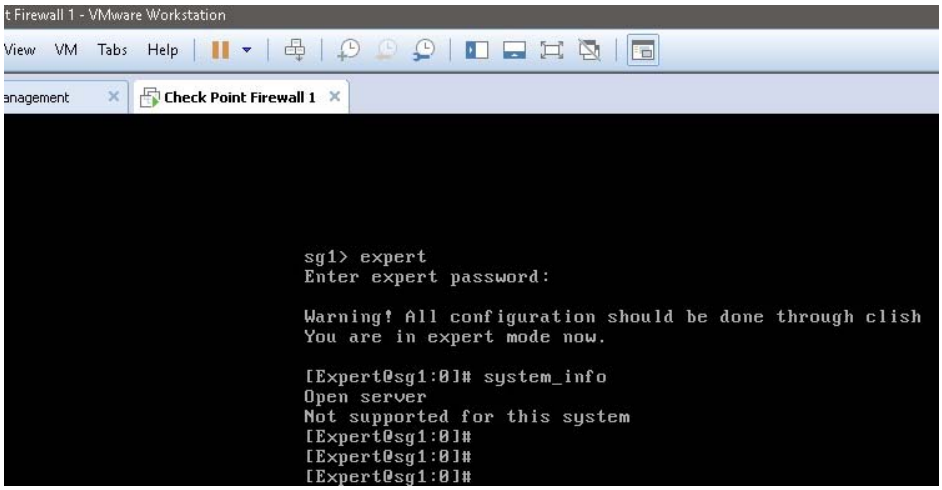

Figura 0.6 Firewall Checkpoint

El segundo componente base es el antivirus en el caso del presente prototipo se utilizó Kaspersky ya que cuenta con una consola que permite la gestión centralizada de actualizaciones, así como la distribución de políticas de seguridad.

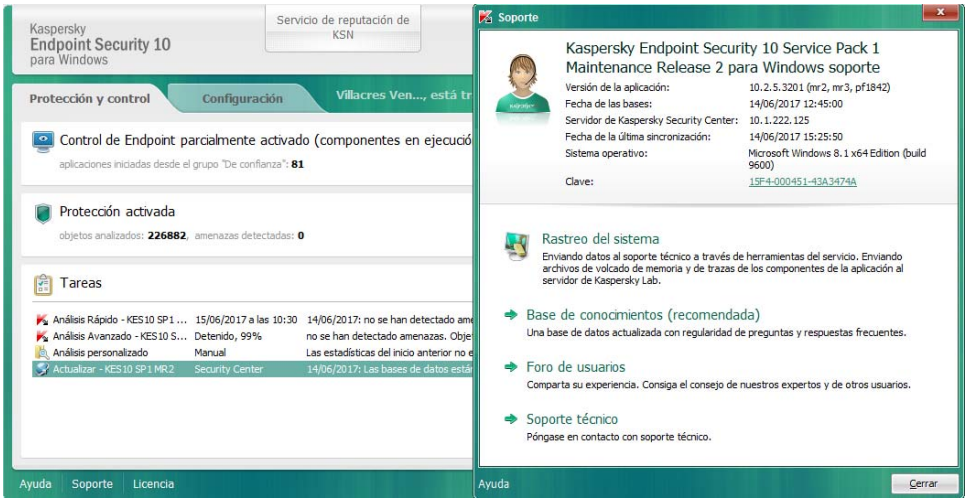

Figura 0.7 Kaspersky Endpoint

5.2.2 Componentes propuestos

Primer componente: se instala el antimalware malwarebytes, y el agente de FIM Ossec en las máquinas de los usuarios.

Segundo componente: se implementa un NAC, instalado de manera no intrusiva, el switch de core tiene configurado dos puertos para conectar el componente: un puerto espejo para la redirección del tráfico y otro en modo trunk, del lado del NAC la interface hacia el puerto espejo es configurado de solo lectura, es decir, no es posible alterar el tráfico, mientras que la interface que va hacia el puerto trunk no es de solo lectura, permite inyectar paquetes esta interfaz se denomina de remediación.

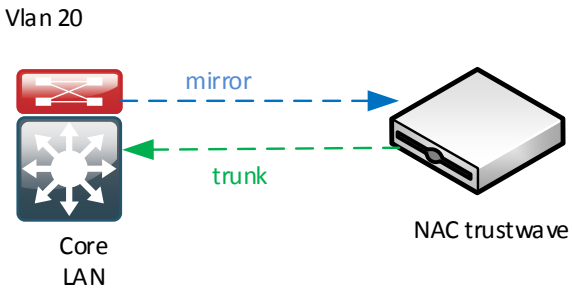

Figura 0.8 Integración NAC

Los puertos del switch del core se configuran de la siguiente manera:

Configuración del puerto en modo trunk:

```
Sw-core(config-if)# switchport mode trunk
```

Configuración de encapsulacion

```
Sw-core(config-if)# switchport trunk encapsulation dot1q
```

Configuración para que el trunk no use DTP

```
Sw-core(config-if)# switchport nonegotiate
```

Configurar el puerto espejo de la VLAN fuente (usuarios)

```
Sw-core(config)# monitor session 1 source vlan 20
```

Configuración del Puerto espejo destino

```
Sw-core(config)# monitor session 1 destination interface
g1/1
```

Verificar la configuración del mirror

```
Sw-core# show monitor
```

Verificar la configuración del trunk

```
Sw-core# show interface g1/1 trunk
```

Tercer componente: se integra el IPS snort al firewall perimetral, y las configuraciones de VPN client to site.

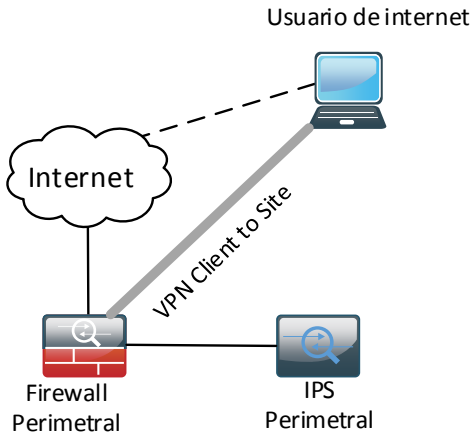

Figura 0.9 Integración IPS Firewall Perimetral

El IPS al ser un sistema orientado a malware contiene reglas snort para bloquear el tráfico malicioso que pasa por el firewall, estas reglas pueden ser importadas con el siguiente comando:

```
[Expert@cpmodule]# SnortConvertor update –f snort.rules
39/39 rules were successfully converted, total of 38 IPS
protections were found.
38/38 IPS protections were updated
Updating database...
Database updated successfully
Please note that for the new configuration to take effect,
you need to make sure that the new protections are activated
and then to install policy.
[Expert@cpmodule]#
```

Cuarto componente: IPS con sandboxing para la protección de acceso al Data Center, debe estar implementado en línea, de forma que pueda realizar restricción directa de tráfico identificado como malicioso, de acuerdo al reporte entregado por parte de la sandbox.

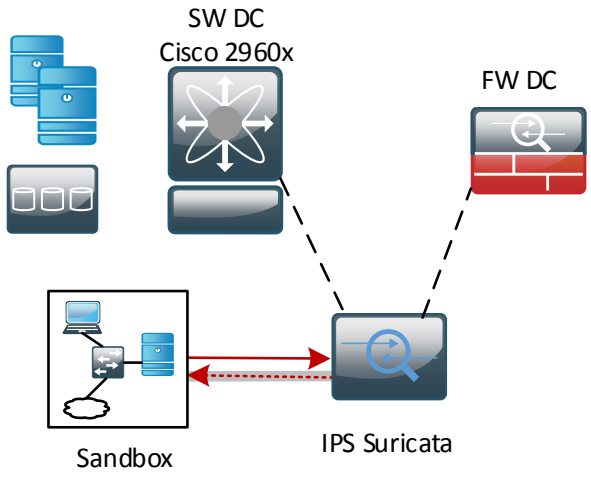

Figura 0.10 Integración IPS DataCenter

Por defecto, Suricata trabaja como un IDS. Ejecutar el siguiente comando para que trabaje como IPS también:

```
apt-get -y install libnetfilter-queue-dev
```

Descarga y configuración de las últimas reglas disponibles en Emerging Threats para Suricata:

```
./configure && make && make install-rules
make install-rules
```

Quinto componente: el firewall GreenSQL se coloca en línea entre la aplicación web (que es visible para los usuarios) y el gestor de bases de datos (que interactúa con la base de datos MySQL) como se puede ver en la siguiente figura:

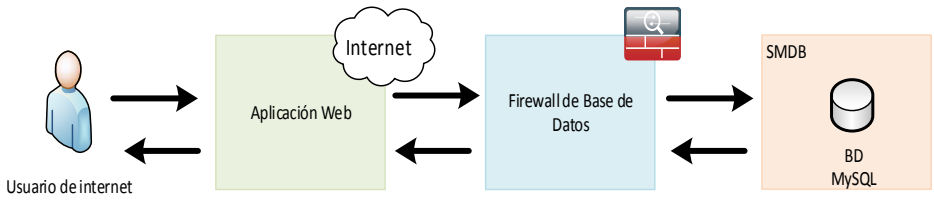

Figura 0.11 Integración firewall de BD

Sexto componente: implementación de un Firewall de Aplicación Web en línea para la DMZ colocado entre el firewall perimetral y los servidores web.

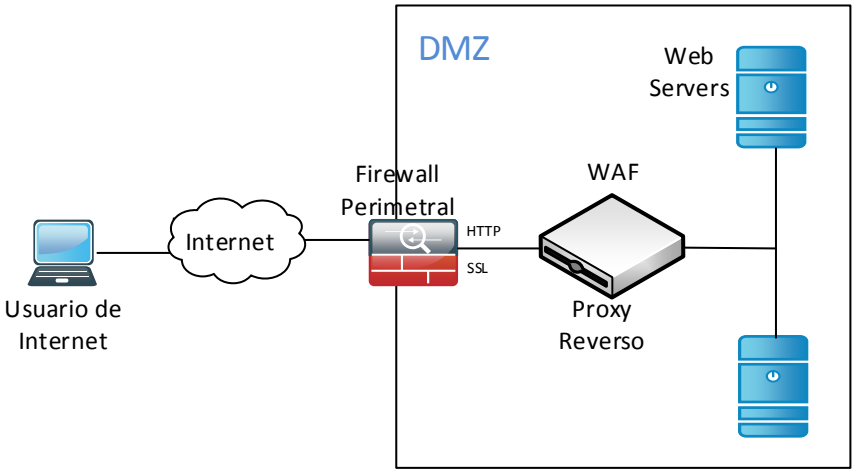

Figura 0.12 Integración WAF

Séptimo componente: con el fin de filtrar el correo electrónico antes que este llegue al servidor de correo interno, se configura el Endian para recibir todos los correos electrónicos de Internet. Así como para escanear y filtrar de forma transparente todo el correo SMTP saliente.

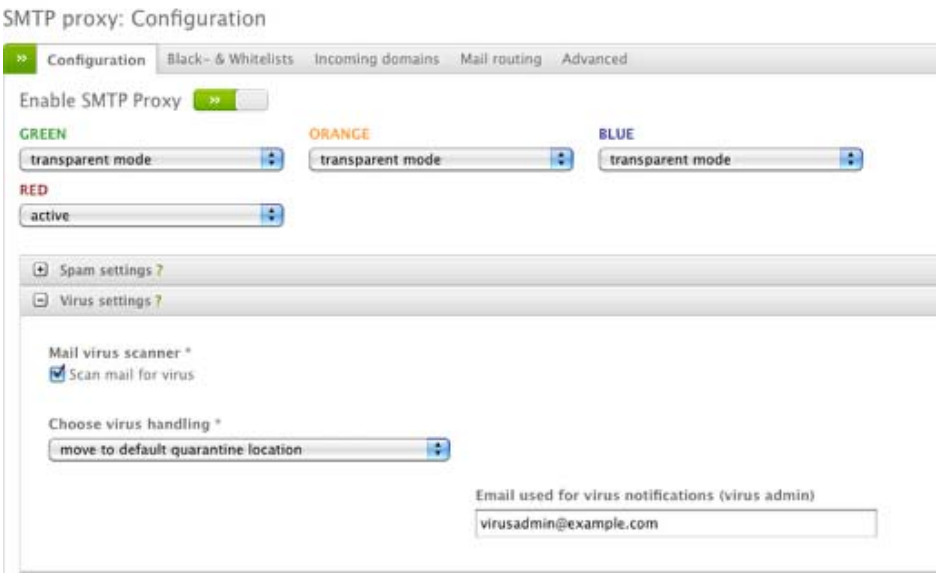

Figura 0.13 Configuración Proxy Correo

Octavo componente: se habilita el sistema de DLP en el firewall checkpoint, además el servicio FIM ossec instalado en los endpoint soporta habilitar DLP.

Noveno componente: tanto el IPS snort como el suricata cuentan con un sistema de detección de brechas, de igual forma el IDS bro instalado en el switch core permite trabajar como BDS.

```
HTTP normalization and anomaly detection.
preprocessor http_inspect
```

```
preprocessor http_inspect_server
```

Décimo componente: se integra splunk instalado en workstation sobre una máquina virtual de Linux Ubuntu 16.04.

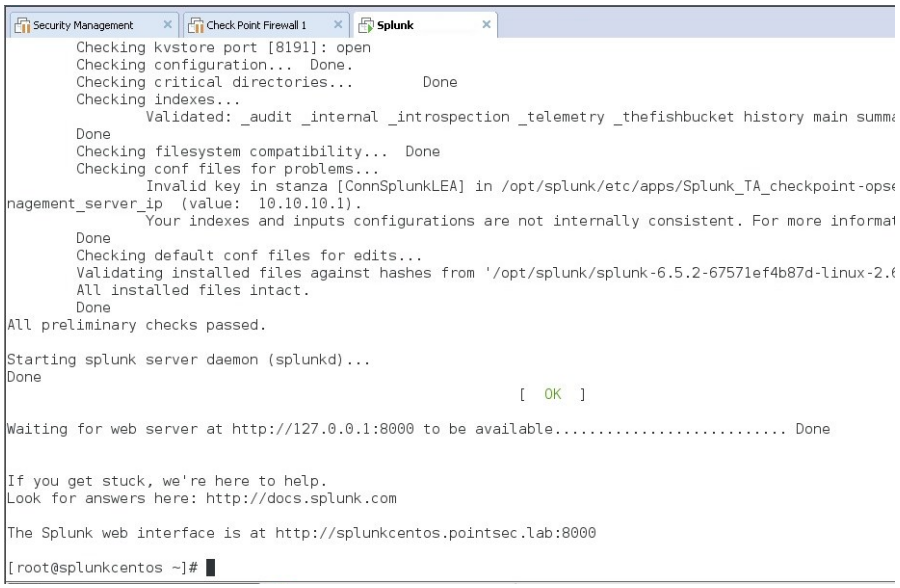

Figura 0.14 Ejecución del Servicio de splunk

Es un sistema recolector de logs procedentes de toda la entidad financiera.

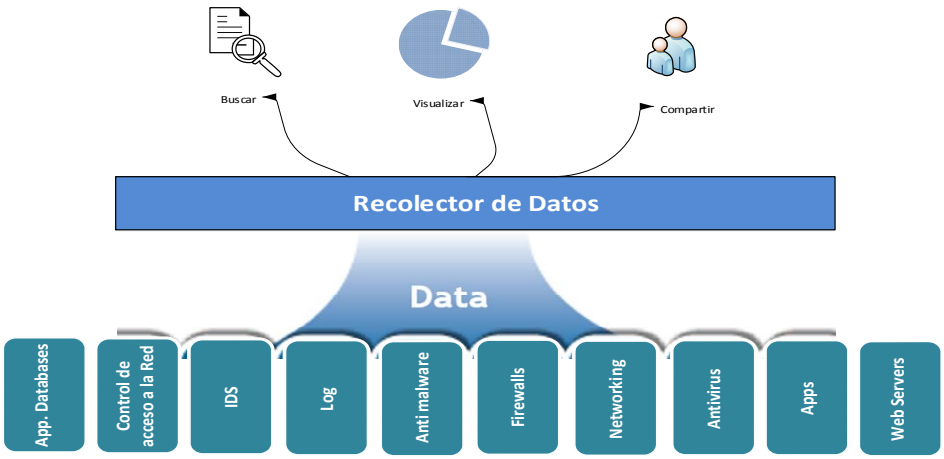

Figura 0.15 Integración SIEM avanzado

Todos los logs llegan al servidor recolector de datos en texto plano, las fuentes como aplicaciones de base de datos, NAC, IDS, Antimalware y FIM, firewalls, equipos de networking (router y switch), antivirus, servidores web y las aplicaciones envían sus logs hacia el SIEM, es decir trabajan con el mecanismo push, mientras que las estaciones de trabajo y los servidores DNS y AD permiten utilizar el mecanismo pull para la obtención de datos.

5.3 Ejecución de ataques.

Se colocan las muestras bajo la arquitectura simulada con su nombre exacto, la siguiente tabla muestra las funcionalidades de cada una de ellas:

Tabla 14 Funcionalidades de malware

Archivo	Funcionalidades

61780002.exe	1. Backdoor 2. Troyano 3. Spyware 4. Comando y control 5. Botnet
9C4DB7EE	1. Clicker 2. Troyano 3. Downloader 4. Rootkit
06377098cd6228696598e66d b000f1b0	1. DDoS 2. Troyano 3. Backdoor 4. Downloader 5. Técnicas de Ofuscación 6. Proceso de evasión 7. Rootkit 8. Un proceso intentó detener el Firewall de Windows
4c8208ae.vbs	1. Downloader 2. Virus 3. VBScript puede llamar a Shell 4. Script contiene URL

Taskxpl.wsf	1. Exploits 2. Gusano 3. Troyano 4. Backdoor 5. Downloader
cd3bc0bedb3900ff3c64.exe	1. Ransomware 2. Botnet 3. Troyano 4. Backdoor
Igfxper.exe	1. Spyware 2. Troyano 3. Keylogger 4. Downloader
63780001.9287.dbe	1. Troyano 2. DoS 3. Proceso de evasión

Para realizar el ataque se ha colocado la muestra en una máquina de usuario final, como lo muestra la Figura 0.16 Colocar muestra en el Entorno:

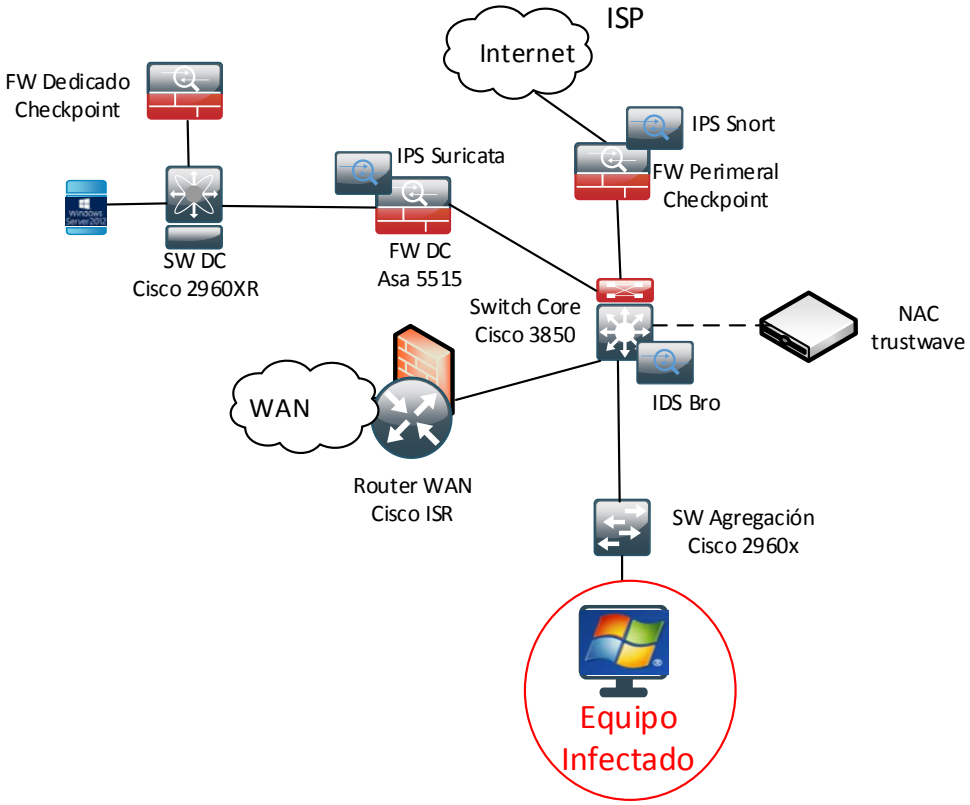

Figura 0.16 Colocar muestra en el Entorno

Se realiza snapshot del equipo infectado para regresar a un punto en el que la máquina se encuentre limpia para el siguiente ataque.

5.4 Pruebas de funcionalidad

El objetivo de esta sección es observar el resultado de los controles de seguridades convencionales aplicadas por las entidades financieras, junto con los componentes propuestos.

Tabla 15 Pruebas de funcionalidad

Muestra A: 61780002.exe|	
Herramienta	**Resultado**
Antivirus del paciente cero	No lo detectó debido a que es una APT de día cero
Firewall	El ataque inicia desde la red interna, por lo que inicialmente el firewall no detecta la amenaza
Cuarto componente	Muestra hecha de comunicaciones IRC salientes hacia la IP 170.178.191.18 con el puerto 49157. El malware modifica los archivos en el directorio \Windows\kernel32.exe para ocultar registros u otras pruebas
Muestra B: 9C4DB7EE	
Herramienta	**Resultado**
Antivirus del paciente cero	Este ataque fue controlado por el antivirus debido a que la firma SHA256 9ff37d103bd84c2bd168e70528b7141c4fe7477e3d9dde44b22748ef0b5280ff formaba parte de su base de datos. El

	antivirus alerta al usuario y solicita la eliminación o cuarentena del archivo, deteniendo así el análisis del ataque
Firewall	El ataque inicia desde la red interna, por lo que inicialmente el firewall no lo detecta
Primer componente	El tráfico de red DNS se envió al servidor ace.tpmk.info que no es el servidor DNS asignado del sistema.
Segundo componente	Intenta evitar el filtrado de la red u ocultar el tráfico de red malicioso.
Muestra C: 06377098cd6228696598e66db000f1b0	
Herramienta	**Comportamiento**
Antivirus del paciente cero	Firma de malware contenida en la base de datos del antivirus con SHA256 fdd762878a7b157facd75577ecf632edef8795ce0da7fe0bac8265452b7e4191. Además desactiva el firewall de Windows.
Firewall	El ataque inicia desde la red interna, por lo que inicialmente el firewall no lo detecta

Tercer componente	El malware intenta contactar a su controlador svchost.exe a través de mensajes HTTP POST, el sistema analiza el tipo de paquete de red y bloquea la comunicación.
Muestra D: 4c8208ae.vbs	
Herramienta	**Resultado**
Antivirus del paciente cero	La muestra no es vista como amenaza para el antivirus
Tercer componente	Existe una URL codificada dentro del script. La URL http://kakssoftwareat.info/PCDefenderSilentSetup.msi ha sido determinada como maliciosa.
Sexto componente	El WAF identifica que el malware usa esta URL para descargar la siguiente etapa de un ataque
Muestra E: Taskxpl.wsf	
Herramienta	**Resultado**
Antivirus del paciente cero	Es una APT de día cero, por lo que su firma no es parte de la base de datos del antivirus
Segundo componente	Lo aísla de la red cuando intenta obtener acceso privilegiado al sistema operativo utilizando

	C:\Windows\System32\svchost.exe -k LocalServiceNetworkRestricted en la línea de comandos
Tercer componente	IPS Snort lo ha señalado como potencialmente malintencionado.
Muestra F: cd3bc0bedb3900ff3c64.exe	
Herramienta	**Resultado**
Antivirus del paciente cero	No es visto como amenaza para el antivirus
Cuarto componente	Contiene C:\Perl\perl\vendor\lib\auto\Tie\EncryptedHash*.* dentro de su código para encriptar los datos del directorio y HKEY_CLASSES_ROOT\.crypted dentro de sus claves de registro
Muestra G: lgfxper.exe	
Herramienta	**Resultado**
Antivirus del paciente cero	La firma forma parte de la base de datos del antivirus, por lo que es controlado por él, bloqueando el análisis del ataque
Cuarto componente	Bloquea la comunicación con el dispositivo, contiene process: {u'process_id': 1592, u'process_name':

	u'igfxper.exe'} en su base de datos de firmas, esto abre una ventana para capturar pulsaciones
Muestra H: 63780001.9287.dbe	
Herramienta	**Resultado**
Antivirus del paciente cero	Controlado por el antivirus, bloqueando el análisis del ataque
Primer componente	El malware altera la extensión de un archivo ejecutable al crear el archivo en disco con el fin de ocultar el tipo de archivo. C:\WINDOWS\system32\msctfime.ime C:\Documents and Settings\User\Application Data\A3D\license.lua

Cada uno de estos sistemas interpreta y gestiona los datos de forma independiente

CAPÍTULO 6

ANÁLISIS DE RESULTADOS

6.1 Correlación de Eventos

La correlación de eventos se utiliza para la detección de comportamiento anómalo que no es detectado por los componentes de seguridad como una amenaza, ya que se trata de actividades cotidianas como el inicio de sesión.

En la siguiente sección se detalla el proceso de correlación de eventos manual que se realiza en la herramienta splunk para la detección de un malware.

La primera búsqueda que hacemos es de falla de autenticación.

```
fail* password
```

La siguiente figura permite identificar el sistema donde ocurrieron las fallas, en splunk se lo conoce como *sourcetype* al tipo de aplicación.

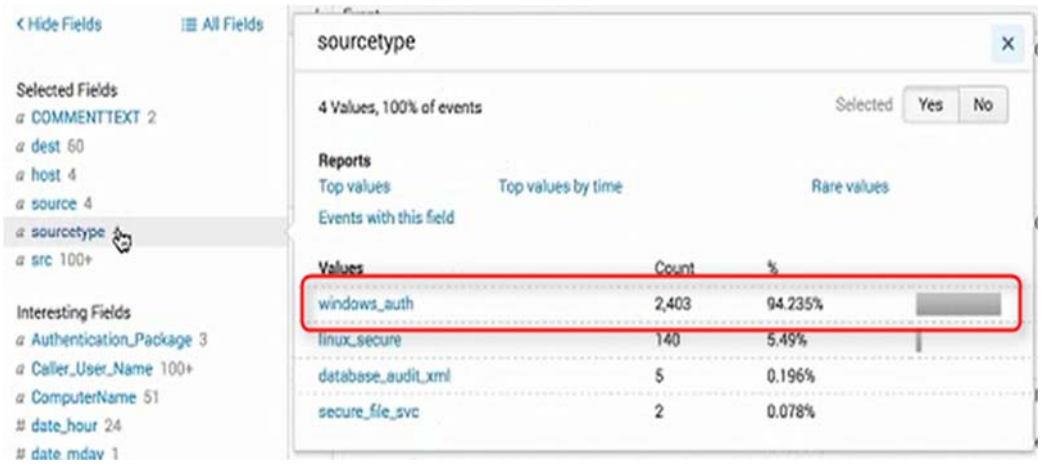

Figura 0.1 Correlación por tipo de aplicación

Splunk hace la agregación y muestra el total de los eventos por aplicación, el 94% ocurrieron en Windows.

En la figura siguiente se conoce el servidor donde existieron las fallas de autenticación, ECOMMERCE-03 contiene más del 50% de estas fallas.

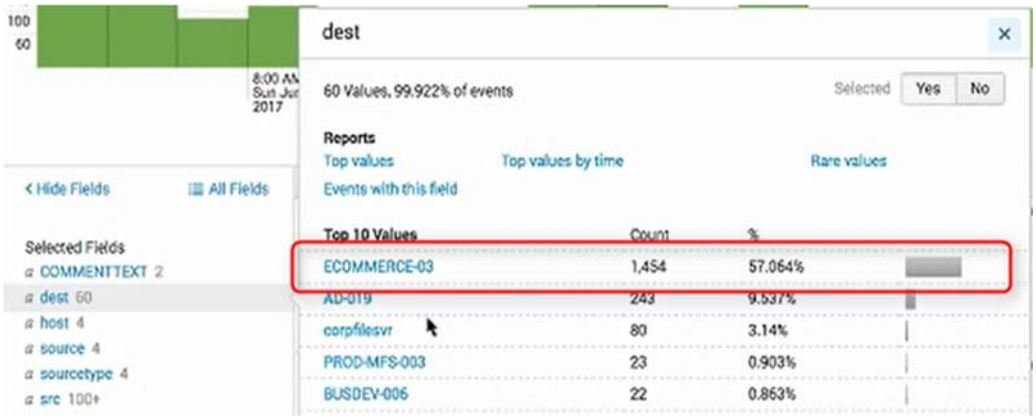

Figura 0.2 Correlación por destino

Es importante conocer desde que máquina se intenta autenticar, la siguiente figura tiene información que la IP 10.11.36.20 tiene un gran número de fallas de autenticación, lo cual nos lleva a pensar que se trata de una máquina infectada.

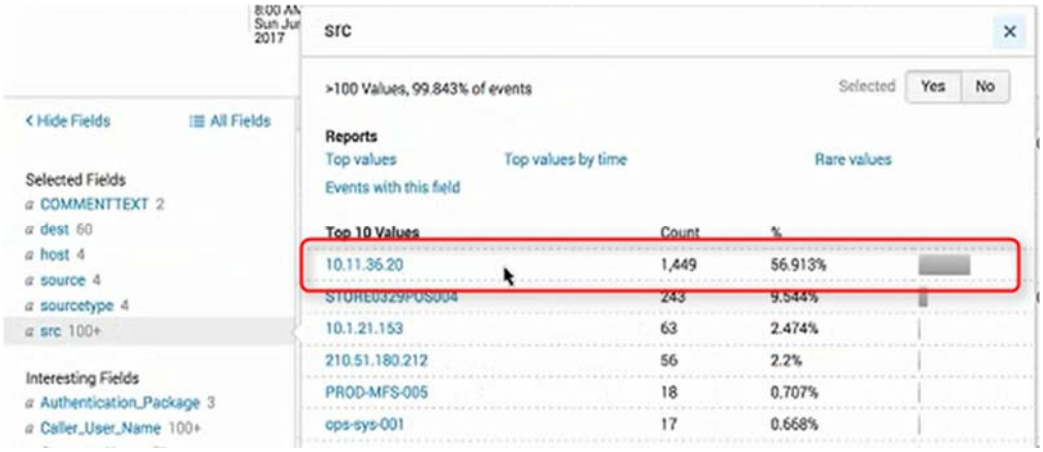

Figura 0.3 Correlación por origen

Spunk permite identificar cual fue el usuario que estaba registrado en esas máquinas cuando se hace la falla de autenticación.

Figura 0.4 Correlación por usuario

Para agilitar las búsquedas splunk maneja comandos que se pueden usar para procesar información, la integración de los siguientes comandos a la búsqueda nos da la información por origen, destino, tipo de aplicación y usuario, contada y ordenada de forma descendente.

```
fail* password
| stats count by src, dest, sourcetype, user
| sort - count
```

El resultado de la consulta es una tabla que permite analizar y entender la información antes identificada de manera independiente, ahora presentada de manera agrupada

src	dest	sourcetype	user	count
10.11.36.20	ECOMMERCE-03	windows_auth	Hax0r	1442
[redacted]	[redacted]	[redacted]	[redacted]	[redacted]
210.51.180.212	corpfilesvr	linux_secure	root	56
65.49.34.15	corpfilesvr	linux_secure	root	8
10.1.21.153	web_cloud_01	linux_secure	manager	4

Figura 0.5 Correlación eventos utilizando comandos

Entonces se determina que es un ataque que tiene un diccionario de cuentas, y posiblemente está tratando con un script automatizado de autenticarse utilizando miles de combinaciones para la clave.

6.2 Nivel de Detección

El nivel de detección se define de acuerdo a las siguientes características

Tabla 16 Descripción de los Niveles de Detección

Nivel de detección	**Descripción**
Alto	El componente de seguridad fue capaz de detectar todas las funcionalidades de la muestra de malware
Medio	El componente de seguridad fue capaz detectar la funcionalidad principal y funcionalidades secundarias de la muestra de malware

Bajo	El componente de seguridad fue capaz de detectar al menos una funcionalidad de la muestra de malware
Nulo	El componente de seguridad no fue capaz de detectar ninguna funcionalidad de las muestra de malware

Cada uno de los ataques cuenta con diferentes niveles de detección, de acuerdo a la efectividad de los componentes del prototipo al momento de detectar las funcionalidades de cada una de las muestras. En la siguiente tabla se detalla el nivel de efectividad de cada uno de los ataques.

Tabla 17 Nivel de Detección por muestra

Archivo	**Funcionalidad**	**Detección**	**Nivel de Detección**
Muestra A	1, 3	No detectado	Medio
	2	Noveno Componente	
	4	Tercer Componente	
	5	Cuarto componente	

Muestra B	1, 2	Antivirus componente base, Primer componente y Decimo componente	Alto
	3	Octavo componente y Segundo componente	
	4	Primer componente	
Muestra C	1,5	Tercer componente	Medio
	2	Antivirus componente base y decimo componente	
	3, 8	Primer componente y decimo componente	
	4	Cuarto componente	
	6, 7	No detectado	
Muestra D	2	No detectado	Medio
	3	Primer componente	
	4, 1	Tercer componente y Sexto componente	
Muestra E	1	Segundo componente	Medio

	2, 5	No detectado	
	3, 4	Tercer componente, Cuarto componente y sexto componente	
Muestra F	1, 2	Cuarto componente	Alto
	3, 4	Primer componente	
Muestra G	1	Componente base, tercer componente y octavo componente	Alto
	2,4	Componente base y primer componente	
	3	Primer componente, Tercer componente y cuarto componente	
Muestra H	1,3	Antivirus componente base, primer componente, noveno componente	Medio
	2	No detectado	

6.3 Análisis Descriptivo

En este análisis se describen los datos obtenidos respecto al nivel de detección, alcance de las muestras, respuesta de los componentes propuestos, los cuales

serán agrupados y representados de forma ordenada con la finalidad de identificar el comportamiento de los mismos.

La siguiente tabla agrupa los componentes para cada nivel de detección:

Tabla 18 Nivel de Detección por componente

Nivel de detección	Cantidad de componentes
Alta	8
Media	7

Los resultados representados en la siguiente figura:

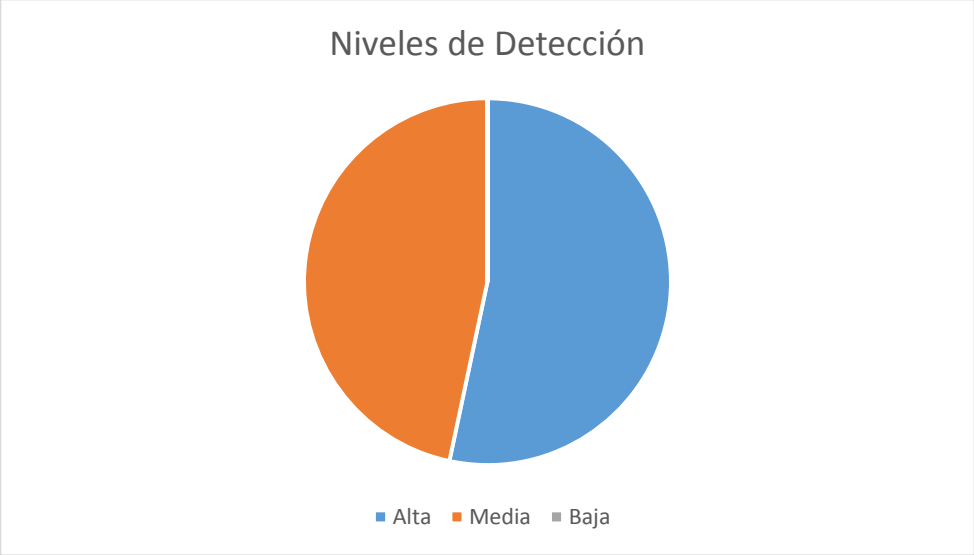

Figura 0.6 Nivel de Detección por Componente

Esto nos indica que el prototipo cuenta con componentes en su mayoría con niveles de detección ALTA, no existió nivel de detección bajo o nulo.

En la siguiente tabla se detallan los componentes ordenados según la cantidad de funcionalidades de malware que fueron capaces de detectar.

Tabla 19 Funcionalidades detectadas

Componente	**Número de funcionalidades detectadas**
Primer Componente	8
Segundo Componente	2
Tercer Componente	6
Cuarto Componente	5
Sexto componente	2
Octavo componente	2
Noveno componente	2
Decimo componente	3

En la siguiente figura de barras se listan en orden los componentes según la cantidad de funcionalidades que han sido detectadas de mayor a menor.

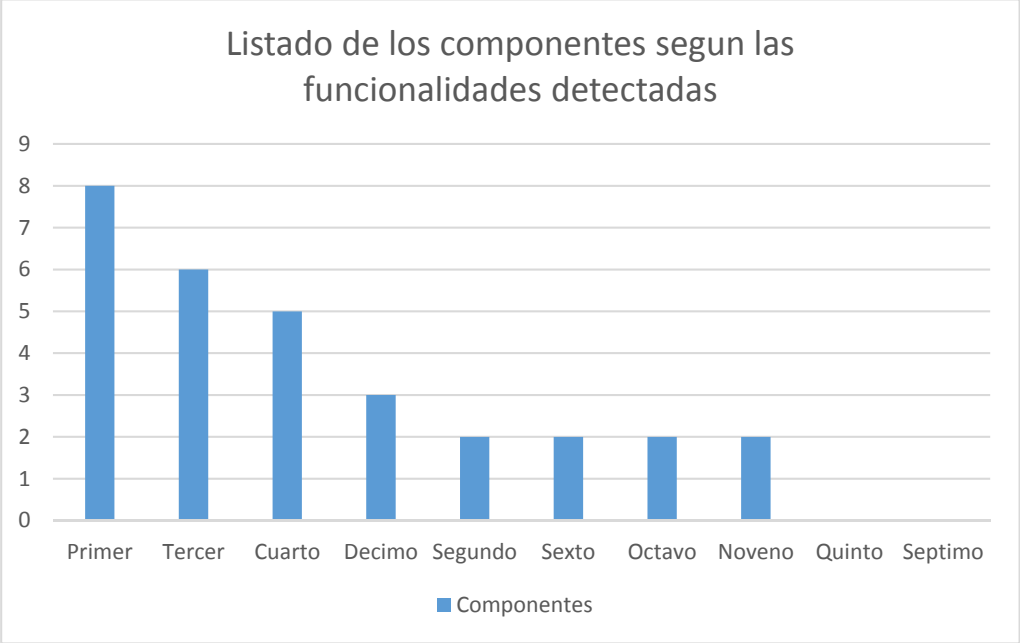

Figura 0.7 Funcionalidades detectadas por los componentes

El componente con mayor número de detección es el primero, se debe a que el ataque inicia desde una máquina de usuario interno, por el mismo motivo los componentes quinto y séptimo, no tienen registro de detección.

En la siguiente tabla se listan las muestras de Malware con el número total de funcionalidades identificadas, junto con el número de funcionalidades detectadas por los componentes del prototipo

Tabla 20 Porcentaje de detección

Malware	**Funcionalidades Identificadas**	**Funcionalidades detectadas**	**Porcentaje de detección**

Muestra A	5	3	60%
Muestra B	4	4	100%
Muestra C	8	6	75%
Muestra D	4	3	75%
Muestra E	5	3	60%
Muestra F	4	4	100%
Muestra G	4	4	100%
Muestra H	3	2	66,67%

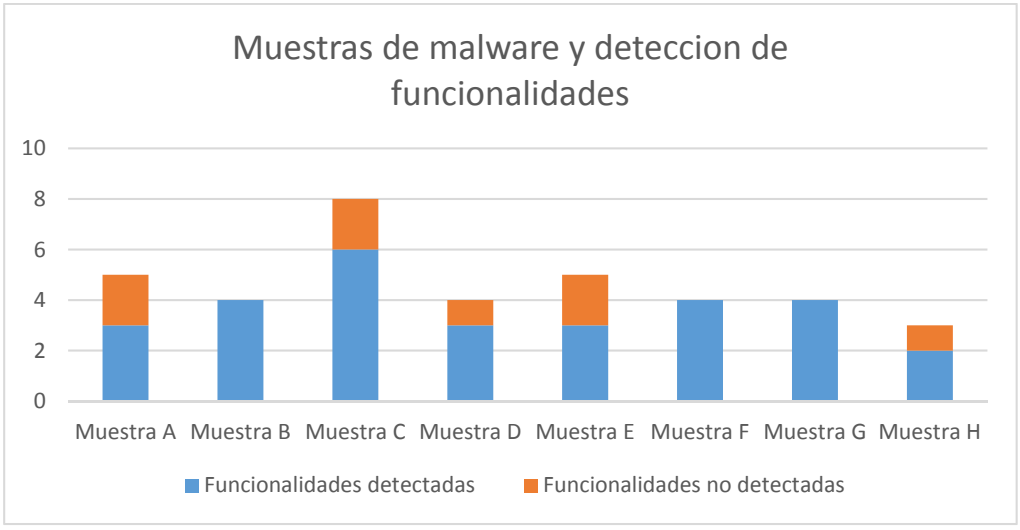

Figura 0.8 Porcentaje de detección

Esta tendencia nos indica que el malware que tuvo más éxito en pasar los controles de seguridad implementados en el modelo de seguridad fueron la muestra A y la muestra E, debido a que de las cinco funcionalidades dos de ellas no fueron detectadas.

CONCLUSIONES Y RECOMENDACIONES

Conclusiones

1. Se logró generar una arquitectura donde se integraron varios componentes de seguridad que se complementan en las tareas de detección y cubren los diferentes vectores de acción sobre los que actúa el malware.

2. Se evidenció que cada componente de la arquitectura detecto funcionalidades diferentes en cada muestra de malware, de ahí que se consideran necesarios controles a nivel de host para el análisis de malware en su entorno de ejecución, así como controles a nivel de red para el análisis de malware en su entorno de transporte.

3. Se comprobó que los componentes de seguridad agregados al modelo detectaron amenazas que no fueron detectadas por los controles de seguridad tradicionales como el anti virus de host y el firewall perimetral.

4. Se demostró que los controles de seguridad actúan de manera individual para ejecutar sus motores de detección, pero todos los controles permiten el envío de logs a un repositorio central de logs.

5. Se expuso que la detección de amenazas que realizan actividades no maliciosas se pueden detectar por medio de herramientas como el SIEM que es capaz de correlacionar eventos cotidianos que finalmente resultan como maliciosos cuando se los analiza en una línea de tiempo como un ataque de fuerza bruta con diccionario de claves aplicado de manera periódica.

6. El análisis manual estático y dinámico requiere de mayor tiempo y conocimientos especializados en arquitectura de sistemas operativos, y resulta más meticuloso que las herramientas automatizad en la investigación de muestras de malware.

Recomendaciones

1. Es conveniente para una institución financiera contar con el personal, las herramientas y el entorno adecuado para realizar un análisis estático y dinámico de malware con la finalidad de verificar la información proporcionada por los motores de análisis automático que poseen los componentes de la arquitectura. Y que les permita realizar un análisis mucho más minucioso de los hallazgos identificados.

2. No resulta viable instalar varios controles de seguridad que realicen el mismo análisis a nivel de host ya que estos degradan el rendimiento del sistema, de igual manera a nivel de red no se recomienda colocar más de un control de acceso a red ya que esto puede generar retardo en la comunicación.

3. Las herramientas de código abierto que se implementan como controles de seguridad, se deben seleccionar en base al respaldo que tengan para su continuidad, ya sea este el respaldo de una comunidad estable o una empresa reconocida.

4. Para un entorno de producción es importante realizar un dimensionamiento correcto del rendimiento requerido de recursos de hardware o recursos virtuales requerido para que el análisis de los controles no genere cuellos de botella o retardos en las operaciones de la entidad.

5. El prototipo propuesto no reemplaza las seguridades tradicionales, sino que es una ayuda para la protección de los datos, y al estar orientado a APT permite generar información para los analistas de malware.

BIBLIOGRAFÍA

[1] S. Gorman, «Cyber Crime Costs Projected To Reach $2 Trillion by 2019,» The Wall Street Journal, 22 Julio 2013. [En línea]. Available: http://www.forbes.com/sites/stevemorgan/2016/01/17/cyber-crime-costs-projected-to-reach-2-trillion-by-2019/#2edc454f3bb0. [Último acceso: 20 Septiembre 2016].

[2] M. Steve, «Cyber Crime Costs Projected To Reach $2 Trillion by 2019,» The Wall Street Journal, 26 Mayo 2016. [En línea]. Available: http://www.wsj.com/articles/SB10001424127887324328904578621880966242990. [Último acceso: 11 Noviembre 2016].

[3] «welivesecurity,» 21 enero 2010. [En línea]. Available: https://www.welivesecurity.com/la-es/2010/01/21/que-es-operacion-aurora/.

[4] Agencia EFE, «El 98,5 por ciento de los riesgos bancarios en América Latina son informáticos,» Agencia EFE, 15 Octubre 2013. [En línea]. Available: http://www.efe.com/efe/america/economia/el-98-5-por-ciento-de-los-riesgos-bancarios-en-america-latina-son-informaticos/20000011-2738875. [Último acceso: 12 diciembre 2016].

[5] Khandelwal, Swati, «The Hacker News,» 20 Mayo 2016. [En línea]. Available: http://thehackernews.com/2016/05/swift-banking-hack.html.

[6] Kaspersky, «CARBANAK APT THE GREAT BANK ROBBERY,» febrero 2015. [En línea]. Available: https://securelist.com/files/2015/02/Carbanak_APT_eng.pdf.

[7] Peter Stancik, «Soluciones a los desafíos de los antivirus actuales,» eset, 24 octubre 2013. [En línea]. Available: https://www.somoseset.com/2013/10/24/soluciones-desafios-antivirus-actuales/.

[8] Superintendencia de Bancos y Seguros del Ecuador, «Normas generales para la aplicación de la Ley General de Instituciones del Sistema Financiero,» Quito, 2012.

[9] PCI Security Standards Council, «Industria de Tarjetas de Pago (PCI) Norma de seguridad de datos,» 2016.

[10] D. Bestuzhev, «El nuevo ecosistema del cibercrimen en Latinoamérica,» Análisis & Opción, 29 septiembre 2011. [En línea]. Available: https://www.americaeconomia.com/analisis-opinion/el-nuevo-ecosistema-del-cibercrimen-en-latinoamerica.

[11] Jordi Medina, «APTs o Amenazas Persistentes Avanzadas,» 24 febrero 2015. [En línea]. Available: https://www.ibm.com/blogs/think/es-es/2015/02/24/cuidado-con-las-apts/.

[12] Marcelo Rivero, «infospyware,» 01 octubre 2016. [En línea]. Available: https://www.infospyware.com/articulos/que-son-los-malwares/.

[13] incibe, «CIBERSEGURIDAD PARA AUTÓNOMOS Y MICROEMPRESAS,» vol. 1, p. 36, 2017.

[14] Kaspersky Lab, «Seguridad 101: Los tipos de malware,» 2016. [En línea]. Available: http://support.kaspersky.com/sp/viruses/general/614.

[15] infySEC Global, Head - Cyber Security Research, «Virus, worm, trojan, backdoor & antivirus-malware and Security».

[16] MARCELO RIVERO, «QUÉ SON LOS MALWARES,» infospyware, 01 octubre 2016. [En línea]. Available: https://www.infospyware.com/articulos/que-son-los-malwares/.

[17] «CISSP Security Mechanisms,» CISSP, 03 marzo 2016. [En línea]. Available: http://asmed.com/cissp-security-mechanisms/.

[18] Maria Garnaeva, Jornt van der Wiel, Denis Makrushin, Anton Ivanov, Yury Namestnikov, «Kaspersky Security Bulletin 2015,» Kaspersly, 15 diciembre 2015. [En línea]. Available: https://securelist.com/kaspersky-security-bulletin-2015-overall-statistics-for-2015/73038/.

[19] O. S. Adebayo, «Techniques for Analysing Android Malware,» *bibliotecavirtual UPS,* p. 7, 2016.

[20] netbiosx, «Internet Explorer Aurora Exploit,» Pentesting Field, 12 marzo 2012. [En línea]. Available: https://pentestlab.blog/2012/03/12/internet-explorer-aurora-exploit/.

[21] Chema Alonso, «USB Rubber Ducky: Un teclado malicioso como un pendrive,» elladodelmal, 26 mayo 2014. [En línea]. Available: http://www.elladodelmal.com/2014/05/usb-rubber-ducky-un-teclado-malicioso.html.

[22] Hak5, «Hakshop by Hak5,» 2017. [En línea]. Available: http://hakshop.myshopify.com/products/usb-rubber-ducky-deluxe?variant=353378649.

[23] «Github,» 2017. [En línea]. Available: https://github.com/hak5darren/USB-Rubber-Ducky/wiki/Payloads.

[24] «netmarketshare,» diciembre 2016. [En línea]. Available: https://www.netmarketshare.com/.

[25] wikipedia, «Registro (hardware),» wikipedia, 30 junio 2017. [En línea]. Available: https://es.wikipedia.org/wiki/Registro_(hardware).

[26] wikipedia, «Pila (informática),» 07 agosto 2017. [En línea]. Available: https://es.wikipedia.org/wiki/Pila_(inform%C3%A1tica).

[27] wikipedia, «Offset (informática),» 07 febrero 2017. [En línea]. Available: https://es.wikipedia.org/wiki/Offset_(inform%C3%A1tica).

[28] «Our Exploit Acquisition Program,» zerodium, 2017. [En línea]. Available: https://www.zerodium.com/program.html.

[29] SpiderLabs Research, «Zero Day Auction for the Masses,» SpiderLabs Blog, 9 junio 2016. [En línea]. Available: https://www.trustwave.com/Resources/SpiderLabs-Blog/Zero-Day-Auction-for-the-Masses/. [Último acceso: 12 agosto 2016].

[30] Kaspersky, 26 mayo 2016. [En línea]. Available: https://usa.kaspersky.com/about/press-releases/2016_charging-mobile-devices-could-put-data-at-risk.

[31] cert.ro, «Monitoring DNS traffic for Security Threats,» botfree, 2015. [En línea]. Available: http://www.botfree.ro/articles/pages/en/2015-05-27-article-monitoring-dns-traffic-for-security-threats.html.

[32] A. M. Matteo Casenove, «Botnet over Tor: The Illusion of Hiding,» p. 10, 2014.

[33] «Tor documentation,» [En línea]. Available: https://www.torproject.org/docs/tor-hidden-service.html.en.

[34] Manage Engine, «Windows File Integrity Monitoring Software,» Manage Engine, 2017. [En línea]. Available: https://www.manageengine.com/products/active-directory-audit/windows-file-integrity-monitoring.html.

[35] D. Oktavianto y I. Muhardianto, «Getting Started with Automated Malware Analysis using Cuckoo Sandbox,» de *Cuckoo Malware Analysis*, 2013.

[36] «about TLS Callback in windows,» stack overflow, 2013. [En línea]. Available: https://stackoverflow.com/questions/14538159/about-tls-callback-in-windows.

[37] M. Sikorski y A. Honig, «Anexo A,» de *Practical Malware Analysis*, 2012.

[38] «malwr,» [En línea]. Available: https://malwr.com/submission/.

[39] R. Cyrus, «Detecting Malicious SMB Activity Using Bro,» SANS Institute InfoSec Reading Room, 2016.

[40] Peter Kacherginsky, «FakeNet-NG: Next Generation Dynamic Network Analysis Tool,» Advanced Malware, 03 agosto 2016. [En línea]. Available: https://www.fireeye.com/blog/threat-research/2016/08/fakenet-ng_next_gen.html.

[41] «The BSD syslog Protocol,» Agosto 2001. [En línea]. Available: http://www.ietf.org/rfc/rfc3164.txt.

[42] International Organitation for Standardization: ISO 27001 - information security management, «ISO/IEC 27001 - Information security management,» 2013. [En línea]. Available: http://www.iso.org/iso/home/standards/management-standards/iso27001.htm. [Último acceso: 01 05 2016].

[43] Algo del Estado, «Título de la página WEB,» Autor, Quito, 2015.

[44] Alex Drozhzhin, «El Robo del Siglo: Hackers Se Llevan Mil Millones de Dólares,» 18 febrero 2015. [En línea]. Available: https://latam.kaspersky.com/blog/el-robo-del-siglo-hackers-se-llevan-mil-millones-de-dolares/5023/.

[45] Cengage Learning, Ethical Hacking Countermeasures: Threats & Defense Mechanisms, Unated States of America, 2010.

GLOSARIO

Amenaza.- Persona, organización o entidad que desee o pueda detonar un evento donde se pueda ver comprometida la seguridad de un activo.

Antivirus.- Programa de seguridad que se instala en la computadora o dispositivo móvil para protegerlo de infecciones por malware

Ataque.- Situación donde una persona intenta romper la seguridad de un sistema.

Buenas Prácticas.- Conjunto coherente de acciones que han rendido o excelente servicio en un determinado contexto.

Código malicioso.- Hardware, software o firmware que es intencionalmente introducido en un sistema con un fin malicioso o no autorizado

DDoS.- Distributed Denial of Service (ataque distribuido denegación de servicio), se ataca al servidor desde muchos ordenadores para que deje de funcionar

Filtrado de contenido.- Bloqueo de información indeseable de Internet.

Firewall.- Sistema o combinación de sistemas que refuerzan los límites entre dos o más redes. Un firewall regula el acceso entre las redes de acuerdo a una política de seguridad específica.

Log.- Es un registro oficial de eventos durante un rango de tiempo en particular.

Riesgo.- Valor de las pérdidas a que se exponen las empresas por la ocurrencia de eventos perjudiciales.

Sniffer.- Programa empleado para escuchar todo el tráfico que pasa por una red.

Vulnerabilidad.- Son puntos débiles del software que permiten que un atacante comprometa la integridad, disponibilidad o confidencialidad del mismo