UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

Análisis de Malware con Cuckoo SandBox

TESIS

Que para obtener el título de

Ingeniero en Computación

P R E S E N T A

Alejandro Bárcenas Godínez

DIRECTOR DE TESIS

Ing. Aldo Jiménez Arteaga

Ciudad Universitaria, Cd. Mx., Octubre 2016

Dedicado a la memoria de mi padre, quién me proporcionó su apoyo y amor de manera incondicional y me impulsó a seguir adelante y a nunca rendirme.

Reconocimientos

Agradezco a Dios por haberme dado la sabiduría, fuerza y paciencia para poder concluir mis estudios de ingeniería en computación.

Agradezco a mis padres, quiénes han forjado lo mejor de ellos en mí, a través de su incondicional amor y apoyo en todo momento y me han impulsado a seguir adelante para nunca desistir a pesar de lo difícil de las adversidades. Los AMO.

A mi cousin Julio, quién me apoyó a lo largo de mi trayectoria universitaria y ha compartido conmigo sus conocimientos para poder ser mejor persona, estudiante y profesionista.

A mi amigo Humberto, por sus valiosos consejos de vida y su apoyo en los primeros semestres de la licenciatura.

A mi amigo José Alberto, por aportar ideas durante la realización de esta tesis, compartir conmigo un poco de su conocimiento, ser mi sensei, y por brindarme su apoyo para poder seguir desarrollándome como un profesionista altamente competitivo.

Agradezco a Vane, por haberme permitido ser parte de tu vida y ser una excelente compañera y amiga durante la carrera.

A mi director de tesis Ing. Aldo Jiménez Arteaga, por haber dedicado su tiempo para la realización y culminación de esta tesis.

A la honorable Facultad de Ingeniería de la Universidad Nacional Autónoma de México, por la excelente formación académica que se imparte en sus aulas.

A la Escuela Nacional Preparatoria 5, por la educación de calidad recibida y las gratas experiencias que viví durante mí estancia en ella.

México, Pumas, Universidad!

Resumen

El malware es definido como un software malicioso que tiene como propósito infiltrarse en las computadoras y servidores de diferentes organizaciones alrededor del mundo. Este tipo de software altera y modifica programas y/o archivos, y en algunos casos puede permitir que los equipos infectados sean controlados remotamente por piratas informáticos para obtener ventajas y beneficios de esta situación.

Hoy en día, el software malicioso, tiene su mayor auge desde su creación. Este puede llegar a ser tan sofisticado, que puede ocasionar daño en el mundo físico; tal y como es el caso de Stuxnet, un malware espía que reprograma sistemas industriales, afectando a instalaciones críticas como centrales nucleares.

Actualmente, en la Secretaría de Salud de la Ciudad de México (SEDESA) se detectó que algunos equipos de cómputo que componen su infraestructura informática, se encuentran infectados con software malicioso, poniendo en peligro la información que reside en ellos, atentando contra su confidencialidad, integridad y disponibilidad.

Por la situación descrita anteriormente, se propuso el desarrollo de este proyecto, el cual tiene como objetivo identificar, desde un sistema centralizado, los equipos que potencialmente pudieran estar infectados, alertando al administrador de red para que tome las medidas necesarias con el fin de contener y evitar la propagación hacia otros activos de información de dicha Secretaría.

Se decidió implementar una SandBox para el análisis automatizado de malware; debido a que estos entornos son seguros para la realización de pruebas en ambientes aislados, evitando la propagación de los archivos maliciosos ejecutados en ésta. Se optó por implementar específicamente Cuckoo SandBox, ya que de manera automática, realiza un análisis sobre una muestra, ya sea un archivo o una URL, arrojando información relevante acerca del comportamiento durante el análisis. Los resultados del análisis de la muestra son consultados a través de cualquier navegador web moderno.

Por otra parte, se empleó *Ubuntu 12.04 Precise Pangolin* como Sistema Operativo de la SandBox y del sistema centralizado encargado de detectar, a través de la red de

datos interna de la Secretaría de Salud, los equipos con comportamientos anómalos. Además de tener la ventaja de ser open source, Ubuntu se caracteriza por su robustez, estabilidad y rapidez para realizar las tareas que son ejecutadas en él. Este Sistema Operativo al ser un Linux, cuenta con el apoyo y soporte de miles de programadores a nivel mundial.

Índice general

Índice de figuras

Índice de tablas

Capítulo 1

Conceptos Generales

1.1. Redes de Datos

1.1.1. Definición

Una red de datos se define como un grupo de dispositivos, medios y servicios que trabajan en forma conjunta mediante reglas establecidas, para que exista un proceso de comunicación entre los diferentes dispositivos, con la finalidad de compartir información y recursos.

1.1.2. Importancia de las redes de datos en la actualidad

Uno de los elementos esenciales en la existencia del ser humano y que es tan importante como el respirar o comer, es la necesidad de poder interactuar y comunicarnos con los nuestros.

La forma en que nos comunicamos ha seguido un proceso evolutivo, el cual pasó de una comunicación individual y presencial, a una forma de comunicación de manera remota; en la que básicamente podemos comunicarnos sin importar la ubicación geográfica en la que nos encontremos.

Al igual que ha pasado con otros avances tecnológicos, la evolución de las tecnologías de la comunicación permitió que la naturaleza con la que interactuábamos socialmente cambiara y se adoptara a una escala global. Como ejemplo de esta evolución tecnológica, fue la creación e interconexión de redes de datos sólidas.

Las redes de datos en la actualidad son un elemento importante para poder llevar a cabo esta comunicación distante, ya que además de poder interactuar con otras personas, podemos compartir contenido multimedia, lo que implica flujos de video, texto y gráficos. Algunos ejemplos de las herramientas de comunicación más populares

comprenden la mensajería instantánea, los blogs, las wikis, los podcasts, radio online, herramientas de colaboración, video en demanda, entre otros.

Además de poder establecer una comunicación remota entre uno o varios individuos, otro uso tradicional de las redes de datos es la compartición de recursos, lo cual significa que la información, servidores, programas y medios de almacenamiento se encuentren disponibles para todos aquellos equipos de cómputo que estén conectados a una red de datos, sin importar la ubicación física del recurso y el usuario.

Las redes de datos también facilitan la manera en que aprendemos hoy en día. Los recursos multimedia de e-learning, a diferencia del método tradicional, no se limita a dos fuentes de conocimiento: el libro de texto y el instructor. Estos recursos pueden contener videos, voz y datos interactivos; que pueden ser consultados en cualquier momento sin importar la localización del educando. Es por ello que actualmente el acceso a una educación de calidad, ya no se limita a vivir cerca del lugar en donde se imparte la instrucción.

Las redes de datos han tenido un impacto positivo en nuestra sociedad. En el mundo actual, estamos conectados como nunca antes y sí alguien tiene una idea, puede compartirla con otras personas para materializarla y hacerla realidad. Las noticias y nuevos descubrimientos fluyen como nunca antes y en cuestión de segundos se dan a conocer por todo el mundo. La figura 1.1 ilustra este concepto.

Figura 1.1: Las redes de datos minimizan las fronteras geográficas. Fuente: Netacad CISCO.

1.1.3. Dispositivos de red

Los dispositivos de red se definen como un conjunto de hardware que proporciona conectividad, para garantizar que los datos fluyan a través de la red. Estos dispositivos pueden interconectar varias redes para formar una internetwork. Según su función se clasifican como acceso a la red (switches, hubs y puntos de acceso inalámbrico), internetworking (routers) y seguridad (firewalls).

Switch

El switch es utilizado para conectar varios dispositivos a través de la misma red de un edificio u oficina. Éste actúa como controlador y permite a los diferentes equipos comunicarse entre sí y compartir información.

Existen dos tipos de switches: los no administrables y los administrables. Los primeros no necesitan una configuración adicional y funcionan inmediatamente una vez se les instale dentro de la red, mientras que los switches administrables, pueden ser supervisados de manera local o remota para establecer una configuración, para poderlo adaptar a las necesidades de la organización. En la figura 1.2 se muestra un switch administrable.

Figura 1.2: Switch Cisco Serie 2600X de 48 puertos (Administrable). Fuente: cisco.com.

Hub

Los hubs son dispositivos que trabajan en la capa 1 (capa física) del modelo OSI y en la capa de acceso al medio en el modelo TCP/IP. Estos dispositivos concentran las conexiones, es decir que el grupo de nodos que se encuentren conectados a éste, la red los tratará como una sola unidad.

Estos dispositivos no pueden dirigir los datos para quién van destinados, por lo que los datos son enviados a todos los puertos, para que todos los dispositivos puedan recibirlos. Actualmente se encuentran de salida del mercado y de la implementación de las redes de datos actuales. La figura 1.3 muestra un hub de cuatro puertos.

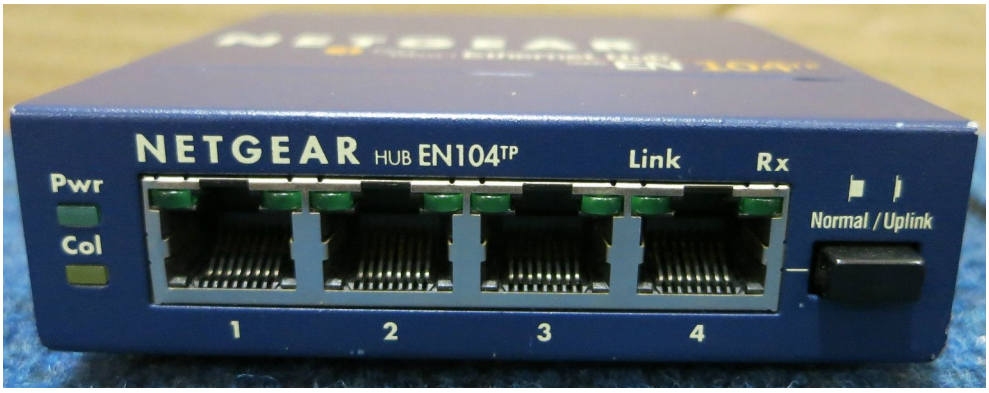


Figura 1.3: Hub de cuatro puertos Netgear EN104TP. Fuente: netgear.com.

Puntos de acceso inalámbrico

Un punto de acceso inalámbrico es un dispositivo que permite extender la conectividad de una red hacia dispositivos móviles de cómputo inalámbricamente (laptops, tabletas, celulares, etc).

Los AP, generalmente, se encuentran conectados de manera alámbrica a otros equipos de comunicación y permiten transmitir datos entre los dispositivos conectados a la red cableada y los dispositivos móviles. En la figura 1.4 se muestra un Access Point.

Figura 1.4: Access Point Linksys WAP54G. Fuente: linksys.com.

Router

Un router es un dispositivo que tiene como principal función establecer la conexión de muchas redes. Estos dispositivos analizan los datos que van a ser enviados o recibidos para poder seleccionar la mejor ruta de desplazamiento de la información, para su transmisión eficaz. La figura 1.5 muestra un router.

Figura 1.5: Router Cisco Serie 1900. Fuente: cisco.com.

Firewall

Estos dispositivos especializados cumplen una función importante en la seguridad de la red al examinar y filtrar la información recibida, según su dirección de origen y destino, protegiendo la red de posibles ataques. En la figura 1.6 se muestra un firewall.

Figura 1.6: Firewall Fortinet Fortigate – 100D. Fuente: fortinet.com.

1.1.4. Topologías de red

Una topología de red se define como la conexión entre un conjunto de nodos, así como la forma en que se encuentran conectados éstos; para que pueda existir una cadena de comunicación y pueda existir el proceso de transmisión de información.

1.1.4.1. Topologías físicas

Cuando se habla de la manera en que están configurados los nodos y la forma en que intercambian datos entre ellos mediante conexiones físicas, se dice que es una topología física de red. A continuación se explicarán las diferentes topologías físicas de red que existen.

Malla

La topología de malla presenta la característica de que cada nodo se encuentra conectado a todos los demás nodos que conforman la red. Al tener un enlace punto a punto y dedicado con los demás nodos que conforman la red, permite llevar los mensajes por diferentes caminos; eliminado la desventaja de los medios de transmisión compartidos, lo que evita que la comunicación y transmisión se vea afectada. Lo mismo pasa sí uno de los nodos llegase a fallar. Comparada con las demás topologías físicas de red, esta topología tiene la desventaja de requerir una mayor cantidad de cable, haciendo que el costo de implementación sea alto. La figura 1.7 describe esta topología.

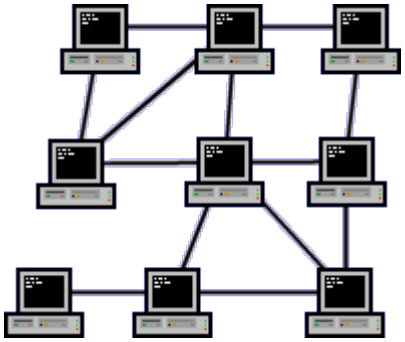


Figura 1.7: Topología tipo Malla. Fuente: wikipedia.com.

Estrella

En la topología de estrella, los nodos de una red se conectan a un nodo central, en la que el nodo central puede ser un hub o un switch. Una ventaja de esta topología es permitir que todos los nodos se comuniquen de manera conveniente. Si un nodo o el cable se dañan, no interrumpe el desempeño de la red. La implementación de esta topología es sencilla. La desventaja de esta topología es que sí el nodo central falla, toda la red quedaría sin comunicación. La figura 1.8 ilustra una topología de red tipo estrella.

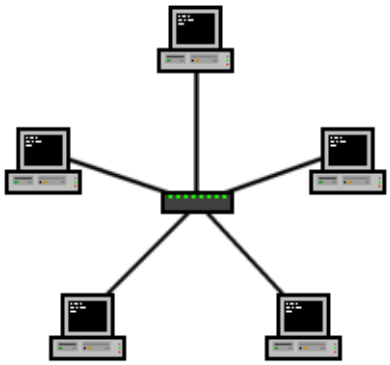


Figura 1.8: Topología tipo Estrella. Fuente: wikipedia.com.

Árbol

En una topología tipo árbol, los nodos se encuentran distribuidos mediante una configuración jerárquica, en la que el medio de transmisión es un cable ramificado. Pueden existir una o varias ramificaciones de cable que se conectan a un punto inicial llamado raíz o cabecera. A su vez, éstas pueden contener más ramificaciones, lo cual da lugar a una topología más compleja. Para poder implementar este tipo de topología, los dispositivos de red que se emplean son concentradores o switches.

Como principal desventaja de esta topología, es sí alguno de los dispositivos de red llegase a fallar, dejaría sin transmitir información hacia esa ramificación y todas las demás que se extienden debajo de ésta.

Esta topología tiene la ventaja de presentar una estructura con un orden jerárquico, lo cual proporciona facilidad para la resolución de problemas en caso de fallas. En la figura 1.9 se muestra una topología de red tipo árbol.

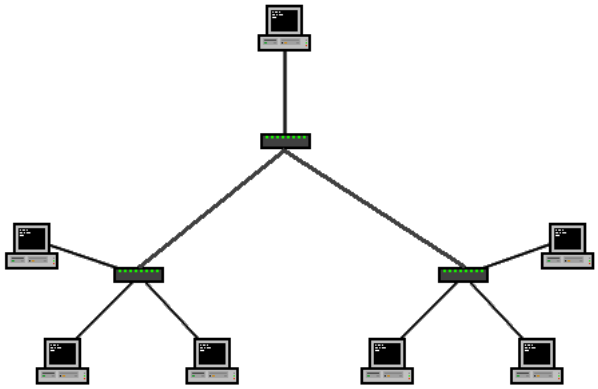


Figura 1.9: Topología tipo Árbol. Fuente: wikipedia.com.

Bus

En una topología tipo bus, todos los nodos se conectan directamente a un canal único digital, llamado bus o backbone, en una configuración multipunto. Básicamente cualquier nodo puede transmitir datos hacia otro nodo, propagándose por todo el medio. Esta topología, presenta tres principales desventajas. La primer desventaja es que al momento de transmitir información desde un nodo origen hacia un nodo destino, todos los nodos conectados al bus reciben dicha transmisión.

La segunda desventaja es sí dos nodos deciden transmitir al mismo tiempo, las señales que son enviadas por el medio colisionarán y serán erróneas. También se debe considerar el hecho de que un nodo esté continuamente transmitiendo información a través del medio.

Finalmente, sí falla el backbone, todos los nodos dejarán de transmitir y toda la red dejará de funcionar.

Para corregir estos problemas, los nodos envían información en pequeñas porciones de datos llamadas tramas. Cada una de estas tramas contiene el identificador del nodo de destino, para que así únicamente el nodo para quién está dirigida, reciba dicha trama y los demás nodos la ignoren. Al reducir el tamaño total de la información a transmitir en tramas, permite que por el mismo bus se intercalen estas tramas. A esto se le llama multiplexación.

Debido a que la arquitectura de esta topología es simple, las ventajas de ésta es una facilidad en la implementación y puesta en funcionamiento. En la figura 1.10 se muestra dicha topología.

Figura 1.10: Topología tipo Bus. Fuente: wikipedia.com.

Anillo

Esta topología se caracteriza por la forma de bucle cerrado que forman todos los nodos que se encuentren en ella. Los enlaces en esta topología son unidireccionales, lo que significa que el sentido de la transmisión puede realizarse en el sentido de las agujas del reloj o en el sentido contrario. La principal desventaja de esta topología es que sí un nodo llega a fallar, se rompe el anillo y la red se queda sin comunicación.

Existe una segunda versión de esta topología: anillo doble, la cual es similar a la topología de anillo, con la diferencia de que existe un segundo enlace redundante. En la topología de anillo doble, un anillo es utilizado como enlace principal, mientras que el otro se emplea de reserva. En otros casos, la configuración de esta topología es utilizada para que la transmisión de la información sea en ambas direcciones. La ventaja de esta variante es una mayor tolerancia a fallos, lo cual garantiza una mayor disponibilidad de la red. La figura 1.11 muestra una topología de red en anillo.

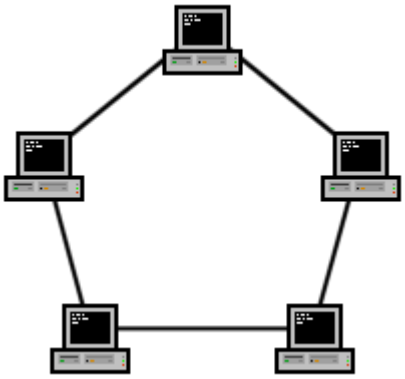


Figura 1.11: Topología tipo Anillo. Fuente: wikipedia.com.

Híbridas

Una topología híbrida es una combinación de las topologías descritas anteriormente. Un ejemplo de esta topología puede ser una compuesta por las siguientes: estrella-árbol, bus-estrella, etc. La implementación de ésta dependerá de las necesidades de cada una de las organizaciones. Esta topología permite cubrir el aumento en el número de nodos o dispositivos, debido a los componentes que la pueden conformar. Generalmente su costo es muy elevado, mientras que la administración es compleja. La figura 1.12 describe esta topología de red.

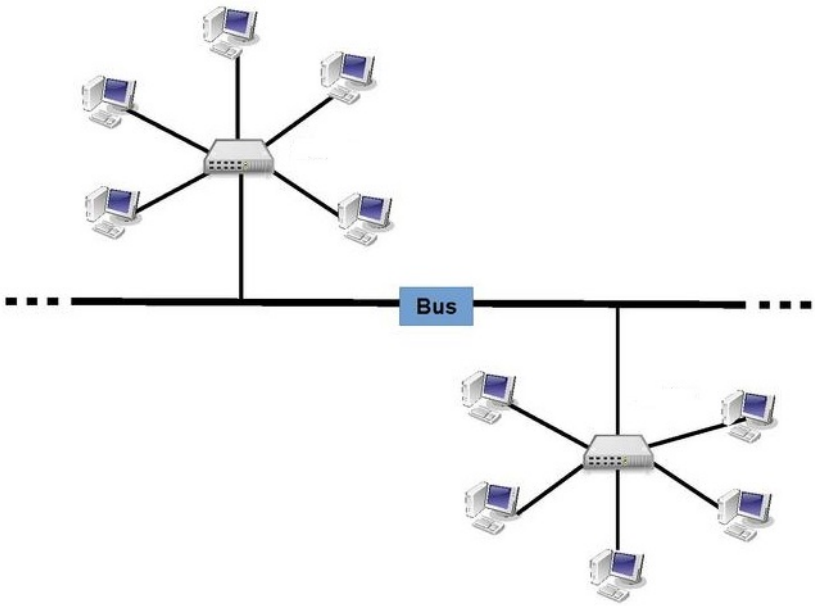


Figura 1.12: Topología tipo Híbrida (Estrella-Bus-Estrella). Fuente: google.com.

1.1.4.2. Topologías lógicas

Cuando se habla de la manera en que un nodo transmite una trama hacia al siguiente nodo, se trata de una topología lógica de red. Esta configuración emplea conexiones virtuales, independientemente de la distribución física y son los protocolos de la capa de enlace de datos que establecen estas conexiones virtuales. Las topologías más comunes son la broadcast y transmisión de tokens.

Broadcast
También conocida como bus, en esta topología cada host transmite sus datos hacia los demás host que están conectados en la red. No hay un orden, por lo que el acceso se otorga hacia el primer host que transmita sus datos al medio.

Tokens
En esta topología se controla el acceso a la red de los host mediante un token electrónico, el cual es transmitido secuencialmente a cada uno de ellos. Cuando un host recibe el token, significa que tiene la oportunidad de transmitir datos a través de la red. Sí dicho host no tiene ninguna información para enviar, transmite el token hacia el siguiente host, repitiéndose el proceso nuevamente. Los ejemplos más populares de la transmisión de tokens son Token Ring y la Interfaz de Datos Distribuida por Fibra (FDDI). En el caso de esta última, el token circula entre los equipos a velocidades muy altas.

1.1.5. Tipos de red

El tamaño e infraestructura de una red de datos varía en gran medida al número de usuarios que necesiten conectarse, el área geográfica cubierta y los diferentes servicios que se encuentren disponibles a través de ella. De acuerdo a lo anterior éstas se pueden clasificar como redes LAN, MAN y WAN.

Una red LAN o también llamada Red de Área Local, cubre un área geográfica única, proporcionando conectividad a estaciones de trabajo en una organización como puede ser una empresa, una biblioteca o una escuela de tamaño mediano. La extensión es limitada, ya que la conectividad entre una estación de trabajo y otra es de 100 metros. Dicha red es administrada por una única organización. La velocidad de transmisión en la que operan son de 10 Mbps, 100 Mbps y 1 Gbps. La figura 1.13 ilustra una red LAN.

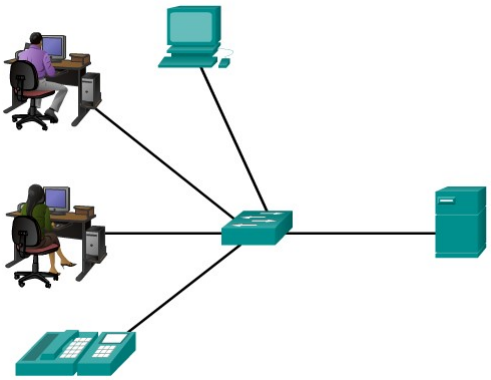


Figura 1.13: Red LAN. Fuente: Netacad CISCO.

1.1.5.1. MAN

Una red MAN o Red de Área Metropolitana recibe su nombre debido a que proporciona cobertura a un área geográfica extensa como por ejemplo una ciudad. Los medios de transmisión empleados en este tipo de redes pueden ser una combinación de fibra óptica y par trenzado. Las velocidades de transmisión a las que opera este tipo de redes sobre par trenzado son 10 Mbps, 20 Mbps, 45 Mbps y 75 Mbps; mientras que en fibra óptica las velocidades son de 100 Mbps, 1 Gbps y 10 Gbps. La figura 1.14 describe este tipo de red.

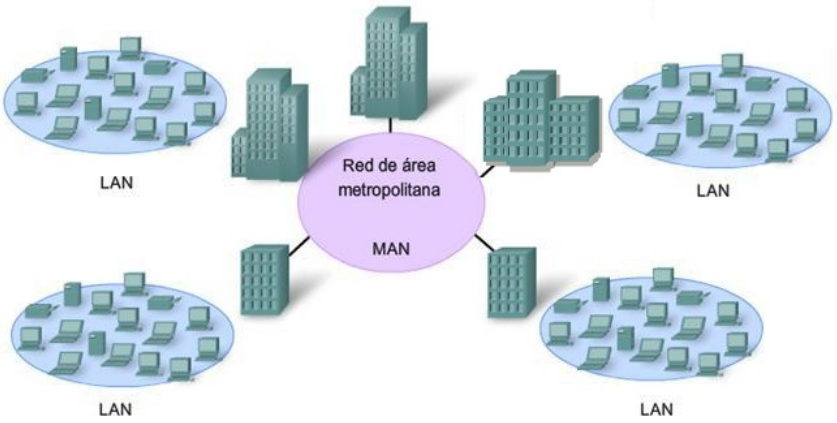


Figura 1.14: Red MAN. Fuente: Netacad CISCO.

1.1.5.2. WAN

Una red WAN o Red de Área Amplia permite interconectar varias redes LAN y MAN que se encuentran en distintas ubicaciones. La cobertura geográfica de este tipo de redes ofrece distancias desde los 100 Km hasta los 1000 Km; lo cual permite comunicar varios países e incluso hasta continentes enteros. Estas redes las utilizan organizaciones particulares y por los ISP para proporcionar servicios a sus clientes. La figura 1.15 ilustra una red WAN.

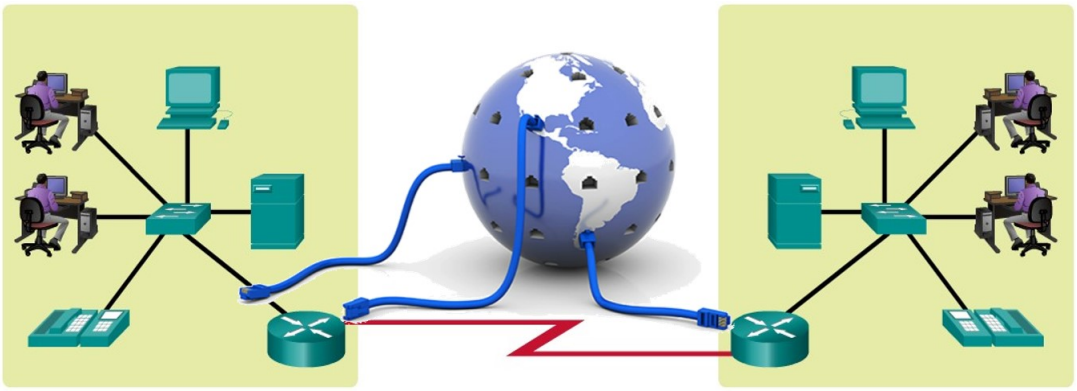


Figura 1.15: Red WAN. Fuente: Netacad CISCO.

1.1.5.3. Redes inalámbricas

Las redes inalámbricas proporcionan conectividad y comunicación sin el uso de una conexión física, a dispositivos portátiles de cómputo, tales como laptops, tabletas, celulares, televisores, entre otros dispositivos electrónicos. Este tipo de redes en particular, proporciona conectividad a usuarios que se desplazan dentro de un área geográfica en particular.

La conexión se establece a través de ondas electromagnéticas. De la misma manera en que las redes cableadas se clasifican de acuerdo al área geográfica que cubren, las redes inalámbricas se clasifican de la siguiente manera:

Wireless Personal Area Network (WPAN)
Las redes de área personal inalámbricas, tiene un alcance entre 1 a 10 metros. El uso de estas redes es para conectar dispositivos periféricos como impresoras, smartphones, ratones inalámbricos, sistemas de audio e inclusive conectar dos computadoras cercanas; sin emplear una conexión cableada. El estándar de red en una WPAN es el IEEE 802.15, que comúnmente es llamado Bluetooth. Ofrece una velocidad de hasta 55 Mbps (estándar IEEE 802.15.3 - 2003).

Wireless Local Area Network (WLAN)

Las redes de área local inalámbricas, ofrecen una cobertura al equivalente de una LAN en una compañía; y representan una opción para aquellas organizaciones en las que el cableado no es una solución. El estándar de una LAN inalámbrica es el IEEE 802.11, comúnmente llamada Wi-Fi. Éste estándar ofrece una velocidad máxima de 600 Mbps en varios cientos de metros (estándar IEEE 802.11n).

Wireless Metropolitan Area Network (WMAN)

Las redes de área metropolitana inalámbricas, utilizan tecnologías basadas en el estándar IEEE 802.16, conocidas comúnmente como WiMAX (Worldwide Interoperability for Microwave Access). WiMAX es similar a Wi-Fi, pero ofrece una mayor cobertura y ancho de banda. Esta tecnología emplea una topología punto a multipunto, para proporcionar acceso a servicios de banda ancha inalámbrica. La cobertura puede ser de hasta 50 km.

Wireless Wide Area Network (WWAN)

Las redes de área extendida inalámbricas emplean tecnologías de red celular como LTE, WiMAX, UMTS, CDMA2000, GSM, entre otras; para la transmisión de datos. Estas tecnologías son ofrecidas a nivel regional, nacional e incluso a nivel mundial, las cuales son proporcionadas por un proveedor de servicios inalámbricos.

A continuación, en la figura 1.16 se ilustran las diferentes tecnologías de redes inalámbricas.

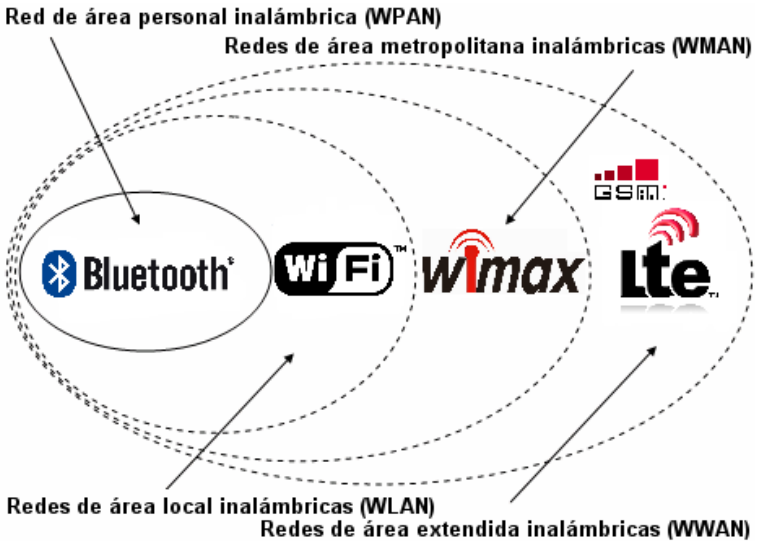


Figura 1.16: Tecnologías de las redes inalámbricas. Fuente: ccm.net.

1.1.6. Protocolos de red

Para que pueda existir una adecuada conectividad y se pueda llevar a cabo un proceso de comunicación exitoso entre los diferentes tipos de dispositivos que conforman la red, es necesario que existan reglas predeterminadas denominadas protocolos. Éstos se interrelacionan unos con otros y se encuentran implementados tanto en el software como en el hardware.

Los protocolos se encuentran organizados en una jerarquía de capas, en el cual cada capa interactúa con la capa superior o inferior, por lo que sí los datos se van a transmitir; cada una de las capas tienen la tarea de agregar una cabecera. Éstos, más la cabecera pasarán a la capa inferior hasta que los datos sean transmitidos por el medio físico. Sí los datos son recibidos, llegan por el medio físico por la capa inferior y subirán por la pila de capas, en la que cada una de éstas leerán las cabeceras correspondientes y al mismo tiempo realizará el proceso de desencapsulación de los datos, transmitiéndolos a la capa superior hasta llegar a la capa de destino. Es por ello que cada uno de los servicios de una capa de nivel superior, depende de la funcionalidad y servicios establecidos en las capas de niveles inferiores.

Actualmente el modelo de referencia más conocido es el OSI, el cual se emplea como referencia para los protocolos de red como el TCP/IP. Tanto el modelo OSI como el TCP/IP son los principales modelos empleados para el funcionamiento de una red. También los diseñadores de protocolos de red, pueden desarrollar sus propios modelos.

1.1.6.1. Modelo OSI

El llamado modelo OSI fue desarrollado por la ISO, la cual es la Organización de Estándares Internacionales, en el año de 1980. Este modelo propone una pila de 7 capas para la correcta comunicación en una red de datos y describe la interacción de cada capa con las que se encuentran por encima y debajo de éstas. A continuación se describirán cada una de las 7 capas empezando por la capa superior.

En la figura 1.17 se ilustra la pila de capas del modelo OSI.

Figura 1.17: Modelo OSI. Fuente: Netacad CISCO.

Capa de Aplicación

La capa 7 del modelo OSI, ofrece a las aplicaciones la manera de acceder a los servicios de las capas inferiores. También establece varios protocolos que son utilizados frecuentemente por los usuarios finales, para el intercambio de datos.

Capa de Presentación

La capa de presentación también es denominada capa 6 del modelo OSI. Dicha capa se encarga de la sintaxis y la semántica de la información transmitida, de tal manera que aunque distintos equipos tengan diferentes representaciones internas de caracteres, la información sea recibida y transmitida de manera legible. Así mismo, esta capa se encarga de comprimir y descomprimir la información, así como su cifrado y descifrado de los mismos.

Capa de Sesión

La capa 5 del modelo OSI, permite que ambos extremos de la comunicación establezcan una sesión, manteniendo y controlando el enlace establecido entre dos máquinas diferentes. Esta capa da seguimiento sobre quién puede transmitir, impide que ambos extremos traten de hacer una acción crítica al mismo tiempo y permite reanudar una sesión establecida en caso de una interrupción.

Capa de Transporte

La capa de transporte o capa 4 del modelo OSI, divide los datos recibidos de las capas superiores en unidades más pequeñas denominados segmentos sí es a través del protocolo TCP, o datagramas en el caso del protocolo UDP. Éstos son transmitidos a la capa de red y se asegura de que todas estas unidades lleguen correctamente al otro

extremo de la comunicación.

Capa de Red

La capa 3 del modelo OSI determina el camino a seguir de un paquete desde su origen hasta su destino, a pesar de que ambos no estén conectados directamente. A esta acción se le denomina enrutamiento y puede basarse en tablas definidas por un administrador de red (enrutamiento estático) o cuando los dispositivos de red llamados routers intercambian información de las tablas de rutas (enrutamiento dinámico). Los protocolos de enrutamiento más empleados son los siguientes:

Routing Information Protocol (RIP)

Es un protocolo de ruteo classful de gateway interior que emplea como métrica el conteo de saltos. Este protocolo determina cuál es la mejor ruta desde un origen hacia un destino mediante el empleo del algoritmo vector-distancia. Realiza el intercambio de información de su tabla de ruteo a través del protocolo UDP por el puerto 520.

Actualmente existen dos variantes del protocolo RIP original: RIPv2 y RIPng. El primero es una mejora del protocolo RIP ya que es un protocolo classless, lo que quiere decir que soporta VLSM y el segundo soporta IPv6.

Open Shortest Path First (OSPF)

OSPF es un protocolo de ruteo classless de estado-enlace desarrollado como reemplazo del protocolo RIP. Probablemente es el protocolo de gateway interior más empleado en redes de gran escala. La métrica de este protocolo denominada costo, se basa principalmente en el ancho de banda.

Emplea el algoritmo Dijkstra para determinar la mejor ruta entre un origen y destino de un sistema autónomo. Las ventajas de este protocolo es una rápida convergencia y una escalabilidad mucho mayor de implementaciones de grandes redes de datos, a diferencia de RIP.

Interior Gateway Routing Protocol (IGRP)

IGRP es un protocolo classful de gateway interior, desarrollado por Cisco Systems en respuesta a las limitaciones del protocolo RIP. Dicho protocolo emplea el algoritmo vector-distancia y a veces también considera el algoritmo estado-enlace, el cual determina la mejor ruta entre un origen y destino considerando el ancho de banda, el retardo, confiabilidad y carga del enlace. Actualmente este protocolo no es soportado por el sistema operativo de los dispositivos Cisco.

Enhanced Interior Gateway Routing Protocol (EIGRP)

Este protocolo de ruteo por vector-distancia avanzado, es una versión mejorada de IGRP. EIGRP es un protocolo classless de gateway interior. Utiliza el algoritmo DUAL (Difussing Update Algorithm), el cual garantiza una excepcional y rápida convergencia

de red. DUAL implementa características que no se encuentran en los protocolos de enrutamiento por vector-distancia, como es el envío de actualizaciones periódicas.

Capa de Enlace de Datos

La capa de enlace de datos es denominada también capa 2. Esta capa prepara los paquetes recibidos de las capas superiores para ser transmitidos, además de controlar el acceso a los medios físicos. Los paquetes dejan de llamarse así y se convierten en tramas. La trama de la capa de enlace de datos incluye lo siguiente:

- *Encabezado:* Está ubicado al inicio de la trama, y contiene información del inicio de la trama, direcciones de origen y destino de la trama así como control de flujo.
- *Datos:* El paquete de la capa de red.
- *Cola o tráiler:* Es información de control al final de la trama para la detección de errores e indicación del fin de la trama.

A su vez la capa de enlace de datos se subdivide en dos capas: control de enlace lógico (LLC) y control de acceso al medio (MAC).

Control de enlace lógico

El control de enlace lógico (LLC) coloca información en la trama para identificar el protocolo de la capa de red que está usando la trama. Esto permite que diferentes protocolos de la Capa 3 utilicen la misma interfaz de red y los mismos medios de transmisión.

Control de acceso al medio

El control de acceso al medio (MAC) proporciona información acerca del medio de transmisión que se emplea en la comunicación, así como la delimitación de datos de acuerdo con los requisitos de señalización física del medio.

Capa Física

La capa 1 ó capa física del modelo OSI, se encarga de transmitir en un ambiente físico las cadenas de bits, que van desde un origen hasta un destino, de acuerdo a las características mecánicas, físicas, eléctricas y funcionales del medio. Se consideran dos tipos de medios para la transmisión de datos: medios guiados y no guiados. Los medios guiados hacen uso del cable como por ejemplo cable coaxial, par trenzado y fibra óptica. Los medios no guiados o sin cable comprenden microondas, satélites, ondas de radio, etc.

1.1.6.2. Protocolo TCP/IP

TCP/IP es el modelo que se usó en la red experimental ARPANET, el cual hace posible la comunicación entre diferentes computadoras que utilizan diferente sistema operativo sobre una red LAN o en una red WAN. La familia de protocolos TCP/IP tiene una amplia suite de protocolos que actualmente son estándares de Internet.

Este modelo de protocolo consta de 4 capas como se muestra en la figura 1.18:

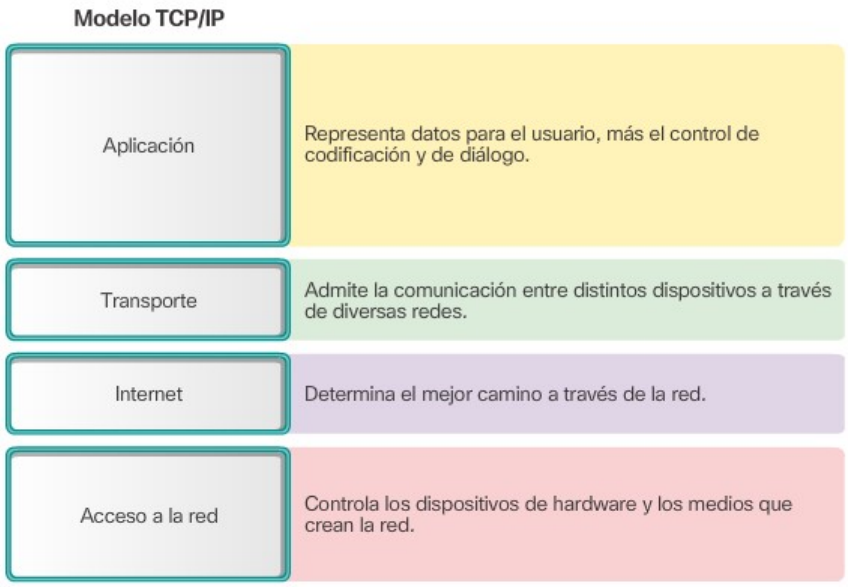


Figura 1.18: Modelo TCP/IP. Fuente: Netacad CISCO.

Capa de Aplicación

Al igual que la capa de aplicación es la capa superior en el modelo OSI, también lo es en el modelo TCP/IP. Ésta sirve de enlace entre los usuarios, las aplicaciones que son utilizadas para establecer una comunicación y la red subyacente en la cual son transmitidos los mensajes.

Un protocolo de la capa de aplicación es el HTTP (Hypertext Transfer Protocol), el cual permite la transferencia de información en la World Wide Web. Básicamente cuando un navegador web desea una página web, emplea este protocolo para hacer una solicitud al servidor del recurso solicitado. Otros protocolos de la capa de aplicación son el POP (Post Office Protocol) y SMTP (Simple Mail Transfer Protocol) para el correo electrónico o el DNS (Domain Name System) para la resolución de nombres de dominio.

Capa de Transporte

En la capa de transporte se encuentran las reglas o procedimientos que permiten garantizar una transmisión segura entre un origen y destino, sin importar la naturaleza de las aplicaciones que intercambian datos. Por otra parte, los protocolos más reconocidos para el transporte de datos se explican a continuación:

El protocolo TCP (Transmission Control Protocol) es orientado a la conexión lo que asegura que los datos que son transmitidos sean recibidos por el otro lado de la comunicación sin errores y en el mismo orden en el que fueron enviados, todo gracias a la negociación en tres pasos conocido como Three-way handshake. Dicha negociación consiste en que el destino envía un ACK (acuse de recibo) por cada paquete recibido al origen, y sí algún paquete está dañado o malformado se solicita al origen que se envíe nuevamente ese paquete.

El protocolo UDP (User Datagram Protocol) permite el envío de datagramas sin que exista una conexión previa ni acuse de recibido, por lo que no se garantiza la entrega de todos los paquetes al destino. Tampoco cuenta con control de flujo, por lo que los paquetes pueden llegar al destino en diferente orden al que fueron enviados desde el origen.

Capa de Internet

Al igual que la capa de Red del modelo OSI, esta capa se encarga de recibir y transferir paquetes por el mejor camino a través de la red, para que puedan llegar a su destino. Es probable que los paquetes hayan sido recibidos en distinto orden al que fueron enviados, lo cual las capas superiores se encargarán de ordenar.

El Protocolo IP (Internet Protocol) se incluye en esta capa. Se encuentra implementado tanto en los equipos finales (PC's, servidores, tablets, celulares, etc) como en los equipos intermediarios (routers). Dicho protocolo:

- Establece las convenciones de direcciones IP, ya que introduce las direcciones IPv4 y direcciones IPv6.
- Establece la ruta que debe seguir un paquete con base a la dirección IP del destinatario.
- Agrega un encabezado IP a los paquetes, adicional a la información de los protocolos TCP o UDP. El encabezado contiene las direcciones IP tanto de origen como de destino, la longitud del datagrama y el número de secuencia del datagrama.
- Fragmenta un paquete si es demasiado grande para su transmisión a través del medio de red. El protocolo IP del destinatario reconstruye los fragmentos, para formar el paquete original.

Capa de Acceso a la Red

Dicha capa se encarga del intercambio de datos entre el receptor y la red a la cual

se está conectando el emisor. También especifica las características del hardware de red que se utilizará para la red y las características físicas del medio de comunicaciones, para que los datos puedan ser transmitidos.

1.2. Seguridad

El concepto de seguridad es una característica de cualquier sistema ya sea informático o no, en el cual se busca protegerlo de sufrir algún peligro o daño, permitiendo garantizar su adecuada y normal operación.

Dentro del concepto de seguridad, existen dos definiciones fundamentales y que aunque pueden ser parecidos, en realidad no lo son: seguridad de la información y seguridad informática.

1.2.1. Tipos de seguridad

Seguridad de la Información
Considerando desde el enfoque de los sistemas computacionales, la información se define como: *"conjunto fundamental de datos organizados y procesados, los cuales son intercambiados por un individuo y un sistema informático, cambiando el estado de conocimiento de quién recibe estos datos"*.

Partiendo de la definición anterior, el concepto de seguridad de la información se define como: *"las acciones y mecanismos que son aplicados, a fin de prevenir cualquier acción que ponga en un estado endeble la información, garantizando su integridad, confidencialidad y disponibilidad"*.

Seguridad Informática
La seguridad informática tiene sus orígenes por allá en tiempos de la Segunda Guerra Mundial, cuando a los primeros mainframes desarrollados para hacer cálculos, se les protegía la confidencialidad e integridad de su información mediante candados, chapas, así como reconocimiento facial del personal autorizado que podía acceder. Conforme fue aumentando la necesidad de salvaguardar la información que contenían, se emplearon métodos más complejos y tecnológicamente más sofisticados.

La seguridad informática se define como *"la necesidad de proteger al hardware, software y la ubicación física de los sistemas informáticos de amenazas, a fin de garantizar la confidencialidad, integridad y disponibilidad de la información que reside en dichos sistemas"*.

Las técnicas de intrusión que son empleadas día a día, para comprometer a los sistemas actuales han sido más agresivas, lo que ocasiona un mayor impacto para quién las recibe. Un ejemplo de esto, es el gusano Stuxnet. Stuxnet se propagaba a través de sistemas de control industrial, para reprogramar estos sistemas y que los atacantes pudieran tomar control de éstos, sin que los operadores pudieran percatarse de las actividades ilícitas que eran realizadas.

Fue la primera amenaza que permitió pasar de un daño virtual a un daño físico, debido a que estaba programado para hacer dos ataques. El primero estaba dirigido a las instalaciones de enriquecimiento de uranio en Irán, en el cual era capaz de manipular la velocidad de las partes mecánicas del proceso de enriquecimiento, lo que ocasionaría que se agrietara el rotor principal, destruyendo el centrifugado, mientras que el segundo ataque contenido en Stuxnet, pretendía atacar la central eléctrica de Bushehr de Irán, en el cual destruiría las turbinas exteriores de la planta, imposibilitando el abastecimiento de energía eléctrica a ese país.

Es por ello que el presente proyecto tiene como objetivo la implementación de una SandBox, en la cual se analizarán los archivos que fueron detectados como maliciosos con ayuda de un IDS, el cual proporcionará dichas muestras.

Antes de poder proporcionar seguridad de la información, se debe contestar las siguientes interrogantes: ¿Qué queremos proteger? ¿Para qué se quiere proteger? ¿De qué o quiénes nos queremos proteger? ¿Cómo nos podemos proteger?, para tener un panorama mucho más amplio, y así saber cuáles son las necesidades de seguridad en una organización, así como los recursos que se deberán emplear para poderla proporcionar.

Seguridad de la red

La seguridad de la red se puede definir como la protección de los equipos y dispositivos de red que almacenan y transmiten información dentro y fuera de una organización.

En la actualidad, establecer medidas de seguridad en la red es un control básico, debido a que la infraestructura de red de la mayoría de las organizaciones, tiene puntos de contacto con la red pública. La red de una organización, al interactuar con las redes públicas, permite aumentar el número de intrusiones a los sistemas de información de la organización, ya que además tienen que lidiar con los ataques informáticos que se generan dentro de ésta.

En un ataque informático externo, el atacante busca la manera de poder infiltrarse en la red de una organización, a través de una debilidad o falla de seguridad. Por otro lado, en un ataque informático interno, el usuario malicioso no se preocupa sobre cómo acceder a la red de una organización porque ya se encuentra en ella, por lo cual le facilita realizar acciones que atente contra los sistemas y a la información que logre tener acceso.

Los ataques informáticos externos comprenden casi tres cuartas partes de los ataques totales en una organización, mientras que los ataques internos representan el porcentaje restante. Sin embargo, los ataques internos son los que tienen un mayor impacto negativo en los intereses de una organización, comparado con los ataques externos.

Seguridad física

La seguridad física se refiere a la implementación de barreras físicas y mecanismos de control (diseño, implementación y mantenimiento) que permitan proteger de amenazas, los recursos físicos de una organización, los cuales interactúan con la información en todos sus estados (transmisión, almacenamiento o procesamiento). La seguridad física analiza como principales amenazas las catástrofes naturales y artificiales, así como las amenazas humanas.

Una catástrofe natural y artificial es la amenaza que tiene una menor probabilidad de que pueda ocurrir, aunque dependiendo de la ubicación geográfica que se tome como referencia, puede aumentar la probabilidad de ocurrencia. El hecho de que este tipo de amenazas sean las menos probables de ocurrir, no implica que no se tomen medidas básicas, ya que sí no se hicieran, tendrían un impacto negativo mucho mayor para la organización en cuestión.

Como ejemplo de una amenaza humana, se puede mencionar la interrupción del suministro eléctrico, un acceso no autorizado o no monitoreado, el uso de dispositivos electrónicos no autorizados, derrame accidental de líquido sobre los equipos de cómputo, entre otros.

De acuerdo a la norma *ISO 27002:2013*, el dominio de seguridad física y ambiental establece una serie de objetivos de control y actividades de control para la implementación de seguridad física en una organización, los cuales se describen a continuación:

Áreas seguras

Los activos que procesen información crítica o confidencial deben ser ubicados en áreas seguras, protegidos por los perímetros de seguridad definidos, con las adecuadas barreras de seguridad y controles apropiados. Este objetivo de control previene un acceso físico no autorizado, daño e interferencia en las instalaciones y en la información de la organización y establece los siguientes controles:

Perímetro de seguridad físico. Los perímetros de seguridad físico deben ser definidos y usados para proteger áreas que contengan información crítica o sensible, así como las instalaciones de procesamiento de información.

Controles de acceso físico. Las áreas seguras deben ser protegidas por controles de entrada apropiados para asegurar que sólo personal autorizado cuenta con permiso de acceso.

Seguridad de oficinas, pisos e instalaciones. Se debe diseñar y aplicar un sistema de seguridad física para oficinas, pisos e instalaciones de la organización.

Protección contra las amenazas externas y ambientales. Control en el que se analiza

la manera de cómo prevenir daño por fuego, inundación, terremoto, explosión, disturbios civiles y otros desastres naturales o que pueden ser provocados por el hombre.

Trabajo en áreas seguras. Como área segura se define el lugar físico en el que se encuentra la información crítica de una organización. Se deben establecer pautas para permitir el trabajo en un área segura.

Áreas de entrega y carga. Los puntos de entrega y carga de una organización deben ser identificados y separados de los activos de información, a fin de evitar accesos no autorizados a la organización.

Seguridad de los equipos

Los equipos deben ser protegidos contra amenazas físicas y ambientales, para prevenir pérdida, daño, robo o compromiso de los activos, así como interrupciones en las operaciones de la organización. Este objetivo de control proporciona las siguientes actividades:

Ubicación y protección del equipamiento. Los equipos deben ser ubicados y protegidos para reducir los riesgos de amenazas ambientales y peligros, así como de accesos no autorizados.

Instalaciones de suministros. El equipo debe ser protegido de fallas de suministro eléctrico y otras interrupciones causadas por fallas en la instalación de suministros.

Seguridad del cableado. Tiene como objetivo proteger contra la intercepción, interferencia o posibles daños al cableado eléctrico y de telecomunicaciones, que transportan información y/o apoyan a los servicios de información.

Mantenimiento del equipo. Se le debe proporcionar un adecuado mantenimiento al equipo de una organización, para garantizar la continua disponibilidad e integridad de la información con la que interactúan.

Eliminación de los activos. Tiene como objetivo prevenir la eliminación de los activos por lo que éstos, la información y el software no deben ser retirados de su sitio sin previa autorización.

Seguridad de los equipos y activos fuera de las instalaciones. Se deben aplicar controles de seguridad a los activos que se encuentran fuera de la organización, pero que contienen información de la misma, tomando en cuenta el riesgo que representa que ésta pueda ser expuesta.

Seguridad en la reutilización o eliminación del equipo. Establece que todos los equipos y activos que contienen medios de almacenamiento, deben ser verificados para

asegurar y garantizar que ninguna información sensible y software licenciado ha sido eliminado o sobrescrito de manera segura antes de su eliminación o reutilización.

Equipo de usuario desatendido. Los usuarios se deben asegurar que los equipos que no son supervisados, cuentan con la protección apropiada.

Política de escritorio limpio y bloqueo de pantalla. El personal de la organización debe adoptar una política, en la cual se establezcan controles para que los lugares de trabajo se encuentren despejados de cualquier tipo de documentación en papel, así como de cualquier medio de almacenamiento extraíble.

1.2.2. Servicios de Seguridad

Un servicio de seguridad se define como la acción que coadyuva a proteger el flujo de información entre los diferentes sistemas de información de una organización, empleando modelos, métodos o mecanismos, evitando un ataque informático.

Confidencialidad
La confidencialidad se define como la protección de la divulgación o exposición de la información a usuarios o sistemas no autorizados. Cuando ocurre lo contrario se dice que la confidencialidad es incumplida. Para garantizar la confidencialidad de la información se establecen mecanismos como clasificación de la información, cifrado de la información, entre otros.

En la actualidad, muchas organizaciones esperan que la información interna crítica que manejan y la que intercambian con otras se mantenga confidencial, debido al valor de mercado de la mayoría de éstas; por lo que confían que la información de la organización se mantenga a salvo de individuos no autorizados para su conocimiento, pudiendo tener un impacto negativo en los intereses financieros de éstas.

Algunos ejemplos que atentan contra la confidencialidad de la información son el malware, intentos de intrusión, ingeniería social, redes sin el nivel de seguridad adecuado y sistemas pobremente administrados.

Integridad
El servicio de integridad permite asegurar que la información no sufra una modificación o alteración en el contenido de ésta, sin una debida autorización. Así mismo, durante la transmisión de información también debe existir integridad, por lo cual en la secuencia de datos que son enviados por un emisor deben ser recibidos tal cual fueron enviados por un receptor.

La integridad de la información es amenazada cuando la información es expuesta a un daño, como puede ser la destrucción o la corrupción de datos.

Existen mecanismos para validar la integridad de un archivo, el cual es llamado valor hash. Este elemento es un indicador que permite saber sí un archivo ha sufrido alguna modificación desde su origen, hasta el momento en que se encuentra en el destino.

Disponibilidad

La disponibilidad permite a los usuarios o sistemas autorizados acceder a la información sin ninguna interferencia u obstrucción. Sí una información es requerida o solicitada y ésta no es desplegada en un corto lapso de tiempo, se considera que no hay disponibilidad de la misma.

Los ataques que atentan contra la disponibilidad de información son conocidos como denegación de servicio (DoS), los cuales consisten en enviar un gran número de conexiones o grandes solicitudes de información hacia un sistema objetivo, por lo que llega a sobrecargarse y no responde a solicitudes legítimas de información. También existe una variante del ataque DoS llamado denegación de servicio distribuido (DDoS), en el cual un conjunto de conexiones o peticiones se coordinan desde diferentes ubicaciones y son enviadas al mismo tiempo hacia un objetivo.

Autenticación

El servicio de autenticación consiste en validar y comprobar la identidad de una entidad quién dice ser, mediante alguno de los siguientes tres métodos de autenticación:

¿Qué sabes?: Tiene como objetivo, validar y comprobar algo que únicamente debe saber un individuo o un sistema autorizado, comparándolo con la información que cuenta en la base de datos al sistema que se desea acceder. Este método hace uso de contraseñas, NIPs, códigos secretos, etc. Sin embargo, este método no es considerado como fuerte y no es adecuado para sistemas que requieren un alto nivel de seguridad.

¿Qué tienes?: Consiste en emplear llaves para el desbloqueo de una barrera física o virtual, para poder tener acceso al espacio o sistema deseado. Un ejemplo de desbloqueo de una barrera física, es el uso de una llave que permite abrir un candado o puertas. Por otro lado, el uso de un token o una tarjeta magnética permite desbloquear una barrera virtual y por lo tanto, permite tener acceso al sistema o recurso solicitado. El riesgo de este método, es que las llaves se pueden perder, dañar o ser robadas.

¿Qué eres?: Este método se refiere a la autenticación biométrica, en la cual usa las características del ser humano, como son las huellas digitales, la voz o la retina de los ojos; las cuales sirven para identificar a un individuo de otro. La desventaja de este método es que requiere de una previa instalación, configuración y mantenimiento.

Cuando son empleados adecuadamente y en conjunto los métodos descritos anteriormente, éstos contribuyen a una adecuada robustez en la forma de autenticación.

Control de Acceso

El control de acceso es un método que determina cómo un individuo o sistema es admitido para interactuar con la información. Los controles de acceso son dependientes del servicio de autenticación, ya que en primera instancia se tiene que determinar si al individuo o sistema se le otorga permiso al recurso solicitado. Este servicio cuenta con tres modelos principales de control de acceso lógico, los cuales se explican a continuación:

Control de Acceso Discrecional (DAC): Es el modelo de control de acceso más ampliamente usado. En este, el dueño de la información impone un conjunto de restricciones sobre los individuos que deseen tener acceso a la información que él posee, mediante lo que ellos pueden hacer: leer, escribir, borrar, renombrar, mover, etc.

Control de Acceso Obligatorio (MAC): Se emplean esquemas de niveles de clasificación de información (por ejemplo pública, restringido, confidencial, secreta y ultra secreta), para tomar una decisión de acuerdo a un conjunto de políticas de seguridad de la información y por los dueños de la información, las restricciones en los controles de acceso.

Control de Acceso Basado en Roles (RBAC): Los permisos y derechos hacia los recursos de la información se establecen de acuerdo a los roles o perfiles, en lugar de usuarios individuales. Este modelo permite una administración más flexible, así como el cumplimiento de los controles de acceso en organizaciones con un gran número de usuarios.

No repudio

El no repudio es un servicio de seguridad que actúa tanto en el emisor como en el receptor de la información, en el cual tanto el emisor como el receptor no pueden refutar que la información ha sido transmitida. Los tipos de no repudio que a continuación se mencionan, se encuentran definidos en el estándar internacional *ISO 14516 "Guía para el uso y administración de proveedores de servicio de confianza electrónicos"*, en el que pueden ser demostrados a una tercera entidad:

- *Aprobación:* Ofrece pruebas de quién es responsable de la aprobación del contenido del mensaje.
- *Envío:* Ofrece pruebas de quién fue el que envió el mensaje.
- *Origen:* Ofrece pruebas del origen del mensaje. Es una combinación de aprobación y envío.
- *Sumisión:* Ofrece pruebas de que un agente de entrega ha aceptado el mensaje para su transmisión.
- *Transporte:* Ofrece pruebas de cualquier intento de negar que los datos no se transmitieron.

- *Recepción:* Ofrece pruebas de cualquier intento de negar por parte del receptor que la información no fue recibida, con lo cual se brinda protección al emisor.
- *Conocimiento:* Proporciona pruebas de que el receptor conoce el contenido del mensaje transmitido.
- *Entrega:* Proporciona pruebas de que el receptor recibió el mensaje y también conoce el contenido del mensaje. Es una combinación de recepción y conocimiento.

1.2.3. Amenaza y vulnerabilidad

Vulnerabilidad

Vulnerabilidad, en el ámbito de seguridad informática, se define como una debilidad en el diseño o implementación de algún sistema informático, que puede ser utilizado por un atacante, violando la seguridad del mismo, para ocasionar algún daño sí el ataque es intencionado. Existen vulnerabilidades que son muy reconocidas en los sistemas informáticos, para los cuales se cuenta con la solución, ya sea una actualización de versión o un parche de seguridad. Las vulnerabilidades que más afectan son las 0-day, ya que no existe una solución conocida, pero sí se conoce la forma de explotarla.

Amenaza

Una amenaza es todo elemento, acción o evento capaz de atentar contra la seguridad de la información, causando una alteración total y/o parcial a la información de la organización, generando un impacto negativo de tipo material, económico, informativo o prestigio de esta. Las amenazas se clasifican en dos grupos:

Amenazas intencionales: Son aquellas que deliberadamente pueden ocasionar un daño hacia una organización (robo de información, divulgación de información, propagación de malware, suplantación de identidad, entre otros).

Amenazas no intencionales: Las amenazas no intencionales, se originan debido a la falta de medidas, omisión o incumplimiento de acciones, que ponen en riesgo el desempeño de los activos de información, impactando de manera negativa en la organización. Como ejemplo de amenazas no intencionales, se tienen los fenómenos naturales, caídas en los medios de comunicación, discontinuidad en el suministro eléctrico, entre otros.

1.2.4. Análisis de riesgos

El análisis de riesgos es un proceso en el cual se logran identificar las amenazas y vulnerabilidades a las que se encuentran expuestos los activos de alguna organización, la probabilidad de ocurrencia y además se estima el impacto que supondría en caso de que se llegaran a concretar. Una vez que es obtenida esta información, se establecen

políticas o procedimientos para combatirlos.

Es importante mencionar, que ningún sistema o infraestructura se encuentra completamente seguro, por lo que un riesgo puede ser mitigado, aceptado o transferido.

Los siguientes conceptos proporcionan un panorama general en la comprensión del análisis de riesgos:

Activo: Un activo es aquello que tiene un valor para una organización, que compone el proceso de comunicación y que por ende necesita de protección; para que no atenten contra su confidencialidad, integridad y disponibilidad. Éstos pueden ser hardware, software, información personal, medios de almacenamiento, servidores, bases de datos, infraestructura.

Riesgo: Un riesgo es una probabilidad de que ocurra un evento y se presente algún daño o pérdida para los activos de la organización. Básicamente es la posibilidad de que se concrete una amenaza.

Riesgo residual: Cuando las vulnerabilidades han sido controladas en la medida de lo posible, aún sigue existiendo un riesgo que no ha sido completamente eliminado, transferido o previsto. Este resto es lo que se conoce como riesgo residual.

El manejo del riesgo residual debe ser juzgado de acuerdo a la zona de confort de la organización, ya que el objetivo de la seguridad de la información no es llevar el riesgo residual a cero, sino manejarlo de acuerdo a los intereses de la organización.

Aceptación: La estrategia de aceptar un riesgo es la decisión que ha tomado una organización, en la que no hay nada por hacer para proteger una vulnerabilidad y se acepta el resultado de su explotación. Generalmente esta decisión se toma basada en una conclusión, en la que el costo de proteger un activo no justifica los gastos de seguridad. Por otro lado, se encuentran los legacy-systems, los cuales son sistemas anticuados, pero se continúan empleando por los usuarios; y no es posible aplicar actualizaciones o parches de seguridad de una manera sencilla; por lo que también se decide emplear la estrategia de aceptación.

Transferencia: Esta estrategia, transfiere el riesgo a otros activos, procesos u otras organizaciones. Esto se puede lograr cuando son reconsiderados cómo se ofrecen los servicios, revisando los modelos de despliegue del Cloud Computing, el outsourcing a otras organizaciones, adquiriendo seguros o implementando contratos de servicios con proveedores.

Mitigación: La estrategia de mitigación de un riesgo, intenta reducir o eliminar el impacto para una organización, causado por la explotación de una vulnerabilidad a

través de un proceso de planeación y preparación.

Análisis del riesgo: Permite establecer una clasificación o puntuación de un riesgo para los activos de información. Es aquí en donde se evalúa que determinados eventos no deseados pueden ocurrir, así como el impacto de las consecuencias en caso de que sucedieran.

Administración de riesgos: Este proceso es usado para identificar, controlar y minimizar el impacto de eventos inciertos, por lo cual la principal función de la administración de riesgos es reducir los mismos, hasta que alcancen un nivel aceptable para la organización.

Impacto: El impacto establece que tan malo puede ser para una organización que se concrete la actividad de una amenaza. El impacto se categoriza como se muestra a continuación:

- *Cumplimiento:* Que la amenaza afecte con el cumplimiento de requerimientos normativos establecidos en la organización.
- *Operativo:* Que la amenaza pueda afectar la operación de los sistemas de información en su confidencialidad, integridad y/o disponibilidad.
- *Imagen:* Que la amenaza afecte la imagen de la organización ante sus clientes, de tal manera que estos decidan no usar los servicios que ofrece la organización afectada.
- *Financiero:* Que la amenaza impacte de manera negativa en la salud financiera o flujo financiero de la organización.

1.2.5. Trazabilidad

En muchas ocasiones, cuando las organizaciones tienen un incidente informático, éstas quieren obtener respuestas como: ¿Cuándo ocurrió? ¿Quién lo hizo? ¿Por qué lo hizo? La trazabilidad permite realizar un seguimiento de las acciones realizadas en el tiempo, entre un origen y destino, por lo que se puede avanzar hacia una investigación que permitan responder las preguntas planteadas anteriormente.

La manera en que se puede establecer trazabilidad es mediante evidencias. Estas evidencias son los registros de actividades, también llamados logs.

Logs

La seguridad no sólo radica en la implementación de controles y medidas de prevención, sino también en controles de identificación, en el caso de que un sistema haya sido comprometido por un intruso.

Las aplicaciones y sistemas registran eventos y acciones de un determinado periodo de tiempo, en un conjunto de archivos llamados logs. Generalmente las actividades que más se almacenan en este conjunto de archivos, es el registro de los accesos hacia algún sistema, aunque también permiten registrar información sobre quién, qué, cuándo, dónde y por qué de determinado evento.

Para que los logs puedan considerarse confiables, su integridad debe ser asegurada de una manera razonable; por lo que sí alguien puede escribir y/o borrar eventos de los logs, se puede decir que no son lo suficientemente confiables como mecanismo de identificación de actividades anómalas y maliciosas.

Como recomendaciones de las mejores prácticas, los logs deben ser protegidos contra la manipulación, accesos no autorizados, deben ser cifrados y revisados periódicamente en busca de identificar alguna actividad sospechosa.

1.2.6. Sistemas de Detección de Intrusos

Para poder entender el concepto de un Sistema de Detección de Intrusos, primero se debe tener presente la definición de intrusión, la cual se define como el acto de entrar en algún lugar sin invitación, autorización, derecho o bienvenida.

Un Sistema de Detección de Intrusos es una tecnología que funciona de la misma manera en que lo hace una alarma antirrobo; el cual se configura para monitorear actividades sospechosas o accesos no autorizados en un host o a una red. Según la forma en que reaccionan se les puede clasificar de la siguiente manera:

NIDS
Un Sistema de Detección de Intrusos basado en Red, monitorea el tráfico de toda una red. Normalmente, la tarjeta de red de una computadora funciona en modo no promiscuo, lo que significa que en este modo de operación, sólo los paquetes que van destinados para la dirección MAC de la tarjeta de red del host se analizarán. Sin embargo, para el adecuado y correcto funcionamiento del NIDS, su tarjeta de red debe operar en modo promiscuo, para que pueda capturar y analizar los paquetes que no van dirigidos a su dirección MAC.

Así, en este modo, el NIDS puede vigilar sigilosamente todas las comunicaciones del segmento de red. El modo promiscuo es necesario para protección de la red. En la figura 1.19 se observa la implantación de 3 NIDS en segmentos estratégicos de red, lo que permite monitorear el tráfico de red para todos los dispositivos de los segmentos.

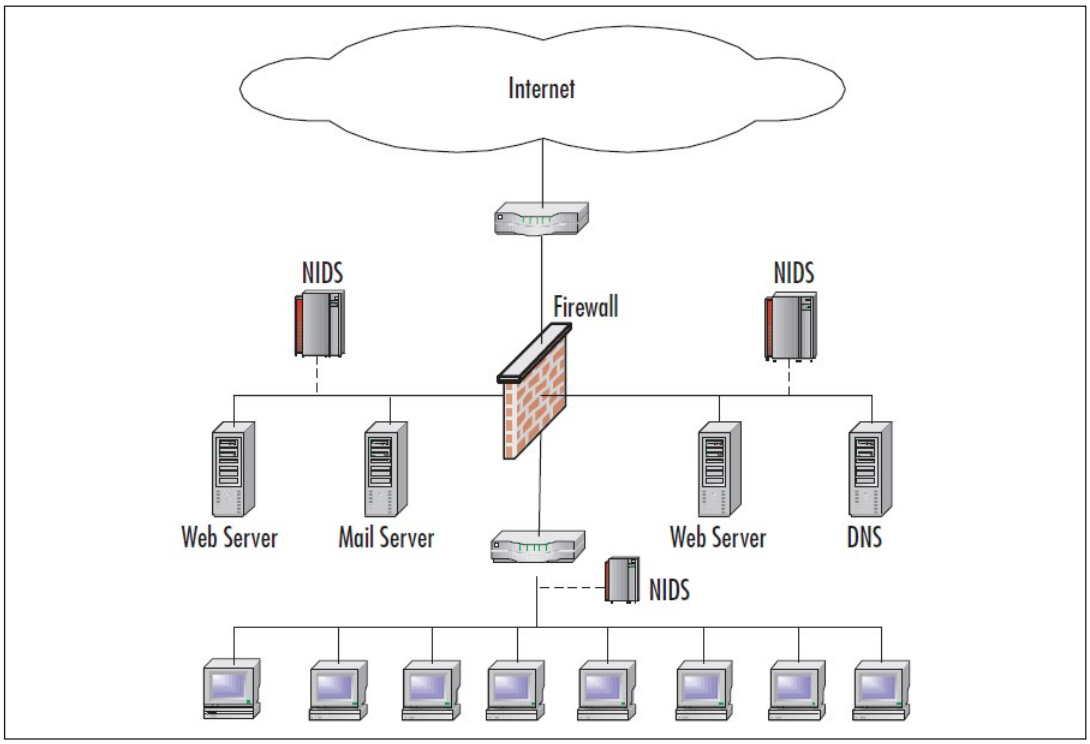


Figura 1.19: Red usando 3 NIDS implementados en segmentos de red estratégicos.

Fuente: *Snort IDS and IPS Toolkit, 2007.*

HIDS

Un Sistema de Detección de Intrusos basado en Host, a diferencia del NIDS; sólo protege el equipo de cómputo en el que reside y su tarjeta de red no opera en modo promiscuo. Una ventaja del HIDS, a diferencia del NIDS, es que el conjunto de reglas que son empleadas, se pueden adaptar de acuerdo a las necesidades específicas de la organización. La reducción en el número de las reglas permite mejorar el desempeño y reducir una sobrecarga del procesador.

En la figura 1.20 se ilustra la implantación de diferentes Sistemas de Detección de Intrusos basados en Host, en el segmento de servidores, así como en los equipos finales.

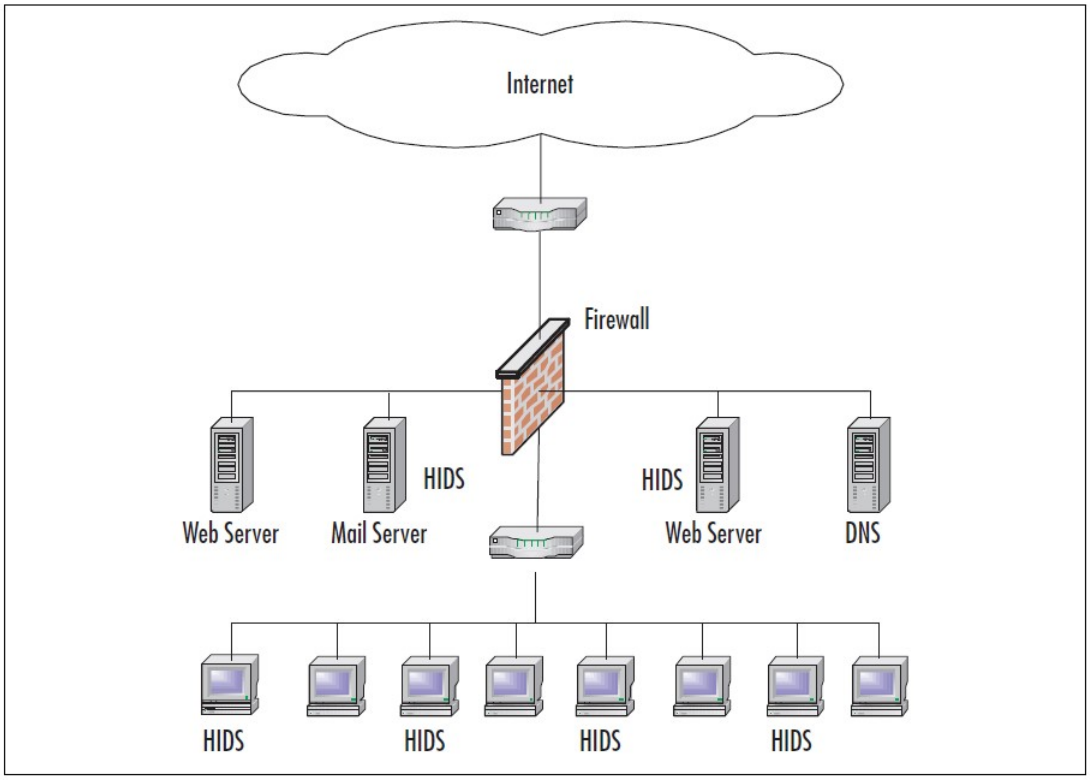


Figura 1.20: Red usando HIDS en servidores y computadoras específicas. Fuente: *Snort IDS and IPS Toolkit, 2007.*

DIDS

En un Sistema de Detección de Intrusos Distribuido, son colocados sensores de detección y reportan a una estación de administración centralizada, en una arquitectura administración/sondeo. Los logs de ataques son continuamente enviados a la estación centralizada y pueden ser almacenados en una base de datos central. Otra ventaja, es que la actualización de reglas para la detección de ataques o tráfico malicioso son descargadas desde el equipo de administración centralizado, para posteriormente ser enviadas a los sensores.

En la figura 1.21, se ilustra la implantación de un DIDS.

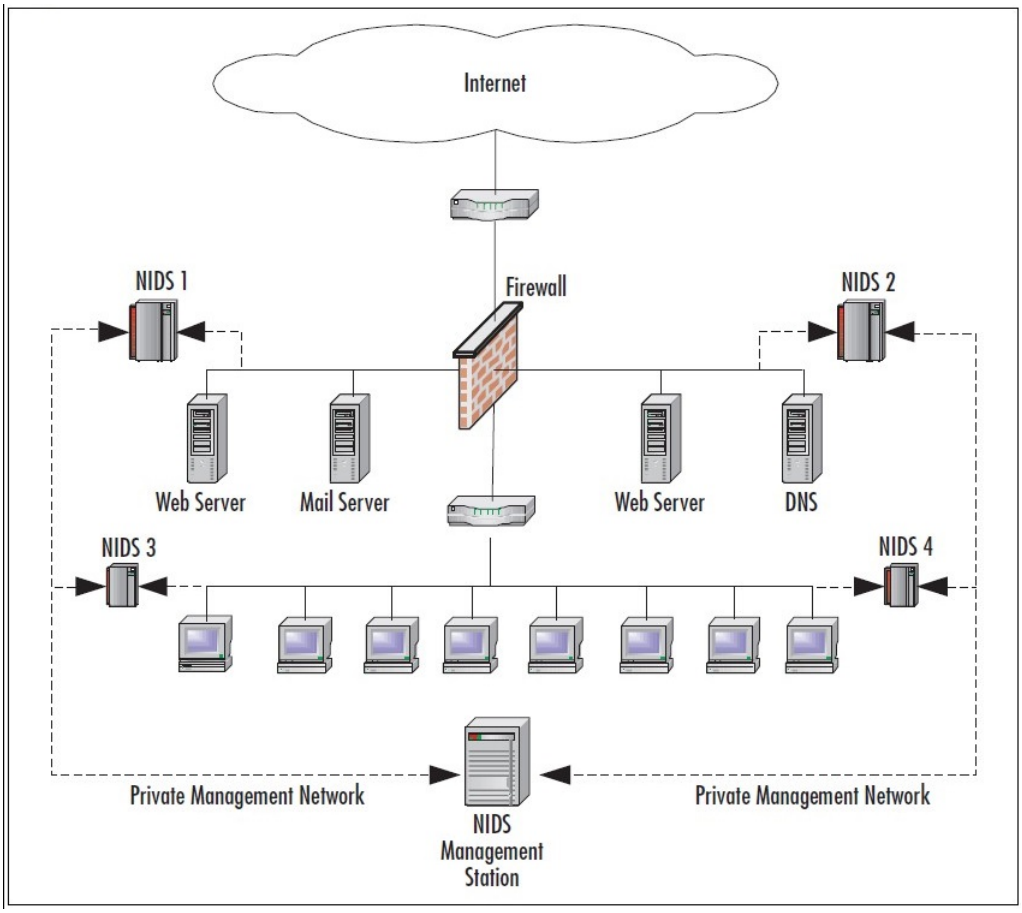


Figura 1.21: Red monitoreada por 4 sensores y una estación de administración centralizada. Fuente: *Snort IDS and IPS Toolkit, 2007.*

IPS

La prevención de intrusos es el siguiente paso en la evolución de la detección de intrusos. Un Sistema de Prevención de Intrusos, a diferencia de los tres sistemas descritos anteriormente, que son pasivos, éstos son reactivos.

Básicamente el concepto de prevención de intrusos es tomar la información reunida durante la detección de intrusiones y actuar en consecuencia a través de un proceso automatizado. Mientras un IDS está diseñado para hacer conciencia de ataques potenciales, el IPS tiene un mayor alcance ya que trabaja activamente para prevenir las intrusiones.

Un IPS actúa de manera similar a un Firewall, pero ofrece muchas ventajas. Puede ser configurado como un IDS, para responder a los ataques de acuerdo a la última versión de firmas de reglas vigentes. Por otro lado, actúa más inteligentemente que un

Firewall, ya que no sólo bloquea la comunicación de una dirección IP específica o un puerto en específico, sino que tiene la habilidad de rechazar los paquetes que contienen un ataque y permitir el tráfico normal.

IDS Snort

Snort es un sistema de detección y prevención de intrusos multiplataforma de código abierto, el cual observa el tráfico de una red privada y detecta anomalías, el cual puede dar indicios sobre algún tipo de comportamiento inusual causado por software malicioso en la intranet. Para ello, emplea una serie de reglas, que contienen patrones, expresiones regulares y coincidencias de malware conocido. La figura 1.22 muestra el imagotipo de Snort.

Figura 1.22: Imagotipo del IDS Snort. Fuente: snort.org.

1.2.7. SandBox

Una SandBox es un mecanismo de seguridad separada de un ambiente de producción, para aislar la ejecución de programas que generalmente están infectados o contienen código malicioso, ejecución de código no probado, usuarios y sitios web que no son de confianza.

Una SandBox provee un conjunto de recursos controlados para que las pruebas se ejecuten en los sistemas invitados, como espacio reservado en disco y memoria, acceso a la red o la habilidad para que puedan inspeccionar el sistema en el que se están ejecutando.

Las implementaciones actuales de una SandBox incluyen las siguientes características:

- Restricciones del acceso a la red.
- Emulación de una computadora a través de una máquina virtual.

- Reglas de ejecución que permite a los administradores tener control total sobre los procesos que se inician o se están ejecutando.

Cuckoo SandBox

Cuckoo es un sistema de análisis de malware automatizado de código abierto. Se emplea para ejecutar y analizar archivos, para después obtener resultados completos del análisis que describen las acciones que realizó el software malicioso, mientras son ejecutadas en un ambiente seguro y aislado. La siguiente información es la que arroja el análisis automatizado de Cuckoo SandBox.

- Rastros de las llamadas realizadas por todos los procesos generados por el programa malicioso.
- Los archivos que son creados, eliminados y descargados por el malware durante su ejecución.
- Volcados de memoria de los procesos del malware.
- Rastros del tráfico de red en formato PCAP.
- Capturas de pantallas tomadas durante la ejecución del malware.
- Volcados de memoria completos de las máquinas.

Cuckoo SandBox está diseñado para ser usado como una aplicación independiente, así como para ser integrado en grandes frameworks, gracias a su diseño modular. Puede ser usado para analizar:

- Archivos ejecutables de Windows
- Archivos DLL
- Documentos PDF
- Documentos de Microsoft Office
- URLs y archivos HTML
- Scripts de PHP
- Archivos CPL
- Scripts de Visual Basic
- Archivos ZIP
- Archivos JAR de Java
- Scripts de Python

Cuckoo SandBox comenzó como un proyecto de Google Summer of Code en 2010, dentro del Proyecto Honeynet. Fue diseñado y desarrollado por Claudio Guarnieri, quién es el desarrollador principal y coordina todos los esfuerzos de los desarrolladores y contribuyentes.

Después del trabajo inicial durante verano de 2010, la primera versión beta se publicó el 5 de febrero de 2011, cuando Cuckoo fue anunciado públicamente y fue distribuido por primera vez.

En marzo de 2011 Cuckoo fue seleccionado de nuevo como un proyecto apoyado durante el Google Summer of Code de 2011 con el Proyecto Honeynet, durante el cual se unió Darío Fernández y amplió su funcionamiento.

El 2 de noviembre de 2011 se hace el lanzamiento de la primera versión estable de Cuckoo, la versión 0.2.

En diciembre de 2011 se liberó Cuckoo v0.3 y a principios de febrero de 2012 surge la versión 0.3.2.

A finales de enero de 2012 Malwr.com es abierto, la cual es una instancia libre y pública que corre sobre Cuckoo Sandbox, provista de una interfaz a través de la cual los usuarios pueden enviar sus archivos para ser analizados y obtener resultados de vuelta.

En marzo de 2012 Cuckoo Sandbox gana la primera ronda en el Magnificent7 organizado por Rapid7.

El 24 de julio de 2012 la versión 0.4 de Cuckoo Sandbox es liberada.

El 20 de diciembre de 2012 la versión 0.5 “To The End Of The World” se libera.

El 15 de abril de 2013 la versión 0.6 es lanzada, poco tiempo después de haber sido lanzada la segunda versión de Malwr.com.

El 9 de enero de 2014 la versión 1.0 de Cuckoo es liberada.

En marzo de 2014 nace la fundación Cuckoo, como una organización sin fines de lucro, dedicada al crecimiento de Cuckoo Sandbox y a proyectos e iniciativas que existan a su alrededor.

El 7 de abril de 2014 la versión 1.1 de Cuckoo Sandbox es liberada. El 7 de octubre de este mismo año, la versión 1.1.1 de Cuckoo SandBox es publicada y liberada.

El 5 de marzo de 2015 se publica la versión 1.2 de Cuckoo SandBox.

Durante el verano de 2015 Cuckoo SandBox comenzó el desarrollo de análisis de malware de Mac OS X como un proyecto de Google Summer of Code como parte del Proyecto Honeynet.

El 21 de enero de 2016, se libera la versión 2.0 RC1, la cual hasta el momento es la última versión estable de Cuckoo SandBox.

La figura 1.23 ilustra el imagotipo de Cuckoo SandBox.

Figura 1.23: Imagotipo de Cuckoo SandBox. Fuente: cuckoosandbox.org.

Capítulo 2

Implementación del Sistema de Detección de Intrusos y Cuckoo SandBox

2.1. Instalación del Sistema de Detección de Intrusos (IDS) y Cuckoo SandBox

La instalación de un IDS, permite a un administrador de red detectar accesos no autorizados y anomalías en el tráfico de la infraestructura de red y activos de una organización.

Una vez que se detectan estas anomalías, el Sistema de Detección Intrusos genera alertas que informan al administrador de la red de dicho comportamiento inusual. Para la realización de esta tesis, el tráfico de red que disparó esas alertas, fue capturado y guardado en un registro (log), para su análisis.

Adicionalmente la SandBox Cuckoo analizó de forma automática, las muestras de malware recabadas a partir del tráfico de red.

A continuación se establece el diagrama de bloques para el proceso de instalación y configuración tanto del IDS Snort como de Cuckoo SandBox. Como se aprecia a continuación en la figura 2.1, el diagrama de bloques para la instalación y configuración del IDS Snort consta de 10 bloques, en el cual cada uno de ellos especifica el proceso de instalación, así como la configuración de paquetes y software complementario, para el óptimo funcionamiento del IDS Snort en la intranet. Esto se explica en la sección 2.1.1.

Figura 2.1: Diagrama de bloques instalación y configuración IDS Snort. Fuente: Elaboración propia.

Por otro lado, en la figura 2.2, se detalla el diagrama de bloques para indicar la configuración que se realizó para la instalación de Cuckoo SandBox. Se compone básicamente por 5 procesos, los cuales son necesarios seguir, para que pueda realizarse de manera eficaz y segura el análisis de muestras maliciosas que hayan sido detectadas por el Sistema de Detección de Intrusos.

El proceso de instalación tanto del software complementario, así como de la propia SandBox puede consultarse a partir de la sección 2.2.1.

Figura 2.2: Diagrama de bloques instalación y configuración Cuckoo SandBox. Fuente: Elaboración propia.

2.1.1. Instalación del Sistema de Detección de Intrusos (IDS)

El Sistema de Detección de Intrusos se instaló en un Sistema Operativo Linux, concretamente sobre *Ubuntu* versión *12.04 Precise Pangolin*. La función principal de este Sistema de Detección de Intrusos es analizar los logs y capturas de datos que fueron generados por las alertas emitidas por dicho sistema.

Para realizar la instalación, primero se agregó una clave GPG. Para ello, se contó con los privilegios de administrador y se creó un directorio para concentrar los archivos descargados con la herramienta wget:

```
# mkdir /usr/InstalacionSnortCuckoo
# cd /usr/InstalacionSnortCuckoo
# wget http://www.dotdeb.org/dotdeb.gpg
```

Se agregó la clave GPG con el siguiente comando:

```
# cat dotdeb.gpg | apt-key add -
```

Snort depende de algunos paquetes adicionales para su correcta instalación, configuración y funcionamiento. Los paquetes que se instalaron fueron:

- apache2
- libapache2-mod-php5
- libwww-perl
- mysql-server
- mysql-common
- mysql-client
- php5-mysql
- libnet1
- libnet1-dev
- libpcre3
- libpcre3-dev
- autoconf
- libcrypt-ssleay-perl
- libmysqlclient-dev
- php5-gd
- php-pear
- libphp-adodb
- php5-cli
- libtool
- libssl-dev
- gcc-4.4
- g++
- automake
- gcc
- make
- flex
- bison
- apache2-doc
- ca-certificates

Para la descarga e instalación de los paquetes necesarios se empleó el gestor de paquetes *apt* y la herramienta *wget*.

```
# apt-get update && apt-get install apache2 libapache2-mod-php5 libwww-
    perl mysql-server mysql-common mysql-client php5-mysql libnet1 libnet1
    -dev libpcre3 libpcre3-dev autoconf libcrypt-ssleay-perl
    libmysqlclient-dev php5-gd php-pear libphp-adodb php5-cli libtool
    libssl-dev gcc-4.4 g++ automake gcc make flex bison apache2-doc ca-
    certificates
```

Durante el proceso de instalación de MySQL se ingresó un password para el usuario root, como se muestra a continuación en la figura 2.3:

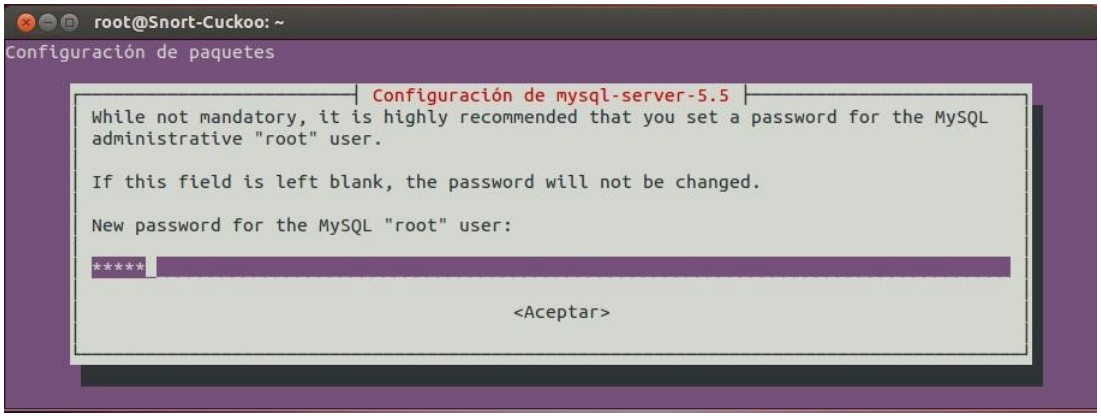


Figura 2.3: Configuración de contraseña para el usuario root de MySQL. Fuente: Captura propia.

Cabe mencionar, si ya se dispone de un servidor Apache configurado con MySQL y PHP, los paquetes correspondientes a dicho software pueden ser omitidos.

Adicionalmente, se instalaron otros paquetes de manera manual que no se encuentran en los repositorios de Ubuntu. Concretamente *daq, libpcap y libdnet.*

A continuación se descargó el paquete *libpcap* en el directorio creado anteriormente. La versión de *libpcap* que se descargó fue la *1.6.1*:

```
# cd /usr/InstalacionSnortCuckoo
# wget http://www.tcpdump.org/release/libpcap-1.6.1.tar.gz
```

Terminada la descarga del archivo, se descomprime y desempaqueta. Hecho lo anterior, se cambió la ubicación del directorio actual al directorio *libpcap-1.6.1.*

```
# tar -zxvf libpcap-1.6.1.tar.gz
# cd libpcap-1.6.1
```

Dentro del directorio *libpcap-1.6.1* se ejecutó el siguiente comando:

```
# ./configure --prefix=/usr --enable-shared
```

El argumento *–prefix=/usr* indica que los archivos de registro (logs) y los directorios de bases de datos se ubicaron en */usr* y no en el directorio */usr/local/var*. El segundo argumento, *–enable-shared*, generó una biblioteca compartida.

Posteriormente se compiló e instaló el paquete en nuestro sistema.

```
# make
# make install
```

El siguiente paquete que se instaló fue *daq*. De la misma forma que *libpcap*, se descargó el archivo dentro del directorio creado para este fin.

```
# cd /usr/InstalacionSnortCuckoo
# wget https://www.snort.org/downloads/snort/daq-2.0.2.tar.gz
```

Se descomprimió y desempaquetó el archivo *daq-2.0.2.tar.gz*. Realizado lo anterior, se cambió la ubicación al directorio creado *(daq-2.0.2)*.

```
# tar -zxvf daq-2.0.2.tar.gz
# cd daq-2.0.2
```

Se ejecutó el siguiente comando.

```
# ./configure
```

Este comando comprueba las características del sistema que afectan a la compilación y a la vez, configuró la compilación según estos valores para crear el archivo makefile.

Se compiló e instaló el paquete en nuestro sistema.

```
# make
# make install
```

Por otro lado, se actualizó el directorio de librerías compartidas de la siguiente manera:

```
# echo >> /etc/ld.so.conf /usr/lib
# echo >> /etc/ld.so.conf /usr/local/lib && ldconfig
```

Por último, se instaló el paquete *libdnet*. La descarga del archivo comprimido se realizó en el directorio *InstalacionSnortCuckoo*.

```
# cd /usr/InstalacionSnortCuckoo
# wget http://libdnet.googlecode.com/files/libdnet-1.12.tgz
```

Éste se descomprimió, desempaquetó y se cambió la ubicación al directorio *libdnet-1.12*.

```
# tar -zxvf libdnet-1.12.tgz
# cd libdnet-1.12
```

Una vez dentro del directorio *libdnet-1.12*, se ejecutó el siguiente comando:

```
# ./configure --prefix=/usr --enable-shared
```

Se compiló e instaló el paquete.

```
# make
# make install
```

A continuación, se instaló la herramienta TCPTrace. TCPTrace es una herramienta escrita por Shawn Ostermann, la cual permite realizar un análisis más preciso de los paquetes generados por Snort con formato tcpdump. Entre la información que podemos obtener con TCPTrace, se encuentran: bytes, segmentos enviados y recibidos, tiempos de vida, retransmisiones, entre otros. Puede realizar gráficos para su posterior análisis.

Esta herramienta se instaló para habilitar la automatización del proceso de análisis de una muestra.

Se agregó el siguiente repositorio en el archivo *sources.list*, el cual se encuentra en */etc/apt*:

```
deb http://us.archive.ubuntu.com/ubuntu precise main universe
```

Los repositorios se actualizaron y mediante el gestor de paquetes *apt*, se instaló TCPTrace.

```
# apt-get update
# apt-get install tcptrace
```

2.1.2. Instalación de Snort

Posterior a la instalación de los paquetes descritos anteriormente, se realizó la instalación de Snort en su versión 2.9.6.2, siguiendo los pasos que se explican a continuación:

Se descargó el archivo *snort-2.9.6.2.tar.gz* y el archivo de configuración de Snort, en el directorio creado anteriormente:

```
# cd /usr/InstalacionSnortCuckoo
# wget http://labs.snort.org/snort/2962/snort.conf
# wget http://www.snort.org/downloads/snort/snort-2.9.6.2.tar.gz
```

Hecho el proceso anterior, se descomprimió el archivo *snort-2.9.6.2.tar.gz* y se cambió la ubicación al directorio *snort-2.9.6.2* como se muestra a continuación:

```
# tar -zxvf snort-2.9.6.2.tar.gz
# cd snort-2.9.6.2
```

Una vez dentro del directorio, se comprobaron las características del sistema, previas a la compilación:

```
# ./configure --enable-sourcefire
```

Hecho lo anterior, se compiló e instaló Snort en su versión 2.9.6.2.

```
# make
# make install
```

A continuación se configuró el servicio para que pueda ser ejecutado fácilmente, para esto se emplearon los siguientes comandos desde consola.

El primer paso que se realizó, fue crear los siguientes directorios y archivos.

```
# mkdir /etc/snort /etc/snort/rules /var/log/snort /var/log/barnyard2 /usr
    /local/lib/snort_dynamicrules
```

```
# touch /etc/snort/rules/white_list.rules /etc/snort/rules/black_list.
    rules
```

Se agregó un nuevo grupo llamado *snort* y un usuario llamado *snort*. Este usuario se agregó al grupo homónimo.

```
# groupadd snort
# useradd -g snort snort
```

Se modificó el usuario y el grupo propietarios (snort, para ambos casos) de los directorios */var/log/snort* y */var/log/barnyard2*.

```
# chown snort:snort /var/log/snort /var/log/barnyard2
```

Una vez realizado el proceso anterior, los archivos con extensión *.conf*, *.config* y *.map* se copiaron al directorio */etc/snort*, así como el archivo *snort.conf*, mediante los siguientes comandos.

```
# cp /usr/InstalacionSnortCuckoo/snort-2.9.6.2/etc/*.conf* /etc/snort
# cp /usr/InstalacionSnortCuckoo/snort-2.9.6.2/etc/*.map* /etc/snort
# cp /usr/InstalacionSnortCuckoo/snort.conf /etc/snort
```

2.1.3. Configuración de Snort

Para la configuración de Snort, es necesario realizar algunos cambios en el archivo *snort.conf*.

A continuación se editaron las variables de configuración de Snort. En la línea 45 se ingresó el segmento de red que deseamos que Snort esté observando el tráfico. Para este caso, la red de análisis emplea un segmento 172.16.0.0/12. En la línea 48, se declaró como red externa, todas las direcciones IP que no se encuentren en el segmento de monitoreo. Las variables se establecieron de la siguiente manera:

```
Linea 45: ipvar HOME_NET 172.16.0.0/12
Linea 48: ipvar EXTERNAL_NET !$HOME_NET
```

Hecho lo anterior, se modificó la línea 104. En esta línea se especificó el directorio donde se encuentran ubicados los archivos de las reglas que Snort empleó, para emitir alertas. Además, se establece la ruta de las reglas de objetos compartidos (línea 105), el set de reglas del preprocesador (línea 106), así como las reglas de lista blanca y negra (líneas 109 y 110 respectivamente).

```
Linea 104: var RULE_PATH ./rules
Linea 105: var SO_RULE_PATH ./so_rules
Linea 106: var PREPROC_RULE_PATH ./preproc_rules
Linea 109: var WHITE_LIST_PATH ./rules
Linea 110: var BLACK_LIST_PATH ./rules
```

De la línea 261 a la 265, se omitieron todas las entradas correspondientes a las variables preprocessor normalize_*.

En la línea 293 después de "decompress_depth 65535" se agregó la siguiente expresión:

```
Linea 293: max_gzip_mem 104857600
```

En la línea 517 se declaró el nombre con el que se almacenaron los logs de Snort, así como el tamaño máximo de cada uno de estos archivos. En este caso, se estableció en 128 Mb. El argumento *output unified2*, se refiere un formato binario optimizado, para que las alertas puedan ser insertadas de forma eficiente y rápida en los logs.

```
Linea 517: output unified2: filename snort.log, limit 128
```

En la línea 528 se editó el módulo *log_tcpdump*, que registra los paquetes generados por las alertas de Snort a un archivo con formato tcpdump. Esto es de suma importancia, debido a que los archivos con formato tcpdump se emplearon para realizar un análisis automatizado post-proceso con el tráfico capturado, que se explica en el capítulo 4.

```
Linea 528: output log_tcpdump: /var/log/log_tcpdump/tcpdump.log
```

Se creó el directorio *log_tcpdump* en el directorio */var/log*. Adicionalmente, se modificó el usuario y el grupo propietarios del directorio.

```
# mkdir /var/log/log_tcpdump
# chown -R snort:snort /var/log/log_tcpdump
```

A partir de la línea 543 se incluyen los ficheros de reglas para Snort. Para este caso, en concreto, se comentaron todos los conjuntos de reglas que no sean de utilidad para la realización de este proyecto. Las reglas que se incluyeron fueron las siguientes:

```
include $RULE_PATH/local.rules
include $RULE_PATH/blacklist.rules
include $RULE_PATH/dos.rules
include $RULE_PATH/exploit-kit.rules
include $RULE_PATH/indicator-compromise.rules
include $RULE_PATH/indicator-obfuscation.rules
include $RULE_PATH/indicator-scan.rules
include $RULE_PATH/indicator-shellcode.rules
include $RULE_PATH/malware-backdoor.rules
include $RULE_PATH/malware-cnc.rules
include $RULE_PATH/malware-other.rules
include $RULE_PATH/malware-tools.rules
include $RULE_PATH/netbios.rules
include $RULE_PATH/os-windows.rules
include $RULE_PATH/server-apache.rules
```

Para verificar el correcto funcionamiento del IDS Snort, fue necesario crear una regla de prueba. Para esta verificación, se creó un archivo llamado *local.rules* de la siguiente manera:

```
# cd /etc/snort/rules
# touch local.rules
```

En este fichero se puede escribir un conjunto de reglas que no se encuentren en los repositorios de las reglas de Snort y que sean realizadas para la captura de tráfico en específico. Para comprobar el correcto funcionamiento del IDS se agregó la siguiente alerta:

```
alert icmp any any -> $HOME_NET any (msg:"Prueba ICMP"; sid: 10000001; rev
    :1;)
```

Se ejecutó Snort en modo consola, y se comprobó que funcionara correctamente. Para ello se realiza un ping desde otro equipo hacia el IDS:

```
# snort -A console -q -u snort -g snort -c /etc/snort/snort.conf -i eth1
```

Lo anterior generó el despliegue de una alerta en tiempo real en consola; como se puede apreciar en la figura 2.4:

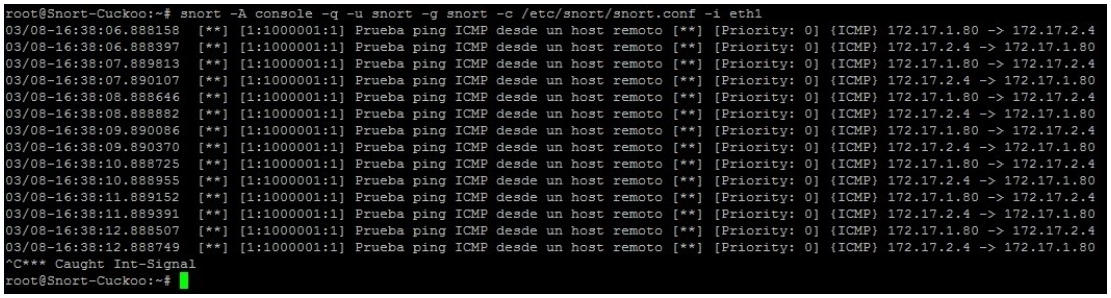


Figura 2.4: Ejecución de Snort en modo consola. Fuente: Captura propia.

Nota: Para terminar con el proceso de Snort, basta con presionar ctrl + c.

2.1.4. Instalación de Barnyard

Para instalar Barnyard, se descargó el archivo *master.tar.gz* en el directorio *InstalacionSnortCuckoo*.

```
# wget https://github.com/firnsy/barnyard2/archive/master.tar.gz
```

Se descomprimió el archivo y se ingresó al directorio creado:

```
# tar -zxvf master.tar.gz
# cd barnyard2-master
```

Fueron actualizados los scripts de configuración:

```
# autoreconf -fvi -I ./m4
```

Se verificaron las características del sistema antes de compilar e instalar.

```
# ./configure --with-mysql --with-mysql-libraries=/usr/lib/i386-linux-gnu
```

Finalmente se compiló e instaló *barnyard* en el sistema.

```
# make
# make install
```

Se movió el archivo *barnyard2.conf* al directorio */etc/snort*.

```
# mv /usr/local/etc/barnyard2.conf /etc/snort
```

Se editó el archivo *barnyard2.conf* para el correcto funcionamiento de Barnyard.

```
La linea 227 fue sustituida:
output alert_fast: stdout

por:
output alert_fast
```

Adicionalmente, al final del archivo, se declaró la siguiente línea. Esta instrucción permite almacenar los eventos generados por Snort en una base de datos, en este caso en MySQL:

```
output database: log, mysql, user=snort password=<Password usuario snort>
    dbname=snort host=localhost
```

2.1.5. Configuración de MySQL

En esta sección se explica el proceso de configuración de la base de datos que usa Snort, para registrar las alertas, y que la aplicación B.A.S.E. utiliza para aprovechar un mayor rendimiento de Snort.

Se ingresó a MySQL con la contraseña del usuario root que se configuró anteriormente:

```
# mysql -u root -p
Enter password: <Password usuario root de MySQL>
```

Una vez que se ingresó como usuario root al manejador de bases de datos MySQL, se creó la base de datos snort. También se debió establecer una contraseña para el usuario snort en MySQL. Por otro lado, fueron proporcionados privilegios sobre la base de datos para el usuario snort creado en Linux.

```
mysql> create database snort;
mysql>  grant CREATE,INSERT,SELECT,DELETE,UPDATE on snort.* to
    snort@localhost;
mysql> SET PASSWORD FOR snort@localhost=PASSWORD('password_usuario_snort')
    ;
```

Una vez que se configuró la base de datos, se crearon las tablas, mediante las

cuales, Snort administra las alertas. Para la creación de dichas tablas se usó el esquema existente en Barnyard2. El nombre de la base de datos es "snort".

```
mysql> source /usr/InstalacionSnortCuckoo/barnyard2-master/schemas/
    create_mysql
```

Se comprobó la creación de las tablas. Para ello, se ingresó a MySQL como usuario root seleccionando la base de datos snort y se realizó un despliegue de las tablas que contiene dicha base de datos.

```
mysql> use snort;
mysql> show tables;
```

En la figura 2.5 se pueden observar las tablas que contiene la base de datos snort.

```
mysql> show tables;
+------------------+
| Tables_in_snort  |
+------------------+
| data             |
| detail           |
| encoding         |
| event            |
| icmphdr          |
| iphdr            |
| opt              |
| reference        |
| reference_system |
| schema           |
| sensor           |
| sig_class        |
| sig_reference    |
| signature        |
| tcphdr           |
| udphdr           |
+------------------+
16 rows in set (0.00 sec)
```

Figura 2.5: Tablas que contiene la base de datos snort. Fuente: Captura propia.

2.1.6. Configuración del servidor apache

En esta sección se explicará cómo configurar el servidor Apache con PHP, para la visualización de las alertas vía web.

Para ello, en el archivo */etc/php5/apache2/php.ini* se editaron los siguientes parámetros:

Se modificó la variable error_reporting (línea 521) con el siguiente valor: error_reporting = E_ALL & E_NOTICE.

Se copió el archivo *default-ssl* en el directorio *sites-enabled*. Y se activó el módulo *ssl*.

```
# cp /etc/apache2/sites-available/default-ssl /etc/apache2/sites-enabled
# a2enmod ssl
```

Adicionalmente, se realizaron algunas configuraciones e instalaron algunos paquetes mediante el framework Pear. Esto garantiza el correcto despliegue de datos numéricos, gráficos y datos en la aplicación B.A.S.E. Se reinició el servidor apache para activar la nueva configuración.

```
# pear config-set preferred_state alpha
# pear channel-update pear.php.net
# pear install -alldeps Image_Color2 Image_Canvas Image_Graph
# /etc/init.d/apache2 restart
```

2.1.7. Instalación de B.A.S.E.

En este apartado se explicará cómo instalar B.A.S.E. versión 1.4.5, para que funcione en conjunto con Snort. Recordar que antes de descargar cada archivo, se usó el directorio creado para tener una instalación óptima y limpia.

Se descargó el archivo *base-1.4.5.tar.gz* de la siguiente url.

```
# wget http://sourceforge.net/projects/secureideas/files/BASE/base-1.4.5/
    base-1.4.5.tar.gz
```

Como se hizo con los archivos anteriores, se descomprimió y desempaquetó.

```
# tar -zxvf base-1.4.5.tar.gz
```

Finalmente se copió el directorio *base-1.4.5* a */var/www*. Una vez que se copió el archivo, se asignaron permisos de lectura, escritura y ejecución. Se modificó el nombre del directorio *base-1.4.5* a *base*. Realizados los pasos anteriores, se concluyó con la instalación de B.A.S.E.

```
# cp -r base-1.4.5 /var/www
# chmod 777 /var/www/base-1.4.5
# cd /var/www
# mv base-1.4.5 base
```

2.1.8. Configuración de B.A.S.E.

A continuación, se explicará la configuración de B.A.S.E.

Se modificó el archivo *default*, ubicado en */etc/apache2/sites-available*. La modificación se realizó para que cuando se desee ejecutar B.A.S.E., únicamente se escriba la dirección IP del servidor. Para ello, en la línea 4 del archivo anterior, se modificó la siguiente línea: *DocumentRoot /var/www/base*.

Se guardaron los cambios del archivo y se reinició nuevamente el servidor Apache.

B.A.S.E. se configuró desde un navegador web. En el navegador web se ingresó la dirección IP de nuestro servidor. Al ingresar mostró la siguiente pantalla, como se observa en la figura 2.6. Lo importante de esta primera pantalla, es la opción *Config Writeable*, ya que debe aparecer Yes. Si aparece No, se deberán de asignar los permisos correspondientes al directorio base ubicado en */var/www*.

Figura 2.6: Pantalla de inicio de configuración de B.A.S.E. Fuente: Captura propia.

En la siguiente ventana, se eligió el idioma y la ruta de adodb, como se muestra en la figura 2.7.

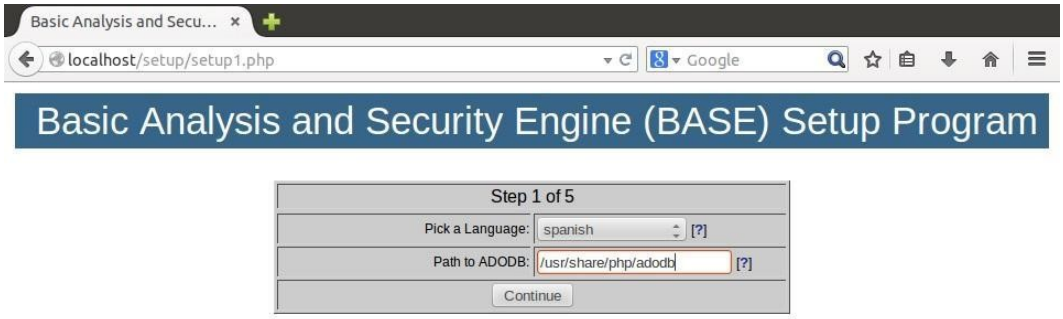


Figura 2.7: Selección del idioma del usuario y la ruta de adodb. Fuente: Captura propia.

A continuación, como se observa en la figura 2.8, se realizaron las configuraciones para la conexión con la base de datos MySQL.

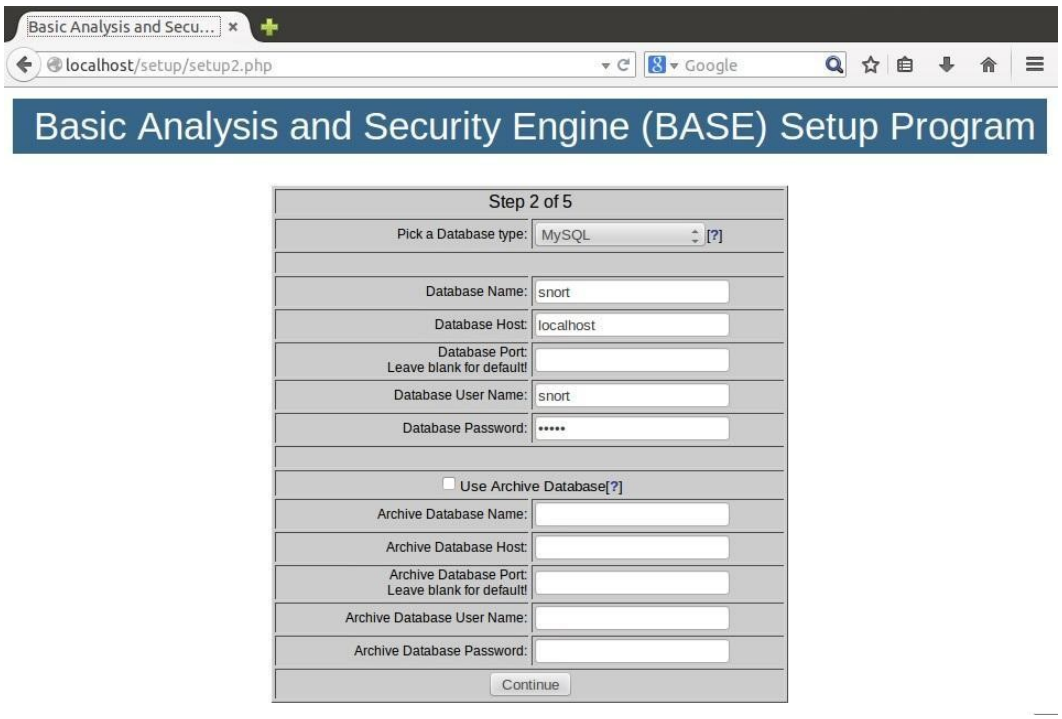


Figura 2.8: Configuración de parámetros para realizar la conexión con la base de datos snort. Fuente: Captura propia.

Se generó una cuenta de administrador para B.A.S.E. Es importante seleccionar la casilla que se encuentra arriba del formulario, para que cada vez que se ingrese a B.A.S.E. solicite credenciales de autenticación. La figura 2.9 ilustra este proceso.

Figura 2.9: Definición de los parámetros de autenticación del sistema B.A.S.E. Fuente: Captura propia.

Como se muestra en la figura 2.10, se agregaron tablas a la base de datos de Snort, para extender el soporte y las funcionalidades de Snort junto a B.A.S.E. Se dio clic en *Create BASE AG*.

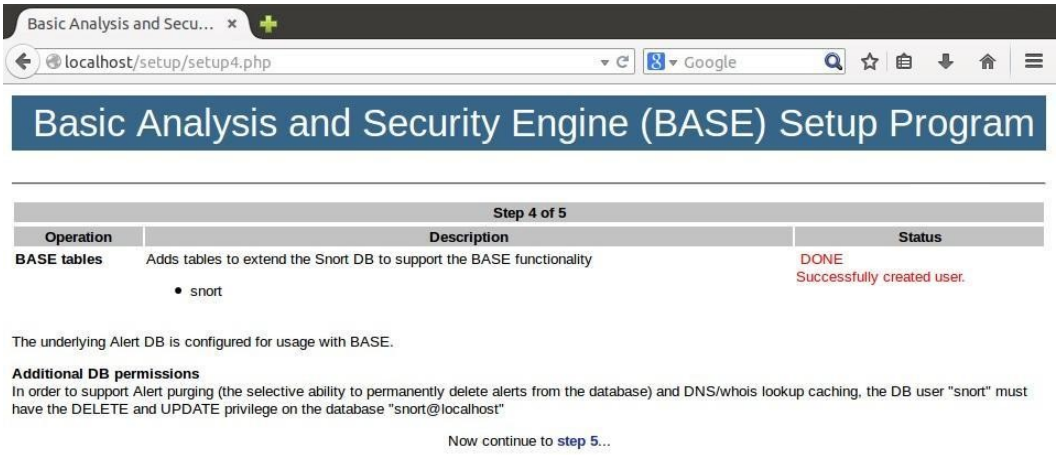


Figura 2.10: Creación de las tablas de B.A.S.E. en la Base de Datos snort. Fuente: Captura propia.

Posteriormente, un mensaje confirma que se crearon correctamente las tablas. Al dar clic en continuar con el paso 5 se debe realizar el proceso de autenticación para ingresar a la aplicación. Una vez finalizados los pasos anteriores se encuentra B.A.S.E. configurado y listo para su ejecución con todas las dependencias necesarias. Ver figura 2.11.

Figura 2.11: Acceso a B.A.S.E. Fuente: Captura propia.

2.1.9. Instalación y actualización de reglas en Snort

En esta sección se expondrán los diferentes conjuntos de reglas que se emplearon para garantizar el correcto desempeño de Snort, una vez puesto en producción.

El primer conjunto de reglas que podremos agregar son las reglas locales. Las reglas locales, son las reglas creadas por el usuario, inicialmente para verificar el correcto funcionamiento de Snort, y posteriormente para encontrar patrones en específico que el usuario desee capturar. Estas reglas se deben colocar en el archivo *local.rules*, ubicado en */etc/snort/rules*. Algunos ejemplos de reglas son:

```
alert icmp any any -> $HOME\_NET any (msg:"Prueba ICMP"; sid:1000001; rev
    :1)
alert tcp any any -> $HOME\_NET 22 (msg: "Conexion SSH por el puerto 22";
    classtype:suspicious-login; sid:1000008;)
alert tcp any any -> $HOME\_NET 23 (msg:"Conexion TELNET por el puerto 23"
    ; sid:1000503;)
```

Otra manera de maximizar la funcionalidad de Snort, fue descargar las reglas creadas por la comunidad de Snort. Este conjunto contiene alrededor de 20000 reglas de diferentes categorías.

Para poder descargar dichas reglas, se ingresó al siguiente enlace *https://www.snort.org/downloads* en el cual tenemos dos opciones de descarga de reglas. En la primera opción deberemos de registrarnos en el sistema de Snort, además de pagar una suscripción por $29.99 dólares al mes. La ventaja de esta opción es tener acceso inmediato a las actualizaciones más recientes de las reglas de Snort.

Para los que no opten por la opción anterior, existe una segunda alternativa en la

cual, simplemente debemos de registrar un correo válido en la plataforma de Snort. La desventaja de este conjunto de reglas, es el desfase de un mes respecto a las que publican mensualmente y qué tienen un costo. En este caso, se empleó el kit de reglas de la versión 2.9.

A continuación se descargó el kit de reglas de Snort versión 2.9. Para poder realizar la descarga, se debe contar con un previo registro. Recordar que una vez que se realizó la descarga del kit de reglas, se ubicaron en el directorio creado anteriormente, para almacenar todas las descargas que hayamos realizado.

Una vez descargado el archivo, se obtuvo el fichero *snortrules-snapshot-2962*, como se muestra a continuación:

```
# tar -zxvf snortrules-snapshot-2962.tar.gz
```

Se copió el contenido de los siguientes directorios: *preproc_rules*, *rules* y *so_rules* a la ubicación donde se crearon dichos directorios, cuando se instaló Snort.

```
# cp preproc_rules/* /etc/snort/preproc_rules/
# cp rules/* /etc/snort/rules/
# cp so_rules/* /etc/snort/so_rules/
```

Se cambió el usuario y grupo propietario a dichos directorios.

```
# chown -R snort:snort /etc/snort
```

Es importante mantener el kit de reglas actualizado, ya que con cierta frecuencia aparece nuevo software malicioso, que son más sofisticados y qué por ende son desconocidos para el Sistema de Detección de Intrusos.

Para mantener actualizado el conjunto de reglas de Snort, se empleó la herramienta PulledPork. PulledPork es un script escrito en Perl, el cual realizará las actualizaciones con el mínimo esfuerzo. Se instalaron los siguientes paquetes y se descargó la última versión de PulledPork en el directorio *InstalacionSnortCuckoo*.

```
# apt-get install libcrypt-ssleay-perl liblwp-useragent-determined-perl -y
# wget https://pulledpork.googlecode.com/files/pulledpork-0.7.0.tar.gz
```

Se descomprimió y desempaquetó el archivo *pulledpork-0.7.0.tar.gz*. Se cambió el nombre del directorio creado *(pulledpork-0.7.0)* a *PulledPork*, para mayor facilidad de uso.

```
# tar -zxvf pulledpork-0.7.0.tar.gz
# mv pulledpork-0.7.0 pulledpork
```

Antes de proseguir fue necesario conseguir un código Oinkcode, que fungió como llave privada, para realizar la actualización de las reglas. Se inició sesión con el usuario qué se creó para descargar el kit de reglas. Posteriormente, se dió clic sobre el nombre de usuario de Snort, como se muestra en la figura 2.12.

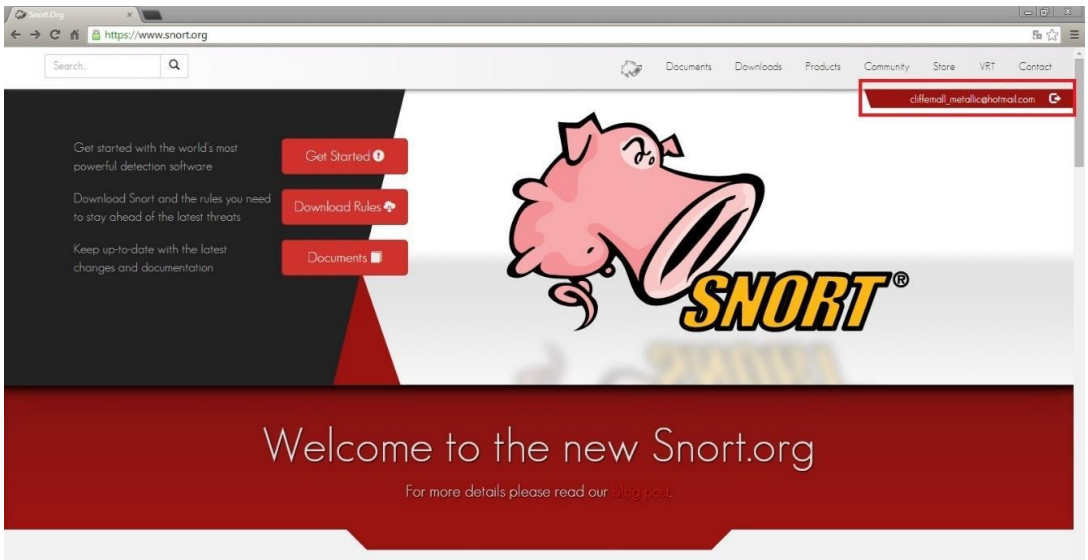


Figura 2.12: Login en la página de Snort. Fuente: snort.org.

A continuación, el usuario se debe dirigir a su perfil. En la sección Oinkcode, se busca el código. La figura 2.13 muestra este proceso:

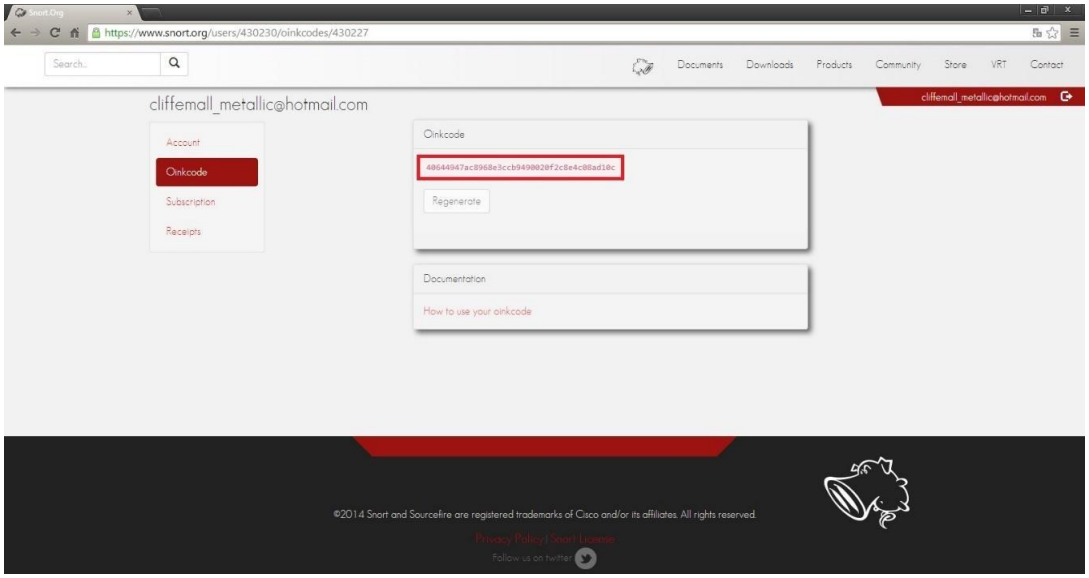


Figura 2.13: Oinkcode proporcionado en la página de Snort. Fuente: snort.org.

Realizado el proceso anterior, se modificó el archivo de configuración de PulledPork *(/usr/src/pulledpork/etc/pulledpork.conf)* y en la línea 19 se escribe la URL de donde son descargadas las reglas incluyendo el Oinkcode. Adicionalmente las líneas 21, 24 y 26 se comentaron.

```
rule_url=https://www.snort.org/rules/|<version reglas.tar.gz>|<oinkcode>
```

Las línea 72, 87, 90, 117, 131 se editaron de la siguiente manera respectivamente:

```
rule path=/etc/snort/rules/snort.rules
local_rules=/etc/snort/rules/local.rules
sid_msg=/etc/snort/sid-msg.map
config_path=/etc/snort/snort.conf
distro=Ubuntu-12.04
```

Se guardaron los cambios realizados en el archivo y se creó el archivo *snort.rules*.

```
# touch /etc/snort/rules/snort.rules
```

Se cambiaron los permisos al archive *pulledpork.pl*.

```
# chmod 755 /usr/InstalacionSnortCuckoo/pulledpork/pulledpork.pl
```

Finalmente, para actualizar el kit de reglas de Snort se ejecutó el siguiente comando que se puede observar en la figura 2.14, indicando la ubicación del archivo de configuración de PulledPork.

```
root@Snort-Cuckoo:~# /usr/InstalacionSnortCuckoo/pulledpork/pulledpork.pl -c /us
r/InstalacionSnortCuckoo/pulledpork/etc/pulledpork.conf

    http://code.google.com/p/pulledpork/
     ,_____ ____
     `----,\    )
      `--==\\  /    PulledPork v0.7.0 - Swine Flu!
       `--==\\/
     .-~~~~~-.Y|\\_  Copyright (C) 2009-2013 JJ Cummings
  @_/        /  66\_  cummingsj@gmail.com
    |    \   \   _(")
     \   /-| ||'--'  Rules give me wings!
      \_\  \_\\
 ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Checking latest MD5 for snortrules-snapshot-2961.tar.gz....
Rules tarball download of snortrules-snapshot-2961.tar.gz....
        They Match
        Done!
Prepping rules from snortrules-snapshot-2961.tar.gz for work....
        Done!
Reading rules...
Generating Stub Rules....
        Done
Reading rules...
Reading rules...
Setting Flowbit State....
        Enabled 39 flowbits
        Done
Writing /etc/snort/rules/snort.rules....
        Done
Generating sid-msg.map....
        Done
Writing v1 /etc/snort/sid-msg.map....
        Done
Writing /var/log/sid_changes.log....
        Done
Rule Stats...
        New:-------355
        Deleted:---23
        Enabled Rules:----5783
        Dropped Rules:----0
        Disabled Rules:---16413
        Total Rules:------22196
No IP Blacklist Changes

Done
Please review /var/log/sid_changes.log for additional details
Fly Piggy Fly!
root@Snort-Cuckoo:~#
```

Figura 2.14: Actualización completa del kit de reglas de Snort con PulledPork. Fuente: Captura propia.

2.2. Instalación de Cuckoo SandBox

La SandBox Cuckoo permitió realizar análisis de malware de manera eficiente. Esto se logró gracias al informe que se generó después de realizar un análisis en un entorno aislado y seguro. Las muestras de malware que obtuvo la SandBox, fueron capturadas por el IDS, al generarse una alerta. Este proceso de automatización se explica en el capítulo 4 de esta tesis.

2.2.1. Instalación de paquetes

En esta sección se explicará la instalación de los paquetes necesarios para el correcto funcionamiento de la SandBox.

Primero, se garantizó que el sistema tuviera instalado Python. En caso de no contar con Python se debe instalar. Para ello se emplea el gestor de paquetes *apt*.

```
# apt-get install python
```

Cuckoo requiere de los paquetes SQLAlchemy y Python BSON. Para la instalación del paquete *sqlalchemy* se realizó de la siguiente manera:

```
# apt-get install python-sqlalchemy
```

Para instalar el paquete *python-bson*, se realizó con el gestor de paquetes *apt* de Linux.

```
# apt-get install python-bson
```

A continuación se instalaron diversos paquetes Python, así como el paquete *build-essential*. Dicho paquete contiene las herramientas necesarias para crear, compilar e instalar programas.

```
# apt-get install python-pip python-dev build-essential
# apt-get install python-dpkt python-jinja2 python-magic python-pymongo
    python-gridfs python-libvirt python-bottle python-pefile python-
    chardet
```

También, se realizó la instalación de paqueterías que son opcionales y que por ende, no son estrictamente requeridas, pero se aconseja instalarlas.

Se instaló el paquete *Django*, para el uso de la interfaz web. Se realizó con la herramienta *pip*, la cual sirve para instalar y administrar paquetes Python:

```
# pip install Django==1.7.1
```

A continuación, se instaló el paquete *Ssdeep*. Dicho paquete permite saber que tan similares son dos o más archivos, mediante la técnica de fuzzy hashing. Se descomprimió con la herramienta *tar*:

```
# wget http://sourceforge.net/projects/ssdeep/files/ssdeep-2.10/ssdeep
    -2.10.tar.gz
# tar -zxvf ssdeep-2.10.tar.gz
```

Se cambió al directorio creado, se comprobaron las características del sistema y se compiló e instaló:

```
# cd ssdeep-2.10
# ./configure
# make
# make check
# make install
```

Pydeep es el siguiente paquete que se instaló. El paquete *Ssdeep* debe estar instalado para poder usar *Pydeep*.

Se cambió la ubicación al directorio *InstalacionSnortCuckoo*, y desde ahí se realizó la descarga. Se descomprimió con la herramienta *unzip* y se cambió al directorio creado. Se compiló e instaló el paquete con Python.

```
# wget https://github.com/kbandla/pydeep/archive/master.zip
# unzip master.zip
# cd pydeep-master
# python setup.py build
# python setup.py install
```

Para la instalación de *Yara* se requieren los siguientes paquetes: *libpcre3* y *libpcre3-dev*.

```
# apt-get install libpcre3 libpcre3-dev
```

Posterior a la instalación de los paquetes mencionados anteriormente, se descargó e instaló el paquete Yara, la cual es una herramienta que sirve para identificar y clasificar muestras de malware. El archivo Yara se descomprimió y se cambió al directorio *yara-3.1.0* para comprobar las características del sistema para su compilación e instalación.

```
# wget https://github.com/plusvic/yara/archive/v3.1.0.tar.gz
# tar -zxvf v3.1.0.tar.gz
# cd yara-3.1.0
# ./bootstrap.sh
# ./configure
# make
# make check
# make install
```

Para construir e instalar el paquete *yara-python* se realiza mediante el siguiente proceso:

```
# cd /usr/InstalacionSnortCuckoo/yara-3.1.0/yara-python
# python setup.py build
# python setup.py install
```

También se realizó la instalación de Distorm. Se descargó de la siguiente manera:

```
# wget ftp://ftp.cn.debian.org/gentoo/distfiles/distorm3-1.0.zip
# unzip distorm3-1.0.zip && cd distorm3-1.0
# cd distorm3-1.0
```

Se compiló e instaló con Python.

```
# python setup.py build
# python setup.py install
```

Otro paquete que se instaló fue Pycrypto. Se descargó con la herramienta *wget*. Se procedió a descomprimirlo y se cambió al directorio *pycrypto-2.6.1*.

```
# wget http://ftp.dlitz.net/pub/dlitz/crypto/pycrypto/pycrypto-2.6.1.tar.
    gz
# tar -xvzf pycrypto-2.6.1.tar.gz
# cd pycrypto-2.6.1/
```

De la misma forma en qué se compiló e instaló Distorm, se realizó con Pycrypto.

```
# python setup.py build
# python setup.py install
```

El siguiente paquete que se instaló fue *volatility*. Se descargó el archivo comprimido de *volatility* y se desempaquetó y descomprimió. Hecho lo anterior, se compiló e instaló con Python.

```
# wget https://volatility.googlecode.com/files/volatility-2.3.1.tar.gz
# tar xvzf volatility-2.3.1.tar.gz
# cd volatility-2.3.1
# python setup.py build
# python setup.py install
```

Finalmente, se instaló TCPdump, el cual es un analizador de paquetes que circulan a través de una red. Dicha herramienta requiere de privilegios de administrador. Pero cómo no es recomendable ejecutar Cuckoo como administrador, es necesario especificar en los binarios de Linux dicho requerimiento. Para ello se descargó la librería *libcap2-bin*, la cual proporciona la herramienta *setcap*, requerida para aplicar privilegios de administrador a Cuckoo.

```
# apt-get install tcpdump
# apt-get install libcap2-bin
```

Se configuraron las capacidades del sistema:

```
# setcap cap_net_raw,cap_net_admin=eip /usr/sbin/tcpdump
```

Finalmente se verificaron los resultados con el siguiente comando:

```
# getcap /usr/sbin/tcpdump
```

El resultado del comando anterior mostró lo siguiente, lo cual establece que ha surtido efecto el cambio realizado anteriormente:

```
/usr/sbin/tcpdump = cap_net_admin,cap_net_raw+eip
```

2.2.2. Configuración de la Base de Datos

En esta sección se explicará la configuración de la Base de Datos para que se puedan almacenar los análisis que vaya generando la SandBox, y que posteriormente se utilizaron para estudiar el comportamiento de dichas muestras.

El paquete *python-mysqldb* se instaló, el cual se define como una interfaz que permite trabajar con bases de datos MySQL desde Python. A continuación, se ingresó a MySQL como usuario root.

```
# apt-get install python-mysqldb -y
# mysql -u root -p
```

Posteriormente se creó la base de datos cuckoo y se proporcionaron todos los privilegios sobre la base de datos cuckoo para el usuario Snort y se recargaron todas las tablas manualmente. La conexión a la base de datos se establece en el archivo de configuración *cuckoo.conf*, que se encuentra en el apéndice A.

```
mysql> create database cuckoo;
mysql> grant all privileges on cuckoo.* to snort@localhost identified by '
    cuckoo';
mysql> flush privileges;
mysql> quit;
```

2.2.3. Instalación de VirtualBox

Una vez que se contó con todas las librerías anteriores en el servidor, se instaló VirtualBox. En VirtualBox se creó el sistema invitado, en dónde se ejecutó malware de manera segura. Se descargó VirtualBox versión 4.3.14 desde el siguiente enlace, en el directorio creado para el concentrado de todos los archivos descargados:

```
# wget http://download.virtualbox.org/virtualbox/4.3.20/virtualbox-4.3_4
    .3.20-96996~Ubuntu~precise_i386.deb
```

A continuación se empleó la herramienta *dpkg*, la cual permite instalar, compilar o eliminar un paquete Debian. En este caso se utilizó la instalación de VirtualBox.

```
# dpkg -i virtualbox-4.3_4.3.20-96996~Ubuntu~precise_i386.deb
```

Sí al momento de realizar la instalación, es mostrado un mensaje diciendo que Vir-

tualBox depende de paquetes adicionales, se puede solucionar el problema ejecutando el siguiente comando:

```
# apt-get -f install
```

Después de ejecutar el comando anterior, automáticamente se instalaron los paquetes que hacían falta para poder realizar la instalación de VirtualBox. Nuevamente se ejecutó la herramienta *dpkg* y esta vez se instaló VirtualBox correctamente.

2.2.4. Instalación y preparación de la máquina huésped

A continuación se explicará la correcta instalación y configuración del sistema huésped, el cual realizó el análisis de las muestras de malware. Es importante contar con un disco imagen de Windows XP Service Pack 3 de 32 bits.

Inicialmente se ejecuta VirtualBox. Una vez inicializado, se creó una nueva máquina virtual. Se le asignó el nombre de 'cuckoo1'. El Tipo: 'Microsoft Windows' y la versión 'Windows XP (32 bits)'. Seleccionada la versión del sistema operativo, así como sus características, se continuó con la siguiente ventana de configuración.

En esta ventana, se estableció el tamaño de la memoria RAM, que fue de 1024 MB, ya que es la cantidad sugerida en la documentación oficial de Cuckoo. Prosiguiendo con la configuración de la máquina huésped, se creó un disco duro virtual. En la siguiente ventana, se determinó un tamaño fijo para el archivo del disco duro virtual, ya que éste fue creado con su máximo tamaño y no permite expandir su tamaño.

En la siguiente ventana, se eligió el nombre del archivo del disco duro virtual. Nuevamente ingresamos 'cuckoo1'. En la parte de abajo se indicó el tamaño del disco duro virtual, el cual se fijó en 10 GB, debido a que es el tamaño que recomienda la guía de instalación de VirtualBox.

Nota: El proceso de la creación del disco duro virtual dependerá de las características de la máquina anfitrión.

Una vez que se encendió por primera vez la máquina huésped, se especificó la ruta en donde se encuentra la imagen de nuestro sistema operativo. Una vez seleccionado el archivo .iso, se continuó con una instalación típica de un sistema Windows XP.

Finalizado el proceso de instalación de la máquina huésped, se realizaron algunas adecuaciones en la máquina virtual, para que pudiera realizar el análisis de malware con el mejor desempeño posible.

Inicialmente, se ajustó el rendimiento de la máquina virtual huésped. Para ello, se ubicaron las 'Propiedades de mi PC' y se eligió la pestaña de 'Opciones Avanzadas'. Se seleccionó la sección qué se llama 'Rendimiento' y se accedió sobre el botón que dice 'Configuración'.

Se desplegó una nueva ventana con las opciones de rendimiento qué se requirieron configurar. En este caso, se adaptó el rendimiento de efectos visuales, seleccionando la opción 'Ajustar para obtener el mejor rendimiento'.

Posteriormente se realizó la configuración de alertas en el Centro de Seguridad de Windows XP. Para acceder al centro de Seguridad en Windows XP, se pulsó la tecla de Inicio de Windows e ingresar al Panel de Control. Una vez dentro del Panel de Control, se dio clic sobre "Centro de Seguridad" y a continuación se abrió la ventana del Centro de Seguridad.

Ahí, del lado derecho, se ubicó la sección "Recursos" y se seleccionó al quinto y última opción de configuración "Cambiar la forma en que el Centro de seguridad me alerta". Esta acción hizo que se desplegara una nueva ventana, en la cual se desmarcaron las tres casillas de verificación, las cuales corresponden al Firewall, actualizaciones automáticas y protección antivirus. La ventana de configuración de alertas puede observarse a continuación en la figura 2.15.

Figura 2.15: Configuración de alertas en el sistema huésped. Fuente: Captura propia.

Posteriormente los servicios de actualizaciones automáticas y el Firewall de Windows fueron deshabilitados. Se abrió una ventana de Ejecutar (tecla Windows + R) y se escribió services.msc la cual mostró todos los servicios locales.

En el servicio "Actualizaciones automáticas", se dio clic derecho sobre dicho servicio y se seleccionó 'propiedades'. Desplegó una ventana nueva y en la pestaña 'General', se detuvo el servicio. En el apartado "Tipo de inicio", se seleccionó Deshabilitado. Se hizo exactamente lo mismo con el servicio de "Firewall de Windows/Conexión compartida a Internet (ICS)".

La siguiente configuración que se realizó, fue deshabilitar el protector de pantalla y no seleccionar ningún fondo de pantalla. Esto se realizó dando clic derecho sobre cualquier parte de la pantalla, seleccionando la pestaña de pantalla. Las configuraciones anteriores se realizaron en las pestañas de "Protector de Pantalla" y "Escritorio" respectivamente.

A continuación Python tuvo que ser instalado, debido a que es un requerimiento

estricto para el sistema huésped Cuckoo, para que pueda ejecutarse adecuadamente el análisis de muestras de malware. Python se descargó desde su sitio web oficial *https://www.python.org/downloads/*. Se recomienda la versión 2.7 de Python. Adicionalmente se instaló la librería Python Image, la cual es usada para obtener capturas de pantallas, mientras se realiza un análisis de malware. También se instaló software adicional, para que las muestras pudieran tener interacción con aplicaciones. Para ello se obtuvo Adobe Reader v6.0, Java 1.5.0.12, Office 2003 y Mozilla Firefox 1.0 desde el siguiente enlace: *http://www.oldapps.com/*.

El siguiente software que se instaló, fue un agente (Agent.py), que se ejecuta dentro del huésped; el cual es el responsable de la comunicación y transferencia de datos con el sistema anfitrión (Ubuntu 12.04). Dicho agente se encuentra en la ruta */opt/cuckoo/agent*. Una vez que se encontró, se copió en *C:\Python27*.

Cuando se ejecutó el agente en Windows con la extensión *.py*, abrió una ventana de Python. Para evitar que dicha ventana sea mostrada y el agente se ejecute en background, basta con cambiar la extensión del archivo *Agent.py* por *Agent.pyw*.

El siguiente paso fue añadir un nuevo valor alfanumérico en el Editor del Registro. Para ingresar al Editor del Registro de Windows, nuevamente se debe presionar la combinación de teclas Windows + R y escribir la palabra *regedit*. Para agregar el nuevo valor alfanumérico, debe realizarse en la siguiente ruta:

```
HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run
```

Ubicada la carpeta 'Run', se dio clic derecho y desplegó un pequeño menú; donde se seleccionó un 'Nuevo Valor Alfanumérico'. Una vez que se creó, se modificó su nombre, y se llamó 'Agent'. Del lado derecho, donde se encuentran todos los valores alfanuméricos de la carpeta 'Run', se modificó el valor que se creó. La información del valor será: *"C:\Python27\agent.pyw"*. Este valor se debe agregar, para cuando es encendido el sistema huésped, automáticamente el agente de Python se ejecute, el cual abre el puerto 8000.

Se ejecutó el agente de Python y se verificó que el puerto 8000 esté a la escucha por cualquier dirección IP. Este proceso se ilustra a continuación, en la figura 2.16.

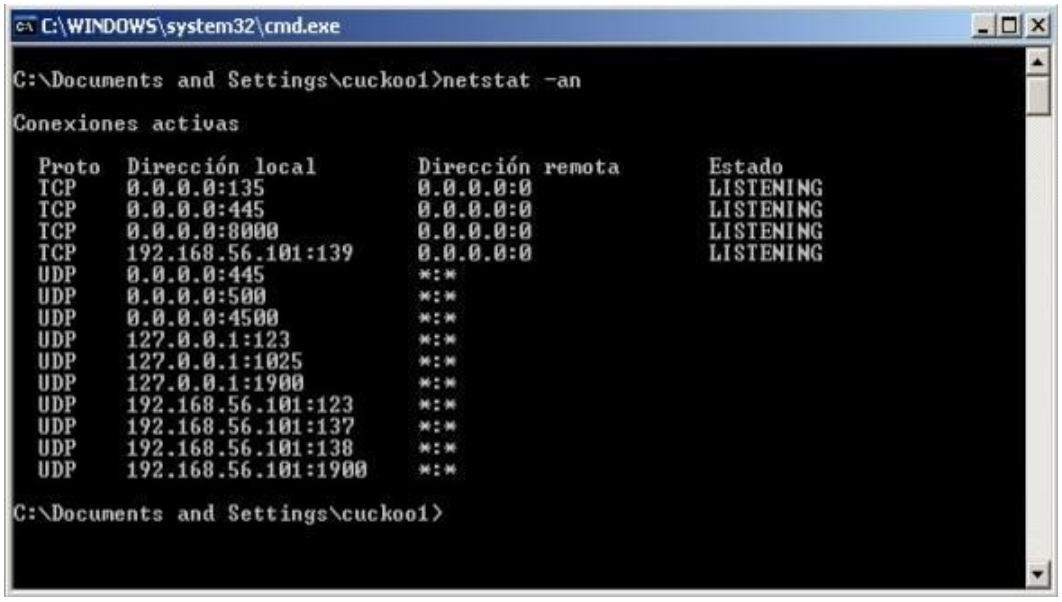


Figura 2.16: El agente Python escucha por el puerto 8000. Fuente: Captura propia.

Una vez que se comprobó el estado del puerto, se realizó una captura instantánea al sistema invitado. Esta acción se realizó por línea de comandos desde el sistema anfitrión ejecutando los siguientes comandos:

```
# VBoxManage snapshot "cuckoo1" take "cuckoo" --pause
# VBoxManage controlvm "cuckoo1" poweroff
# VBoxManage snapshot "cuckoo1" restorecurrent
```

Terminado el proceso anterior, se configuró una interfaz virtual, la cual comunica al sistema anfitrión con el sistema invitado, permitiendo enviar las muestras de malware recolectadas para su análisis.

Cuando se terminó de guardar la captura instantánea del equipo, en la pestaña Archivo de VirtualBox se seleccionó la opción 'Preferencias'. Se desplegó una nueva ventana y se seleccionó la opción 'Red'. Se seleccionó la pestaña "Redes solo-anfitrión", como se muestra en la figura 2.17, y del lado derecho y se seleccionó el ícono verde. La función de este ícono verde es crear la interfaz virtual 'vboxnet0'.

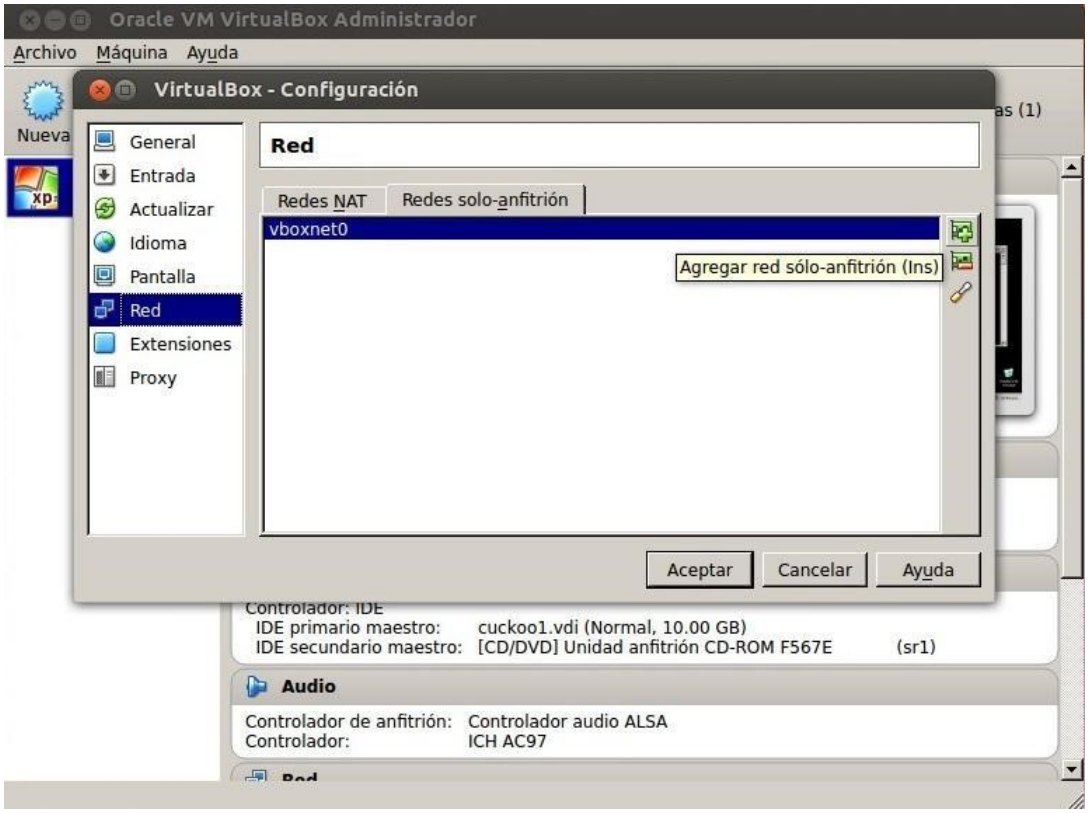


Figura 2.17: Creación de la interfaz virtual vboxnet0 en VirtualBox. Fuente: Captura propia.

Finalmente se configuró una dirección IP y una máscara de red al sistema invitado. La dirección IP es '192.168.56.1' y la máscara '255.255.255.0'.

Ahora en el sistema anfitrión, se modificó el firewall de Linux y se habilitó el acceso a internet para la máquina huésped, utilizando las siguientes reglas de iptables:

```
# iptables -A FORWARD -o eth1 -i vboxnet0 -s 192.168.56.0/24 -m conntrack
    --ctstate NEW -j ACCEPT
# iptables -A FORWARD -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT
# iptables -A POSTROUTING -t nat -j MASQUERADE
# sysctl -w net.ipv4.ip_forward=1
```

2.2.5. Instalación de la SandBox Cuckoo

Para la correcta instalación de Cuckoo, se empleó la herramienta wget. El archivo *cuckoo_1.1.tar.gz* se descargó en el directorio *opt*.

```
# cd /opt
# wget http://downloads.cuckoosandbox.org/1.1/cuckoo_1.1.tar.gz
```

Una vez que se descargó Cuckoo, el archivo fue desempaquetado y descomprimido. Esto generó un directorio llamado Cuckoo.

Para activar la Sandbox y validar su correcto funcionamiento se siguió la ruta */opt/cuckoo*. Desde ahí se activó la SandBox Cuckoo. Este proceso de activación de Cuckoo SandBox fue automatizado y se explica en el capítulo 4 de este tema.

```
# tar -zxvf cuckoo_1.1.tar.gz
# cd /opt/cuckoo
# python cuckoo.py
```

Al ejecutar Cuckoo estará a la espera de que se envíen muestras de malware para su análisis, como se muestra en la figura 2.18.

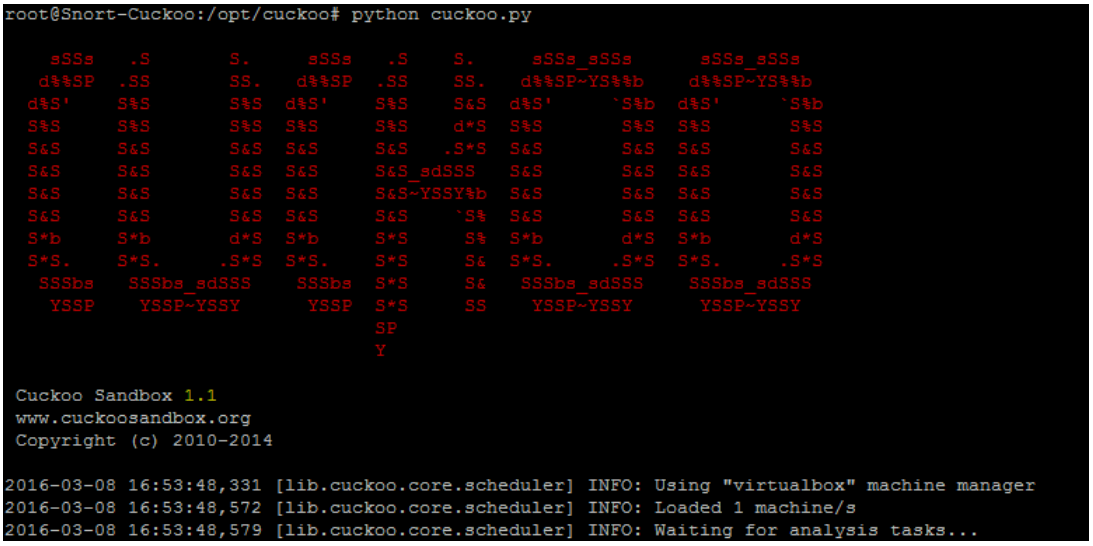


Figura 2.18: Ejecución de Cuckoo SandBox. Fuente: Captura propia.

Para poder enviar muestras de malware a Cuckoo de manera manual, se realiza de la siguiente manera:

```
# cd /opt/cuckoo/utils
# python submit.py ruta_muestras_malware
```

2.2.6. Archivos de configuración

En este apartado se explica la función de cada uno de los archivos de configuración de Cuckoo que se empleó en este proyecto. Para consultar el contenido de los archivos de configuración, referirse al apéndice A.

auxiliary.conf
Este archivo se debe editar si se desea activar o desactivar el uso de un analizador de paquetes (sniffer) externo, en este caso tcpdump.

cuckoo.conf
En este archivo se elige sí cada vez que inicia Cuckoo compruebe que se tenga la última versión instalada. Si se desea hacer esto, Cuckoo se conectará a una locación remota y verificará que la última versión que estemos ejecutando es la versión más actual. También en este archivo se configura la conexión con la base de datos, y se define la dirección IP y el puerto que Cuckoo va a emplear para devolver los resultados obtenidos.

memory.conf
En este archivo de configuración se activa o desactiva el uso de volatility, la cual es una herramienta que nos ayudará a realizar análisis forense sobre volcados de memoria.

processing.conf
En este archivo se configura la disponibilidad de los módulos de procesamiento. Estos módulos se encuentran en *modules/processing*. Se define como se asimilan los datos en bruto obtenidos durante el análisis.

reporting.conf
Este archivo de configuración habilita o deshabilita la generación automatizada de reportes. Genera un reporte con formato *jsondump* y un reporte con formato *html* por cada muestra de malware analizada.

virtualbox.conf
En este archivo de configuración se especifica la forma en que se ejecuta VirtualBox y la forma en que interactuará con Cuckoo, ya que puede ser por línea de comandos o con interfaz gráfica de usuario. En este proyecto se configuró el modo de interfaz gráfica.

2.3. Puesta en producción

En este apartado se explicarán los requerimientos de software y hardware necesarios para su funcionamiento, el proceso de instalación y puesta en producción del Sistema de Detección de Intrusos y la SandBox Cuckoo, así como la manera en que trabajan ambos sistemas.

2.3.1. Requerimientos de hardware y software

Se debe contar con el software y hardware necesario, para llevar a cabo la implementación del IDS y la SandBox, así como tener en cuenta la ubicación dentro de la red en la que fueron instalados dichos servicios.

El hardware necesario para la implementación del sistema es el siguiente:

- Un servidor o PC. Debe contar con al menos 4 GB en memoria RAM, ya que dentro de ella se ejecutan diversos procesos y servicios. El tener una cantidad óptima de memoria RAM, garantiza que no se produzcan cuellos de botella al momento de realizar las capturas de tráfico, generar alertas y analizar muestras de malware. Adicionalmente se recomienda tener un espacio de almacenamiento por lo menos de 500 GB, para poder almacenar el tráfico capturado, las muestras de malware obtenidas a partir de éste y los resultados del análisis de malware.

- Además, la PC debe de contar con dos tarjetas de red, ya que una interfaz se emplea para la captura de tráfico, y otra para la administración remota.

Los requerimientos de software necesarios que se necesitan son los siguientes:

- Linux Ubuntu (a partir de la versión 12.04). Este sistema operativo es la base en donde se implementaron los servicios del IDS y SandBox. Las ventajas que posee este sistema operativo es la gran estabilidad y el software que se necesita es gratuito. Otra ventaja de usar Ubuntu, es que cuenta con un firewall, el cual se puede modificar mediante el uso de iptables.

Este proyecto se basa en el IDS Snort, el cual es el sistema principal para poder realizar la captura del tráfico de red y su posterior análisis.

La SandBox Cuckoo es el sistema que se encarga de realizar el análisis de las muestras obtenidas, a partir del tráfico capturado por el IDS.

2.3.2. Configuración de port mirroring

La configuración de un port mirroring o puerto espejo fue necesario para la realización de esta tesis, debido a que el IDS Snort requiere monitorear el tráfico de toda una red, en busca de tráfico que pueda ser catalogado como malicioso.

Para ello en la configuración del switch core de dicha Secretaría, se estableció al puerto 20 como el puerto de destino del tráfico proveniente de las VLAN *Medicos, Medicos2 y Gobierno*; las cuales pertenecen al puerto 23 del switch, que es el enlace troncal.

En la figura 2.19 se aprecia el proceso de configuración del port mirroring en el switch core, así como el estatus de la configuración del puerto espejo y el proceso de guardado de la configuración actual.

```
* HEBDMDFC.14 # enable mirroring to port 20
* HEBDMDFC.15 # configure mirroring add port 23 vlan Medicos
* HEBDMDFC.16 # configure mirroring add port 23 vlan Medicos2
* HEBDMDFC.17 # configure mirroring add port 23 vlan Gobierno
* HEBDMDFC.18 # show mirroring
Mirroring Mode: Enhanced
Mirror port: 20 is up
Number of Mirroring filters:3
Mirror Port configuration:
        Port number 23 in vlan Medicos
        Port number 23 in vlan Medicos2
        Port number 23 in vlan Gobierno
* HEBDMDFC.19 # save
The configuration file primary.cfg already exists.
Do you want to save configuration to primary.cfg and overwrite it? (y/N) Yes
Saving configuration on master ..... done!
Configuration saved to primary.cfg successfully.
HEBDMDFC.20 #
```

Figura 2.19: Configuración del port mirroring. Fuente: Captura propia.

2.3.3. Beneficios

Con la implementación de un IDS en la red, se detecta en tiempo real, principalmente tráfico sospechoso, peticiones de resolución de nombres de dominio malicioso y ataques de denegación de servicios. Esto genera alertas, las cuales mantienen al administrador de red informado, para que tome las medidas necesarias, a fin de mitigar posibles fallas en los servicios y en la red.

Debido a que el IDS Snort, almacena las alertas generadas en una base de datos dedicada, se puede realizar un análisis posterior con mayor detalle, acerca del comportamiento del tráfico que generó dicho alerta.

Al implementar la SandBox Cuckoo con un Sistema de Detección de Intrusos, el beneficio de tener un IDS aumenta exponencialmente; ya que al mismo tiempo que el IDS genera alertas, se puede analizar el comportamiento en un ambiente aislado y controlado, y poder saber en cuestión de unos pocos minutos sí se trató de un falso positivo o realmente es una potencial amenaza.

Así mismo, una vez que termina el análisis en la SandBox, se genera un reporte, en el cual se muestra diversa información acerca del comportamiento del malware. Un apartado importante en el reporte, es que se conecta a la base de datos de Virus Total, y muestra cuántos motores antivirus detectaron la muestra como maliciosa.

2.3.4. Topología de red

La instalación del Sistema de Detección de Intrusos, garantiza conocer, capturar y analizar el tráfico de datos que circulan en una red en específico. Su ubicación dentro de la red, debe ser la idónea, ya que debe de estar implementado estratégicamente, a fin de que pueda capturar todo el tráfico que circula en los segmentos de red que serán monitoreados.

Una vez que el Sistema de Detección de Intrusos se encuentra en producción, se obtienen las capturas de tráfico necesarios, para poder analizar cuál es el comportamiento del tráfico de red.

El Sistema de Detección de Intrusos, cuando se encuentra en producción, no manipula o procesa la información que viaja por la red, simplemente la captura.

La ubicación del IDS se realizó en la Zona Desmilitarizada (DMZ) de la red, ya que es ahí donde se encuentran diversos servidores de filtrado de contenido, resguardados por un firewall. Esta ubicación tiene la ventaja de poder reunir en único puerto espejo, el tráfico de los servidores de producción junto al IDS, y poder detectar todo el tráfico de la institución, ya que el puerto reflejado, es la única puerta de salida de todo ese tráfico. Esta arquitectura tiene la ventaja, además de detectar anomalías en el tráfico de la red interna, detectar patrones anómalos en el tráfico entrante.

En la figura 2.20 se muestra la arquitectura de red empleada para la instalación del IDS. El IDS, además de contar con una interfaz que captura el tráfico del puerto espejo del switch de core, cuenta con otra interfaz, mediante la cual se puede tener acceso a él para su administración; ya que la otra interfaz se encuentra en modo pasivo y no cuenta con una dirección IP establecida.

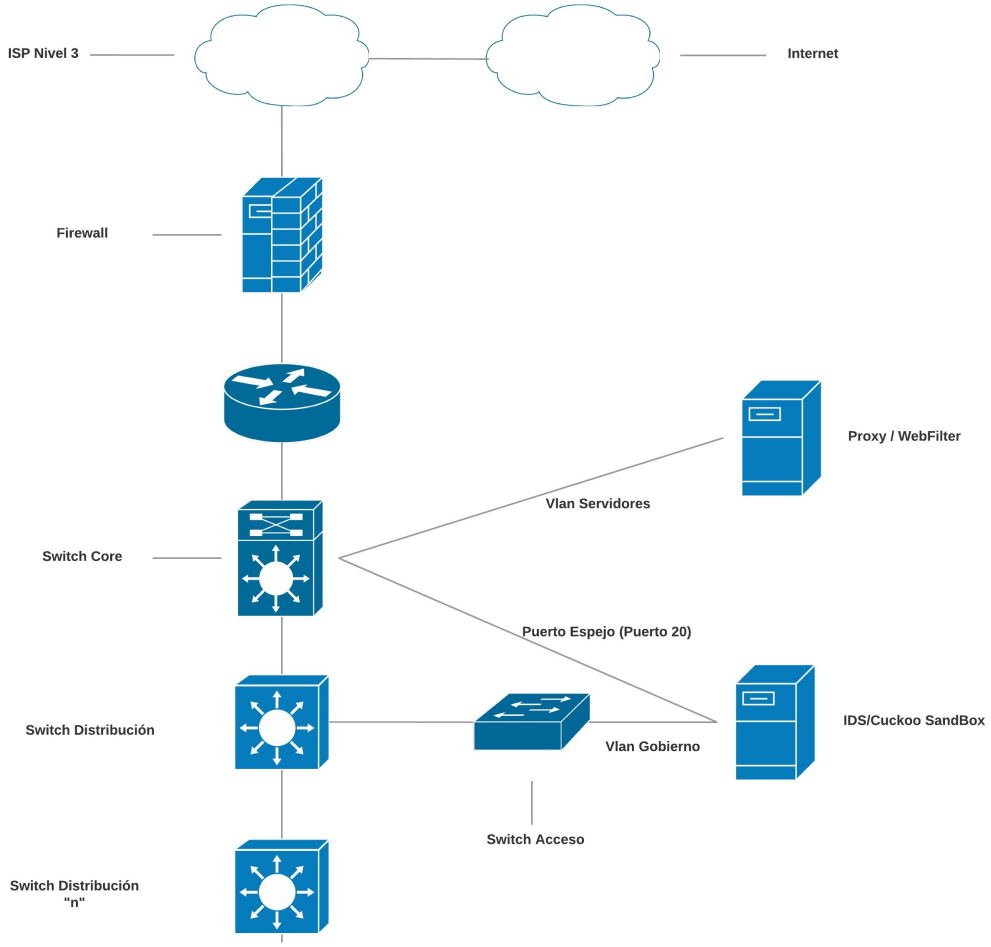


Figura 2.20: Arquitectura de red de la instalación del IDS Snort. Fuente: Elaboración propia.

2.3.5. Funcionamiento

En el sistema anfitrión, por cada alerta emitida, el Sistema de Detección de Intrusos genera un archivo que se almacena en */var/log/log_tcpdump/*. Éstos se almacenan con el nombre de *tcpdump.log.identificador*, en donde el identificador, es determinado por la fecha y hora con el que se crearon. Es importante mencionar que estos eventos que se encuentran registrados en los archivos mencionados anteriormente, también son almacenados en la base de datos snort.

Una vez que se obtuvieron suficientes alertas, el programa para la automatización del proceso de análisis de un evento (referirse al capítulo 4 de esta tesis) analizó los datos de tráfico referentes al tráfico HTTP, para poder obtener las direcciones URL

que fueron solicitadas y proceder a descargar, en caso de existir la petición de descarga de un archivo. Éstas son las muestras de malware. En caso de que no se haya realizado una petición para descargar algún archivo, también la URL puede ser analizada con Cuckoo SandBox.

La SandBox Cuckoo se encarga de la ejecución y análisis de las muestras. Funciona con el hypervisor VirtualBox. Cada muestra, se envía a una máquina virtual (invitada) en un ambiente aislado y controlado, a fin de evitar la propagación de la muestra.

Cuckoo está compuesto por el equipo anfitrión, en donde también se encuentra el IDS, y un par de sistemas invitados, para realizar el análisis.

El sistema anfitrión ejecuta el componente principal de la SandBox, el cual es iniciar el análisis, capturar tráfico y la generación de reportes. Los sistemas invitados, son entornos aislados, donde se envían las muestras de malware, para posteriormente ser ejecutadas y analizadas. El comportamiento de la muestra, es enviado al sistema anfitrión para la generación del reporte.

En la figura 2.21 se muestra la arquitectura de red que fue empleada, para la realización de esta tesis.

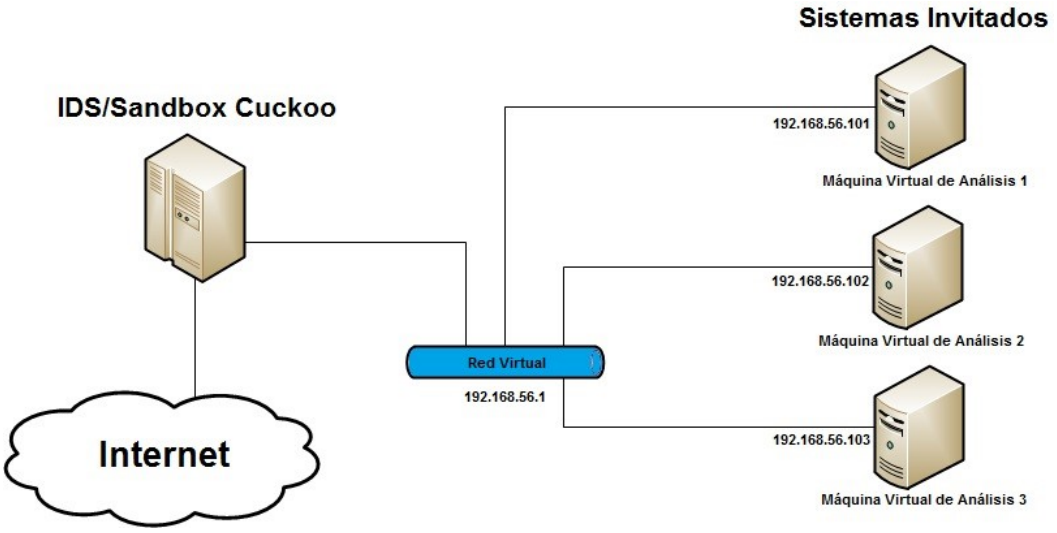


Figura 2.21: Arquitectura de red de Cuckoo SandBox. Fuente: docs.cuckoosandbox.org/ elaboración propia.

Capítulo 3

Recolección de Evidencia

En este capítulo se explicará el funcionamiento de un Sistema de Detección de Intrusos y la arquitectura de las reglas de Snort, las cuales sin ellas, no podrían generar alertas y notificar al administrador de red de probables comportamientos anómalos y/o sospechosos en el tráfico de red.

También, este capítulo explica la evidencia capturada con ayuda del IDS Snort, y los reportes generados por Cuckoo SandBox a partir de esta evidencia.

3.1. Funcionamiento de un IDS

3.1.1. Métodos para la recolección de datos

Para poder comprender el funcionamiento de la suite de reglas de Snort, es necesario saber cómo funciona un IDS.

La primera parte que se debe entender, es que un IDS simplemente está observando el tráfico de una red. En general existen 3 tipos de IDS, lo cual determinará el tipo de datos que ingresarán a él:

- Información específica de aplicaciones, como el correcto flujo de datos de éstas.
- Información específica de hosts, como contenido de logs y permisos del sistema de archivos.
- Información específica de red, como el contenido de paquetes en la red.

Un IDS no tiene un método eficaz para la recolección de información, cada uno cuenta con sus propias ventajas y desventajas; y cada método empleado es adecuado para diferentes tareas.

Sniffeo de paquetes

Cualquier tipo de IDS que observa el tráfico de una red, realiza un sniffeo de paquetes. El sniffeo de paquetes es un método clásico para realizar la detección de intrusos, e inclusive existen técnicas de evasión de IDS; por ejemplo los ataques de fragmentación, en los cuales un atacante cambia la manera en que los paquetes están fragmentados y en consecuencia al reensamblarlos, quedan fragmentos vacíos o superpuestos.

Análisis de logs

Muchos IDS extraen datos de los logs del sistema y alertan sí se observan comportamientos anómalos. Las implementaciones originales de IDS emplearon este método para recolectar datos. Dicho método permite obtener las huellas que un atacante dejó en los registros del sistema.

Monitoreo de llamada al sistema

Los HIDS se autoajustan como un residente en el kernel del sistema operativo, y está observando (o en algunos casos interceptando), llamadas potencialmente maliciosas al sistema. Sí el HIDS determina que la llamada al sistema puede ser potencialmente maliciosa, como solicitar cambiar de un usuario de menor privilegios a uno de mayor privilegios, genera una alerta.

En el caso de algunos HIDS, como el Linux Intrusion Detection System (LIDS), no permite realizar la llamada al sistema.

Monitoreo del sistema de archivos

Otro método empleados por los HIDS es observar los atributos y tamaños de los ficheros del sistema de archivos. Sí el kernel del sistema operativo espontáneamente modifica estos valores y ningún administrador realizó dicho cambio, es probable que hayan sido troyanizados. Observando estas alertas, ayuda a los administradores a determinar una posible actividad maliciosa.

3.1.2. Recolección de información e intentos de intrusión

Cualquier Sistema de Detección de Intrusos va a recolectar una gran cantidad de datos, debido al constante flujo de información en la red, así como una pequeña cantidad de ruido eléctrico. Para tener la mayor eficacia, es necesario contar con algoritmos para determinar cuál de todo ese tráfico vale la pena alertar a los administradores.

Existen dos tácticas básicas para obtener los mejores resultados. Para ello, el tráfico se puede ajustar a una política de seguridad determinada, dictada por las necesidades particulares de una organización o de la misma red. Mientras algunos administradores eligen permitir sólo el tráfico que ellos saben que es bueno, otros optan por bloquear el tráfico que ellos saben que es malo. Para realizar la mejor decisión para la organización, se debe tener en cuenta el tipo de tráfico que probablemente se observe, la cantidad de

personal que tiene que hacer frente a las alertas, así como el nivel de paranoia generado por las alertas.

La estrategia de permitir el tráfico bueno, implicará enfrentarse a una gran cantidad de falsos positivos y a un gran volumen de alertas generadas. En cambio, la estrategia de alertar cuando se sospeche de tráfico malicioso, significa que el volumen de alertas generadas será menor. Esto se debe, a que las reglas pueden ser muy específicas sobre la definición de algún comportamiento anómalo. Este método es más preciso, ya que cuando una alerta permanece activa por un largo tiempo, significa que una actividad maliciosa puede estar ocurriendo.

La elección de la estrategia es un análisis de beneficio/costo, así como considerar el tiempo y los recursos que están dispuestos a dedicarse al correcto funcionamiento de un IDS, con la eficacia de controlar el máximo número de ataques posibles.

3.1.3. Respuesta de un IDS ante un intento de ataque

Algunos sistemas son capaces de lanzar contraataques a las intrusiones detectadas, aunque las mejores prácticas indican que la identificación automatizada y el rastreo de llamadas al sistema, son los mecanismos más importantes.

Existen diferentes enfoques de las acciones a tomar por parte del IDS, cuando se detecta un intento de intrusión.

Respuesta pasiva
Como ya se mencionó, los IDS observan el tráfico de la red, y éstos se configuran para enviar y/o mostrar alertas a un administrador y/o almacenarlas en un archivo. Estas alertas pueden ser de diversas maneras, como por ejemplo, notificaciones por medio de e-mail, páginas o mensajes de texto para el administrador, incluso llamadas telefónicas automatizadas.

Generalmente, estos tipos de IDS se configuran con una interfaz de administración independiente de la interfaz que monitorea la red. La interfaz que está escuchando no tiene configurada ni siquiera una dirección IP, ya que es una interfaz pasiva y discreta, configurada para no responder; evitando revelar su presencia.

Respuesta activa
Los IPSs y los IDSs con capacidad de respuesta activa, se comportan como un IDS de respuesta pasiva en cuanto a la detección. Sin embargo, cuando detectan un intento de ataque, se pueden configurar para tomar medidas proactivas en contra de éstos, en lugar de simplemente alertar al administrador y esperar a que él tome medidas. Ellos pueden ser colocados, de tal manera que el tráfico atraviesa las interfaces, descartando el tráfico que ven como malicioso, para que sólo el tráfico confiable pueda ser enviado al

destinatario (modo inline). Este enfoque ofrece prevención y protección, debido a que el sensor puede detener un ataque antes de que alcance a su objetivo, que es algo que los IDSs de respuesta pasiva no pueden realizar.

Otro tipo de respuesta, es que ellos pueden enviar mensajes inalcanzables de Internet Control Message Protocol (ICMP) al origen, en un esfuerzo de convencimiento de que el sistema objetivo es inalcanzable.

La ventaja de la respuesta activa es que no tiene que estar un administrador de sistema viendo el flujo de información en tiempo real. Lo peligroso, son las consecuencias que una mala configuración puede provocar.

3.1.4. Motor de detección

El motor de detección de Snort, es el componente principal del IDS, el cual es un grupo de herramientas de detección y prevención de amenazas, que trabajan conjuntamente para reensamblar tráfico, evitar evasiones y amenazas, entre otros. En un NIDS, es el encargado de tomar datos de los paquetes capturados, comparándolos con el conjunto de reglas que han sido configuradas.

Snort, a diferencia de otros IDS, genera sólo una alerta por cada paquete que generó un evento. Esto se debe a que la primera regla que coincide con el payload del paquete, es la única que generará la alerta.

El motor de detección de Snort primero determina el conjunto de reglas establecidas, para así poder realizar la comparación entre las reglas y los paquetes. Éstas se almacenan en memoria dentro de las estructuras RTN (Rule Tree Node) y OTN (Options Tree Node), formando una matriz enlazada como se observa en la figura 3.1. En los RTNs se almacena la información de las cabeceras de las reglas, mientras que en los OTNs se almacenan las opciones de la regla.

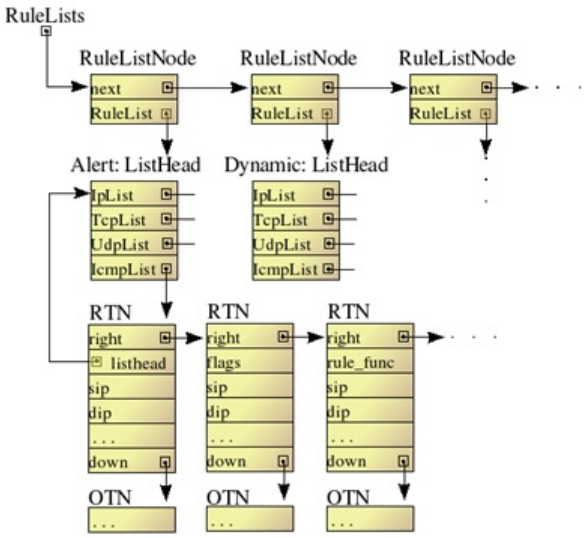


Figura 3.1: Matriz enlazada. Fuente: Optimización de Sistemas de Detección de Intrusos en Red, 2009.

Una vez que se ha generado una alerta, el paquete es clasificado de acuerdo al protocolo (TCP, UDP, ICMP, IP), para encontrar características propias de cada protocolo. Según el tipo de protocolo, se selecciona un árbol RTN y se inicia con un recorrido de los diferentes nodos RTN, verificando las opciones de los encabezados de las reglas hasta encontrar una coincidencia. Una vez que se seleccionó un árbol RTN, se inicia el recorrido de los nodos OTN, en donde están almacenadas las opciones de las reglas.

Finalmente, se determina sí un nodo OTN coincide con la información de los datos del paquete, comparando las opciones de esa regla contra el paquete.

3.1.5. Módulos de salida

Estos módulos son ejecutados cuando el sistema de alertas se activa, tomando la salida para almacenarla en diferentes formatos. A continuación se detalla cada módulo de Snort.

- **alert_syslog:** Este módulo permite enviar las alertas al syslog.
- **alert_fast:** Muestra información de manera rápida de los principales campos de la alerta. Estos son: tiempo, mensajes de la alerta, clasificación, prioridad y sockets de origen y destino.
- **alert_full:** Además de los campos anteriores, el módulo alert_full muestra información completa de las cabeceras de los paquetes capturados.
- **alert_unixsock:** Crea un socket para el envío de reportes de alertas.

- **log_tcpdump:** Almacena los paquetes en un archivo con formato tcpdump.
- **csv:** Permite escribir alertas en un formato más fácil de importar a una base de datos.
- **unified 2:** Este módulo permite registrar los datos en formato binario básico. Almacena paquetes y alertas para su posterior análisis.
- **log null:** Permite que se generen alertas para cierto tráfico, pero sin que se capturen los paquetes.
- **Database:** Snort permite el registro de alertas en una base de datos. Estas pueden ser: MySQL, Oracle, PostgreSQL y unixODBC.

3.2. Estructura de las reglas de Snort

Snort emplea un lenguaje de descripción de reglas ligero y simple, que es flexible y muy potente. Las reglas de Snort están divididas en dos secciones lógicas: encabezado de la regla y las opciones de la regla. El encabezado de la regla contiene la acción de la regla, protocolo, un operador, ip o red de origen y destino, así como los puertos de origen y destino. Las opciones de la regla contienen mensajes de alerta e información de en qué parte del paquete se debe inspeccionar para determinar sí la regla debe entrar en acción.

Aquí se encuentra la forma general de una regla de Snort:

```
acción protocolo ip/red_origen puerto_origen dirección_operador ip/
    red_destino  puerto_destino (opciones)
```

3.2.1. Encabezado de la regla

El encabezado de la regla contiene la información del origen y destino de la comunicación.

Acciones de la regla

La acción de la regla le indica a Snort como debe actuar cuando encuentra un paquete que equivale a los criterios de la regla. Las acciones por defecto disponibles en Snort son cinco: alert, log, pass, activate y dynamic. Adicionalmente, sí el IDS se configura en modo inline, existen tres opciones que son: drop, reject y sdrop.

- **alert:** Snort genera una alerta, usando el método de alerta seleccionado, para posteriormente capturar el paquete.
- **log:** captura el paquete.

- **pass:** el paquete es ignorado.
- **activate:** se genera una alerta y una regla dinámica es invocada.
- **dynamic:** se activa por una regla de activación.
- **drop:** es empleado en el modo inline; el paquete es bloqueado y capturado.
- **reject:** es empleado en el modo inline; el paquete es bloqueado y capturado. Se envía un reset TCP sí el protocolo es TCP o un mensaje ICMP de puerto inalcanzable sí el protocolo es UDP.
- **sdrop:** es empleado en el modo inline; el paquete es bloqueado y no se guarda ningún registro de él.

Protocolos

Permite establecer el protocolo de comunicación, el cual Snort analiza por comportamientos sospechosos: TCP, UDP, ICMP e IP.

Direcciones IP

Permite establecer el origen y destino de la comunicación a nivel de capa 3 del modelo OSI. Este campo se puede indicar de las siguientes formas:

- Indicando la IP de un host (p.e. 10.123.32.45).
- Indicando la dirección de un segmento de red (p.e. 10.123.32.0/24).
- Indicando un conjunto de direcciones de red, empleando corchetes (p.e. [10.123.32.44, 10.123.32.54, 10.123.32.60]).
- Empleando el uso de variables. Las variables por defecto en Snort es $EXTERNAL_NET (red externa), $HOME_NET (red local) y ANY (cualquier red). Sí es necesario, también se pueden definir nuevas variables en el fichero */etc/snort/snort.conf*. El formato es el siguiente:

```
var NET_a 10.123.32.0/24                    // red específica
var NET_b !10.123.32.0/24                   // diferente a una red específica
var NET_c [10.123.32.0/24, 10.123.33.0/24]  // conjunto de redes
```

Número de puertos

Permite establecer el origen y destino de la comunicación. Este campo se puede especificar de las siguientes formas:

- Indicando un rango de puertos (p.e. alert udp any any -> $HOME_NET 1:1024).
- Indicando un puerto en específico (p.e. log tcp any any -> 192.168.1.0/24 80).
- Indicando un rango de puertos menores o iguales a un puerto en específico (p.e. log tcp $EXTERNAL_NET any -> $HOME_NET :6000).

- Indicando un rango de puertos mayores o iguales a un puerto en específico (p.e. alert tcp any any -> 10.0.0.0/8 500:).
- Indicando la captura de todos los puertos menores a 2004 excepto el 1945 (p.e. log tcp any any -> $HOME_NET !1945:2004).

Dirección del operador

La dirección del operador indica el sentido de la comunicación que la regla debe aplicar. El operador -> indica que las direcciones IP y puertos de la izquierda son el host de origen del flujo de información, mientras que las direcciones IP y puertos de la derecha son el destino. El operador bidireccional <> indica a Snort que debe considerar las direcciones IP y puertos tanto de origen como de destino. Este operador es útil para el análisis de ambos lados de una comunicación, como por ejemplo sesiones telnet o POP3.

La siguiente regla analiza el tráfico que se origina de un servidor Telnet en la red 172.16.0.0/12, que contiene la palabra confidential:

```
alert tcp 172.16.0.0/12 23 -> any any (content: "confidential"; msg: "
    Detected confidential";)
```

La regla anterior puede adaptarse para analizar el tráfico saliente o entrante de cualquier servidor Telnet en la red:

```
alert tcp 172.16.0.0/12 23 <> any any (content: "confidential"; msg: "
    Detected confidential";)
```

Reglas activadoras/dinámicas

La combinación de reglas activadoras/dinámicas proporcionan al IDS un mejor desempeño. Se puede contar con una regla que active a otra cuando la primera lleve a cabo una acción para un número determinado de paquetes. La regla activadora se desempeña sólo como una regla de alerta, a menos de que contenga un campo de opción requerido: *activates*. En cambio, las reglas dinámicas actúan sólo como regla de registro, y tienen un campo de opción diferente: *activated_by*.

En el siguiente ejemplo se le indica al IDS que debe alertar, cuando detecte un desbordamiento de búfer IMAP y capture los siguientes 50 paquetes con cabeceras para el puerto 143. Así mismo, el origen de dichos paquetes tiene que ser una red distinta a la local, mientras que el destino tiene que ser la red local. Sí el desbordamiento de búfer fue exitoso, existe una gran posibilidad de que los datos útiles se encuentren dentro de los siguientes 50 paquetes, por lo que esos paquetes son valiosos para su posterior análisis.

```
activate tcp !$HOME_NET any -> $HOME_NET 143 (flags:PA; content:"|
    E8C0FFFFFF|/bin"; activate:1; msg "Desbordamiento de bufer IMAP!";)

dynamic tcp !$HOME_NET any -> $HOME_NET 143 (activated_by:1; count:50;)
```

3.2.2. Opciones de la regla

En las opciones de las reglas de Snort, se establecen los mensajes; con los cuales se muestran las alertas, un identificador de regla, las decisiones que elige la regla, los valores de los campos que contiene el paquete para que se cumpla la condición del encabezado, entre otros. Todas las opciones de la regla se deben separar una de otra usando un punto y coma (;). Las palabras clave de las opciones de la regla, deben ir separadas de sus argumentos con dos puntos (:).

Existen 4 categorías para las opciones de la regla:

- *general:* Estas opciones proporcionan información sobre la regla, pero no tienen ningún efecto durante la detección.
- *payload:* Estas opciones buscan datos de carga útil dentro del paquete.
- *non-payload:* Estas opciones buscan datos de carga no útil.
- *post-detection:* Estas opciones son disparadores de reglas específicas, que ocurren después de que una regla se ha ejecutado.

Opciones generales

En la tabla 3.1 se muestran las opciones generales de las reglas de Snort:

Nombre	Descripción
msg	Esta opción muestra una breve descripción cuando una regla es activada.
reference	Permite incluir referencias a las reglas de sistemas de identificación externos.
gid	Identifica la parte de Snort que genera un evento, cuando una regla en particular se activa.
sid	El sid (Snort ID) se emplea para identificar fácilmente cada regla de Snort. Los rangos de sid son los siguientes: $\leq$ 999 999 Reservado para las reglas oficiales de Snort.org y el conjuto de reglas VRT. 1, 000,000 - 1, 999,999 Reservado para las reglas locales. 2, 000,000 - 2, 999,999 Reservado para el repositorio de Bleeding Edge Threats. $\geq$ 3, 000,000 Reservado para futuro uso.
rev	La opción rev, indica la versión y/o revisión de la regla. La opción rev en combinación con sid, sólo identifica una regla de Snort; relacionando el ID de la regla individual con el número de revisión de la regla.
classtype	Se usa para clasificar una regla e identifica el tipo de ataque que contiene el paquete.
priority	Se emplea para asignar un nivel de gravedad a las reglas.
metadata	Permite insertar información adicional acerca de la regla.

Tabla 3.1: Opciones generales

Opciones payload
En la tabla 3.2 se muestran las opciones payload de las reglas de Snort:

Nombre	Descripción
content	Permite indicarle a Snort la búsqueda de contenido específico en el campo de datos del paquete. Sí los datos del paquete concuerdan exactamente con los ingresados en la regla, el resto de las opciones de la regla se ejecutan.
protected_ content	Permite buscar contenido en un paquete, sin revelear el contenido en la regla.
hash	Específica el algoritmo hash que se ocupa cuando se observa una regla de contenido protegido. Acepta MD5, SHA256 y SHA512.
nocase	Permite buscar contenido sin distinguir entre mayúsculas y minúsculas.
rawbytes	Permite observar los datos del paquete en crudo, ignorando cualquier decodificación que haya sido realizado por los preprocesadores.
depth	Específica un rango de "n"bytes de un paquete, en los cuales Snort debe inspeccionar el patrón.
offset	Se indica a Snort donde debe empezar a inspeccionar el patrón en un paquete.
distance	Se indica a Snort la cantidad de bytes que debe ignorar, antes de empezar a inspeccionar el patrón en un paquete.
within	Establece que Snort debe inspeccionar el patrón a partir de "n"bytes desde la última coincidencia.
uricontent	Se establece una búsqueda de contenido en las peticiones (URI) que se realizan a los servidores HTTP.
urilen	Indica la mínima y máxima longitud; así como el rango de tamaño de la URI para que coincida.
isdataat	Comprueba que la carga útil tiene datos en una ubicación específica.
pcre	Permite usar reglas escritas en perl que son compatibles con expresiones regulares.
byte_test	Analiza un campo de un byte contra un valor específico.
byte_jump	Se le indica a una regla leer la longitud de una porción de datos.
ftpbounce	Detecta ataques de FTP Bounce.
asn1	Decodifica un paquete o una porción de un paquete y comprueba sí existen codificaciones maliciosas.

Tabla 3.2: Opciones payload

Opciones non-payload
En la tabla 3.3 se muestran las opciones non-payload de las reglas de Snort:

Nombre	Descripción
fragoffset	Permite comparar el desplazamiento del fragmento IP contra un valor decimal.
ttl	Es usado para verificar el valor time-to-live de un paquete IP. Se emplea para detectar intentos de trazar la ruta. El rango de valores que emplea es de 0 a 255.
tos	Verifica el valor del campo ToS (Type of Service) en el encabezado IP, el cual indica una serie de parámetros sobre la calidad de servicio durante el recorrido por una red.
id	Se emplea para comprobar el valor del campo id del datagrama. Este se emplea en caso de que el datagrama sea fragmentado. Herramientas como scanners y exploits, emplean este campo para diversos propósitos.
ipopts	Esta opción se emplea para verificar el campo opción de la cabecera del datagrama IP.
fragbits	Es usado para verificar sí los bits reservados y de fragmentación se encuentran establecidos en el encabezado IP.
dsize	Permite establecer el tamaño del payload. Se emplea para verificar tamaños anómalos de paquetes, que puedan causar desbordamiento de búfer.
flags	Se usa para comprobar sí los bits específicos de las banderas TCP están activos. Los bits que verifica son: F -FIN S -SYN R -RST P -PSH A -ACK U -URG C -CWR E –ECE 0 -Ninguna bandera TCP establecida.
flow	Es empleada para determinar la dirección del flujo del tráfico.
flowbits	Esta opción es usada para seguir los estados de las sesiones del protocolo de transporte, junto con la opción flow.
seq	Permite verificar por un número de secuencia TCP.

Continúa en la siguiente página

Tabla 3.3 – Continuación de la página previa

Nombre	Descripción
ack	Verifica el valor ACK de un paquete TCP.
window	Verifica el tamaño de una ventana TCP específica.
itype	Identifica el tipo de mensaje ICMP.
icode	Identifica el valor del código ICMP.
icmp_id	Obtiene el identificador ICMP.
icpm_seq	Permite identificar una determinada secuencia de paquetes ICMP.
rpc	Se emplea para identificar aplicaciones Rich Client Platform (RPC), tales como Eclipse, NetBeans, Visual Studio, etc.
ip_proto	Obtiene el protocolo del encabezado IP.
sameip	Realiza una comparación entre las direcciones IP de origen y destino; y determina sí las IP de origen y destino son las mismas.
stream_ reassemble	Permite a una regla activar o desactivar el reensamblado del flujo TCP, en el tráfico que coincidió con las reglas.
stream_size	Permite a una regla capturar tráfico, de acuerdo a la coincidencia del número de bytes observados.

Tabla 3.3: Opciones non-payload

Opciones post-detection
En la tabla 3.4 se muestran las opciones post-detection de las reglas de Snort:

Nombre	Descripción
logto	Se le indica a Snort que guarde el paquete que activó esa alerta.
session	Permite extraer datos de usuario de sesiones TCP, por ejemplo telnet, ftp, entre otros.
resp	Permite a Snort terminar conexiones de protocolo basadas en las reglas que se activan. Por ejemplo, puede enviar un paquete específico TCP o ICMP.
react	Permite una respuesta activa, que permite el envío a una página web u otro servicio al cliente, para después terminar la conexión.
tag	Registra un número en específico de paquetes después de que una alerta se ha activado.
activates	Permite especificar a una regla que agregue otra, cuando un evento específico de red ocurrió.
activated_by	Permite activar dinámicamente una regla, cuando una alerta en específico se activó.
count	Especifica el número de paquetes que debe dejar pasar la regla, antes de que ésta se active. Debe ser empleada con la opción "activated_by".
replace	Es una opción especial para el modo inline de Snort. Se reemplazará el contenido coincidente por una cadena de la misma longitud.
detection_filter	Establece un rango, en el que debe ser excedido por un host de origen o destino, antes de que la regla pueda generar una alerta.

Tabla 3.4: Opciones post-detection

3.3. Eventos generados (Alertas)

En el periodo que se dejó funcionando el Sistema de Detección de Intrusos, se generaron las siguientes alertas, mostradas en la figura 3.2.

< Firma >	< Clasificación >	< Total # >
[snort] Snort Alert [1:2002750:10]	bad-unknown	79103(19%)
[snort] Snort Alert [1:28806:2]	trojan-activity	177(0%)
[snort] http_inspect: MESSAGE WITH INVALID CONTENT-LENGTH OR CHUNK SIZE	unknown	528(0%)
[snort] sensitive_data: sensitive data global threshold exceeded	sdf	22420(5%)
[snort] Snort Alert [1:2001683:3]	*desclasificado*	12674(3%)
[snort] Snort Alert [1:5001684:99]	*desclasificado*	31085(7%)
[snort] http_inspect: UNKNOWN METHOD	unknown	32969(8%)
[snort] Snort Alert [1:2002749:4]	bad-unknown	11144(3%)
[snort] http_inspect: HTTP RESPONSE GZIP DECOMPRESSION FAILED	unknown	3838(1%)
[snort] stream5: Reset outside window	bad-unknown	1181(0%)
[snort] sensitive_data: sensitive data - eMail addresses	sdf	22959(5%)
[snort] http_inspect: LONG HEADER	bad-unknown	47062(11%)
[snort] stream5: Limit on number of overlapping TCP packets reached	bad-unknown	7716(2%)
[snort] http_inspect: NO CONTENT-LENGTH OR TRANSFER-ENCODING IN HTTP RESPONSE	unknown	20103(5%)
[snort] stream5: TCP Small Segment Threshold Exceeded	bad-unknown	45082(11%)
[snort] sensitive_data: sensitive data - Credit card numbers	sdf	194(0%)
[snort] stream5: Bad segment, overlap adjusted size less than/equal 0	bad-unknown	13834(3%)
[snort] ssh: Protocol mismatch	non-standard-protocol	59988(14%)
[snort] Snort Alert [1:3000003:99]	misc-activity	54(0%)
[snort] Snort Alert [1:30211:1]	trojan-activity	3120(1%)
[snort] Snort Alert [1:28039:4]	trojan-activity	81(0%)
[snort] Snort Alert [1:31683:1]	trojan-activity	45(0%)
[snort] Snort Alert [1:28801:2]	trojan-activity	547(0%)
[snort] http_inspect: OVERSIZE REQUEST-URI DIRECTORY	bad-unknown	259(0%)
[snort] http_inspect: CHUNKED ENCODING - EXCESSIVE CONSECUTIVE SMALL CHUNKS	unknown	312(0%)
[snort] Snort Alert [1:28423:1]	trojan-activity	77(0%)
[snort] http_inspect: NON-RFC DEFINED CHAR	bad-unknown	1140(0%)
[snort] Snort Alert [1:21860:3]	successful-user	90(0%)
[snort] Snort Alert [1:27919:3]	trojan-activity	90(0%)
[snort] http_inspect: UNESCAPED SPACE IN HTTP URI	unknown	28(0%)
[snort] http_inspect: POST W/O CONTENT-LENGTH OR CHUNKS	unknown	12(0%)
[snort] Snort Alert [1:32125:1]	trojan-activity	2(0%)
[snort] Snort Alert [1:31527:1]	trojan-activity	4(0%)

Figura 3.2: Alertas emitidas por el IDS Snort. Fuente: Captura propia.

Para la realización de este proyecto solo se toman en cuenta las alertas que indican un comportamiento anómalo por los equipos finales de los usuarios. Estas se indican con la clasificación *trojan-activity*, omitiendo las demás alertas que se generaron.

El primer evento que se generó es el que contiene la firma "Snort Alert [1:28806:2]", el cual generó 177 eventos. Esto se observa en la figura 3.3.

[snort] Snort Alert [1:28806:2]	trojan-activity	177(0%)

Figura 3.3: Alerta de Snort 1:28806:2. Fuente: Captura propia.

La siguiente alerta generada contiene la firma "Snort Alert [1:30211:1]" generando un total de 3120 eventos. Esta alerta se observa a continuación en la figura 3.4.

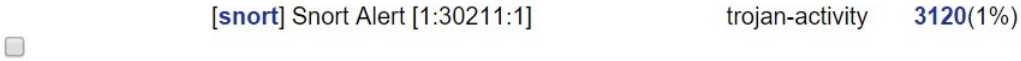


Figura 3.4: Alerta de Snort 1:30211:1. Fuente: Captura propia.

El siguiente evento generado contiene la firma "Snort Alert [1: 28039:4]". Esta alerta tuvo una ocurrencia de 81 eventos, tal y como se muestra en la figura 3.5.

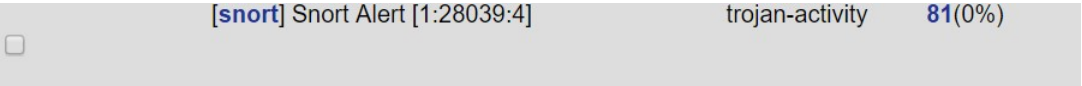


Figura 3.5: Alerta de Snort 1:28039:4. Fuente: Captura propia.

El evento que contiene la firma "Snort Alert [1:31683:1]", generó un total de 45 alertas. La figura 3.6 muestra este evento.

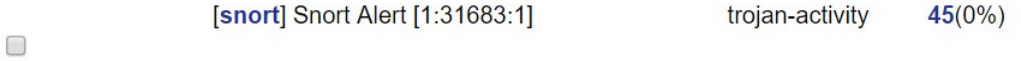


Figura 3.6: Alerta de Snort 1:31683:1. Fuente: Captura propia.

El evento generado con la firma "Snort Alert [1:28801:2]", tuvo un total de 547 alertas. Esto se observa en la figura 3.7.

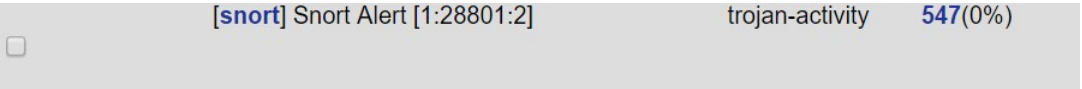


Figura 3.7: Alerta de Snort 1:28801:2. Fuente: Captura propia.

La alerta con la firma "Snort Alert [1:28423:1]", disparó 77 alertas. Los detalles de este evento se observan en la figura 3.8.

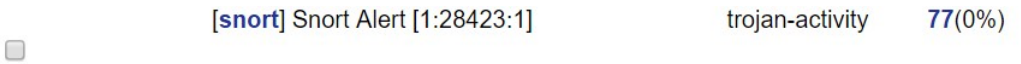


Figura 3.8: Alerta de Snort 1:28423:1. Fuente: Captura propia.

El evento con la firma "Snort Alert [1:27919:3], generó un total de 90 alertas, como se aprecia a continuación en la figura 3.9.

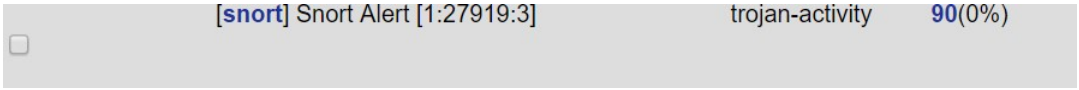


Figura 3.9: Alerta de Snort 1:27919:3. Fuente: Captura propia.

En la figura 3.10 se muestra el evento con la firma "Snort Alert [1:32125:1]", la cual disparó 2 alertas.

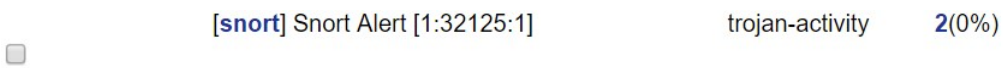


Figura 3.10: Alerta de Snort 1:32125:1. Fuente: Captura propia.

Finalmente, el evento con la firma "Snort Alert [1:31527:1]", tuvo una ocurrencia de 4 eventos, como se muestra en la figura 3.11.

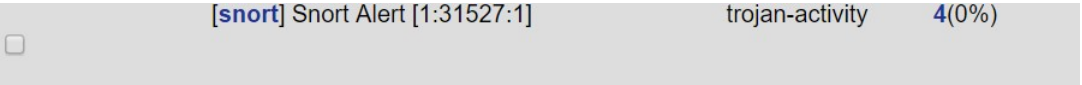


Figura 3.11: Alerta de Snort 1:31527:1. Fuente: Captura propia.

Cabe recordar que estos eventos, se generaron gracias a la ayuda de las reglas que se descargaron de la página oficial de Snort. El conjunto de reglas que se empleó fue el que proporciona la comunidad de Snort de manera gratuita.

3.3.1. Contenido de las reglas disparadas

A continuación se explicarán cada una de las reglas que generaron los eventos anteriormente descritos.

Snort Alert [1:28806:2]

La regla que generó el evento "Snort Alert [1:28806:2]", se encuentra ubicada en el fichero */etc/snort/rules/local.rules*. El contenido de la regla es el siguiente:

```
alert tcp $HOME_NET any -> $EXTERNAL_NET $HTTP_PORTS (msg:"INDICATOR-
    COMPROMISE potential malware download - single digit .exe file
    download"; flow:to_server,established; urilen:6; content:".exe";
    fast_pattern:only; pcre:"/\/[a-z0-9]\.exe$/Ui"; metadata:impact_flag
    red, policy security-ips drop, ruleset community, service http;
    reference:url,urlquery.net/search.php?q=%5C%2F%5Ba-zA-Z%5D%5C.%5BEe%5D
    %5BXx%5D%5BEe%5D%24&type=regexp&start=2013-09-07&end=2013-12-06&max
    =400; classtype:trojan-activity; sid:28806; rev:2;)
```

- **Encabezado de la regla:**
  Esta regla es una alerta y establece que el protocolo que va a observar va a ser TCP. El origen de la comunicación es cualquier host que pertenezca a la red 172.16.0.0/12 y el puerto de origen puede ser cualquiera.

  El destino de la comunicación es cualquier dirección IP que no se encuentre en la red local, y los puerto de destino pueden ser cualquiera de los que a continuación se mencionan: 36, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 311, 383, 555, 591, 593, 631, 801, 808, 818, 901, 972, 1158, 1220, 1414, 1533, 1741, 1830, 1942, 2231, 2301, 2381, 2809, 2980, 3029, 3037, 3057, 3128, 3443, 3702, 4000, 4343, 4848, 5000, 5117, 5250, 5600, 6080, 6173, 6988, 7000, 7001, 7071, 7144, 7145, 7510, 7770, 7777, 7778, 7779, 8000, 8008, 8014, 8028, 8080, 8081, 8082, 8085, 8088, 8090, 8118, 8123, 8180, 8181, 8222, 8243, 8280, 8300, 8333, 8344, 8500, 8509, 8800, 8888, 8899, 8983, 9000, 9060, 9080, 9090, 9091, 9111, 9290, 9443, 9999, 10000, 11371, 12601, 13014, 15489, 29991, 33300, 34412, 34443, 34444, 41080, 44449, 50000, 50002, 51423, 53331, 55252, 55555, 56712. Esta lista de puertos se encuentra en el fichero snort.conf. En este fichero a la variable HTTP_PORTS se le pueden agregar o eliminar puertos, según sean las necesidades de la organización. En este caso, se empleó la configuración de puertos HTTP por defecto de Snort.

- **Opciones de la regla:**
  El argumento de la opción *msg* (INDICATOR-COMPROMISE potential malware download - single digit .exe file download), va a mostrar dicho mensaje que

identifica el por qué esa alerta fue disparada.

Los argumentos de la opción *flow* (to_server, established), indican que se aplique la regla solo a un sentido de la comunicación y que sean peticiones hacia un servidor. El argumento established indica que deben existir conexiones TCP establecidas.

El argumento *urilen* (6), establece la longitud de 6 bytes de las URI's que comparará Snort.

El argumento de la opción *content* (.exe), establece que Snort busque la palabra .exe en el payload del paquete, el cual es la extensión de un archivo ejecutable.

El argumento de la opción *fast_pattern* (only), establece que debe usar el contenido ".exe" por el Fast Pattern Matcher, para que no se evalúe como la opción "content" de la regla.

El argumento de la opción *pcre* (/\/[a-z0-9]\.exe$/Ui), sirva para indicar el uso de una expresión regular. Dicha expresión regular establece que el valor a buscar en el payload del paquete debe de contener cualquier palabra que empiece con "/" formado por una letra minúscula o un número, seguido de la extensión .exe. Los siguientes son delimitadores que indican que debe distinguir entre mayúsculas y minúsculas.

Los argumentos de la opción *metadata* (impact_flag red, policy security-ips drop, ruleset community y service http) establecen información adicional acerca de la regla, sin que afecte los patrones de búsqueda. Los primeros dos argumentos se emplean en el modo inline de Snort. El tercer argumento proporciona información acerca de que esta regla pertenece al grupo de reglas de la comunidad de Snort. El cuarto argumento identifica a la regla como un servicio HTTP.

El argumento de la opción *reference* (url), establece una referencia externa, de sistemas de identificación externos.

El argumento de la opción *classtype* (trojan-activity), clasifica e identifica el tipo de ataque que contiene el paquete.

El argumento de la opción *sid* (28806), establece el identificador único para esta regla en específico. El identificador de esta regla es el 28806.

El argumento de la opción *rev* (2), indica que es la versión 2 de dicha regla.

Snort Alert [1:30211:1]

La regla que generó al evento "Snort Alert [1:30211:1]" se ubica en el fichero */etc/snort/rules/malware-cnc.rules*. El contenido de la regla es el siguiente:

```
alert tcp $EXTERNAL_NET $HTTP_PORTS -> $HOME_NET any (msg:"MALWARE-CNC Win
    .Trojan.ZeusVM embedded image config file download"; flow:to_client,
    established; flowbits:isset,file.jpeg; file_data; content:"|FF FE 3F
    10 00 00|"; fast_pattern:only; pcre:"/\xFF\xFE\x3F\x10\x00\x00.{14}[\
    x2Bx\x2Fa-z0-9]{20}/smi"; metadata:impact_flag red, policy balanced-
    ips drop, policy security-ips drop, service http; reference:url,www.
    virustotal.com/en/file/
    C003CA9C9694489F202E5A77FBD4973ADF7286C414EB98D525A8BFBC582D8962/
    analysis/; classtype:trojan-activity; sid:30211; rev:1;)
```

- **Encabezado de la regla:**
  Esta regla es una alerta y establece que el protocolo que va a observar va a ser TCP. El origen de la comunicación es cualquier dirección IP que no pertenezca al segmento de la intranet. Los puertos de origen de la comunicación podrán ser los puertos HTTP que se mencionaron en la alerta 28806.

  El destino de la comunicación es cualquier dirección IP que pertenezca al segmento de la intranet y el puerto destino puede ser cualquiera.

- **Opciones de la regla:**
  El argumento de la opción *msg* (MALWARE-CNC Win.Trojan.ZeusVM embedded image config file download), mostrará este mensaje cuando dicha alerta haya sido disparada, permitiendo identificar esta alerta con mayor facilidad.

  Los argumentos de la opción *flow* (to_client, established), indican que se aplique la regla solo a un sentido de la comunicación y que sean peticiones hacia un cliente. El argumento established indica que deben existir conexiones TCP establecidas.

  Los argumento de la opción *flowbits* (isset,file.jpeg), le permite asegurar a Snort que debe buscar un archivo JPEG.

  La opción *file_data*, en conjunto con el argumento de la opción *content* (|FF FE 3F 10 00 00|); asegura que el contenido en hexadecimal se encuentra en el flujo

del archivo actual con extensión JPEG.

El argumento de la opción *fast_pattern* (only), establece que debe usar el contenido en hexadecimal por el Fast Pattern Matcher, para que no se evalúe como la opción "content" de la regla.

El argumento de la opción *pcre* (/\xFF\xFE\x3F\x10\x00\x00.{14}[\x2Bx\x2Fa-z0-9]{20}/smi), sirve para indicar el uso de una expresión regular. Esta expresión regular indica que debe de buscar las referencias hexadecimales de caracteres en regex. El punto indica que puede ser cualquier carácter, excepto saltos de línea. El valor dentro de las llaves (14), indica que cualquier carácter se debe repetir 14 veces, seguida de la repetición de 20 veces de una cadena compuesta por una combinación de una referencia hexadecimal, el carácter "x", una letra minúscula o un número. A continuación los delimitadores "smi"; s, indica que debe de considerar los espacios en blanco como espacios en blanco; m, el modo debe ser multilínea y la i que distinga entre mayúsculas y minúsculas.

Los argumentos de la opción *metadata* (impact_flag red, policy balanced-ips drop, policy security-ips drop y service http) establecen información adicional acerca de la regla, sin que afecte los patrones de búsqueda. Los primeros tres argumentos se emplean en el modo inline de Snort. El cuarto argumento identifica a la regla como un servicio HTTP.

El argumento de la opción *reference* (url), establece una referencia externa, de sistemas de identificación externos.

El argumento de la opción *classtype* (trojan-activity), clasifica e identifica el tipo de ataque que contiene el paquete.

El argumento de la opción *sid* (30211), establece el identificador único para esta regla en específico. El identificador de esta regla es el 30211.

El argumento de la opción *rev* (1), indica que es la versión 1 dicha regla.

Snort Alert [1:28039:4]

La regla que generó al evento "Snort Alert [1:28039:4]" se ubica en el fichero */etc/snort/rules/indicator-compromise.rules*. El contenido de la regla es el siguiente:

```
alert udp $HOME_NET any -> any 53 (msg:"INDICATOR-COMPROMISE Suspicious .
    pw dns query"; flow:to_server; content:"|01 00 00 01 00 00 00 00 00 00
    00|"; depth:10; offset:2; content:"|02|pw|00|"; distance:0;
    fast_pattern; metadata:policy balanced-ips alert, policy security-ips
    drop, service dns; classtype:trojan-activity; sid:28039; rev:4;)
```

- **Encabezado de la regla:**
  Esta regla es una alerta y establece que el protocolo que va a observar va a ser UDP. El origen de la comunicación es cualquier dirección IP que pertenezca al segmento 172.16.0.0/12. Los puertos de origen de la comunicación pueden ser cualquiera.

  El destino de la comunicación es cualquier dirección IP y el puerto destino es el 53.

- **Opciones de la regla:**
  El argumento de la opción *msg* (INDICATOR-COMPROMISE Suspicious .pw dns query), mostrará este mensaje cuando dicha alerta haya sido disparada, permitiendo identificarla con mayor facilidad.

  El argumento de la opción *flow* (to_server), indican que se aplique la regla solo a un sentido de la comunicación y que sean peticiones hacia un servidor, en este caso un DNS.

  El argumento de la opción *content* (|01 00 00 01 00 00 00 00 00 00|), junto con el argumento de la opción depth (10) y el argumento de la opción offset (2); establece la búsqueda en el payload del paquete después de los primeros 2 bytes. Con la opción depth se le indica a Snort que busque en los siguientes 10 bytes el contenido "|01 00 00 01 00 00 00 00 00 00|".

  El argumento de la opción *content* (|02|pw|00|) junto con la opción distance y su respectivo argumento (0); establece que debe empezar a buscar el contenido "|02|pw|00|" a partir del byte 0.

  La opción *fast_pattern*, establece que debe usar el contenido por el Fast Pattern Matcher, para que no se evalúe como la opción "content" de la regla.

  Los argumentos de la opción *metadata* (policy balanced-ips alert, policy security-ips drop, service dns) establecen información adicional acerca de la regla, sin que

afecte los patrones de búsqueda. Los argumentos policy balanced-ips alert y policy security-ips drop; se emplean en el modo inline de Snort. El tercer argumento identifica a la regla como un servicio DNS.

El argumento de la opción *classtype* (trojan-activity), clasifica e identifica el tipo de ataque que contiene el paquete.

El argumento de la opción *sid* (28039), establece el identificador único para esta regla en específico.

El argumento de la opción *rev* (4), indica que es la versión 4 dicha regla.

Snort Alert [1:31683:1]

La regla que generó al evento "Snort Alert [1:31683:1]" se ubica en el fichero */etc/snort/rules/malware-cnc.rules*. El contenido de la regla es el siguiente:

```
alert tcp $HOME_NET any -> $EXTERNAL_NET $HTTP_PORTS (msg:"MALWARE-CNC Win
    .Trojan.Badur variant outbound connection"; flow:to_server,established
    ; content:"/get/?data="; depth:11; http_uri; content:"User-Agent:
    win32|0D 0A|"; fast_pattern:only; http_header; metadata:impact_flag
    red, policy balanced-ips drop, policy security-ips drop, ruleset
    community, service http; reference:url,www.virustotal.com/en/file/840
    b3b76030696b1ce9eccd5ee6d55dd79c0120871094cb9266769c09f03029c/analysis
    /; classtype:trojan-activity; sid:31683; rev:1;)
```

- **Encabezado de la regla:**
  Esta regla es una alerta y establece que el protocolo que va a observar va a ser TCP. El origen de la comunicación es cualquier dirección IP que pertenezca al segmento de la intranet. Cualquier puerto puede ser el origen.

  El destino de la comunicación es cualquier dirección IP que sea diferente al segmento de la intranet y los puertos de destino son los puertos HTTP que se mencionaron en la alerta 28806.

- **Opciones de la regla:**
  La opción *msg* con su respectivo argumento (MALWARE-CNC Win.Trojan.Badur variant outbound connection), informa al administrador el motivo por el cual esa alerta ha sido disparada. Además permite identificar de una forma más sencilla esa alerta.

Los argumentos de la opción *flow* (to_server, established), indican que se aplique la regla sólo a un sentido de la comunicación y que sean peticiones hacia un servidor. El argumento established indica que deben existir conexiones TCP establecidas.

El argumento de la opción *content* (/get/?data=), junto con el argumento de la opción depth (11) y la opción http_uri; establece la búsqueda del contenido "/get/?data=" en los primeros 11 bytes de la URI normalizada.

El argumento de la opción *content* (User-Agent: win32|0D 0A|), el argumento de la opción *fast_pattern* (only) y la opción *http_header*; establecen que Snort realice la búsqueda de "User-Agent: win32|0D 0A|" en las cabeceras de las solicitudes HTTP. Además, el contenido es usado por el Fast Pattern Matcher, para que no se evalúe como la opción "content" de la regla. Esto es de gran ayuda ya que permite realizar un menor esfuerzo al momento de evaluar la regla, ya que permite ubicar el contenido en el payload, independientemente de su ubicación en el flujo de la comunicación.

Los argumentos de la opción *metadata* (impact_flag red, policy balanced-ips drop, policy security-ips drop, ruleset community, service http) proporcionan información adicional, sin que interfiera en las demás opciones establecidas. Los primeros tres argumentos se emplean en el modo inline de Snort. El cuarto argumento proporciona información acerca de que esta regla pertenece al grupo de reglas de la comunidad de Snort. El último argumento identifica a la regla como un servicio HTTP.

El argumento de la opción *reference* (url), establece una referencia externa, de sistemas de identificación externos, respecto a esta actividad maliciosa.

El argumento de la opción *classtype* (trojan-activity), clasifica e identifica el tipo de ataque que contiene el paquete.

El argumento de la opción *sid* (31683), establece el identificador único para esta regla en específico.

El argumento de la opción *rev* (1), indica la versión actual de la regla.

Snort Alert [1:28801:2]

La regla que generó al evento "Snort Alert [1:28801:2]", se ubica en el fichero */etc/snort/rules/local.rules*. El contenido de la regla es el siguiente:

```
alert tcp $HOME_NET any -> $EXTERNAL_NET $HTTP_PORTS (msg:"DELETED MALWARE
    -CNC Win.Trojan.Bancos outbound connection attempt"; flow:to_server,
    established; content:".exe HTTP/1.1|0D 0A|Accept: */*|0D 0A|Accept-
    Encoding: gzip, deflate|0D 0A|User-Agent: "; fast_pattern:only;
    content:"|3B| MSIE "; http_header; content:!"Accept-Language:";
    http_header; reference:url,www.virustotal.com/en/file/26
    c60976776d212aefc9863efde914059dd2847291084c158ce51655fc1e48d0/
    analysis/1382620137/; classtype:trojan-activity; sid:28801; rev:2;)
```

- **Encabezado de la regla:**
  Esta regla es una alerta y establece que el protocolo que va a observar va a ser TCP. El origen de la comunicación es cualquier dirección IP que pertenezca al segmento de la intranet. Cualquier puerto puede ser el origen.

  El destino de la comunicación es cualquier dirección IP que sea diferente al segmento de la intranet y los puertos de destino son los puertos HTTP que se mencionaron en la alerta 28806.

- **Opciones de la regla:**
  La opción *msg* con su respectivo argumento (DELETED MALWARE-CNC Win.Trojan.Bancos outbound connection attempt); informa al administrador el motivo por el cual esa alerta ha sido disparada. Además permite identificar de una forma más sencilla esa alerta.

  Los argumentos de la opción *flow* (to_server, established), indican que se aplique la regla solo a un sentido de la comunicación y que sean peticiones hacia un servidor. El argumento established indica que deben existir conexiones TCP establecidas.

  El argumento de la opción *content* (.exe HTTP/1.1|0D 0A|Accept: */*|0D 0A| Accept-Encoding: gzip, deflate|0D 0A|User-Agent:) junto con la opción *fast_pattern* y su argumento (only); establece que debe usar el contenido anteriormente mostrado en el argumento de la opción content, por el Fast Pattern Matcher, para que no se evalúe como la opción "content" de la regla.

  La opción *content* con su respectivo argumento y la opción *http_header*, le indica a Snort que debe de realizar la búsqueda de "|3B| MSIE" en la cabecera de las solicitudes HTTP.

La opción *content* con su argumento (Accept-Language:) y la opción *http_header*, le indica a Snort que debe realizar la búsqueda de una cadena diferente a la del argumento de la opción *content*, en los encabezados de las solicitudes HTTP.

El argumento de la opción *reference* (url), establece una referencia externa, de sistemas de identificación externos, respecto a esta actividad maliciosa.

El argumento de la opción *classtype* (trojan-activity), clasifica e identifica el tipo de ataque que contiene el paquete.

El argumento de la opción *sid* (28801), establece el identificador único para esta regla en específico. Así, el identificador de esta regla es el 28801.

El argumento de la opción *rev* (2), indica que la versión actual de la regla es la 2.

Snort Alert [1:28423:1]

La regla que generó al evento "Snort Alert [1:28423:1]", se ubica en el fichero */etc/snort/rules/exploit-kit.rules*. El contenido de la regla es el siguiente:

```
alert tcp $EXTERNAL_NET $HTTP_PORTS -> $HOME_NET any (msg:"EXPLOIT-KIT
    Multiple exploit kit single digit exe detection"; flow:to_client,
    established; content:"filename="; http_header; content:".exe"; within
    :6; fast_pattern; http_header; pcre:"/filename=[\x22\x27]?\d\.exe[\x22
    \x27]?/Hi"; metadata:policy balanced-ips drop, policy security-ips
    drop, service http; classtype:trojan-activity; sid:28423; rev:1;)
```

- **Encabezado de la regla:**
  Esta regla es una alerta y establece que el protocolo que va a observar va a ser TCP. El origen de la comunicación es cualquier dirección IP externa. El puerto de origen son los puertos HTTP que se mostraron en la alerta 28806.

  El destino de la comunicación es cualquier dirección IP que pertenezca al segmento de la intranet y los puertos de destino pueden ser cualquiera.

- **Opciones de la regla:**
  La opción *msg* junto con su argumento (EXPLOIT-KIT Multiple exploit kit single digit exe detection), aporta información acerca de la actividad maliciosa que activó esa alerta.

  Los argumentos de la opción *flow* (to_client, established), indican que se aplique la regla solo a un sentido de la comunicación y que sean peticiones hacia un cliente.

El argumento established indica que deben existir conexiones TCP establecidas.

La opción *content* con su respectivo argumento y la opción *http_header*, le indica a Snort que debe de realizar la búsqueda de "filename=" en la cabecera de las solicitudes HTTP.

La opción *content* con el argumento (.exe), junto con las opciones *within* y el argumento (6), la opción *fast_pattern* y *http_header*, establece que Snort debe de inspeccionar en las cabeceras de las solicitudes HTTP el contenido ".exe" a partir de 6 bytes desde que encontró la última coincidencia. Además se le indica que ese contenido es usado por el Fast Pattern Matcher.

La opción *pcre* y el argumento (/filename=[\x22\x27]?\d\.exe[\x22\x27]?/Hi), sirve para indicar el uso de una expresión regular. Esta expresión regular establece que el contenido a buscar debe de contener la cadena "filename=", seguido de una cadena compuesta por cualquiera de las dos referencias hexadecimales. El signo de interrogacón, indica que no puede o que puede coincidir la cadena solo una vez; pero no más. Posteriormente se establece que la coincidencia va ir proseguido de cualquier número del 0 al 9. Después debe de existir la palabra ".exe", seguido a una cadena compuesta por cualquiera de las dos referencias hexadecimales. Nuevamente se establece que no puede existir ninguna cadena o solo una. A continuación los delimitadores "Hi". El delimitador "H" establece una representación en hexadecimal y la "i" que distinga entre mayúsculas y minúsculas.

Los argumentos de la opción *metadata* (policy balanced-ips drop, policy security-ips drop, service http) proporcionan información adicional, sin que interfiera en las demás opciones establecidas. Los primeros dos argumentos se emplean en el modo inline de Snort. El tercer argumento identifca a la regla como un servicio HTTP.

El argumento de la opción *classtype* (trojan-activity), clasifica e identifica el tipo de ataque que contiene el paquete.

El argumento de la opción *sid* (28423), establece el identificador único para esta regla en específico.

El argumento de la opción *rev* (1), indica que la versión actual de la regla es la 1.

Snort Alert [1:27919:3]

La regla que generó al evento "Snort Alert [1:27919:3]", se ubica en el fichero */etc/snort/rules/malware-cnc.rules*. El contenido de la regla es el siguiente:

```
alert tcp $HOME_NET any -> $EXTERNAL_NET $HTTP_PORTS (msg:"MALWARE-CNC Win
    .Trojan.Zeus encrypted POST Data exfiltration"; flow:to_server,
    established; content:"Accept-Encoding|3A| identity, *|3B|q=0|0D 0A|";
    fast_pattern:only; http_header; content:"|3B| MSIE "; http_header;
    pcre:"/[^ -~\r\n]{4}/P"; metadata:impact_flag red, policy balanced-ips
     drop, policy security-ips drop, ruleset community, service http;
    reference:url,www.virustotal.com/en/file/8825
    abfca1a6d843ce5670858886cb63bb1317ddbb92f91ffd46cfdcaba9ac00/analysis
    /; classtype:trojan-activity; sid:27919; rev:3;)
```

- **Encabezado de la regla:**
  Esta regla es una alerta y establece que el protocolo que va a observar va a ser TCP. El origen de la comunicación es cualquier dirección IP que pertenezca a la intranet. El puerto de origen puede ser cualquiera.

  El destino de la comunicación es cualquier dirección IP que sea diferente al segmento de la intranet y los puertos HTTP de destino pueden ser los que se mostraron en la alerta 28806.

- **Opciones de la regla:**
  La opción *msg* junto con su argumento (MALWARE-CNC Win.Trojan.Zeus encrypted POST Data exfiltration), aporta información acerca de la actividad maliciosa que activó esta alerta.

  Los argumentos de la opción *flow* (to_server, established), indican que se aplique la regla solo a un sentido de la comunicación y que sean peticiones hacia un servidor. El argumento established indica que deben existir conexiones TCP establecidas.

  El argumento de la opción *content* (Accept-Encoding|3A| identity, *|3B|q=0|0D 0A|), el argumento de la opción *fast_pattern* (only) y la opción *http_header*; establecen que Snort realice la búsqueda de "Accept-Encoding|3A| identity, *|3B|q=0|0D 0A|" en las cabeceras de las solicitudes HTTP. Además, el contenido es usado por el Fast Pattern Matcher, para que no se evalúe como la opción "content" de la regla.

  El argumento de la opción *content* (|3B| MSIE), junto con la opción *http_header* indica que Snort debe de realizar la búsqueda de dicho contenido en los encabezados de las solicitudes HTTP.

La opción *pcre* y el argumento (/[^ -~\r\n]{4}/P), sirve para indicar el uso de una expresión regular. Esta expresión regular establece que coincida cualquier carácter que no se encuentre dentro del rango "espacio en blanco" hasta "~" (código de caracteres del 32 al 126), y que no sea un retorno de carro y un salto de línea. Además debe de existir una repetición de 4 veces de la cadena anterior que no contenga los caracteres especificados.

Los argumentos de la opción *metadata* (impact_flag red, policy balanced-ips drop, policy security-ips drop, ruleset community, service http) proporcionan información adicional, sin que interfiera en las demás opciones establecidas. Los primeros tres argumentos se emplean en el modo inline de Snort. El cuarto argumento proporciona información acerca de que esta regla pertenece al grupo de reglas de la comunidad de Snort. El quinto argumento es un identificador propio de la regla en la que establece el tipo de servicio que Snort va a observar.

El argumento de la opción *reference* (url), establece una referencia externa, de sistemas de identificación externos, respecto a esta actividad maliciosa.

El argumento de la opción *classtype* (trojan-activity), clasifica e identifica el tipo de ataque que contiene el paquete.

El argumento de la opción *sid* (27919), establece el identificador único para esta regla en específico.

El argumento de la opción *rev* (3), indica que la versión actual de la regla es la 3.

Snort Alert [1:32125:1]

La regla que generó al evento "Snort Alert [1:32125:1]", se ubica en el fichero */etc/snort/rules/blacklist.rules*. El contenido de la regla es el siguiente:

```
alert tcp $HOME_NET any -> $EXTERNAL_NET any (msg:"BLACKLIST User-Agent
    known malicious user-agent string - update - Win.Backdoor.Upatre";
    flow:to_server,established; content:"User-Agent: update|0D 0A|";
    fast_pattern:only; metadata:impact_flag red, policy balanced-ips drop,
     policy security-ips drop, service http; reference:url,www.virustotal.
    com/en/file/8
    f98fce6c20dbbe8a156e5a5b671066ccd0db240140e81d69d1a7205457605cb/
    analysis/; classtype:trojan-activity; sid:32125; rev:1;)
```

- **Encabezado de la regla:**
  Esta regla es una alerta y establece que el protocolo que va a observar va a ser

TCP. El origen de la comunicación es cualquier dirección IP que pertenezca a la intranet. El puerto de origen puede ser cualquiera.

El destino de la comunicación es cualquier dirección IP que sea diferente al segmento de la intranet y el puerto de destino puede ser cualquiera.

- **Opciones de la regla:**
  La opción *msg* junto con su argumento (BLACKLIST User-Agent known malicious user-agent string - update - Win.Backdoor.Upatre), aporta información acerca de la actividad maliciosa que activó esta alerta.

  Los argumentos de la opción *flow* (to_server, established), indican que se aplique la regla solo a un sentido de comunicación y que sean peticiones hacia un servidor. El argumento established indica que deben existir conexiones TCP establecidas.

  El argumento de la opción *content* (User-Agent: update|0D 0A|), establece que Snort busque "User-Agent: update|0D 0A|" en el payload del paquete, el cual es la extensión de un archivo ejecutable. El argumento de la opción *fast_pattern* (only), establece que debe usar el contenido por el Fast Pattern Matcher, para que no se evalúe como la opción "content" de la regla.

  Los argumentos de la opción *metadata* (impact_flag red, policy balanced-ips drop, policy security-ips drop, service http) proporcionan información adicional, sin que interfiera en las demás opciones establecidas. Los argumentos impact_flag red, policy balanced-ips drop y policy security-ips drop se emplean cuando el IDS se configura en modo inline. El último argumento identifica que el tipo de servicio que Snort monitoreará será HTTP.

  El argumento de la opción *reference* (url), establece una referencia externa, de sistemas de identificación externos, respecto a esta actividad maliciosa.

  El argumento de la opción *classtype* (trojan-activity), clasifica e identifica el tipo de ataque que contiene el paquete.

  El argumento de la opción *sid* (32125), establece el identificador único para esta regla en específico. El identificador de esta regla es el 32125.

  El argumento de la opción *rev* (1), indica que la versión actual de la regla es la 1.

Snort Alert [1:31527:1]

La regla que generó al evento "Snort Alert [1:31527:1]", se ubica en el fichero */etc/snort/rules/malware-cnc.rules*. El contenido de la regla es el siguiente:

```
alert tcp $HOME_NET any -> $EXTERNAL_NET [443,446,447] (msg:"MALWARE-CNC
    Win.Trojan.Ramnit variant outbound detected"; flow:to_server,
    established; dsize:6; content:"|00 FF 01 00 00 00|"; fast_pattern:only
    ; metadata:policy balanced-ips drop, policy security-ips drop, service
     ssl; reference:url,www.virustotal.com/en/file/83
    F75C8D52B84795A526CA7DAEA29186CDC2CDD4A33871A942BB00D673BB0E20/
    analysis/; classtype:trojan-activity; sid:31527; rev:1;)
```

- **Encabezado de la regla:**
  Esta regla es una alerta y establece que el protocolo que va a observar va a ser TCP. El origen de la comunicación es cualquier dirección IP que pertenezca a la intranet. El puerto de origen puede ser cualquiera.

  El destino de la comunicación es cualquier dirección IP que sea diferente al segmento de la intranet y el puerto de destino puede ser el 443, el 446 o el 447.

- **Opciones de la regla:**
  La opción *msg* junto con su argumento (MALWARE-CNC Win.Trojan.Ramnit variant outbound detected), aporta información acerca de la actividad maliciosa que activó esta alerta.

  Los argumentos de la opción *flow* (to_server, established), indican que se aplique la regla solo a un sentido de comunicación y que sean peticiones hacia un servidor. El argumento established indica que deben existir conexiones TCP establecidas.

  El argumento de la opción *dsize* (6), establece el tamaño del payload. Esta opción se emplea para verificar tamaños anómalos en el paquete.

  El argumento de la opción *content* (|00 FF 01 00 00 00|) y el argumento de la opción *fast_pattern* (only), establecen que el contenido "|00 FF 01 00 00 00|" es usado por el Fast Pattern Matcher, para que no se evalúe como la opción "content" de la regla.

  Los argumentos de la opción *metadata* (policy balanced-ips drop, policy security-ips drop, service ssl) proporcionan información adicional, sin que interfiera en las demás opciones establecidas. Los primeros dos argumentos se emplean cuando el IDS está en modo inline. El último argumento establece el identificador de servicio que la regla va a estar monitoreando.

El argumento de la opción *reference* (url), establece una referencia externa, de sistemas de identificación externos, respecto a esta actividad maliciosa.

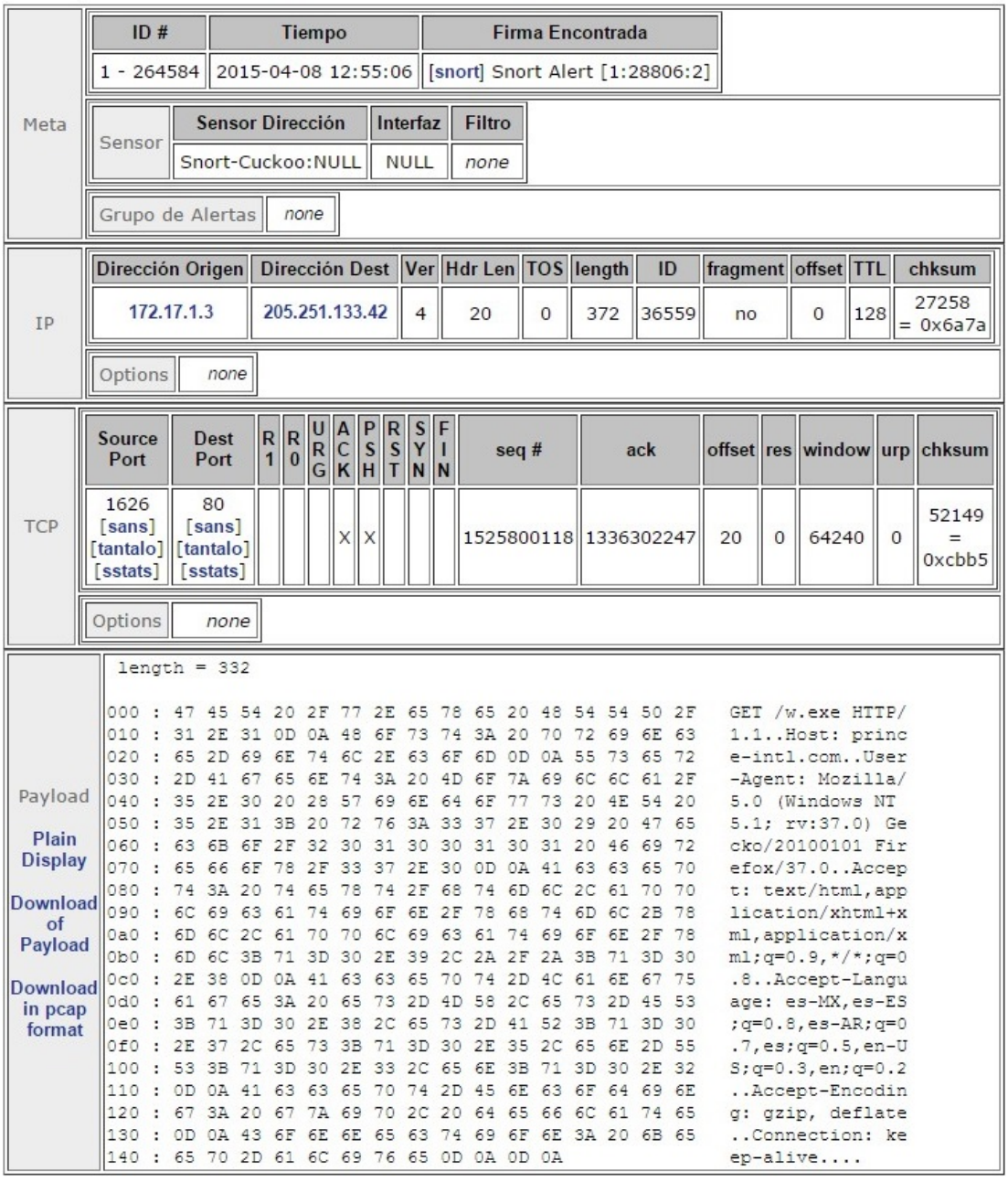
El argumento de la opción *classtype* (trojan-activity), clasifica e identifica el tipo de ataque que contiene el paquete.

El argumento de la opción *sid* (31527), establece el identificador único para esta regla en específico. El identificador de esta regla es el 31527.

El argumento de la opción *rev* (1), indica que la versión actual de la regla es la 1.

3.3.2. Contenido y análisis del payload de los paquetes capturados

Snort Alert [1:28806:2]



Meta

ID #	Tiempo	Firma Encontrada
1 - 264584	2015-04-08 12:55:06	[snort] Snort Alert [1:28806:2]

Sensor

Sensor Dirección	Interfaz	Filtro
Snort-Cuckoo:NULL	NULL	*none*

Grupo de Alertas: *none*

IP

Dirección Origen	Dirección Dest	Ver	Hdr Len	TOS	length	ID	fragment	offset	TTL	chksum
172.17.1.3	205.251.133.42	4	20	0	372	36559	no	0	128	27258 = 0x6a7a

Options: *none*

TCP

Source Port	Dest Port	R 1	R 0	URG	ACK	PSH	RST	SYN	FIN	seq #	ack	offset	res	window	urp	chksum
1626 [sans] [tantalo] [sstats]	80 [sans] [tantalo] [sstats]				X	X				1525800118	1336302247	20	0	64240	0	52149 = 0xcbb5

Options: *none*

Payload

Plain Display

Download of Payload

Download in pcap format

```
 length = 332

000 : 47 45 54 20 2F 77 2E 65 78 65 20 48 54 54 50 2F   GET /w.exe HTTP/
010 : 31 2E 31 0D 0A 48 6F 73 74 3A 20 70 72 69 6E 63   1.1..Host: princ
020 : 65 2D 69 6E 74 6C 2E 63 6F 6D 0D 0A 55 73 65 72   e-intl.com..User
030 : 2D 41 67 65 6E 74 3A 20 4D 6F 7A 69 6C 6C 61 2F   -Agent: Mozilla/
040 : 35 2E 30 20 28 57 69 6E 64 6F 77 73 20 4E 54 20   5.0 (Windows NT
050 : 35 2E 31 3B 20 72 76 3A 33 37 2E 30 29 20 47 65   5.1; rv:37.0) Ge
060 : 63 6B 6F 2F 32 30 31 30 30 31 30 31 20 46 69 72   cko/20100101 Fir
070 : 65 66 6F 78 2F 33 37 2E 30 0D 0A 41 63 63 65 70   efox/37.0..Accep
080 : 74 3A 20 74 65 78 74 2F 68 74 6D 6C 2C 61 70 70   t: text/html,app
090 : 6C 69 63 61 74 69 6F 6E 2F 78 68 74 6D 6C 2B 78   lication/xhtml+x
0a0 : 6D 6C 2C 61 70 70 6C 69 63 61 74 69 6F 6E 2F 78   ml,application/x
0b0 : 6D 6C 3B 71 3D 30 2E 39 2C 2A 2F 2A 3B 71 3D 30   ml;q=0.9,*/*;q=0
0c0 : 2E 38 0D 0A 41 63 63 65 70 74 2D 4C 61 6E 67 75   .8..Accept-Langu
0d0 : 61 67 65 3A 20 65 73 2D 4D 58 2C 65 73 2D 45 53   age: es-MX,es-ES
0e0 : 3B 71 3D 30 2E 38 2C 65 73 2D 41 52 3B 71 3D 30   ;q=0.8,es-AR;q=0
0f0 : 2E 37 2C 65 73 3B 71 3D 30 2E 35 2C 65 6E 2D 55   .7,es;q=0.5,en-U
100 : 53 3B 71 3D 30 2E 33 2C 65 6E 3B 71 3D 30 2E 32   S;q=0.3,en;q=0.2
110 : 0D 0A 41 63 63 65 70 74 2D 45 6E 63 6F 64 69 6E   ..Accept-Encodin
120 : 67 3A 20 67 7A 69 70 2C 20 64 65 66 6C 61 74 65   g: gzip, deflate
130 : 0D 0A 43 6F 6E 6E 65 63 74 69 6F 6E 3A 20 6B 65   ..Connection: ke
140 : 65 70 2D 61 6C 69 76 65 0D 0A 0D 0A               ep-alive....
```

Figura 3.12: Paquete de la firma 28806. Fuente: Captura propia.

En el paquete mostrado en la figura 3.12, se muestran los encabezados meta, IP,

TCP, así como el payload del paquete que fue capturado. En el encabezado metadata se establece la fecha (2015-04-08) y hora (12:55:06) en la que fue emitida la alerta, así como al identificador del evento (1-264584) que generó dicha regla.

En la cabecera IP y de acuerdo a la regla establecida; se comprueba que existe una conexión TCP desde una dirección local (172.17.1.3) hacia una dirección IP remota (205.251.133.42). En la cabecera TCP se muestra el puerto de origen (1626) y destino (80).

En el payload del paquete se validan las búsquedas de contenido específico que se describieron en el cuerpo de la regla, el cual encontró que la longitud de la URI es de 6 bytes, así como el contenido específico de búsqueda (/w.exe). En dicho payload se muestra el método GET del protocolo HTTP para la descarga de un archivo ejecutable.

Snort Alert [1:30211:1]

Meta

ID #	Tiempo	Firma Encontrada
1 - 36902	2015-04-04 12:29:32	[snort] Snort Alert [1:30211:1]

Sensor	Sensor Dirección	Interfaz	Filtro
	Snort-Cuckoo:NULL	NULL	*none*

Grupo de Alertas	*none*

IP

Dirección Origen	Dirección Dest	Ver	Hdr Len	TOS	length	ID	fragment	offset	TTL	chksum
195.238.181.21	172.17.1.44	4	20	0	1500	42726	no	0	128	26612 = 0x67f4

Options	*none*

TCP

Source Port	Dest Port	R 1	R 0	U R G	A C K	P S H	R S T	S Y N	F I N	seq #	ack	offset	res	window	urp	chksum
80 [sans] [tantalo] [sstats]	2492 [sans] [tantalo] [sstats]				X					1353060776	4019464097	20	0	64240	0	1125 = 0x465

Options	*none*

Payload

Plain Display

Download of Payload

Download in pcap format

```
length = 1460

000 : 48 54 54 50 2F 31 2E 31 20 32 30 30 20 4F 4B 0D   HTTP/1.1 200 OK.
010 : 0A 53 65 72 76 65 72 3A 20 6E 67 69 6E 78 2F 31   .Server: nginx/1
020 : 2E 31 2E 31 39 0D 0A 44 61 74 65 3A 20 53 61 74   .1.19..Date: Sat
030 : 2C 20 30 34 20 41 70 72 20 32 30 31 35 20 31 38   , 04 Apr 2015 18
040 : 3A 32 39 3A 31 38 20 47 4D 54 0D 0A 43 6F 6E 74   :29:18 GMT..Cont
050 : 65 6E 74 2D 54 79 70 65 3A 20 69 6D 61 67 65 2F   ent-Type: image/
060 : 6A 70 65 67 0D 0A 43 6F 6E 74 65 6E 74 2D 4C 65   jpeg..Content-Le
070 : 6E 67 74 68 3A 20 38 31 37 35 32 0D 0A 4C 61 73   ngth: 81752..Las
080 : 74 2D 4D 6F 64 69 66 69 65 64 3A 20 54 75 65 2C   t-Modified: Tue,
090 : 20 31 30 20 4D 61 72 20 32 30 31 35 20 31 36 3A    10 Mar 2015 16:
0a0 : 33 38 3A 33 38 20 47 4D 54 0D 0A 43 6F 6E 6E 65   38:38 GMT..Conne
0b0 : 63 74 69 6F 6E 3A 20 63 6C 6F 73 65 0D 0A 41 63   ction: close..Ac
0c0 : 63 65 70 74 2D 52 61 6E 67 65 73 3A 20 62 79 74   cept-Ranges: byt
0d0 : 65 73 0D 0A 0D 0A FF D8 FF E0 00 10 4A 46 49 46   es..........JFIF
0e0 : 00 01 02 00 00 64 00 64 00 00 FF EC 00 11 44 75   .....d.d......Du
0f0 : 63 6B 79 00 01 00 04 00 00 00 3C 00 00 FF EE 00   cky.......<.....
100 : 0E 41 64 6F 62 65 00 64 C0 00 00 00 01 FF DB 00   .Adobe.d........
110 : 84 00 06 04 04 04 05 04 06 05 05 06 09 06 05 06   ................
120 : 09 0B 08 06 06 08 0B 0C 0A 0A 0B 0A 0A 0C 10 0C   ................
130 : 0C 0C 0C 0C 0C 10 0C 0E 0F 10 0F 0E 0C 13 13 14   ................
140 : 14 13 13 1C 1B 1B 1B 1C 1F 1F 1F 1F 1F 1F 1F 1F   ................
150 : 1F 1F 01 07 07 07 0D 0C 0D 18 10 10 18 1A 15 11   ................
160 : 15 1A 1F 1F 1F 1F 1F 1F 1F 1F 1F 1F 1F 1F 1F 1F   ................
170 : 1F 1F 1F 1F 1F 1F 1F 1F 1F 1F 1F 1F 1F 1F 1F 1F   ................
180 : 1F 1F 1F 1F 1F 1F 1F 1F 1F 1F 1F 1F 1F 1F 1F 1F   ................
190 : 1F 1F 1F FF C0 00 11 08 00 28 00 28 03 01 11 00   .........(.(....
1a0 : 02 11 01 03 11 01 FF C4 00 92 00 01 01 00 03 01   ................
1b0 : 00 00 00 00 00 00 00 00 00 00 00 06 07 01 04 05   ................
1c0 : 02 01 01 01 01 01 01 01 00 00 00 00 00 00 00 00   ................
1d0 : 00 00 04 03 05 01 02 00 10 00 02 01 02 05 01 02   ................
1e0 : 0B 06 07 00 00 00 00 00 00 01 02 03 04 05 00 11   ................
1f0 : 12 06 07 13 21 31 41 51 71 81 A1 22 32 52 92 D2   ....!1AQq.."2R..
200 : 14 82 23 33 63 93 B3 A3 C3 64 74 45 16 08 11 00   ..#3c....dtE....
210 : 01 03 02 02 08 04 07 00 00 00 00 00 00 00 00 01   ................
220 : 00 03 04 11 02 12 05 F0 21 41 71 81 D1 32 23 51   ........!Aq..2#Q
230 : 91 C1 13 31 61 A1 E1 F1 22 14 FF DA 00 0C 03 01   ...1a..."....... 
240 : 00 02 11 03 11 00 3F 00 AF 6F FD D9 3A 56 CD 42   ......?..o..:V.B
250 : 92 32 53 40 C2 23 12 12 86 49 34 86 62 CC 3B 74   .2S@.#...I4.b.;t
260 : AE A0 32 C6 9C 38 E0 8C 45 11 F7 48 34 43 7F D9   ..2..8..E..H4C.
270 : AB 7D E6 FD 59 BE 7C 3B F9 ED F0 FA 04 6F 70 AC   .}..Y.|;.....op.
280 : 8D C9 5C C4 2A B3 96 27 20 04 93 12 49 F0 0F 5F   ..\.*..' ...I.._
290 : 1F 7B 16 78 0F 20 BE F7 0A 45 4F 62 DF B3 C6 24   .{.x. ...EOb...$
2a0 : 5A 19 11 4F 68 12 54 32 37 C2 D2 E6 30 62 E3 03   Z..Oh.T27...0b..
2b0 : 68 F2 FB 2B 0B 1C 3F 95 BD B1 F7 4D 52 5C A3 B7   h..+..?....MR\..
2c0 : 4C EC 60 92 43 4F 24 0E DA FA 72 F6 E9 28 C7 B7   L.`.CO$...r..(..
2d0 : 49 2A 41 18 94 B8 F6 E1 C4 17 B6 5D 35 A1 41 F9   I*A........]5.A.
2e0 : 2A F3 47 06 EA AF 82 49 3A 6E B5 2E 3D 60 72 FC   *.G....I:n..=`r.
2f0 : 28 8F 7F DA C2 E0 8E D8 D3 69 51 91 D4 8E 47 70   (.......iQ...Gp
300 : 82 41 9A 4A 8D E4 61 86 51 41 3D E3 3D BD 73 9B   .A.J..a.QA=.=.s.
310 : 70 DB EE 53 D1 48 6D 81 64 96 2A A2 3E EC B0 52   p..S.Hm.d.*.>..R
320 : 10 E7 9F BD DD 80 CD 7A D1 61 B4 1F D9 22 3D 87   .......z.a..."=.
330 : 10 34 D4 A8 9B DA EF BA E8 22 A5 5D BD 6F FA D9   .4.......".].o..
340 : 65 2E 67 72 A5 C2 2A E5 90 C8 32 F6 B6 7E 8C 66   e.gr..*...2..~.f
350 : C6 6D BB AB 8C D1 29 DB AE 1D 21 48 F6 25 E6 19   .m....)...!H.%..
360 : F7 8C 11 33 67 52 F5 D1 89 55 41 C9 5F 53 96 1E   ...3gR...UA._S..
370 : 62 0E 35 26 5B 46 B8 72 44 63 AD 1E E6 24 CF 7A   b.5&[F.rDc...$.z
380 : 5C 7F BA 7F D8 83 14 CB C7 6C 69 B4 AE 48 EA 28   \......li..H.(
390 : 4A 44 3C 58 D1 01 1A AA DF C3 1C 87 70 9A A6 D9   JD<X........p...
3a0 : B3 DA 8E 25 A4 82 09 B2 AA 0C DD 43 A3 39 3D 9F   ...%.......C.9=.
3b0 : 67 BC E3 1F 31 86 00 2E 57 5A 6C 67 89 22 D4 B3   g...1...WZlg."..
3c0 : 94 B9 1E E1 B3 5A DA 29 28 A2 AB FA E1 31 7E AB   .....Z.)(....1~.
3d0 : 32 E9 E9 68 CB 2D 3E 3D 78 1C 28 61 EA D4 D2 8A   2..h.->=x.(a....
```

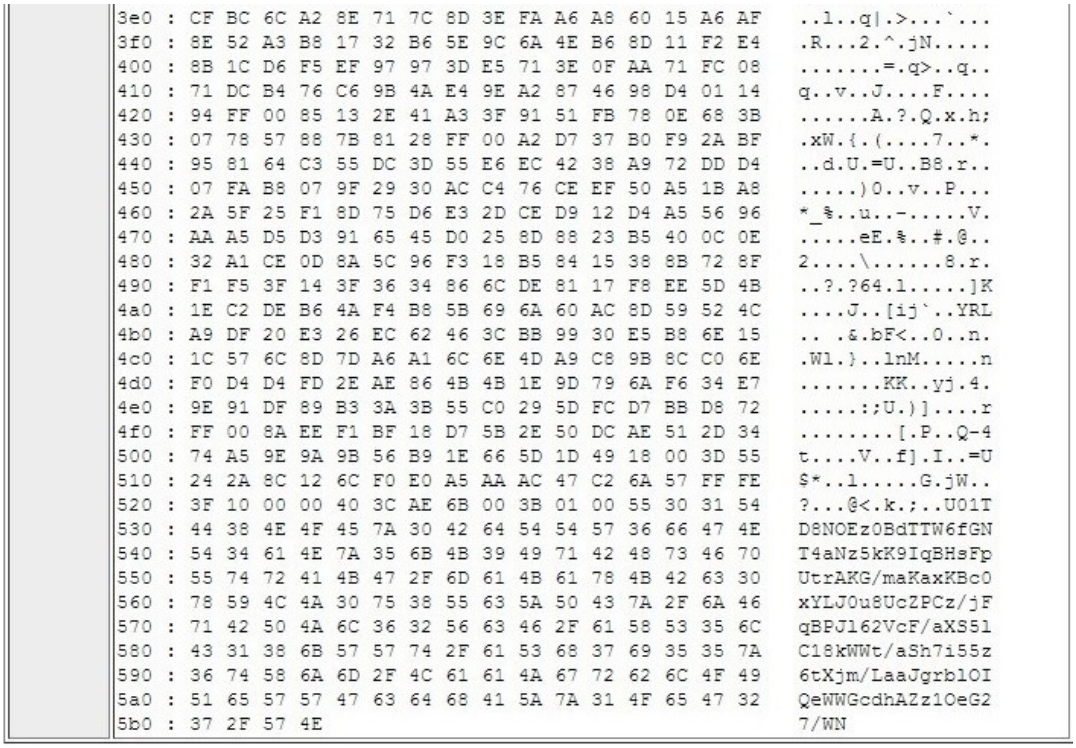


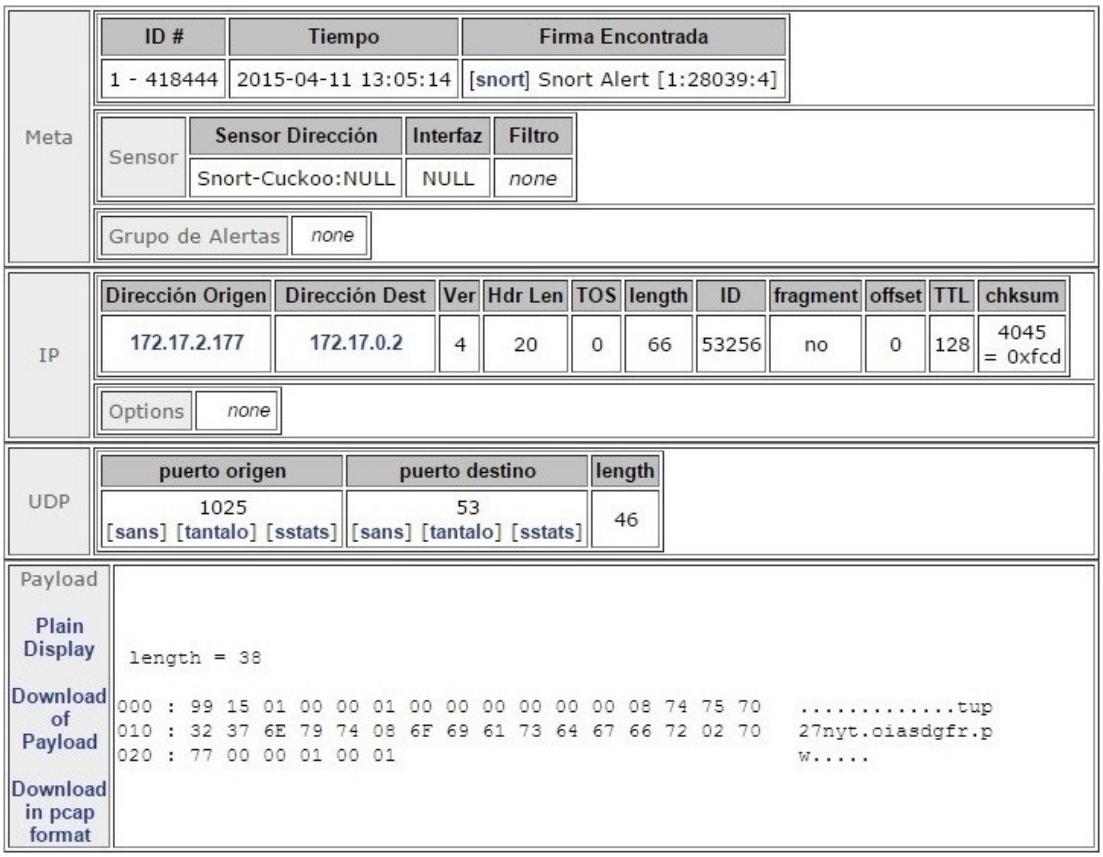
Figura 3.13: Paquete de la firma 30211. Fuente: Captura propia.

En la alerta mostrada en la figura 3.13, se muestran los encabezados metada, IP, TCP, así como el payload del paquete que fue capturado. En el encabezado metadata se establece la fecha y hora en la que fue emitida la alerta, así como al identificador del evento que generó dicha regla.

En la cabecera IP y de acuerdo a la regla correspondiente al evento 30211; se comprueba que existe una conexión TCP desde una dirección externa a la intranet (195.238.181.21) hacia una dirección IP interna (172.17.1.44). En la cabecera TCP se muestra información acerca del puerto de origen (80) y destino (2492).

En el payload del paquete se validan las búsquedas de contenido específico que se describieron en el cuerpo de la regla, el cual encontró coincidencia para el contenido y la expresión regular en la línea 550.

Snort Alert [1:28039:4]



Meta

ID #	Tiempo	Firma Encontrada
1 - 418444	2015-04-11 13:05:14	[snort] Snort Alert [1:28039:4]

Sensor

Sensor Dirección	Interfaz	Filtro
Snort-Cuckoo:NULL	NULL	*none*

Grupo de Alertas: *none*

IP

Dirección Origen	Dirección Dest	Ver	Hdr Len	TOS	length	ID	fragment	offset	TTL	chksum
172.17.2.177	172.17.0.2	4	20	0	66	53256	no	0	128	4045 = 0xfcd

Options: *none*

UDP

puerto origen	puerto destino	length
1025 [sans] [tantalo] [sstats]	53 [sans] [tantalo] [sstats]	46

Payload

Plain Display

Download of Payload

Download in pcap format

```
length = 38

000 : 99 15 01 00 00 01 00 00 00 00 00 00 08 74 75 70   .............tup
010 : 32 37 6E 79 74 08 6F 69 61 73 64 67 66 72 02 70   27nyt.oiasdgfr.p
020 : 77 00 00 01 00 01                                 w.....
```

Figura 3.14: Paquete de la firma 28039. Fuente: Captura propia.

En la alerta mostrada en la figura 3.14, es posible observar los encabezados metadata, IP, UDP y el área de datos. En la sección de metadata es posible observar la fecha y hora en que fue generado este evento.

En la cabecera IP, se muestra la dirección IP de origen y destino que tuvo lugar en esta conexión UDP. La IP de origen pertenece al segmento de la intranet, la cual es la 172.17.2.177. La dirección de destino es una dirección que también pertenece al segmento de red y es la 172.17.0.2. En esta dirección IP se encuentra el servicio DNS, el cual resuelve nombres de dominio. En el encabezado UDP se muestra el puerto de origen y destino de la comunicación, que corresponden al 1025 y 53 respectivamente.

En el payload del paquete se encuentra la petición de resolución de un nombre de dominio, el cual es el *tup27nyt.oiasdgfr.pw*. Actualmente este dominio se encuentra libre para el público en general.

Snort Alert [1:31683:1]

Meta

ID #	Tiempo	Firma Encontrada
1 - 412114	2015-04-11 12:17:15	[snort] Snort Alert [1:31683:1]

Sensor	Sensor Dirección	Interfaz	Filtro
	Snort-Cuckoo:NULL	NULL	*none*

Grupo de Alertas	*none*

IP

Dirección Origen	Dirección Dest	Ver	Hdr Len	TOS	length	ID	fragment	offset	TTL	chksum
172.17.2.177	54.213.128.72	4	20	0	747	62958	no	0	128	39998 = 0x9c3e

Options	*none*

TCP

Source Port	Dest Port	R 1	R 0	U R G	A C K	P S H	R S T	S Y N	F I N	seq #	ack	offset	res	window	urp	chksum
2062 [sans] [tantalo] [sstats]	80 [sans] [tantalo] [sstats]				X	X				3802762029	1722505955	20	0	64240	0	12717 = 0x31ad

Options	*none*

Payload

Plain Display

Download of Payload

```
length = 707

000 : 47 45 54 20 2F 67 65 74 2F 3F 64 61 74 61 3D 79   GET /get/?data=y
010 : 41 37 50 32 63 4B 79 6E 52 50 31 49 74 49 63 64   A7P2cKynRP1ItIcd
020 : 65 71 61 32 36 48 41 49 75 6E 50 48 4E 75 55 34   eqa26HAIunPHNuU4
030 : 50 47 76 6E 72 30 7A 6D 67 6D 65 4E 46 37 42 4F   PGvnr0zmgmeNF7BO
040 : 7A 33 48 74 64 46 34 58 70 37 65 57 34 59 33 58   z3HtdF4Xp7eW4Y3X
050 : 52 6B 31 49 55 6F 4E 38 72 7A 77 38 32 44 53 6D   Rk1IUoN8rzw82DSm
060 : 58 32 2F 67 6A 4C 69 47 4B 30 6F 6B 71 50 57 34   X2/gjLiGK0okqPW4
070 : 68 49 65 6F 68 49 66 5A 58 77 6F 76 70 6F 31 32   hIeohIfZXwovpo12
080 : 72 66 44 30 63 4E 65 79 34 7A 4B 37 51 52 4E 63   rfD0cNey4zK7QRNc
090 : 70 6B 54 69 61 65 48 6E 6C 47 75 30 4F 71 4A 6B   pkTiaeHnlGu0OqJk
0a0 : 46 53 64 62 53 51 45 78 4A 4C 71 73 4B 61 6D 36   FSdbSQExJLqsKam6
0b0 : 67 41 54 2F 33 57 52 51 59 55 52 48 4D 53 63 76   gAT/3WRQYURHMScv
0c0 : 48 53 70 61 4B 69 4D 61 47 30 78 6A 4A 76 56 59   HSpaKiMaG0xjJvVY
0d0 : 36 62 79 38 50 62 32 43 6A 62 73 49 41 46 39 45   6by8Pb2CjbsIAF9E
0e0 : 67 71 58 70 30 4C 36 48 42 63 67 53 33 4E 70 77   gqXp0L6HBcgS3Npw
0f0 : 59 73 38 52 64 38 48 66 37 6E 64 48 35 6B 6E 6C   Ys8Rd8Hf7ndH5knl
100 : 50 4B 31 7A 64 42 78 6D 79 4B 4F 38 32 46 51 41   PK1zdBxmyKO82FQA
110 : 64 30 36 69 4F 64 41 72 6D 68 64 36 2F 64 41 4B   d06iOdArmhd6/dAK
120 : 76 6F 54 57 53 6C 6A 53 4C 2F 45 38 4B 4B 68 2F   voTWSljSL/E8KKh/
130 : 35 6E 2F 55 64 33 45 35 47 71 76 6B 67 65 6A 44   5n/Ud3E5GqvkgejD
140 : 79 6F 59 4D 37 6A 57 78 44 39 4F 35 74 4C 4D 64   yoYM7jWxD9O5tLMd
150 : 54 7A 57 78 39 69 32 41 7A 5A 30 33 68 59 76 6C   TzWx9i2AzZ03hYvl
160 : 72 39 32 30 70 61 35 35 49 67 38 6D 47 66 35 31   r920pa55Ig8mGf51
170 : 44 58 6B 44 74 4F 79 55 5A 76 50 52 46 67 79 5A   DXkDtOyUZvPRFgyZ
```

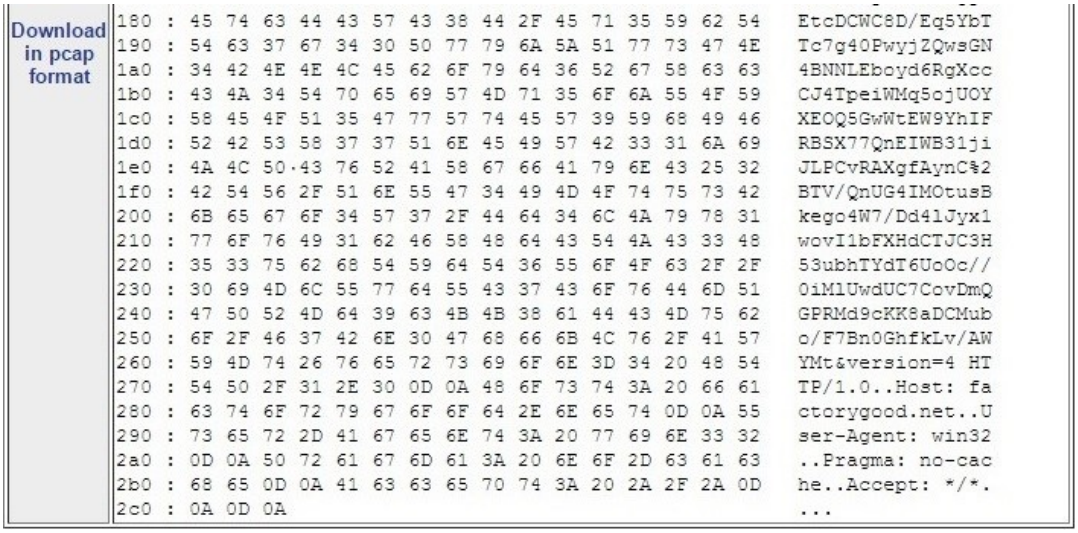


Figura 3.15: Paquete de la firma 31683. Fuente: Captura propia.

En la figura 3.15 se muestra el encabezado metadata, IP, TCP y el contenido de los datos. Este evento muestra información respecto a la fecha y hora en que esta alerta fue disparada, mostrando también el identificador de la regla que generó dicho evento.

En la cabecera IP es posible observar la dirección IP de origen y destino para esta conexión TCP. DE acuerdo a la regla para este evento, la dirección de origen tiene que ser una dirección local, que en este caso es la 172.17.2.177; mientras que la de destino tiene que ser una dirección IP que no pertenezca a la intranet (54.213.128.72). De la misma manera, la regla establece que para que esta sea disparada, es necesario que el puerto de origen sea cualquiera y el puerto de destino sea uno de los que se encuentran establecidos en el archivo de configuración de Snort. En la cabecera TCP se muestra que estos corresponden al 2062 como puerto de origen y el 80 como puerto de destino.

En el payload del paquete es posible realizar la coincidencia con el contenido "/get/?data=" establecido en la regla para su búsqueda. El match se realizó en la línea 000 dentro de los primeros 11 bytes del payload del paquete.

Snort Alert [1:28801:2]

Meta

ID #	Tiempo	Firma Encontrada
1 - 411848	2015-04-11 11:45:06	[snort] Snort Alert [1:28801:2]

Sensor	Sensor Dirección	Interfaz	Filtro
	Snort-Cuckoo:NULL	NULL	*none*

Grupo de Alertas	*none*

IP

Dirección Origen	Dirección Dest	Ver	Hdr Len	TOS	length	ID	fragment	offset	TTL	chksum
172.17.1.43	173.236.50.26	4	20	0	249	28769	no	0	128	64602 = 0xfc5a

Options	*none*

TCP

Source Port	Dest Port	R 1	R 0	URG	ACK	PSH	RST	SYN	FIN	seq #	ack	offset	res	window	urp	chksum
1305 [sans] [tantalo] [sstats]	80 [sans] [tantalo] [sstats]				X	X				2973774564	2026852656	20	0	64240	0	22273 = 0x5701

Options	*none*

Payload

Plain Display

Download of Payload

Download in pcap format

```
length = 209

000 : 47 45 54 20 2F 30 66 66 69 63 65 6B 65 79 73 65   GET /0fficekeyse
010 : 72 69 61 6C 31 35 2E 65 78 65 20 48 54 54 50 2F   rial15.exe HTTP/
020 : 31 2E 31 0D 0A 41 63 63 65 70 74 3A 20 2A 2F 2A   1.1..Accept: */*
030 : 0D 0A 41 63 63 65 70 74 2D 45 6E 63 6F 64 69 6E   ..Accept-Encodin
040 : 67 3A 20 67 7A 69 70 2C 20 64 65 66 6C 61 74 65   g: gzip, deflate
050 : 0D 0A 55 73 65 72 2D 41 67 65 6E 74 3A 20 4D 6F   ..User-Agent: Mo
060 : 7A 69 6C 6C 61 2F 34 2E 30 20 28 63 6F 6D 70 61   zilla/4.0 (compa
070 : 74 69 62 6C 65 3B 20 4D 53 49 45 20 36 2E 30 3B   tible; MSIE 6.0;
080 : 20 57 69 6E 64 6F 77 73 20 4E 54 20 35 2E 31 3B    Windows NT 5.1;
090 : 20 53 56 31 29 0D 0A 48 6F 73 74 3A 20 77 77 77    SV1)..Host: www
0a0 : 2E 62 69 74 63 6F 69 6E 73 6D 73 78 70 72 65 73   .bitcoinsmsxpres
0b0 : 73 2E 63 6F 6D 0D 0A 43 6F 6E 6E 65 63 74 69 6F   s.com..Connectio
0c0 : 6E 3A 20 4B 65 65 70 2D 41 6C 69 76 65 0D 0A 0D   n: Keep-Alive...
0d0 : 0A                                                .
```

Figura 3.16: Figura 3.16: Paquete de la firma 28801. Fuente: Captura propia.

La figura 3.16 muestra los encabezados metadata, IP, TCP y el contenido del paquete; que corresponden al evento 28801. En la cabecera metadata se encuentra la fecha y hora en que se generó dicho evento, junto con su respectivo identificador. En el encabezado IP, se puede consultar la dirección IP de origen y destino. De acuerdo a la regla establecida se indica que la dirección IP de origen pertenece al segmento 172.16.0.0/12, mientras que la dirección IP de destino es cualquier dirección IP externa. Para este evento dichas direcciones IP son la 172.17.1.43 y la 173.236.50.26 respectivamente. En la cabecera TCP se muestran los puertos de origen y destino, los cuales corresponden

al 1305 y al 80 respectivamente.

En el payload del paquete se validan las búsquedas de contenido específico que se describieron en el cuerpo de la regla, el cual establece que debe de buscar en el payload del paquete, la extensión de un archivo ejecutable, seguida de un String de 68 bytes de longitud. Esta coincidencia se puede encontrar a partir de la línea 010. También existe una segunda coincidencia de un string que contenga el byte "3B" o la cadena "MSIE" que se encuentra a partir del renglón 070.

Snort Alert [1:28423:1]

Meta

ID #	Tiempo	Firma Encontrada
1 - 320513	2015-04-06 20:26:00	[snort] Snort Alert [1:28423:1]

Sensor	Sensor Dirección	Interfaz	Filtro
	Snort-Cuckoo:NULL	NULL	*none*

Grupo de Alertas	*none*

IP

Dirección Origen	Dirección Dest	Ver	Hdr Len	TOS	length	ID	fragment	offset	TTL	chksum
116.255.152.168	172.17.2.5	4	20	0	1500	46809	no	0	128	49796 = 0xc284

Options	*none*

TCP

Source Port	Dest Port	R 1	R 0	URG	ACK	PSH	RST	SYN	FIN	seq #	ack	offset	res	window	urp	chksum
80 [sans] [tantalo] [sstats]	1528 [sans] [tantalo] [sstats]				X					67652414	1288751488	20	0	64240	0	45883 = 0xb33b

Options	*none*

```
length = 1460

000 : 48 54 54 50 2F 31 2E 31 20 32 30 30 20 4F 4B 0D   HTTP/1.1 200 OK.
010 : 0A 43 6F 6E 74 65 6E 74 2D 54 79 70 65 3A 20 61   .Content-Type: a
020 : 70 70 6C 69 63 61 74 69 6F 6E 2F 6F 63 74 65 74   pplication/octet
030 : 2D 73 74 72 65 61 6D 0D 0A 43 6F 6E 74 65 6E 74   -stream..Content
040 : 2D 4C 65 6E 67 74 68 3A 20 39 37 32 38 0D 0A 41   -Length: 9728..A
050 : 63 63 65 70 74 2D 52 61 6E 67 65 73 3A 20 62 79   ccept-Ranges: by
060 : 74 65 73 0D 0A 53 65 72 76 65 72 3A 20 48 46 53   tes..Server: HFS
070 : 20 32 2E 33 63 0D 0A 53 65 74 2D 43 6F 6F 6B 69    2.3c..Set-Cooki
080 : 65 3A 20 48 46 53 5F 53 49 44 3D 30 2E 36 34 38   e: HFS_SID=0.648
090 : 32 33 31 34 39 37 35 30 30 30 39 32 3B 20 70 61   231497500092; pa
0a0 : 74 68 3D 2F 3B 0D 0A 4C 61 73 74 2D 4D 6F 64 69   th=/;..Last-Modi
0b0 : 66 69 65 64 3A 20 53 61 74 2C 20 30 34 20 41 70   fied: Sat, 04 Ap
0c0 : 72 20 32 30 31 35 20 31 36 3A 34 36 3A 34 39 20   r 2015 16:46:49
0d0 : 47 4D 54 0D 0A 43 6F 6E 74 65 6E 74 2D 44 69 73   GMT..Content-Dis
0e0 : 70 6F 73 69 74 69 6F 6E 3A 20 61 74 74 61 63 68   position: attach
0f0 : 6D 65 6E 74 3B 20 66 69 6C 65 6E 61 6D 65 3D 22   ment; filename="
100 : 32 2E 65 78 65 22 3B 0D 0A 0D 0A 4D 5A 90 00 03   2.exe";....MZ...
```

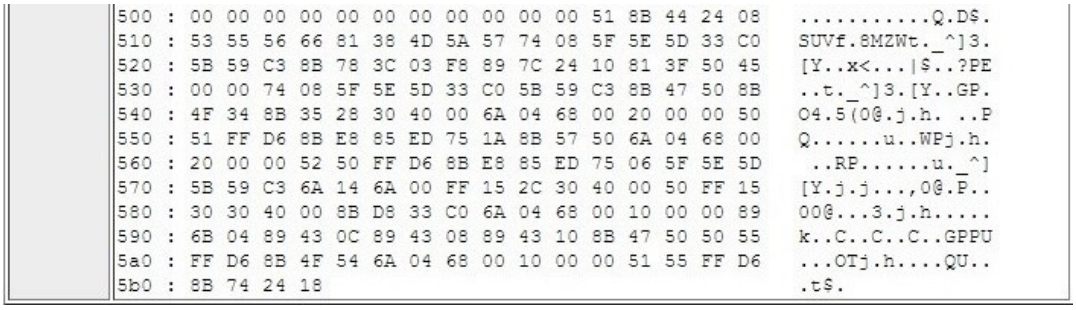
Payload

Plain Display

Download of Payload

Download in pcap format

```
110 : 00 00 00 04 00 00 00 FF FF 00 00 B8 00 00 00 00   ................
120 : 00 00 00 40 00 00 00 00 00 00 00 00 00 00 00 00   ...@............
130 : 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
140 : 00 00 00 00 00 00 00 E0 00 00 00 0E 1F BA 0E 00   ................
150 : B4 09 CD 21 B8 01 4C CD 21 54 68 69 73 20 70 72   ...!..L.!This pr
160 : 6F 67 72 61 6D 20 63 61 6E 6E 6F 74 20 62 65 20   ogram cannot be
170 : 72 75 6E 20 69 6E 20 44 4F 53 20 6D 6F 64 65 2E   run in DOS mode.
180 : 0D 0D 0A 24 00 00 00 00 00 00 00 D1 68 53 42 95   ...$........hSB.
190 : 09 3D 11 95 09 3D 11 95 09 3D 11 56 06 60 11 93   .=...=...=.V.`..
1a0 : 09 3D 11 EE 15 31 11 94 09 3D 11 FA 16 37 11 9E   .=...1...=...7..
1b0 : 09 3D 11 16 15 33 11 94 09 3D 11 FA 16 39 11 96   .=...3...=...9..
1c0 : 09 3D 11 95 09 3C 11 B8 09 3D 11 7D 16 36 11 96   .=...<...=.}.6..
1d0 : 09 3D 11 52 69 63 68 95 09 3D 11 00 00 00 00 00   .=.Rich..=......
1e0 : 00 00 00 00 00 00 00 00 00 00 00 00 50 45 00 00 4C   ............PE..L
1f0 : 01 03 00 3D 80 29 54 00 00 00 00 00 00 00 00 00 E0   ...=.)T.........
200 : 00 0F 01 0B 01 06 00 00 16 00 00 00 0C 00 00 00   ................
210 : 00 00 00 7E 22 00 00 00 10 00 00 00 30 00 00 00   ...~".......0...
220 : 00 40 00 00 10 00 00 00 02 00 00 04 00 00 00 00   .@..............
230 : 00 00 00 04 00 00 00 00 00 00 00 00 50 00 00 00   ............P...
240 : 04 00 00 00 00 00 00 02 00 00 00 00 00 10 00 00   ................
250 : 10 00 00 00 00 10 00 00 10 00 00 00 00 00 00 10   ................
260 : 00 00 00 00 00 00 00 00 00 00 00 00 CC 30 00 00 64   ............0..d
270 : 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
280 : 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
290 : 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
2a0 : 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
2b0 : 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
2c0 : 00 00 00 00 30 00 00 BC 00 00 00 00 00 00 00 00   ....0...........
2d0 : 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
2e0 : 00 00 00 2E 74 65 78 74 00 00 00 16 14 00 00 00   ....text........
2f0 : 10 00 00 00 16 00 00 00 04 00 00 00 00 00 00 00   ................
300 : 00 00 00 00 00 00 00 20 00 00 60 2E 72 64 61 74   ....... ..`.rdat
310 : 61 00 00 AA 04 00 00 00 30 00 00 00 06 00 00 00   a.......0.......
320 : 1A 00 00 00 00 00 00 00 00 00 00 00 00 00 00 40   ...............@
330 : 00 00 40 2E 64 61 74 61 00 00 00 E0 05 00 00 00   ..@.data........
340 : 40 00 00 00 06 00 00 00 20 00 00 00 00 00 00 00   @....... .......
350 : 00 00 00 00 00 00 00 40 00 00 C0 00 00 00 00 00   .......@........
360 : 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
370 : 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
380 : 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
390 : 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
3a0 : 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
3b0 : 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
3c0 : 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
3d0 : 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
3e0 : 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
3f0 : 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
400 : 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
410 : 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
420 : 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
430 : 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
440 : 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
450 : 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
460 : 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
470 : 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
480 : 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
490 : 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
4a0 : 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
4b0 : 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
4c0 : 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
4d0 : 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
4e0 : 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
4f0 : 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
```



```
500 : 00 00 00 00 00 00 00 00 00 00 51 8B 44 24 08   ..........Q.D$.
510 : 53 55 56 66 81 38 4D 5A 57 74 08 5F 5E 5D 33 C0   SUVf.8MZWt._^]3.
520 : 5B 59 C3 8B 78 3C 03 F8 89 7C 24 10 81 3F 50 45   [Y..x<...|$..?PE
530 : 00 00 74 08 5F 5E 5D 33 C0 5B 59 C3 8B 47 50 8B   ..t._^]3.[Y..GP.
540 : 4F 34 8B 35 28 30 40 00 6A 04 68 00 20 00 00 50   O4.5(0@.j.h. ..P
550 : 51 FF D6 8B E8 85 ED 75 1A 8B 57 50 6A 04 68 00   Q......u..WPj.h.
560 : 20 00 00 52 50 FF D6 8B E8 85 ED 75 06 5F 5E 5D    ..RP......u._^]
570 : 5B 59 C3 6A 14 6A 00 FF 15 2C 30 40 00 50 FF 15   [Y.j.j...,0@.P..
580 : 30 30 40 00 8B D8 33 C0 6A 04 68 00 10 00 00 89   00@...3.j.h.....
590 : 6B 04 89 43 0C 89 43 08 89 43 10 8B 47 50 50 55   k..C..C..C..GPPU
5a0 : FF D6 8B 4F 54 6A 04 68 00 10 00 00 51 55 FF D6   ...OTj.h....QU..
5b0 : 8B 74 24 18                                       .t$.
```

Figura 3.17: Paquete de la firma 28423. Fuente: Captura propia.

En la figura 3.17, se muestran los encabezados metadata, IP, TCP y el contenido del paquete; que corresponden al evento 28423. En la cabecera metadata se encuentra información respecto al identificador único de ese evento; así como la fecha y hora.

En la cabecera IP, se puede obtener la información de las direcciones IP de origen y destino que se establecieron en la regla para este evento. La IP de origen no pertenece al segmento de la intranet (116.255.152.168), mientras que la IP de destino pertenece a la red interna (172.17.2.5). En el encabezado TCP, el puerto de origen pertenece al conjunto de puertos de HTTP configurados en el archivo snort.conf (80) y el puerto de destino puede ser cualquiera. En este caso es el 1528.

En el payload del paquete se validan las búsquedas de contenido específico que se describieron en el cuerpo de la regla, el cual encontró el contenido específico de búsqueda (.exe). Esta coincidencia se encuentra en la línea 100. Con el uso de la expresión regular es posible establecer el contenido preciso que debe buscar Snort en el payload del paquete.

Snort Alert [1:27919:3]

Meta

ID #	Tiempo	Firma Encontrada
1 - 336061	2015-04-10 12:37:48	[snort] Snort Alert [1:27919:3]

Sensor	Sensor Dirección	Interfaz	Filtro
	Snort-Cuckoo:NULL	NULL	*none*

Grupo de Alertas	*none*

IP

Dirección Origen	Dirección Dest	Ver	Hdr Len	TOS	length	ID	fragment	offset	TTL	chksum
172.17.1.244	50.62.74.148	4	20	0	320	233	no	0	128	52727 = 0xcdf7

Options	*none*

TCP

Source Port	Dest Port	R 1	R 0	U R G	A C K	P S H	R S T	S Y N	F I N	seq #	ack	offset	res	window	urp	chksum
1052 [sans] [tantalo] [sstats]	80 [sans] [tantalo] [sstats]				X	X				1868230478	1201693267	20	0	64240	0	18652 = 0x48dc

Options	none

Payload

Plain Display

Download of Payload

Download in pcap format

```
length = 280

000 : 50 4F 53 54 20 2F 68 6F 6D 65 2F 70 6F 2F 67 61   POST /home/po/ga
010 : 74 65 2E 70 68 70 20 48 54 54 50 2F 31 2E 30 0D   te.php HTTP/1.0.
020 : 0A 48 6F 73 74 3A 20 77 77 77 2E 61 63 61 63 69   .Host: www.acaci
030 : 61 64 65 70 65 72 75 73 2E 63 6F 6D 2E 62 72 0D   adeperus.com.br.
040 : 0A 41 63 63 65 70 74 3A 20 2A 2F 2A 0D 0A 41 63   .Accept: */*..Ac
050 : 63 65 70 74 2D 45 6E 63 6F 64 69 6E 67 3A 20 69   cept-Encoding: i
060 : 64 65 6E 74 69 74 79 2C 20 2A 3B 71 3D 30 0D 0A   dentity, *;q=0..
070 : 43 6F 6E 74 65 6E 74 2D 4C 65 6E 67 74 68 3A 20   Content-Length:
080 : 32 36 36 0D 0A 43 6F 6E 6E 65 63 74 69 6F 6E 3A   266..Connection:
090 : 20 63 6C 6F 73 65 0D 0A 43 6F 6E 74 65 6E 74 2D    close..Content-
0a0 : 54 79 70 65 3A 20 61 70 70 6C 69 63 61 74 69 6F   Type: applicatio
0b0 : 6E 2F 6F 63 74 65 74 2D 73 74 72 65 61 6D 0D 0A   n/octet-stream..
0c0 : 43 6F 6E 74 65 6E 74 2D 45 6E 63 6F 64 69 6E 67   Content-Encoding
0d0 : 3A 20 62 69 6E 61 72 79 0D 0A 55 73 65 72 2D 41   : binary..User-A
0e0 : 67 65 6E 74 3A 20 4D 6F 7A 69 6C 6C 61 2F 34 2E   gent: Mozilla/4.
0f0 : 30 20 28 63 6F 6D 70 61 74 69 62 6C 65 3B 20 4D   0 (compatible; M
100 : 53 49 45 20 35 2E 30 3B 20 57 69 6E 64 6F 77 73   SIE 5.0; Windows
110 : 20 39 38 29 0D 0A 0D 0A                            98)....
```

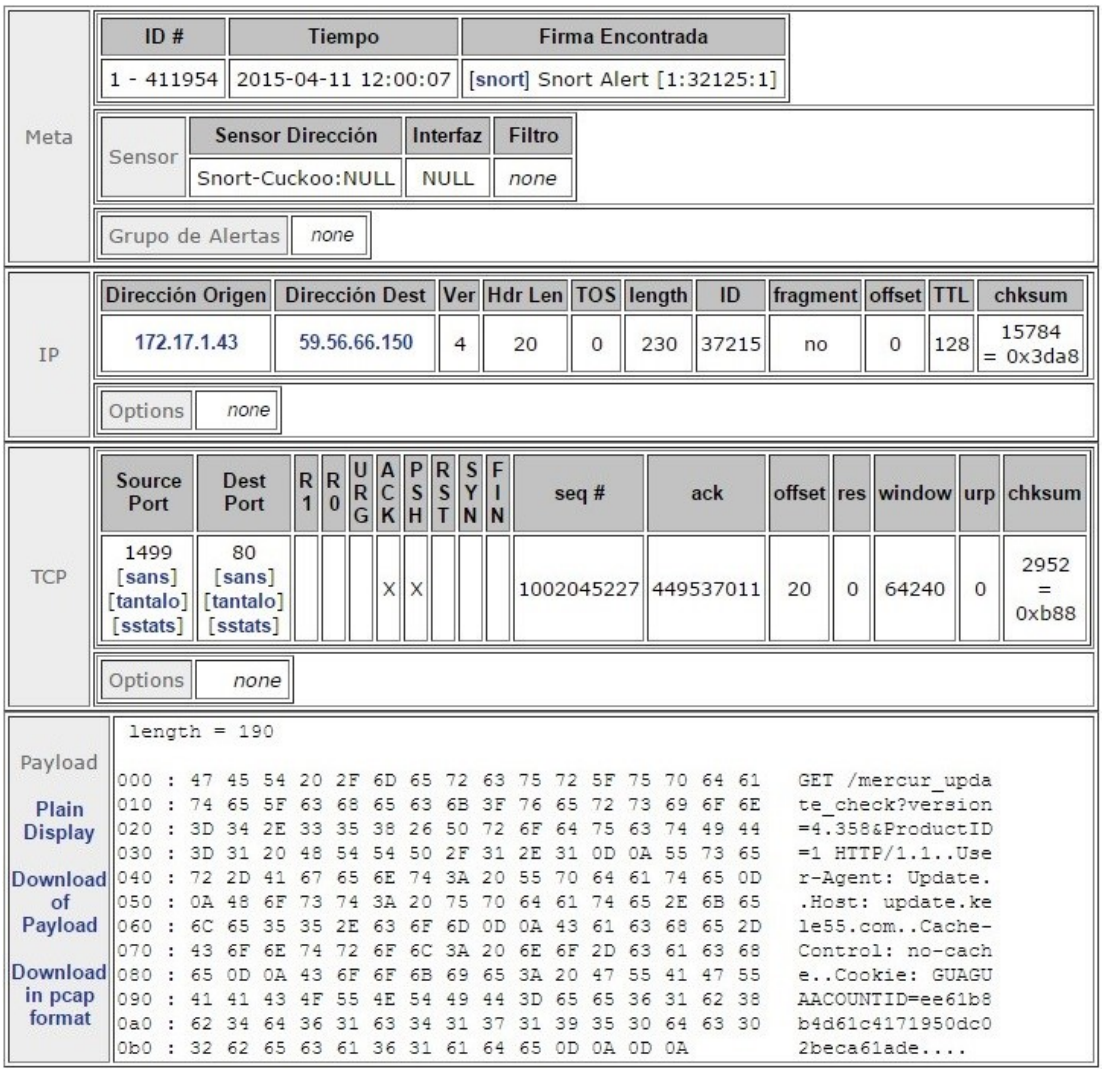
Figura 3.18: Paquete de la firma 27919. Fuente: Captura propia.

En el paquete mostrado en la figura 3.18, en el encabezado metadata se establece la fecha y hora en la que fue emitida la alerta, así como al identificador del evento que generó dicha regla.

En la cabecera IP y de acuerdo a la regla establecida; se comprueba que existe una conexión TCP desde una dirección local (172.17.1.244) hacia una dirección IP remota (50.62.74.148). En la cabecera TCP se muestra el puerto de origen (1052) y destino (80).

En el payload del paquete se validan las búsquedas de contenido específico que se describieron en el cuerpo de la regla. Con el uso de la expresión regular es posible establecer el contenido preciso que debe de buscar Snort en el payload del paquete. Las coincidencias se encuentran a partir de la línea 040.

Snort Alert [1:32125:1]



Meta

ID #	Tiempo	Firma Encontrada
1 - 411954	2015-04-11 12:00:07	[snort] Snort Alert [1:32125:1]

Sensor

Sensor Dirección	Interfaz	Filtro
Snort-Cuckoo:NULL	NULL	*none*

Grupo de Alertas: *none*

IP

Dirección Origen	Dirección Dest	Ver	Hdr Len	TOS	length	ID	fragment	offset	TTL	chksum
172.17.1.43	59.56.66.150	4	20	0	230	37215	no	0	128	15784 = 0x3da8

Options: *none*

TCP

Source Port	Dest Port	R 1	R 0	URG	ACK	PSH	RST	SYN	FIN	seq #	ack	offset	res	window	urp	chksum
1499 [sans] [tantalo] [sstats]	80 [sans] [tantalo] [sstats]				X	X				1002045227	449537011	20	0	64240	0	2952 = 0xb88

Options: *none*

Payload

Plain Display

Download of Payload

Download in pcap format

```
length = 190

000 : 47 45 54 20 2F 6D 65 72 63 75 72 5F 75 70 64 61   GET /mercur_upda
010 : 74 65 5F 63 68 65 63 6B 3F 76 65 72 73 69 6F 6E   te_check?version
020 : 3D 34 2E 33 35 38 26 50 72 6F 64 75 63 74 49 44   =4.358&ProductID
030 : 3D 31 20 48 54 54 50 2F 31 2E 31 0D 0A 55 73 65   =1 HTTP/1.1..Use
040 : 72 2D 41 67 65 6E 74 3A 20 55 70 64 61 74 65 0D   r-Agent: Update.
050 : 0A 48 6F 73 74 3A 20 75 70 64 61 74 65 2E 6B 65   .Host: update.ke
060 : 6C 65 35 35 2E 63 6F 6D 0D 0A 43 61 63 68 65 2D   le55.com..Cache-
070 : 43 6F 6E 74 72 6F 6C 3A 20 6E 6F 2D 63 61 63 68   Control: no-cach
080 : 65 0D 0A 43 6F 6F 6B 69 65 3A 20 47 55 41 47 55   e..Cookie: GUAGU
090 : 41 41 43 4F 55 4E 54 49 44 3D 65 65 36 31 62 38   AACOUNTID=ee61b8
0a0 : 62 34 64 36 31 63 34 31 37 31 39 35 30 64 63 30   b4d61c4171950dc0
0b0 : 32 62 65 63 61 36 31 61 64 65 0D 0A 0D 0A         2beca61ade....
```

Figura 3.19: Paquete de la firma 32125. Fuente: Captura propia.

En la alerta mostrada en la figura 3.19, es posible constatar que en el encabezado metada se establece la fecha en que hubo una coincidencia, el identificador de la regla que generó ese evento; así como el ID del evento. En el encabezado IP, y de acuerdo a la regla establecida, se determina que la dirección de origen de la comunicación es una dirección de la intranet (172.17.1.43) y la dirección de destino es una dirección externa a la red local (59.56.66.150). En la cabecera TCP se observa que el puerto de origen es el 1499 y el puerto destino es el 80.

En el payload del paquete se validan las búsquedas de contenido específico que se

describieron en el cuerpo de la regla, el cual establece que debe de buscar en el payload del paquete, la cadena “User-Agent update |0D 0A|”, en el cual los valores en 0D y 0A, son retorno de carro y salto de línea respectivamente.

Snort Alert [1:31527:1]

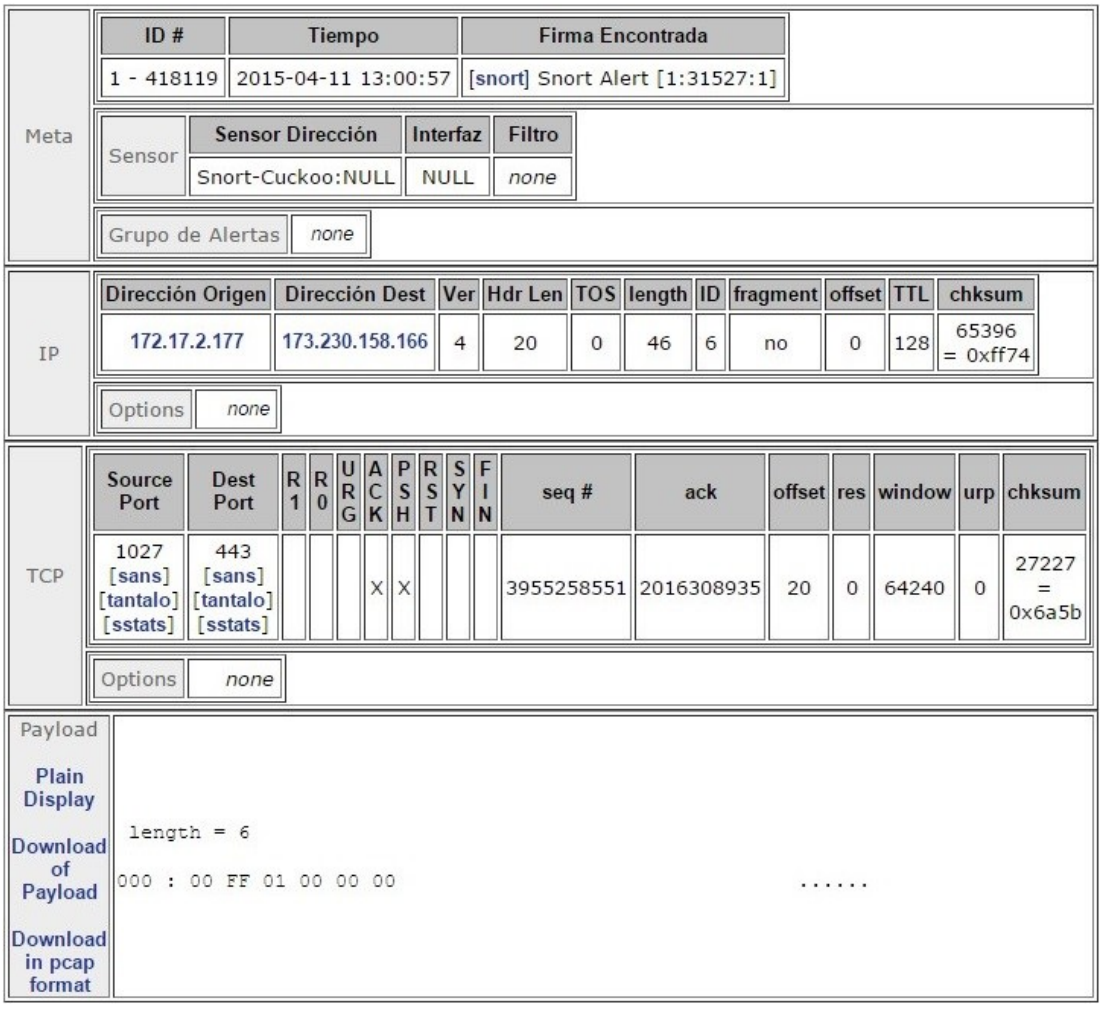


Figura 3.20: Paquete de la firma 31527. Fuente: Captura propia.

En el paquete de la figura 3.20, se observa que en el encabezado metadata se establece la fecha y hora en la que fue emitida la alerta, así como al identificador del evento que generó dicha regla (1-418119).

En el encabezado IP, y de acuerdo a la regla establecida, se determina que la dirección de origen de la comunicación es una dirección de la intranet (172.17.2.177) y la dirección de destino es una dirección externa a la red local (173.230.158.166). En la

cabecera TCP se especifica el puerto de origen 1027 y el puerto destino es el 443.

En el payload del paquete se validan las búsquedas de contenido específico que se describieron en el cuerpo de la regla, el cual establece un tamaño de 6 bytes del payload, en los cuales existan 6 caracteres. Estos valores se establecen en hexadecimal en el patrón a buscar en la regla.

3.3.3. Análisis en Cuckoo SandBox

A continuación se explicará la estructura y el contenido de los reportes generados por Cuckoo Sandbox sobre los archivos y URL obtenidos en cada uno de los eventos anteriores.

Snort Alert [1:28806:2]
El siguiente informe que se muestra en la figura 3.21, corresponde a una conexión que Snort detectó como una actividad de un troyano, que generó el evento 28806. El reporte se encuentra dividido en dos partes.

En la primera parte del reporte de Cuckoo, se muestra el tipo de objeto que fue analizado, la fecha en que se inició y terminó el análisis, la duración del análisis, así como la versión de Cuckoo con la que se realizó el análisis.

A continuación se muestran algunos campos en los cuales es posible identificar la muestra analizada. En estos campos se muestra el nombre del archivo analizado, su tamaño en bytes, el tipo de archivo, el CRC32 y diversas funciones hash como MD5, SHA1, SHA256 y SHA512, que sirven para identificar unívocamente a la muestra analizada. A continuación se muestra un algoritmo de fuzzy hashing (Ssdeep). Después, un campo en el que se despliega información acerca de Yara, la cual es una herramienta de clasificación de malware y finalmente un campo de Virus Total, el cual despliega el número de motores antivirus que detectaron la muestra como una amenaza. Adicionalmente Cuckoo toma capturas de pantalla mientras se realiza el análisis de la muestra.

Category	Started On	Completed On	Duration	Cuckoo Version
FILE	2015-04-25 13:51:44	2015-04-25 13:54:29	165 seconds	1.1

File Details

File name	w.exe
File size	787456 bytes
File type	PE32 executable (GUI) Intel 80386, for MS Windows
CRC32	26267A76
MD5	748361f76fd712a7193ce416180830bd
SHA1	a0af6451d0338d64ae4407cc9729be785192eda6
SHA256	03ccffd2630dad73f8c36198c9868515eec77e402d2ee03bcb1a07f80cfd91cb
SHA512	dd879edef0a31e139aa29e61dc19d9d3858f3cbeecdebc533d677e1506a91f27abc18ba4dfb50e35732471dce59a4fedf9
Ssdeep	12288:Grs1xjRzOrh0Ode5j51p/U0zEKsniVYqLetqChZzeFUG9uqGZoNcSkFbT8bBt4KH:v1/zOXde57enSYeaRe9uqGZoNcS
PEiD	• BobSoft Mini Delphi -> BoB / BobSoft
Yara	None matched
VirusTotal	Permalink VirusTotal Scan Date: 2015-04-23 13:09:03 Detection Rate: 40/55 (Expand)

Signatures

No signatures matched

Screenshots

Static Analysis

Sections
Imports
Strings

Dropped Files

Nothing to display.

Network Analysis

Nothing to display.

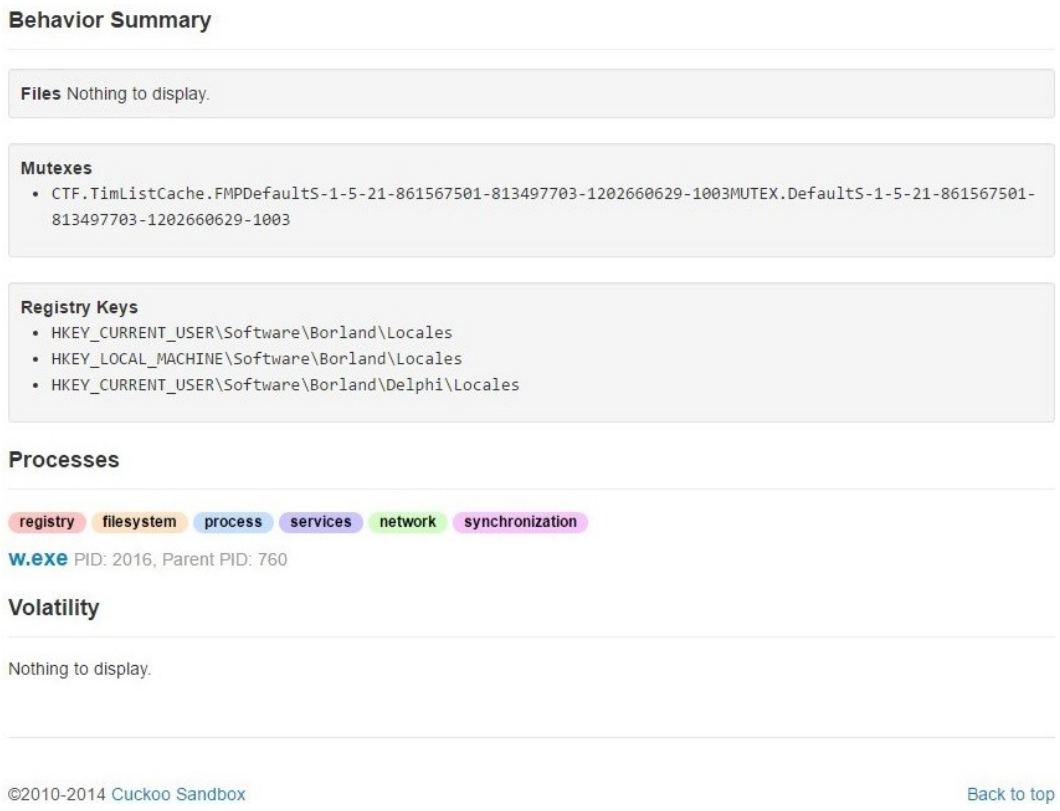


Figura 3.21: Análisis con Cuckoo SandBox del evento 28806. Fuente: Captura propia.

La sección de análisis estático muestra información respecto a la estructura del archivo ejecutable. Aquí es posible encontrar información sobre las secciones que conforman el archivo y las direcciones de memoria. Así mismo, es posible determinar sí el archivo malicioso importa alguna librería durante la ejecución del mismo. Para este archivo se importaron las siguientes librerías:

- winspool.drv
- user32.dll
- advapi32.dll
- version.dll
- gdi32.dll
- comctl32.dll
- comdlg32.dll
- kernel32.dll
- oleaut32.dll
- opengl32.dll
- winmm.dll

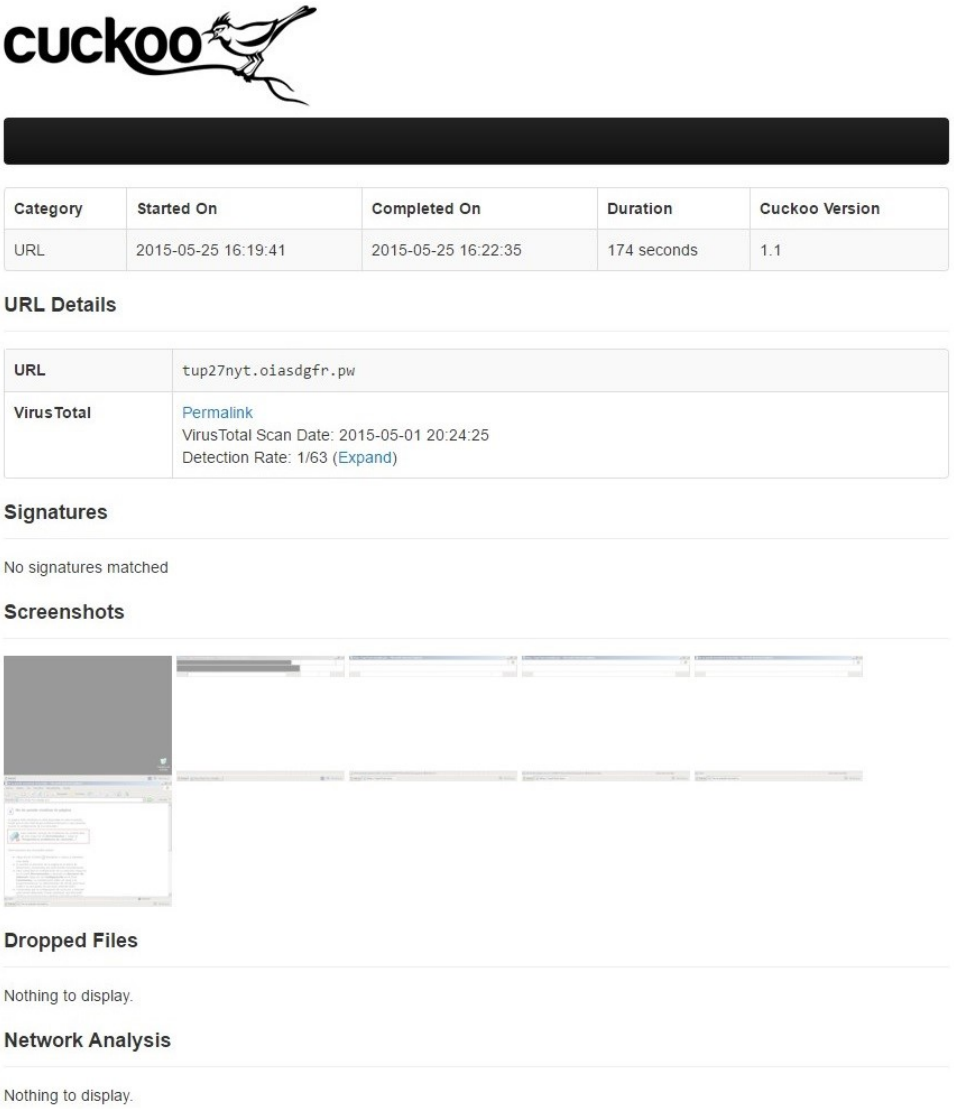
Sí hay información a mostrar en esta sección, bastará con seleccionar las subsecciones localizadas en el área de análisis estático.

En el área Dropped Files se notifica que el archivo *w.exe* no descargó archivos adicionales. Por ende, en el área Network Analysis se informa que dicho archivo malicioso no se conectó a ningún sitio en Internet y que tampoco envía información robada del sistema. En la sección Behavior Summary, se encuentra un historial de aquellos archivos que fueron modificados, el detalle de las exclusiones mutuas (mutexes) así como las

llaves de registro.

Al final del reporte, se selecciona el nombre del archivo analizado, y desplegará una tabla con el resumen de las operaciones realizadas sobre el sistema. Estas operaciones se encuentran separadas por categorías, y se diferencian por un color distinto que está indicado en la sección processes.

Snort Alert [1:28039:4]



Category	Started On	Completed On	Duration	Cuckoo Version
URL	2015-05-25 16:19:41	2015-05-25 16:22:35	174 seconds	1.1

URL Details

URL	tup27nyt.oiasdgfr.pw
VirusTotal	Permalink VirusTotal Scan Date: 2015-05-01 20:24:25 Detection Rate: 1/63 (Expand)

Signatures

No signatures matched

Screenshots

Dropped Files

Nothing to display.

Network Analysis

Nothing to display.

Behavior Summary

Files

- C:\Documents and Settings\cuckoo\Escritorio
- C:\WINDOWS\Registration\R000000000007.clb
- C:\Documents and Settings\cuckoo\Configuraci\xc3\xb3n local\Archivos temporales de Internet
- C:\Documents and Settings\cuckoo\Configuraci\xc3\xb3n local\Historial
- C:\Documents and Settings\cuckoo\Configuraci\xc3\xb3n local\Archivos temporales de Internet\Content.IE5\
- C:\
- C:\Documents and Settings\cuckoo\Configuraci\xc3\xb3n local\Archivos temporales de Internet\Content.IE5\index.dat
- C:\Documents and Settings\cuckoo\Cookies\
- C:\Documents and Settings\cuckoo\Cookies\index.dat
- C:\Documents and Settings\cuckoo\Configuraci\xc3\xb3n local\Historial\History.IE5\
- C:\Documents and Settings\cuckoo\Configuraci\xc3\xb3n local\Historial\History.IE5\index.dat
- C:\WINDOWS\System32\cscui.dll
- shadow
- IDE#CdRomVBOX_CD-ROM_____1.0_____#42562d3231303037333036372020202020202020#{53f5630d-b6bf-11d0-94f2-00a0c91efb8b}
- MountPointManager
- STORAGE#Volume#1&30a96598&0&Signature49B049AFOffset7E00Length27F4DB200#{53f5630d-b6bf-11d0-94f2-00a0c91efb8b}
- C:\Documents and Settings
- C:\Documents and Settings\cuckoo
- C:\Documents and Settings\cuckoo\Favoritos
- C:\Documents and Settings\cuckoo\Favoritos\desktop.ini
- C:\Documents and Settings\cuckoo\Favoritos\V\xc3\xadnculos
- C:\Documents and Settings\cuckoo\Configuraci\xc3\xb3n local
- C:\Documents and Settings\cuckoo\Configuraci\xc3\xb3n local\Archivos temporales de Internet\desktop.ini
- C:\Documents and Settings\cuckoo\Favoritos\V\xc3\xadnculos*.*
- C:\Documents and Settings\cuckoo\Datos de programa\Sun\Java\Deployment\deployment.properties
- C:\Archivos de programa
- C:\Archivos de programa\Internet Explorer
- C:\Archivos de programa\Adobe\Acrobat 6.0\Reader\ActiveX\AcroIEHelper.dll
- PIPE\lsarpc
- C:\WINDOWS\system32\shdocvw.dll
- C:\WINDOWS\system32\stdole2.tlb
- c:\autoexec.bat
- C:\Documents and Settings\All Users\Datos de programa\Microsoft\Network\Connections\Pbk*.pbk
- C:\WINDOWS\system32\Ras*.pbk
- C:\Documents and Settings\cuckoo\Datos de programa\Microsoft\Network\Connections\Pbk*.pbk
- C:\WINDOWS\WindowsShell.manifest
- C:\WINDOWS\WindowsShell.Config
- C:\ARCHIV~1\MICROS~2\OFFICE11\REFBAR.ICO
- C:\ARCHIV~1\MICROS~2\OFFICE11\REFBARH.ICO
- C:\Archivos de programa\Messenger\msmsgs.exe
- C:\WINDOWS\system32\shell32.dll
- C:\WINDOWS\system32\url.dll
- C:\WINDOWS\system32\mshtml.dll
- C:\Archivos de programa\Microsoft Office\OFFICE11\msohev.dll
- C:\notexist.htm
- C:\Archivos de programa\Internet Explorer\iexplore.exe
- C:\WINDOWS\system32\inetcpl.cpl
- C:\Documents and Settings\cuckoo\Configuraci\xc3\xb3n local\Historial\desktop.ini

Mutexes

- CTF.TimListCache.FMPDefaultS-1-5-21-861567501-813497703-1202660629-1003MUTEX.DefaultS-1-5-21-861567501-813497703-1202660629-1003
- Shell.CMruPidlList
- WininetStartupMutex
- _!MSFTHISTORY!_
- c:!documents and settings!cuckoo!configuraci\xc3\xb3n local!archivos temporales de internet!content.ie5!
- c:!documents and settings!cuckoo!cookies!
- c:!documents and settings!cuckoo!configuraci\xc3\xb3n local!historial!history.ie5!
- WininetConnectionMutex
- WininetProxyRegistryMutex
- ShimCacheMutex
- MSCTF.Shared.MUTEX.IMF

Registry Keys

- HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Explorer\Performance
- HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\BrowseNewProcess
- HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Explorer\BrowseNewProcess
- HKEY_LOCAL_MACHINE\Software\Policies\Microsoft\Internet Explorer\Restrictions
- HKEY_CURRENT_USER\Software\Policies\Microsoft\Internet Explorer\Restrictions
- HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\User Shell Folders
- HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\Shell Folders
- HKEY_LOCAL_MACHINE\Software\Microsoft\Internet Explorer
- HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer
- HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\Streams
- HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\
- HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer
- HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer
- HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Policies\System
- HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\Advanced
- HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\CabinetState
- HKEY_LOCAL_MACHINE\Software\Microsoft\Internet Explorer\International
- HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Session Manager\AppCompatibility
- HKEY_LOCAL_MACHINE\Software\Microsoft\COM3
- HKEY_USERS\S-1-5-21-861567501-813497703-1202660629-1003_Classes
- HKEY_LOCAL_MACHINE\Software\Classes
- \REGISTRY\USER
- HKEY_LOCAL_MACHINE\Software\Classes\CLSID
- CLSID\{A5E46E3A-8849-11D1-9D8C-00C04FC99D61}
- CLSID\{A5E46E3A-8849-11D1-9D8C-00C04FC99D61}\TreatAs
- \CLSID\{A5E46E3A-8849-11D1-9D8C-00C04FC99D61}
- \CLSID\{A5E46E3A-8849-11D1-9D8C-00C04FC99D61}\InprocServer32
- \CLSID\{A5E46E3A-8849-11D1-9D8C-00C04FC99D61}\InprocServerX86
- \CLSID\{A5E46E3A-8849-11D1-9D8C-00C04FC99D61}\LocalServer32
- \CLSID\{A5E46E3A-8849-11D1-9D8C-00C04FC99D61}\InprocHandler32
- \CLSID\{A5E46E3A-8849-11D1-9D8C-00C04FC99D61}\InprocHandlerX86
- \CLSID\{A5E46E3A-8849-11D1-9D8C-00C04FC99D61}\LocalServer
- HKEY_CLASSES_ROOT\CLSID\{A5E46E3A-8849-11D1-9D8C-00C04FC99D61}
- HKEY_CLASSES_ROOT\CLSID\{A5E46E3A-8849-11D1-9D8C-00C04FC99D61}\TreatAs
- HKEY_CURRENT_USER\SOFTWARE\Microsoft\Internet Explorer\Security\P3Global
- HKEY_CURRENT_USER\SOFTWARE\Microsoft\Internet Explorer\Security\P3Sites
- HKEY_CLASSES_ROOT\.htm
- HKEY_CLASSES_ROOT\.html
- HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\URL\Prefixes
- HKEY_CURRENT_USER\Software\Microsoft\Internet Explorer\URLSearchHooks
- HKEY_LOCAL_MACHINE\Software\Classes\CLSID\{cfbfae00-17a6-11d0-99cb-00c04fd64497}\InProcServer32

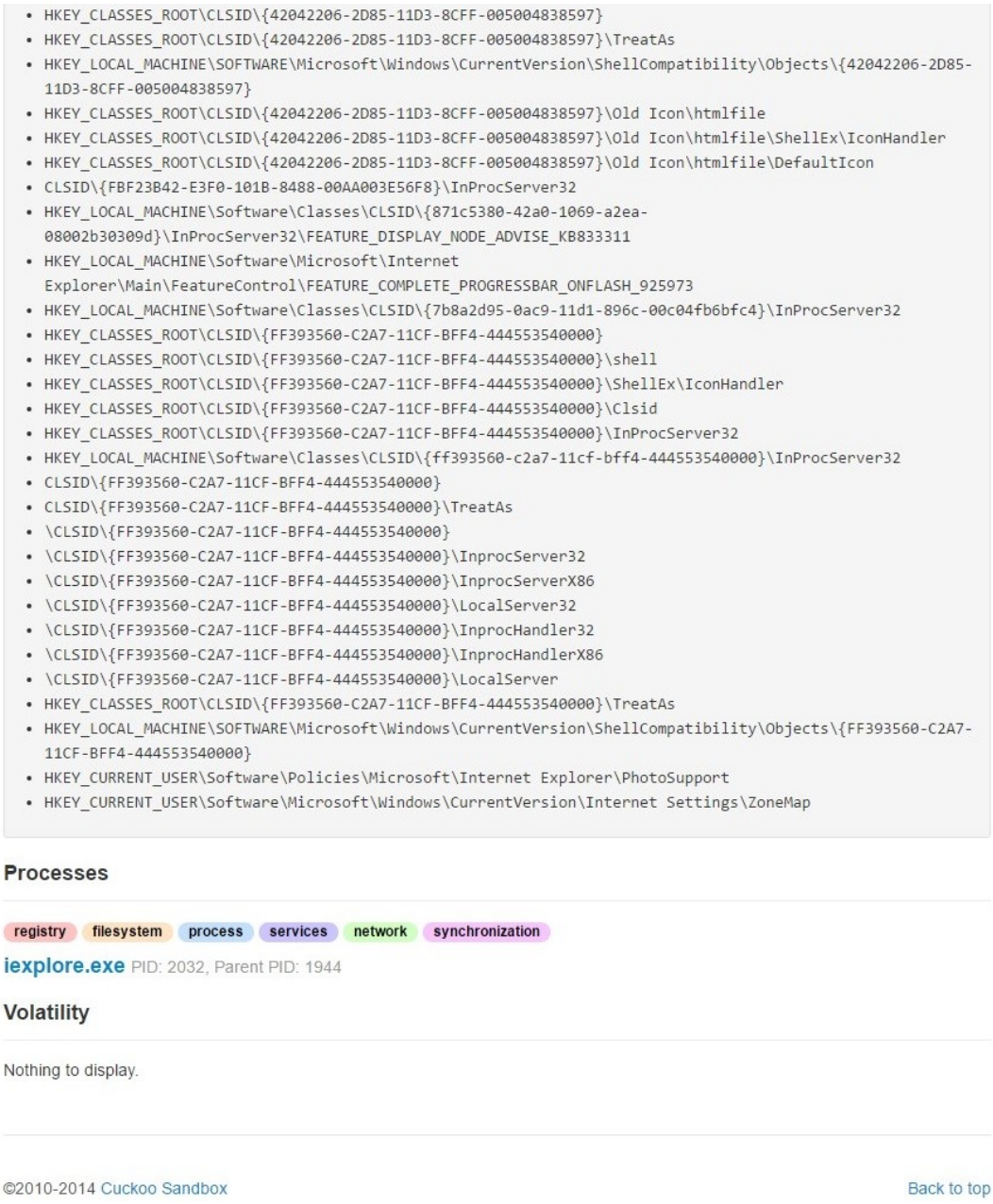


Figura 3.22: Análisis con Cuckoo SandBox del evento 28039. Fuente: Captura propia.

El informe mostrado en la figura 3.22, corresponde al evento con ID 28039 que Snort detectó como una actividad relacionada con un troyano. Esta actividad fue detectada como maliciosa, ya que la muestra analizada fue una resolución del nombre de dominio *tup27nyt.oiasdgfr.pw*.

Como se observa al inicio del reporte, la muestra fue analizada en la versión de Cuckoo SandBox 1.1 con una duración de 174 segundos.

En la sección de los detalles de la URL analizada, se observa que únicamente un antivirus del portal Virus Total detectó el dominio como peligroso.

Posterior a los detalles de la URL analizada, se encuentran las capturas de pantalla realizadas durante la ejecución del análisis de la muestra. Por otro lado, el reporte generado por Cuckoo despliega que no fue descargado ningún archivo durante la ejecución del análisis y que no emplea la técnica drive-by-download, ni se realizaron conexiones a sitios externos durante la ejecución.

Adicionalmente, se muestra el resúmen del comportamiento que tuvo el archivo durante el análisis. Al inicio se encuentra un registro con los archivos que fueron modificados durante la realización del análisis de la URL *tup27nyt.oiasdgfr.pw*.

Debajo de lo anterior, se observan los detalles de los mutexes y las llaves de registro que fueron modificadas.

Al final del reporte se muestra el proceso iexplore.exe lanzado durante el análisis, que corresponde al proceso de Internet Explorer con el identificador de proceso 2031.

Snort Alert [1:31683:1]
El siguiente informe que se muestra en la figura 3.23, corresponde a una conexión que Snort detectó como una actividad de un troyano, que generó el evento 31683. A continuación se explicará el contenido del reporte que arrojó Cuckoo SandBox al analizar el dominio *factorygood.net*.

En la primera parte del reporte de Cuckoo, se muestra la categoría de la muestra que fue analizada, la fecha en que se inició y terminó el análisis, la duración del análisis, así como la versión de Cuckoo con la que se realizó el análisis. En la sección correspondiente al análisis de la URL, se identifica que 5 motores antivirus detectaron el dominio como un sitio web malicioso. Posterior de los detalles anteriores, son mostradas las capturas de pantalla durante el análisis de la URL y se informa que no fue descargado ningún archivo ni que se realizaron conexiones a sitios remotos.

En la sección del resúmen del comportamiento de la muestra se despliegan los archivos que sufrieron modificaciones al analizar la URL, lo que permite realizar una trazabilidad sobre algún comportamiento sospechoso. Por otro lado, continuando con los detalles del reporte generado, son informadas las exclusiones mutuas así como las modificaciones en el registro de Windows.

Para poder realizar el análisis de la URL tuvo que realizarse sobre el navegador

Internet Explorer con su correspondiente proceso iexplore.exe con PID 2016.

cuckoo

Category	Started On	Completed On	Duration	Cuckoo Version
URL	2015-05-25 16:24:12	2015-05-25 16:27:02	170 seconds	1.1

URL Details

URL	factorygood.net
VirusTotal	Permalink VirusTotal Scan Date: 2015-05-20 23:09:36 Detection Rate: 5/63 (Expand)

Signatures

No signatures matched

Screenshots

Dropped Files

Nothing to display.

Network Analysis

Nothing to display.

Behavior Summary

Files

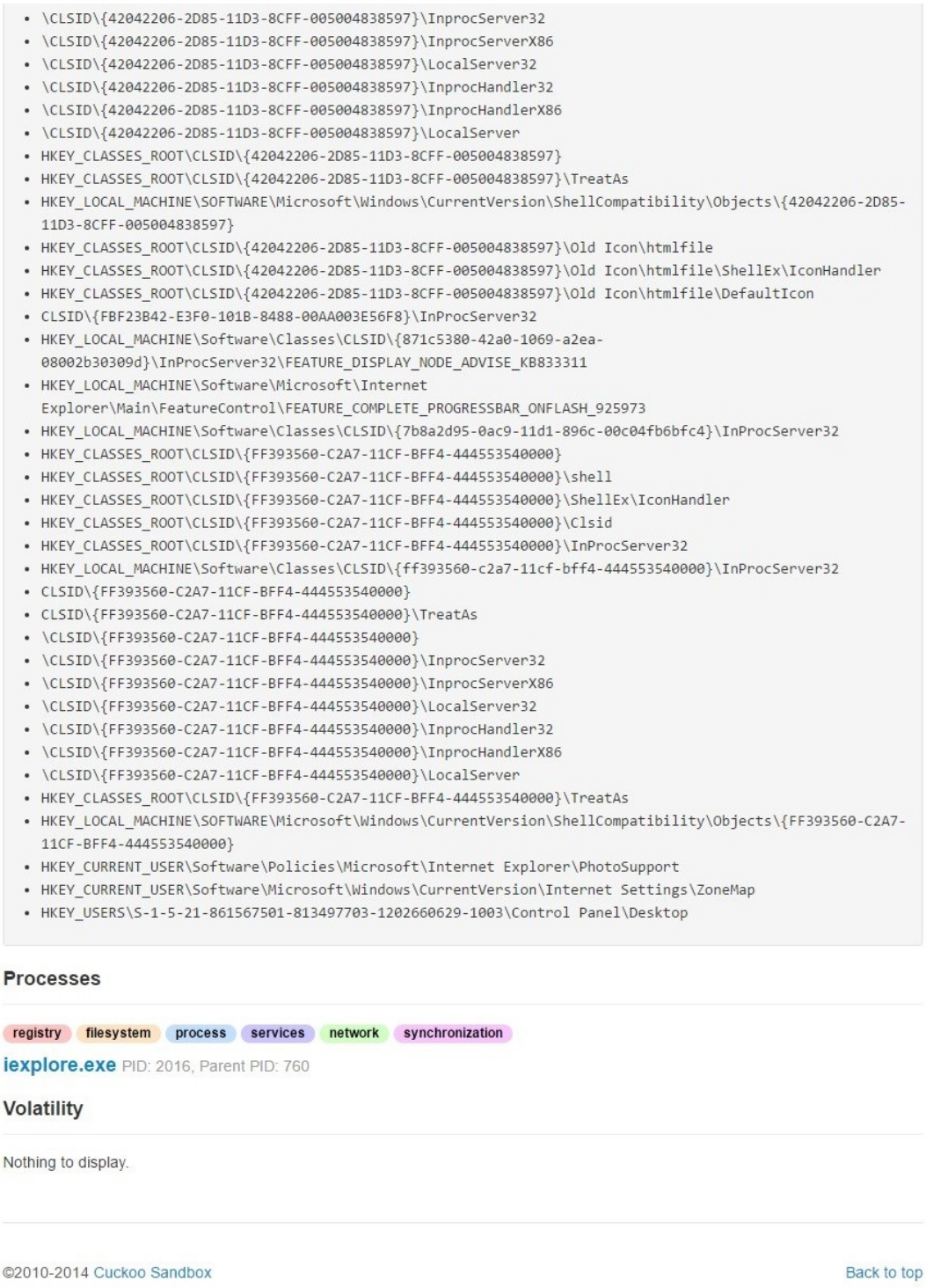
- C:\Documents and Settings\cuckoo\Escritorio
- C:\WINDOWS\Registration\R000000000007.clb
- C:\Documents and Settings\cuckoo\Configuraci\xc3\xb3n local\Archivos temporales de Internet
- C:\Documents and Settings\cuckoo\Configuraci\xc3\xb3n local\Historial
- C:\Documents and Settings\cuckoo\Configuraci\xc3\xb3n local\Archivos temporales de Internet\Content.IE5\
- C:\
- C:\Documents and Settings\cuckoo\Configuraci\xc3\xb3n local\Archivos temporales de Internet\Content.IE5\index.dat
- C:\Documents and Settings\cuckoo\Cookies\
- C:\Documents and Settings\cuckoo\Cookies\index.dat
- C:\Documents and Settings\cuckoo\Configuraci\xc3\xb3n local\Historial\History.IE5\
- C:\Documents and Settings\cuckoo\Configuraci\xc3\xb3n local\Historial\History.IE5\index.dat
- C:\WINDOWS\System32\cscui.dll
- shadow
- IDE#CdRomVBOX_CD-ROM_____1.0_____#42562d3231303037333036372020202020202020#{53f5630d-b6bf-11d0-94f2-00a0c91efb8b}
- MountPointManager
- STORAGE#Volume#1&30a96598&0&Signature49B049AFOffset7E00Length27F4DB200#{53f5630d-b6bf-11d0-94f2-00a0c91efb8b}
- C:\Documents and Settings
- C:\Documents and Settings\cuckoo
- C:\Documents and Settings\cuckoo\Favoritos
- C:\Documents and Settings\cuckoo\Favoritos\desktop.ini
- C:\Documents and Settings\cuckoo\Favoritos\V\xc3\xadnculos
- C:\Documents and Settings\cuckoo\Configuraci\xc3\xb3n local
- C:\Documents and Settings\cuckoo\Configuraci\xc3\xb3n local\Archivos temporales de Internet\desktop.ini
- C:\Documents and Settings\cuckoo\Favoritos\V\xc3\xadnculos*.*
- C:\Documents and Settings\cuckoo\Datos de programa\Sun\Java\Deployment\deployment.properties
- C:\Archivos de programa
- C:\Archivos de programa\Internet Explorer
- C:\Archivos de programa\Adobe\Acrobat 6.0\Reader\ActiveX\AcroIEHelper.dll
- PIPE\lsarpc
- C:\WINDOWS\system32\shdocvw.dll
- C:\WINDOWS\system32\stdole2.tlb
- c:\autoexec.bat
- C:\Documents and Settings\All Users\Datos de programa\Microsoft\Network\Connections\Pbk*.pbk
- C:\WINDOWS\system32\Ras*.pbk
- C:\Documents and Settings\cuckoo\Datos de programa\Microsoft\Network\Connections\Pbk*.pbk
- C:\WINDOWS\WindowsShell.manifest
- C:\WINDOWS\WindowsShell.Config
- C:\ARCHIV~1\MICROS~2\OFFICE11\REFBAR.ICO
- C:\ARCHIV~1\MICROS~2\OFFICE11\REFBARH.ICO
- C:\Archivos de programa\Messenger\msmsgs.exe
- C:\WINDOWS\system32\shell32.dll
- C:\WINDOWS\system32\url.dll
- C:\WINDOWS\system32\mshtml.dll
- C:\Archivos de programa\Microsoft Office\OFFICE11\msohev.dll
- C:\notexist.htm
- C:\Archivos de programa\Internet Explorer\iexplore.exe
- C:\WINDOWS\system32\inetcpl.cpl
- C:\Documents and Settings\cuckoo\Configuraci\xc3\xb3n local\Historial\desktop.ini

Mutexes

- CTF.TimListCache.FMPDefaultS-1-5-21-861567501-813497703-1202660629-1003MUTEX.DefaultS-1-5-21-861567501-813497703-1202660629-1003
- Shell.CMruPidlList
- WininetStartupMutex
- _!MSFTHISTORY!_
- c:!documents and settings!cuckoo!configuraci\xc3\xb3n local!archivos temporales de internet!content.ie5!
- c:!documents and settings!cuckoo!cookies!
- c:!documents and settings!cuckoo!configuraci\xc3\xb3n local!historial!history.ie5!
- WininetConnectionMutex
- WininetProxyRegistryMutex
- ShimCacheMutex
- MSCTF.Shared.MUTEX.IMF

Registry Keys

- HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Explorer\Performance
- HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\BrowseNewProcess
- HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Explorer\BrowseNewProcess
- HKEY_LOCAL_MACHINE\Software\Policies\Microsoft\Internet Explorer\Restrictions
- HKEY_CURRENT_USER\Software\Policies\Microsoft\Internet Explorer\Restrictions
- HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\User Shell Folders
- HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\Shell Folders
- HKEY_LOCAL_MACHINE\Software\Microsoft\Internet Explorer
- HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer
- HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\Streams
- HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\
- HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer
- HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer
- HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Policies\System
- HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\Advanced
- HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\CabinetState
- HKEY_LOCAL_MACHINE\Software\Microsoft\Internet Explorer\International
- HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Session Manager\AppCompatibility
- HKEY_LOCAL_MACHINE\Software\Microsoft\COM3
- HKEY_USERS\S-1-5-21-861567501-813497703-1202660629-1003_Classes
- HKEY_LOCAL_MACHINE\Software\Classes
- \REGISTRY\USER
- HKEY_LOCAL_MACHINE\Software\Classes\CLSID
- CLSID\{A5E46E3A-8849-11D1-9D8C-00C04FC99D61}
- CLSID\{A5E46E3A-8849-11D1-9D8C-00C04FC99D61}\TreatAs
- \CLSID\{A5E46E3A-8849-11D1-9D8C-00C04FC99D61}
- \CLSID\{A5E46E3A-8849-11D1-9D8C-00C04FC99D61}\InprocServer32
- \CLSID\{A5E46E3A-8849-11D1-9D8C-00C04FC99D61}\InprocServerX86
- \CLSID\{A5E46E3A-8849-11D1-9D8C-00C04FC99D61}\LocalServer32
- \CLSID\{A5E46E3A-8849-11D1-9D8C-00C04FC99D61}\InprocHandler32
- \CLSID\{A5E46E3A-8849-11D1-9D8C-00C04FC99D61}\InprocHandlerX86
- \CLSID\{A5E46E3A-8849-11D1-9D8C-00C04FC99D61}\LocalServer
- HKEY_CLASSES_ROOT\CLSID\{A5E46E3A-8849-11D1-9D8C-00C04FC99D61}
- HKEY_CLASSES_ROOT\CLSID\{A5E46E3A-8849-11D1-9D8C-00C04FC99D61}\TreatAs
- HKEY_CURRENT_USER\SOFTWARE\Microsoft\Internet Explorer\Security\P3Global
- HKEY_CURRENT_USER\SOFTWARE\Microsoft\Internet Explorer\Security\P3Sites
- HKEY_CLASSES_ROOT\.htm
- HKEY_CLASSES_ROOT\.html
- HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\URL\Prefixes
- HKEY_CURRENT_USER\Software\Microsoft\Internet Explorer\URLSearchHooks



- \CLSID\{42042206-2D85-11D3-8CFF-005004838597}\InprocServer32
- \CLSID\{42042206-2D85-11D3-8CFF-005004838597}\InprocServerX86
- \CLSID\{42042206-2D85-11D3-8CFF-005004838597}\LocalServer32
- \CLSID\{42042206-2D85-11D3-8CFF-005004838597}\InprocHandler32
- \CLSID\{42042206-2D85-11D3-8CFF-005004838597}\InprocHandlerX86
- \CLSID\{42042206-2D85-11D3-8CFF-005004838597}\LocalServer
- HKEY_CLASSES_ROOT\CLSID\{42042206-2D85-11D3-8CFF-005004838597}
- HKEY_CLASSES_ROOT\CLSID\{42042206-2D85-11D3-8CFF-005004838597}\TreatAs
- HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\ShellCompatibility\Objects\{42042206-2D85-11D3-8CFF-005004838597}
- HKEY_CLASSES_ROOT\CLSID\{42042206-2D85-11D3-8CFF-005004838597}\Old Icon\htmlfile
- HKEY_CLASSES_ROOT\CLSID\{42042206-2D85-11D3-8CFF-005004838597}\Old Icon\htmlfile\ShellEx\IconHandler
- HKEY_CLASSES_ROOT\CLSID\{42042206-2D85-11D3-8CFF-005004838597}\Old Icon\htmlfile\DefaultIcon
- CLSID\{FBF23B42-E3F0-101B-8488-00AA003E56F8}\InProcServer32
- HKEY_LOCAL_MACHINE\Software\Classes\CLSID\{871c5380-42a0-1069-a2ea-08002b30309d}\InProcServer32\FEATURE_DISPLAY_NODE_ADVISE_KB833311
- HKEY_LOCAL_MACHINE\Software\Microsoft\Internet Explorer\Main\FeatureControl\FEATURE_COMPLETE_PROGRESSBAR_ONFLASH_925973
- HKEY_LOCAL_MACHINE\Software\Classes\CLSID\{7b8a2d95-0ac9-11d1-896c-00c04fb6bfc4}\InProcServer32
- HKEY_CLASSES_ROOT\CLSID\{FF393560-C2A7-11CF-BFF4-444553540000}
- HKEY_CLASSES_ROOT\CLSID\{FF393560-C2A7-11CF-BFF4-444553540000}\shell
- HKEY_CLASSES_ROOT\CLSID\{FF393560-C2A7-11CF-BFF4-444553540000}\ShellEx\IconHandler
- HKEY_CLASSES_ROOT\CLSID\{FF393560-C2A7-11CF-BFF4-444553540000}\Clsid
- HKEY_CLASSES_ROOT\CLSID\{FF393560-C2A7-11CF-BFF4-444553540000}\InProcServer32
- HKEY_LOCAL_MACHINE\Software\Classes\CLSID\{ff393560-c2a7-11cf-bff4-444553540000}\InProcServer32
- CLSID\{FF393560-C2A7-11CF-BFF4-444553540000}
- CLSID\{FF393560-C2A7-11CF-BFF4-444553540000}\TreatAs
- \CLSID\{FF393560-C2A7-11CF-BFF4-444553540000}
- \CLSID\{FF393560-C2A7-11CF-BFF4-444553540000}\InprocServer32
- \CLSID\{FF393560-C2A7-11CF-BFF4-444553540000}\InprocServerX86
- \CLSID\{FF393560-C2A7-11CF-BFF4-444553540000}\LocalServer32
- \CLSID\{FF393560-C2A7-11CF-BFF4-444553540000}\InprocHandler32
- \CLSID\{FF393560-C2A7-11CF-BFF4-444553540000}\InprocHandlerX86
- \CLSID\{FF393560-C2A7-11CF-BFF4-444553540000}\LocalServer
- HKEY_CLASSES_ROOT\CLSID\{FF393560-C2A7-11CF-BFF4-444553540000}\TreatAs
- HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\ShellCompatibility\Objects\{FF393560-C2A7-11CF-BFF4-444553540000}
- HKEY_CURRENT_USER\Software\Policies\Microsoft\Internet Explorer\PhotoSupport
- HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Internet Settings\ZoneMap
- HKEY_USERS\S-1-5-21-861567501-813497703-1202660629-1003\Control Panel\Desktop

Processes

registry filesystem process services network synchronization

iexplore.exe PID: 2016, Parent PID: 760

Volatility

Nothing to display.

©2010-2014 Cuckoo Sandbox Back to top

Figura 3.23: Análisis con Cuckoo SandBox del evento 31683. Fuente: Captura propia.

Snort Alert [1:28801:2]

A continuación se muestra el informe realizado por Cuckoo Sandbox, respecto al evento 28801. En la figura 3.24, se detalla que el objeto analizado fue un archivo, la fecha de inicio y finalización del análisis del archivo Officekeyserial15.exe así como la duración total del análisis. Este análisis se realizó en la versión 1.1 de Cuckoo Sandbox.

En el apartado detalles del archivo se muestran las características del archivo analizado para su identificación. En esta sección se muestra el nombre del archivo analizado, el tamaño del archivo (1232165 bytes), se identifica al archivo como un archivo ejecutable. Por otro lado, también se despliega el CRC32 del archivo Officekeyserial15.exe, así como las 4 diferentes funciones hash con las cuales es posible identificarlo.

Se muestra el valor del algoritmo de fuzzy hashing Ssdeep, y Yara no encontró una clasificación para dicho malware. En la sección de Virus Total se muestra que 28 motores antivirus de 57 identificaron la muestra como malicioso. Durante el análisis se realizaron 5 capturas de pantalla.

La sección análisis estático muestra información respecto a la estructura del archivo ejecutable. Aquí se muestra información sobre la versión del archivo, las secciones que conforman el archivo y las direcciones de memoria que tiene cada una de ellas. Al examinar el módulo de importaciones del reporte de Cuckoo SandBox es posible detectar las siguientes librerías:

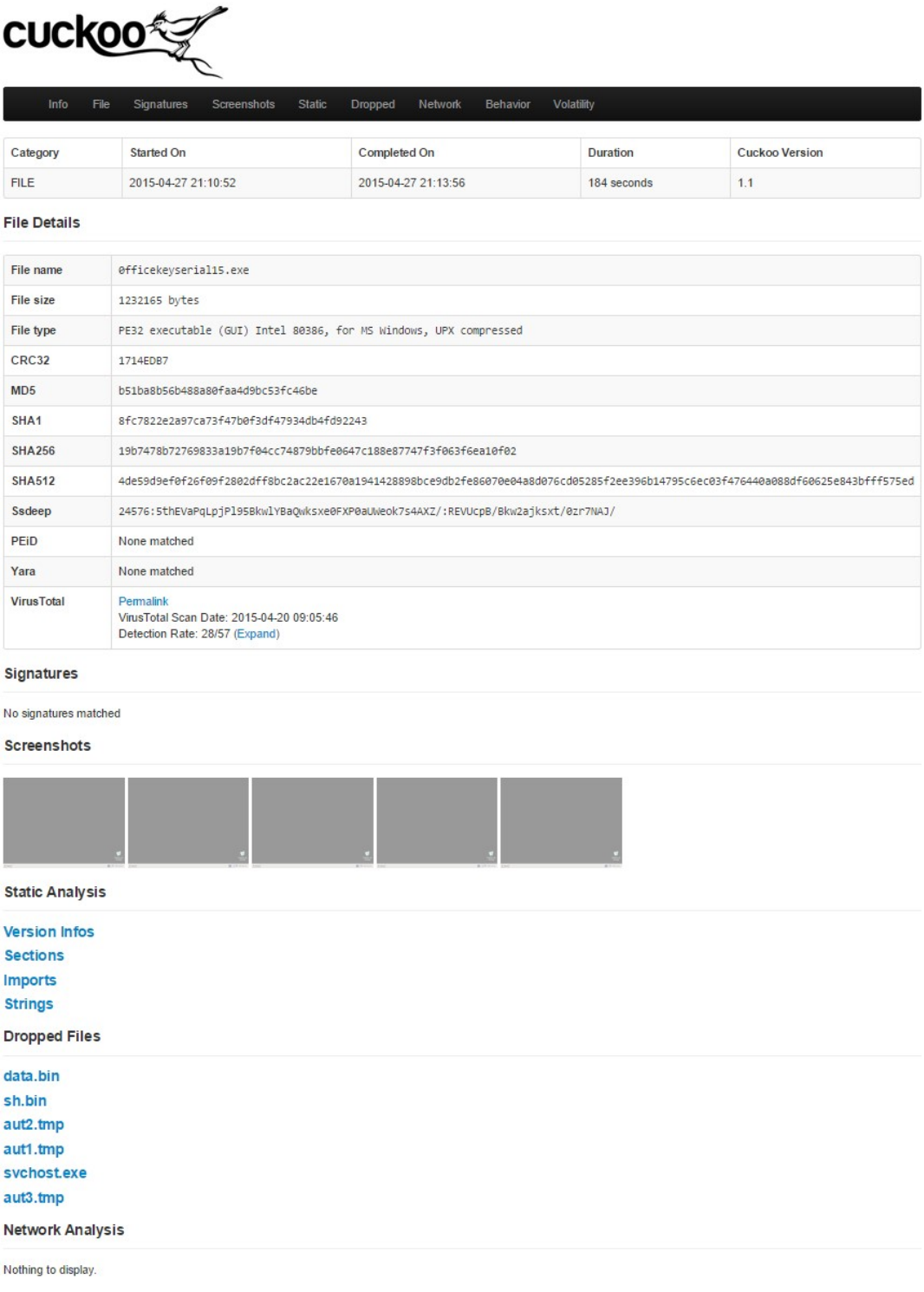
- kernel32.dll
- advapi32.dll
- comctl32.dll
- comdlg32.dll
- gdi32.dll
- mpr.dll
- ole32.dll
- oleaut32.dll
- psapi.dll
- shell32.dll
- user32.dll
- userenv.dll
- versión.dll
- wininet.dll
- winmm.dll
- wsock32.dll

De la misma manera, se observa que el archivo malicioso descargó algunos archivos adicionales. Los archivos descargados son:

- data.bin
- sh.bin
- aut1.tmp
- aut2.tmp
- svchost.exe
- aut3.tmp

No hay ningún comportamiento de red para mostrar en la sección análisis de red. Continuando con la revisión del reporte del evento 28801, es posible ver el historial de

los archivos que fueron modificados, información respecto a los mutexes, así como las llaves de registro.



cuckoo

Info File Signatures Screenshots Static Dropped Network Behavior Volatility

Category	Started On	Completed On	Duration	Cuckoo Version
FILE	2015-04-27 21:10:52	2015-04-27 21:13:56	184 seconds	1.1

File Details

File name	0fficekeyserial15.exe
File size	1232165 bytes
File type	PE32 executable (GUI) Intel 80386, for MS Windows, UPX compressed
CRC32	1714EDB7
MD5	b51ba8b56b488a80faa4d9bc53fc46be
SHA1	8fc7822e2a97ca73f47b0f3df47934db4fd92243
SHA256	19b7478b72769833a19b7f04cc74879bbfe0647c188e87747f3f063f6ea10f02
SHA512	4de59d9ef0f26f09f2802dff8bc2ac22e1670a1941428898bce9db2fe86070e04a8d076cd05285f2ee396b14795c6ec03f476440a088df60625e843bfff575ed
Ssdeep	24576:5thEVaPqLpjPl95BkwlYBaQwksxe0FXP0aUWeok7s4AXZ/:REVUcpB/Bkw2ajksxt/0zr7NAJ/
PEiD	None matched
Yara	None matched
VirusTotal	Permalink VirusTotal Scan Date: 2015-04-20 09:05:46 Detection Rate: 28/57 (Expand)

Signatures

No signatures matched

Screenshots

Static Analysis

Version Infos
Sections
Imports
Strings

Dropped Files

data.bin
sh.bin
aut2.tmp
aut1.tmp
svchost.exe
aut3.tmp

Network Analysis

Nothing to display.

Behavior Summary

Files

- C:\DOCUME~1\cuckoo\CONFIG~1\Temp\0fficekeyserial15.exe
- IDE#CdRomVBOX_CD-ROM_____1.0_____#42562d3231303037333036372020202020202020#{53f5630d-b6bf-11d0-94f2-00a0c91efb8b}
- MountPointManager
- STORAGE#Volume#1&30a96598&0&Signature49B049AFOffset7E00Length27F4DB200#{53f5630d-b6bf-11d0-94f2-00a0c91efb8b}
- C:\DOCUME~1
- C:\Documents and Settings\cuckoo
- C:\Documents and Settings\cuckoo\CONFIG~1
- C:\Documents and Settings\cuckoo\Configuraci\xc3\xb3n local\Temp
- C:\Documents and Settings\cuckoo\Configuraci\xc3\xb3n local\Temp\0fficekeyserial15.exe
- C:\DOCUME~1\cuckoo\CONFIG~1\Temp\test.a3x
- C:\DOCUME~1\cuckoo\CONFIG~1\Temp\aut1.tmp
- C:\Documents and Settings\cuckoo\Configuraci\xc3\xb3n local\Temp\test.a3x
- C:\DOCUME~1\cuckoo\CONFIG~1\Temp\data.bin
- C:\DOCUME~1\cuckoo\CONFIG~1\Temp\aut2.tmp
- C:\DOCUME~1\cuckoo\CONFIG~1\Temp\sh.bin
- C:\DOCUME~1\cuckoo\CONFIG~1\Temp\aut3.tmp
- C:\DOCUME~1\cuckoo\CONFIG~1\Temp\image.png
- c:\windows\system32\MSVBVM60.DLL
- C:\WINDOWS\Registration\R000000000007.clb
- PIPE\lsarpc
- C:\WINDOWS\system32\wbem\wbemdisp.TLB
- C:\WINDOWS\system32\stdole2.tlb
- C:\Documents and Settings\cuckoo\Datos de programa\\svchost.exe
- C:\Documents and Settings\cuckoo\Datos de programa\svchost.exe
- C:\WINDOWS\system32\wshom.ocx
- C:\WINDOWS\ProcessHacker.exe
- C:\WINDOWS\SysWOW64\svchost.exe
- C:\WINDOWS\kprocesshacker.sys

Mutexes

- CTF.TimListCache.FMPDefaultS-1-5-21-861567501-813497703-1202660629-1003MUTEX.DefaultS-1-5-21-861567501-813497703-1202660629-1003

Registry Keys

- HKEY_CURRENT_USER\Control Panel\Mouse
- HKEY_CURRENT_USER\Software\AutoIt v3\AutoIt
- HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer
- HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer
- HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\ShellCompatibility\Objects\{20D04FE0-3AEA-1069-A2D8-08002B30309D}
- HKEY_CLASSES_ROOT\CLSID\{20D04FE0-3AEA-1069-A2D8-08002B30309D}\InProcServer32
- HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\MountPoints2\CPC\Volume
- HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\MountPoints2\CPC\Volume\{48bfff50-781f-11e4-8ef1-806d6172696f}\
- HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\MountPoints2\CPC\Volume\{48bfff52-781f-11e4-8ef1-806d6172696f}\
- HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\MountPoints2\{48bfff52-781f-11e4-8ef1-806d6172696f}\
- HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\MountPoints2\{48bfff50-781f-11e4-8ef1-806d6172696f}\
- HKEY_CLASSES_ROOT\Drive\shellex\FolderExtensions
- HKEY_CLASSES_ROOT\Drive\shellex\FolderExtensions\{fbeb8a05-beee-4442-804e-409d6c4515e9}
- HKEY_CLASSES_ROOT\Directory
- HKEY_CLASSES_ROOT\Directory\CurVer
- HKEY_CLASSES_ROOT\Directory\
- HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer
- HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\
- HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Policies\System
- HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\Advanced
- HKEY_CLASSES_ROOT\Directory\\ShellEx\IconHandler
- HKEY_CLASSES_ROOT\Directory\\Clsid
- HKEY_CLASSES_ROOT\Folder
- HKEY_CLASSES_ROOT\Folder\Clsid
- HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\FileExts
- HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\FileExts\.exe
- HKEY_CLASSES_ROOT\.exe
- HKEY_CLASSES_ROOT\exefile
- HKEY_CLASSES_ROOT\exefile\CurVer
- HKEY_CLASSES_ROOT\exefile\
- HKEY_CLASSES_ROOT\exefile\\ShellEx\IconHandler
- HKEY_CLASSES_ROOT\SystemFileAssociations\.exe
- HKEY_CLASSES_ROOT\SystemFileAssociations\application
- HKEY_CLASSES_ROOT\exefile\\Clsid
- HKEY_CLASSES_ROOT*
- HKEY_CLASSES_ROOT*\Clsid
- HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\FileExts\.a3x
- HKEY_CLASSES_ROOT\.a3x
- HKEY_CLASSES_ROOT\SystemFileAssociations\.a3x
- HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Session Manager
- HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\VBA\Monitors
- HKEY_CLASSES_ROOT\winmgmts
- HKEY_CLASSES_ROOT\winmgmts\CLSID
- HKEY_LOCAL_MACHINE\Software\Microsoft\COM3

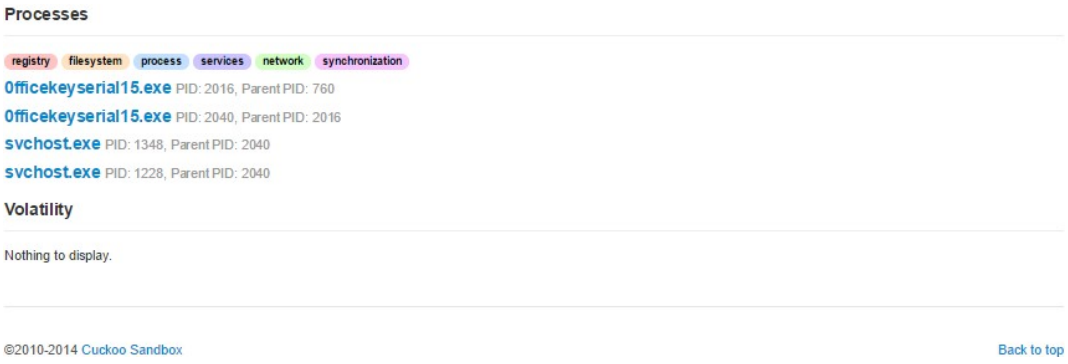


Figura 3.24: Análisis con Cuckoo SandBox del evento 28801. Fuente: Captura propia.

Al final del informe, mostrado en la figura anterior, se encuentran todos los procesos involucrados durante el análisis del binario malicioso. Estos procesos se encuentran separados por categorías. Esta tabla puede ser consultada seleccionando cada uno de los diferentes procesos que fueron creados durante la ejecución de la muestra.

Snort Alert [1:27919:3]

Derivado del evento 27919 generado por el IDS Snort, se realizó un análisis en Cuckoo Sandbox de la URL *www.acaciadeperus.com.br/home/po/gate.php*. Como se observa en la figura 3.25. el informe se realizó en Cuckoo Sandbox versión 1.1, detallando el tipo de objeto que fue analizado, en este caso una URL. También se muestra la fecha de inicio y fin del análisis. El tiempo total del análisis fue de 2 minutos y 54 segundos.

Por tratarse de un objeto de tipo URL, en la sección de Virus Total, al realizar el desglose de los motores antivirus que detectaron el dominio como malicioso, se especifica que 3 motores antivirus de 62 fueron capaces de identificar la URL como un sitio malicioso. Los antivirus que determinaron que el sitio es malicioso fueron:

- BitDefender
- Kaspersky
- Websense ThreatSeeker

Category	Started On	Completed On	Duration	Cuckoo Version
URL	2015-04-27 21:31:54	2015-04-27 21:34:48	174 seconds	1.1

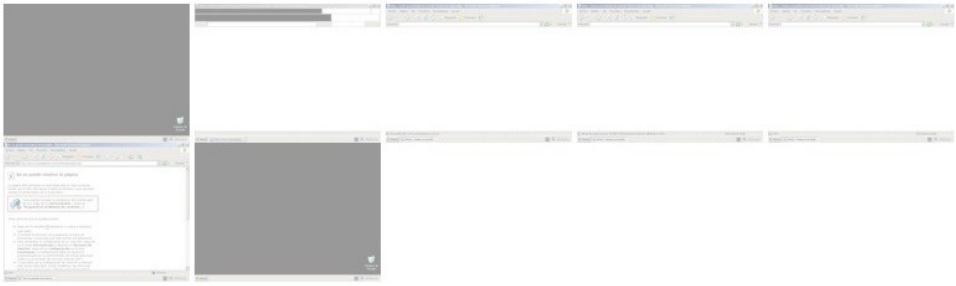
URL Details

URL	www.acaciadeperus.com.br/home/po/gate.php
VirusTotal	Permalink VirusTotal Scan Date: 2015-02-24 05:23:31 Detection Rate: 3/62 (Expand)

Signatures

No signatures matched

Screenshots



Dropped Files

Nothing to display.

Network Analysis

Nothing to display.

Behavior Summary

Files

- C:\Documents and Settings\cuckoo\Escritorio
- C:\WINDOWS\Registration\R000000000007.clb
- C:\Documents and Settings\cuckoo\Configuraci\xc3\xb3n local\Archivos temporales de Internet
- C:\Documents and Settings\cuckoo\Configuraci\xc3\xb3n local\Historial
- C:\Documents and Settings\cuckoo\Configuraci\xc3\xb3n local\Archivos temporales de Internet\Content.IE5\
- C:\
- C:\Documents and Settings\cuckoo\Configuraci\xc3\xb3n local\Archivos temporales de Internet\Content.IE5\index.dat
- C:\Documents and Settings\cuckoo\Cookies\
- C:\Documents and Settings\cuckoo\Cookies\index.dat
- C:\Documents and Settings\cuckoo\Configuraci\xc3\xb3n local\Historial\History.IE5\
- C:\Documents and Settings\cuckoo\Configuraci\xc3\xb3n local\Historial\History.IE5\index.dat
- C:\WINDOWS\System32\cscui.dll
- shadow
- IDE#CdRomVBOX_CD-ROM_____1.0_____#42562d3231303037333036372020202020202020#{53f5630d-b6bf-11d0-94f2-00a0c91efb8b}
- MountPointManager

Mutexes

- CTF.TimListCache.FMPDefaultS-1-5-21-861567501-813497703-1202660629-1003MUTEX.DefaultS-1-5-21-861567501-813497703-1202660629-1003
- Shell.CMruPidlList
- WininetStartupMutex
- _!MSFTHISTORY!_
- c:!documents and settings!cuckoo!configuraci\xc3\xb3n local!archivos temporales de internet!content.ie5!
- c:!documents and settings!cuckoo!cookies!
- c:!documents and settings!cuckoo!configuraci\xc3\xb3n local!historial!history.ie5!
- WininetConnectionMutex
- WininetProxyRegistryMutex
- ShimCacheMutex
- MSCTF.Shared.MUTEX.IMF

Registry Keys

- HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Explorer\Performance
- HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\BrowseNewProcess
- HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Explorer\BrowseNewProcess
- HKEY_LOCAL_MACHINE\Software\Policies\Microsoft\Internet Explorer\Restrictions
- HKEY_CURRENT_USER\Software\Policies\Microsoft\Internet Explorer\Restrictions
- HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\User Shell Folders
- HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\Shell Folders
- HKEY_LOCAL_MACHINE\Software\Microsoft\Internet Explorer
- HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer
- HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\Streams
- HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\
- HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer
- HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer
- HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Policies\System
- HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\Advanced
- HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\CabinetState
- HKEY_LOCAL_MACHINE\Software\Microsoft\Internet Explorer\International
- HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Session Manager\AppCompatibility
- HKEY_LOCAL_MACHINE\Software\Microsoft\COM3
- HKEY_USERS\S-1-5-21-861567501-813497703-1202660629-1003_Classes
- HKEY_LOCAL_MACHINE\Software\Classes
- \REGISTRY\USER
- HKEY_LOCAL_MACHINE\Software\Classes\CLSID
- CLSID\{A5E46E3A-8849-11D1-9D8C-00C04FC99D61}
- CLSID\{A5E46E3A-8849-11D1-9D8C-00C04FC99D61}\TreatAs
- \CLSID\{A5E46E3A-8849-11D1-9D8C-00C04FC99D61}
- \CLSID\{A5E46E3A-8849-11D1-9D8C-00C04FC99D61}\InprocServer32
- \CLSID\{A5E46E3A-8849-11D1-9D8C-00C04FC99D61}\InprocServerX86
- \CLSID\{A5E46E3A-8849-11D1-9D8C-00C04FC99D61}\LocalServer32
- \CLSID\{A5E46E3A-8849-11D1-9D8C-00C04FC99D61}\InprocHandler32
- \CLSID\{A5E46E3A-8849-11D1-9D8C-00C04FC99D61}\InprocHandlerX86
- \CLSID\{A5E46E3A-8849-11D1-9D8C-00C04FC99D61}\LocalServer
- HKEY_CLASSES_ROOT\CLSID\{A5E46E3A-8849-11D1-9D8C-00C04FC99D61}
- HKEY_CLASSES_ROOT\CLSID\{A5E46E3A-8849-11D1-9D8C-00C04FC99D61}\TreatAs
- HKEY_CURRENT_USER\SOFTWARE\Microsoft\Internet Explorer\Security\P3Global
- HKEY_CURRENT_USER\SOFTWARE\Microsoft\Internet Explorer\Security\P3Sites
- HKEY_CLASSES_ROOT\.htm
- HKEY_CLASSES_ROOT\.html
- HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\URL\Prefixes
- HKEY_CLASSES_ROOT\http
- HKEY_CLASSES_ROOT\dummy

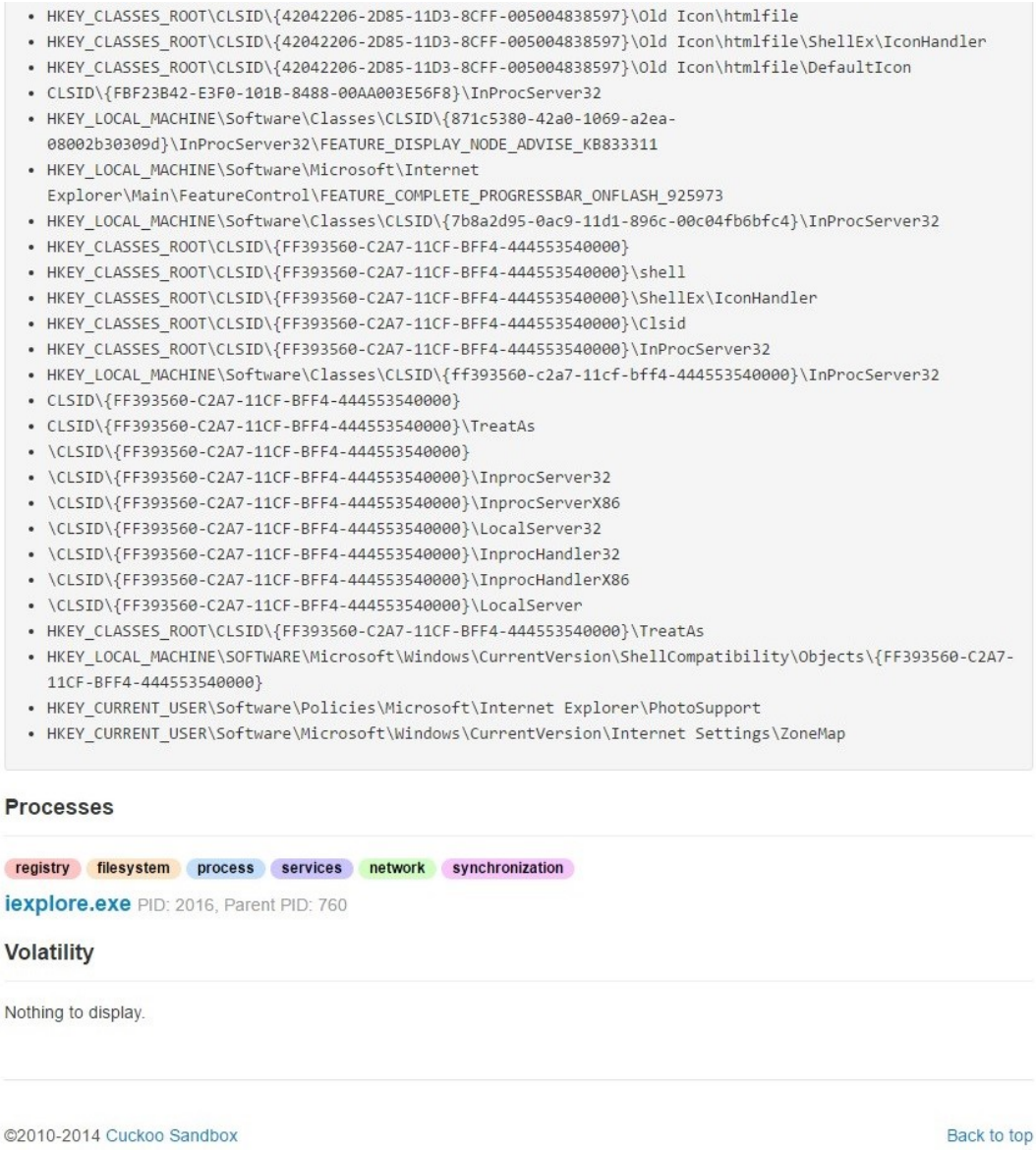


Figura 3.25: Análisis con Cuckoo SandBox del evento 27919. Fuente: Captura propia.

Continuando con el reporte de Cuckoo SandBox, se establece que no fueron descargados archivos adicionales y que por ende la sección de análisis de red no tiene información por mostrar.

En el resúmen del comportamiento de la muestra se despliegan los archivos modificados durante su análisis. Por otra parte, continuando con los detalles del reporte generado, son informadas las exclusiones mutuas así como las modificaciones en el registro de Windows.

Para poder realizar el análisis de la URL tuvo que realizarse sobre el navegor Internet Explorer con su correspondiente proceso iexplore.exe con PID 2016.

Snort Alert [1:32125:1]
El reporte mostrado en la figura 3.26, corresponde al análisis del evento 32125 generado por el IDS Snort. A continuación se explicará el contenido del reporte que arrojó Cuckoo SandBox al analizar el dominio *update.kele55.com*.

En la primera parte del reporte de Cuckoo, se muestran los detalles del análisis realizado sobre la muestra, que en este caso en particular, se trató de una URL. Se establece la fecha de inicio y término así como la duración del análisis de la URL. De la misma manera que los reportes mostrados anteriormente, el análisis de la muestra se realizó sobre la versión 1.1 de Cuckoo SandBox.

En la sección de los detalles del análisis de la URL, se establece que únicamente 1 antivirus detectó el dominio como un sitio web malicioso. Posterior de los detalles anteriores, son mostradas las capturas de pantalla durante el análisis de la URL y se informa que no fue descargado ningún archivo ni que se realizaron conexiones a sitios remotos.

En la sección resúmen del comportamiento de la muestra durante el análisis, es mostrada la información sobre los archivos que sufrieron modificaciones en la máquina huésped durante el proceso de análisis de la URL, así como las exclusiones mutuas y las modificaciones en el registro de Windows.

Al final del reporte de Cuckoo SandBox se muestra el identificador del proceso iexplore.exe, que permitió poder examinar la URL a través de Internet Explorer.

Category	Started On	Completed On	Duration	Cuckoo Version
URL	2015-05-25 16:38:19	2015-05-25 16:41:25	186 seconds	1.1

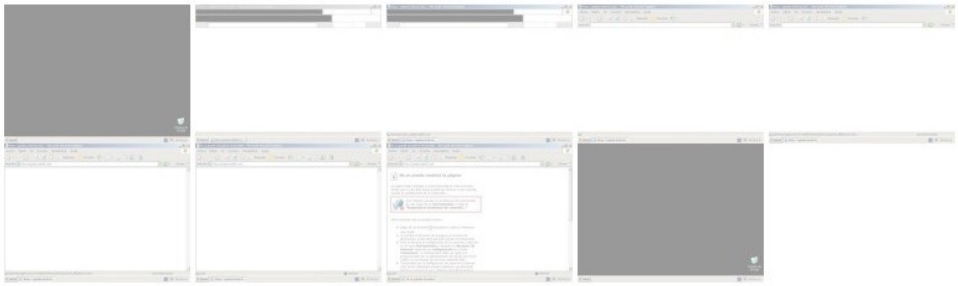
URL Details

URL	update.kele55.com
VirusTotal	Permalink VirusTotal Scan Date: 2015-01-15 08:20:24 Detection Rate: 1/61 (Expand)

Signatures

No signatures matched

Screenshots



Dropped Files

Nothing to display.

Network Analysis

Nothing to display.

Behavior Summary

Files

- C:\Documents and Settings\cuckoo\Escritorio
- C:\WINDOWS\Registration\R000000000007.clb
- C:\Documents and Settings\cuckoo\Configuraci\xc3\xb3n local\Archivos temporales de Internet
- C:\Documents and Settings\cuckoo\Configuraci\xc3\xb3n local\Historial
- C:\Documents and Settings\cuckoo\Configuraci\xc3\xb3n local\Archivos temporales de Internet\Content.IE5\
- C:\
- C:\Documents and Settings\cuckoo\Configuraci\xc3\xb3n local\Archivos temporales de Internet\Content.IE5\index.dat
- C:\Documents and Settings\cuckoo\Cookies\
- C:\Documents and Settings\cuckoo\Cookies\index.dat
- C:\Documents and Settings\cuckoo\Configuraci\xc3\xb3n local\Historial\History.IE5\
- C:\Documents and Settings\cuckoo\Configuraci\xc3\xb3n local\Historial\History.IE5\index.dat
- C:\WINDOWS\System32\cscui.dll
- shadow
- IDE#CdRomVBOX_CD-ROM_____1.0_____#42562d3231303037333036372020202020202020#{53f5630d-b6bf-11d0-94f2-00a0c91efb8b}
- MountPointManager
- STORAGE#Volume#1&30a96598&0&Signature49B049AFOffset7E00Length27F4DB200#{53f5630d-b6bf-11d0-94f2-00a0c91efb8b}
- C:\Documents and Settings
- C:\Documents and Settings\cuckoo
- C:\Documents and Settings\cuckoo\Favoritos
- C:\Documents and Settings\cuckoo\Favoritos\desktop.ini
- C:\Documents and Settings\cuckoo\Favoritos\V\xc3\xadnculos
- C:\Documents and Settings\cuckoo\Configuraci\xc3\xb3n local
- C:\Documents and Settings\cuckoo\Configuraci\xc3\xb3n local\Archivos temporales de Internet\desktop.ini
- C:\Documents and Settings\cuckoo\Favoritos\V\xc3\xadnculos*.*

Mutexes

- CTF.TimListCache.FMPDefaultS-1-5-21-861567501-813497703-1202660629-1003MUTEX.DefaultS-1-5-21-861567501-813497703-1202660629-1003
- Shell.CMruPidlList
- WininetStartupMutex
- _!MSFTHISTORY!_
- c:!documents and settings!cuckoo!configuraci\xc3\xb3n local!archivos temporales de internet!content.ie5!
- c:!documents and settings!cuckoo!cookies!
- c:!documents and settings!cuckoo!configuraci\xc3\xb3n local!historial!history.ie5!
- WininetConnectionMutex
- WininetProxyRegistryMutex
- ShimCacheMutex
- MSCTF.Shared.MUTEX.IMF

Registry Keys

- HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Explorer\Performance
- HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\BrowseNewProcess
- HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Explorer\BrowseNewProcess
- HKEY_LOCAL_MACHINE\Software\Policies\Microsoft\Internet Explorer\Restrictions
- HKEY_CURRENT_USER\Software\Policies\Microsoft\Internet Explorer\Restrictions
- HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\User Shell Folders
- HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\Shell Folders
- HKEY_LOCAL_MACHINE\Software\Microsoft\Internet Explorer
- HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer
- HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\Streams
- HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\
- HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer
- HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer
- HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Policies\System
- HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\Advanced
- HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\CabinetState
- HKEY_LOCAL_MACHINE\Software\Microsoft\Internet Explorer\International
- HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Session Manager\AppCompatibility
- HKEY_LOCAL_MACHINE\Software\Microsoft\COM3
- HKEY_USERS\S-1-5-21-861567501-813497703-1202660629-1003_Classes
- HKEY_LOCAL_MACHINE\Software\Classes
- \REGISTRY\USER
- HKEY_LOCAL_MACHINE\Software\Classes\CLSID
- CLSID\{A5E46E3A-8849-11D1-9D8C-00C04FC99D61}
- CLSID\{A5E46E3A-8849-11D1-9D8C-00C04FC99D61}\TreatAs
- \CLSID\{A5E46E3A-8849-11D1-9D8C-00C04FC99D61}
- \CLSID\{A5E46E3A-8849-11D1-9D8C-00C04FC99D61}\InprocServer32
- \CLSID\{A5E46E3A-8849-11D1-9D8C-00C04FC99D61}\InprocServerX86
- \CLSID\{A5E46E3A-8849-11D1-9D8C-00C04FC99D61}\LocalServer32
- \CLSID\{A5E46E3A-8849-11D1-9D8C-00C04FC99D61}\InprocHandler32
- \CLSID\{A5E46E3A-8849-11D1-9D8C-00C04FC99D61}\InprocHandlerX86
- \CLSID\{A5E46E3A-8849-11D1-9D8C-00C04FC99D61}\LocalServer
- HKEY_CLASSES_ROOT\CLSID\{A5E46E3A-8849-11D1-9D8C-00C04FC99D61}
- HKEY_CLASSES_ROOT\CLSID\{A5E46E3A-8849-11D1-9D8C-00C04FC99D61}\TreatAs
- HKEY_CURRENT_USER\SOFTWARE\Microsoft\Internet Explorer\Security\P3Global
- HKEY_CURRENT_USER\SOFTWARE\Microsoft\Internet Explorer\Security\P3Sites
- HKEY_CLASSES_ROOT\.htm
- HKEY_CLASSES_ROOT\.html
- HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\URL\Prefixes
- HKEY_CURRENT_USER\Software\Microsoft\Internet Explorer\URLSearchHooks
- HKEY_LOCAL_MACHINE\Software\Classes\CLSID\{cfbfae00-17a6-11d0-99cb-00c04fd64497}\InProcServer32

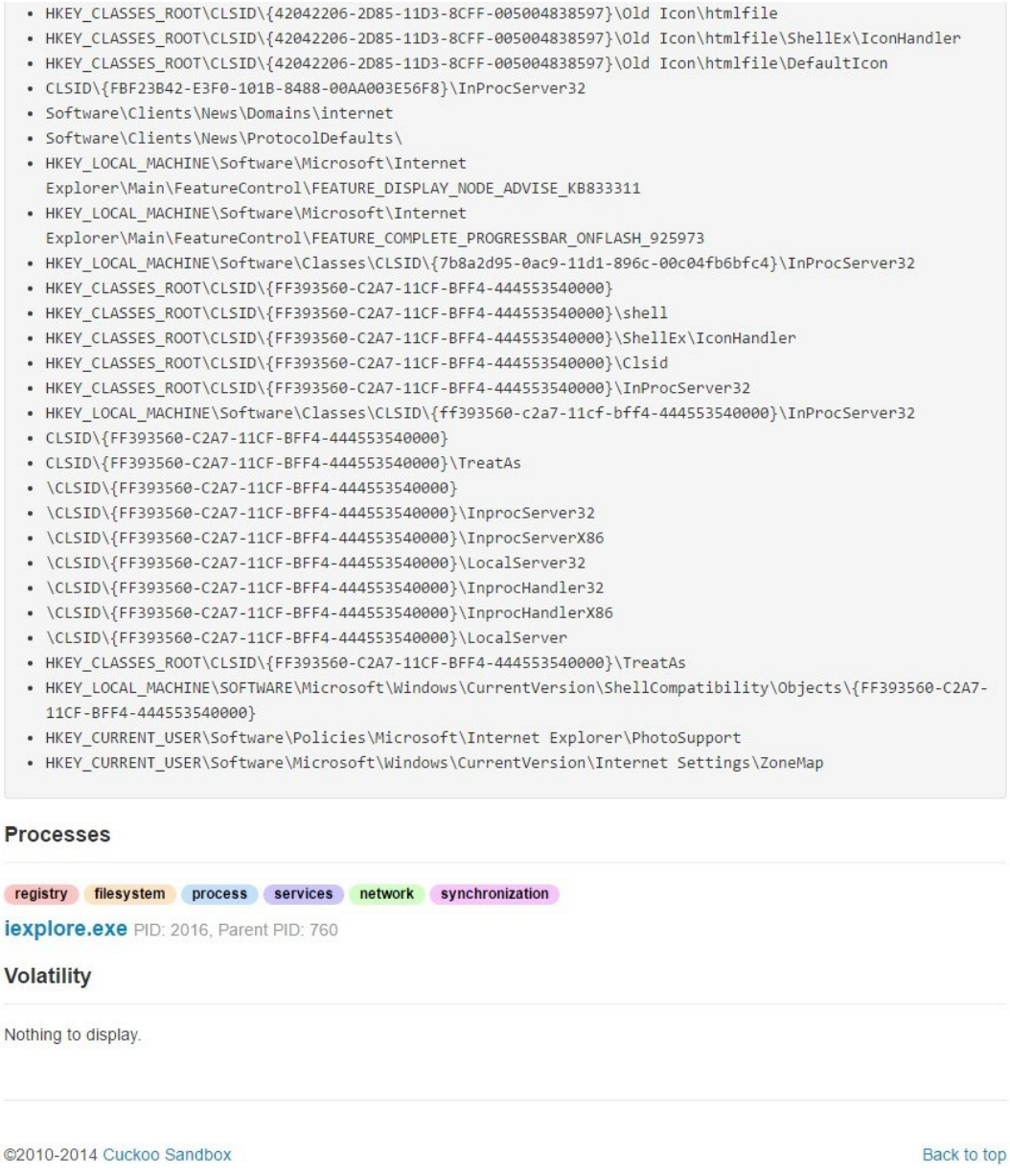


Figura 3.26: Análisis con Cuckoo SandBox del evento 32125. Fuente: Captura propia.

Respecto a los eventos 30211, 28423 y 31527, los cuales fueron alertados por el IDS Snort, en el payload capturado de éstos, no fue encontrada información útil como para poder obtener una muestra y realizar el proceso de análisis de las muestras con la SandBox.

Capítulo 4

Automatización del proceso de análisis de un evento

En este capítulo se explicará el proceso automatizado de análisis con la SandBox Cuckoo de un evento alertado por el IDS Snort.

Para el administrador de una red este proceso simplifica, en gran medida, el análisis de una posible amenaza que atente contra los activos críticos de una organización. Así de una manera sencilla y eficaz, podrá determinar sí se trata de un falso positivo o en realidad se trata de un software malicioso detectado en la intranet de la organización que él administra, para que pueda contenerlo y evitar así su propagación. El análisis de la muestra que contiene el evento alertado por el IDS Snort, es un proceso delicado, ya que sí se realiza en un ambiente físico y de producción; podría propagarse a otros equipos y sistemas; pudiendo poner en riesgo los activos y la información que reside en éstos.

Para alcanzar este objetivo, se realizó un programa en shell de Unix, el cual tiene la función principal de obtener el archivo del payload del paquete descargado desde el IDS de Snort o de un archivo con formato pcap, el cual se obtiene a través de cualquier sniffer de paquetes como Wireshark, Windump entre otros. Una vez que dicho archivo ejecutable ha sido obtenido desde el payload del paquete o desde el archivo pcap, éste puede ser enviado de manera inmediata a la SandBox Cuckoo para su análisis en un entorno seguro y aislado; evitando así su propagación.

Dicho programa fue realizado para que funcione en la versión *2.9.6.2 del IDS de Snort*, *Cuckoo SandBox 1.1*, y que ambos se encuentren instalados en un sistema *Ubuntu 12.04 Precise Pangolin*. Es posible que sí se instalan versiones diferentes de los componentes mencionados anteriormente, el script no funcione correctamente, por lo que se recomienda adaptarlo a nuevas versiones, para su correcto funcionamiento.

Otras tareas que se pueden realizar a través de este programa son las siguientes:

- Eliminar los eventos de la Base de Datos del IDS Snort.
- Comprobar el estado de Cuckoo SandBox.
- Enviar una muestra para su análisis con Cuckoo SandBox.
- Obtener información del repositorio de reportes a partir de las muestras analizadas con Cuckoo SandBox, para su despliegue a través de una página web.
- Habilitar la interfaz web de Cuckoo SandBox.

Para facilitar la lectura de los resultados del análisis de la muestra, se diseñó una página web, en la que se despliegan los datos más relevantes que pueden dar un indicio al administrador de red, que el IDS ha detectado una potencial amenaza, por lo cual debe actuar inmediatamente para su mitigación de los equipos infectados y así evitar que se siga propagando o inclusive que usuarios maliciosos puedan tomar control del equipo infectado.

4.1. Estructura y módulos del programa de automatización

El programa de automatización fue desarrollado en Shell de Unix, debido a su flexibilidad, rapidez y eficiencia con la que se pueden realizar tareas asignadas por parte del usuario. El script hace uso de las herramientas GREP, SED y AWK, las cuales en conjunto permiten el procesamiento de archivos e información basada en texto mediante el uso de expresiones regulares. Como se mencionó anteriormente, el programa consta de 6 módulos y en cada uno de ellos se pueden realizar diferentes tareas, las cuales se pueden observar en la figura 4.1.

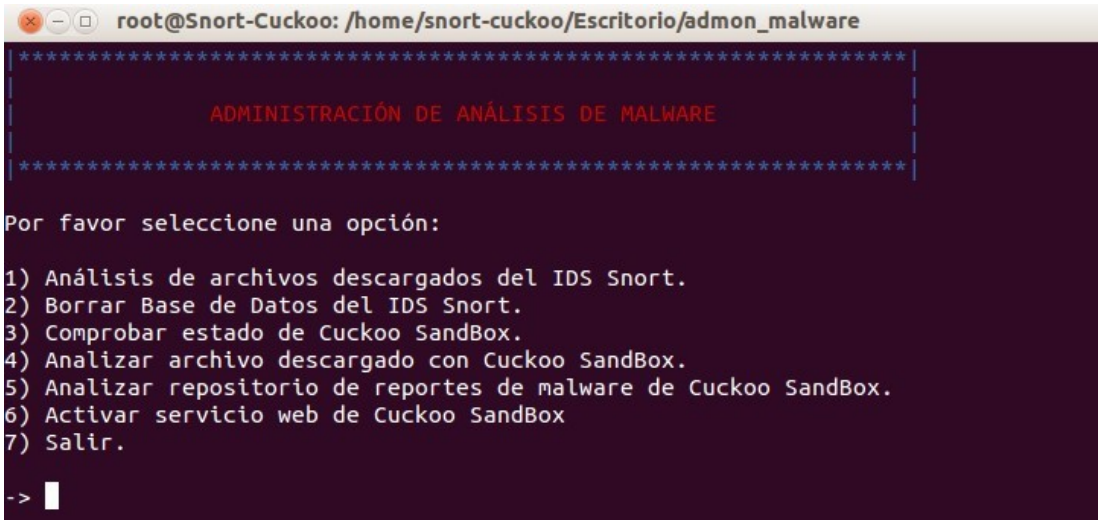


Figura 4.1: Módulos principales. Fuente: Captura propia.

4.1.1. Módulo de análisis de archivos descargados del IDS Snort

Este módulo se encarga de obtener el archivo que se encuentra en el payload de un paquete o desde un archivo con formato pcap. Tanto el payload del paquete como el archivo .pcap son generados de forma automática cuando el IDS emite una alerta de un evento que ha sido detectado como una posible amenaza, por lo que ambos tipos de archivos pueden ser descargados a través de la interfaz gráfica (B.A.S.E.) del IDS Snort.

Como se muestra en la figura 4.2, una vez que el usuario ha seleccionado el módulo de análisis de archivos descargados, debe elegir sí desea analizar el payload de un paquete (archivo .bin) o sí requiere analizar un archivo .pcap.

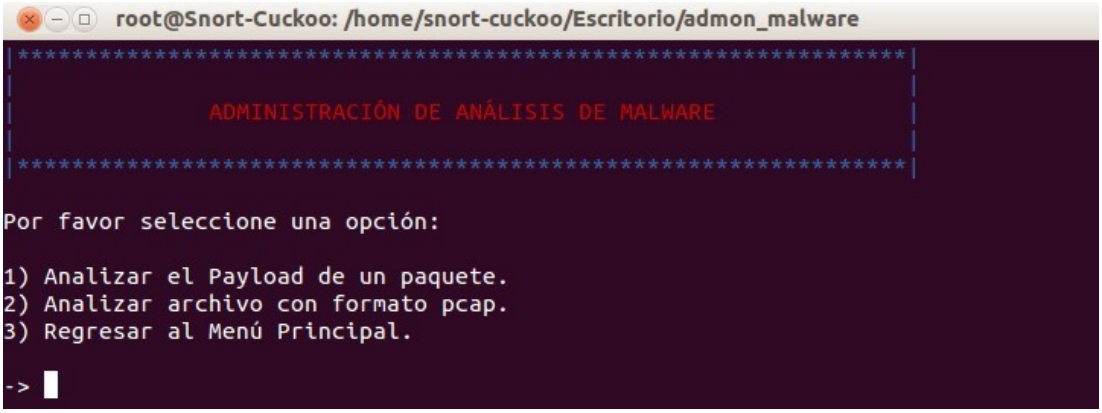


Figura 4.2: Análisis de archivo con extensión .bin o .pcap. Fuente: Captura propia.

En las páginas 205 y 209 del apéndice B se presentan las funciones *payload_paquetes()* y *archivos_pcap()* las cuales se encargan de realizar el análisis de los archivos .bin o .pcap para la descarga del archivo que fue detectado como malicioso. Sí la descarga ha sido exitosa, se le indicará al usuario la carpeta en la que ha sido guardada la muestra, como se muestra a continuación en la figura 4.3:

Figura 4.3: Descarga de archivo desde un archivo .bin y .pcap. Fuente: Captura propia.

Después de 5 segundos, aparecerá un nuevo módulo en el que se le indica al usuario sí desea realizar el análisis de la muestra en la SandBox. Para poder realizar dicho análisis, Cuckoo SandBox debe estar ejecutándose y la máquina virtual invitada en la que se analizará la muestra debe estar escuchando para que le sean asignadas tareas de análisis. Para comprobar lo anterior, el usuario deberá seleccionar el módulo número 3, el cual se explicará más adelante.

Finalmente al terminar este proceso de análisis de archivos descargados del IDS Snort, el usuario elegirá sí desea repetir el proceso para otro archivo .pcap o .bin; por lo que en caso de no repetir el proceso anterior, regresará al menú principal.

4.1.2. Módulo para eliminar los incidentes de la Base de Datos del IDS Snort

Como se mencionó en el capítulo 2, el IDS Snort registra los incidentes en una Base de Datos MySQL, para que puedan ser revisadas y analizadas las alertas a través de la aplicación B.A.S.E.

Este módulo permite eliminar todos los registros de los incidentes almacenados en las tablas que contiene la Base de Datos Snort. Es importante mencionar que una vez que se proceda con esta tarea, no se podrá recuperar ningún evento capturado por el IDS Snort.

Este módulo se basa en la función llamada *borrar_base_de_datos()* (apéndice B, página 216) para poder realizar la eliminación de los registros capturados por el IDS Snort. A su vez, esta función hace uso de dos programas adicionales, los cuales en conjunto

permiten eliminar todos los registros tanto en las tablas de la base de datos, así como los archivos generados por Snort. Uno está realizado en shell de Unix, mientras que el otro está desarrollado en PHP.

El primer programa del que se apoya la función anterior es *eliminar_eventos_ids*, el cual tiene la función de eliminar todos los archivos generados por las alertas del IDS, así como ejecutar el programa realizado en PHP para eliminar los valores de las tablas de la Base de Datos. Por otro lado, el programa *eliminar_tablas_ids.php*, realiza una conexión a MySQL, en la cual se envía una serie de consultas para la eliminación de datos de las siguientes tablas:

- icmphdr
- opt
- event
- data
- signature
- iphdr
- tcphdr
- acid_event

Una vez que se ha seleccionado este módulo, aparecerá un mensaje de confirmación para la eliminación de las tablas como se muestra en la figura 4.4:

Figura 4.4: Mensaje para la eliminación de eventos de las tablas de la Base de Datos de Snort. Fuente: Captura propia.

Cuando se confirma la eliminación de los eventos de la Base de Datos, comenzará el proceso de borrado de los archivos que contienen las alertas del IDS Snort, así como los valores de las tablas de la Base de Datos Snort. El tiempo que se tome en eliminar los eventos detectados por el IDS Snort, dependerá de la cantidad de información que contengan las tablas de la base de datos.

Una vez terminado el proceso anterior, se mostrará un mensaje como el de la figura 4.5, en la que se confirma el número de elementos borrados que contenían las tablas mencionadas anteriormente.

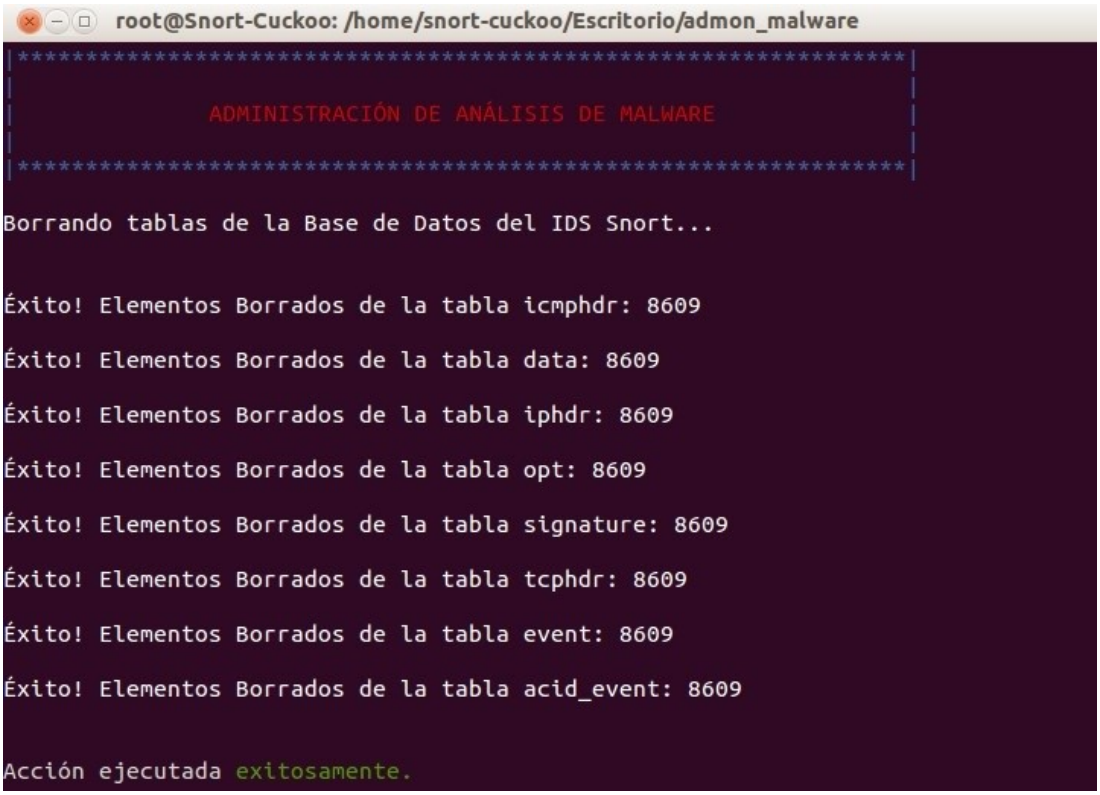


Figura 4.5: Borrado de información de las tablas de la Base de Datos Snort. Fuente: Captura propia.

Finalmente el usuario será dirigido al menú principal para consultar otros módulos y realizar otras tareas.

4.1.3. Módulo para comprobar que Cuckoo SandBox está en ejecución

El módulo para comprobar el estado de Cuckoo SandBox es fundamental para el análisis automatizado de muestras, ya que sí la SandBox no está ejecutándose, o la máquina virtual huésped no se encuentra en espera de tareas de análisis, será imposible realizar el análisis del comportamiento de archivos sospechosos. Mediante este módulo se pueden resolver estos inconvenientes.

Al ejecutar este módulo se permite determinar si los componentes de la SandBox Cuckoo se encuentran listos y preparados para poder realizar el análisis de las muestras obtenidas. En el caso de que Cuckoo SandBox se encuentre ejecutándose, este módulo proporcionará el identificador del proceso para conocer el proceso correspondiente. En la figura 4.6 se comprueba que Cuckoo se está ejecutando correctamente, al igual que la máquina virtual huésped, donde se analizarán las muestras. Cabe aclarar que sí existe

un problema con la interfaz virtual de la máquina huésped o la misma máquina virtual, Cuckoo no puede iniciarse, por lo que no podría tener un PID asociado que identifique su correcta ejecución.

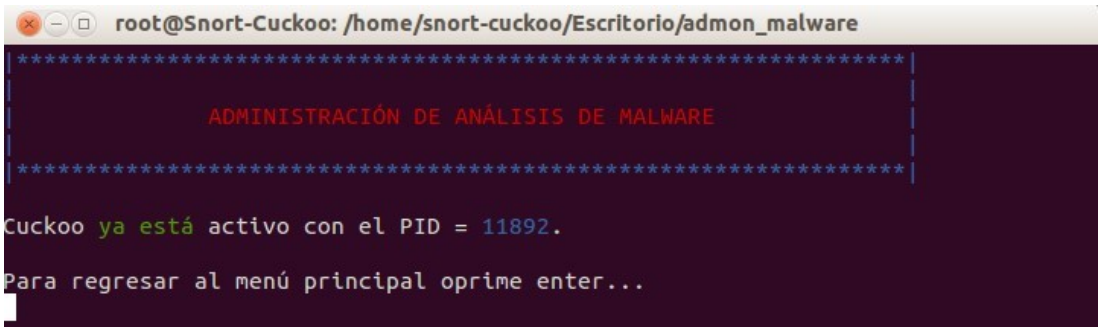


Figura 4.6: Cuckoo SandBox ejecutándose correctamente. Fuente: Captura propia.

Sí la SandBox no se encuentra en ejecución, este módulo permite notificarle esto al usuario por lo que podrá ejecutarse de manera automática, como es mostrado a continuación en la figura 4.7:

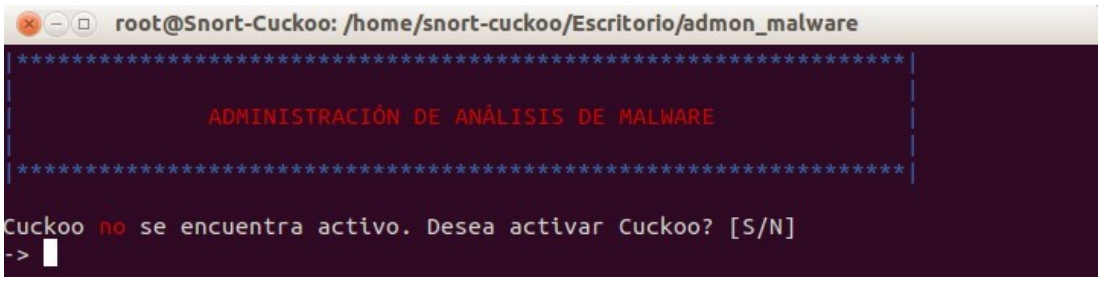


Figura 4.7: Notificación de que Cuckoo SandBox no está actualmente en ejecución.

Fuente: Captura propia.

Una vez que el usuario ha indicado que desea ejecutar Cuckoo SandBox, se desplegará una pantalla como la que se muestra en la figura 4.8, confirmando que la SandBox está lista para poder realizar el proceso de analizar archivos sospechosos automáticamente.

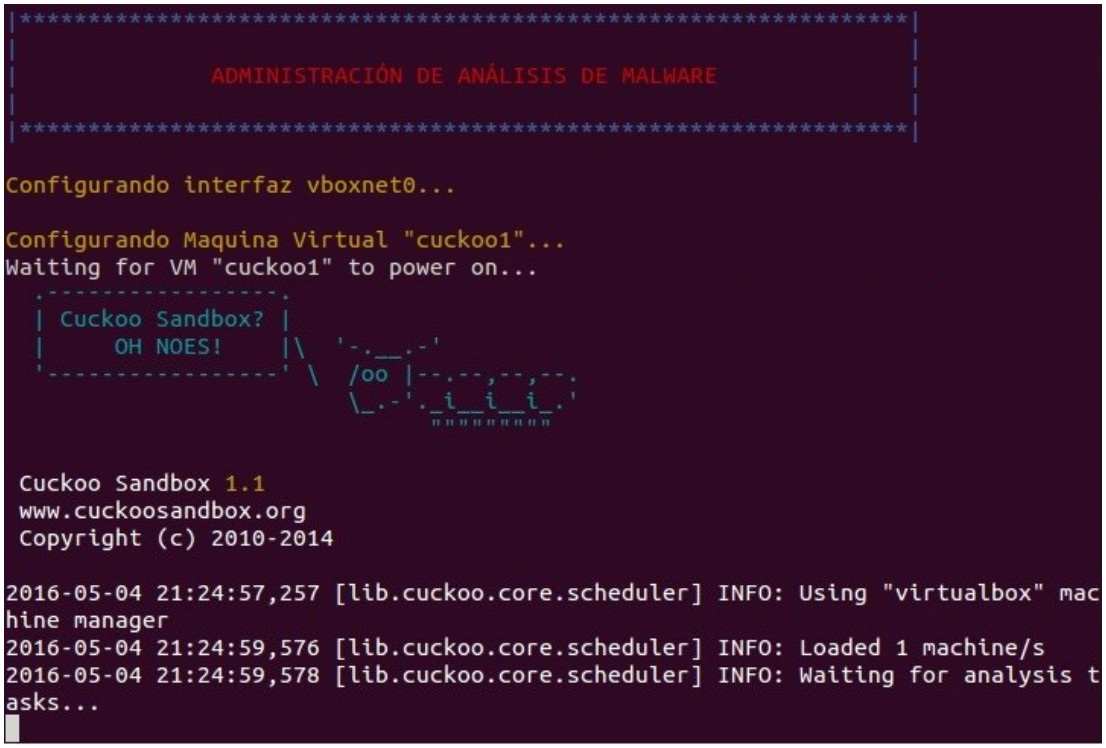


Figura 4.8: Activación de Cuckoo SandBox. Fuente: Captura propia.

Finalmente, la función *status_cuckoo()* proporciona información sobre el PID de Cuckoo y que permite comprobar la correcta de ejecución de Cuckoo SandBox, el cual puede ser consultado en la página 217 del apéndice B.

4.1.4. Módulo para enviar una muestra para su análisis con Cuckoo SandBox

Este módulo, depende de que Cuckoo SandBox se encuentre en ejecución. Como se explicó previamente, la función del módulo anterior es validar la correcta ejecución de Cuckoo SandBox, por lo que será de apoyo para poder enviar una muestra a la SandBox para su análisis.

Sí se hace uso de este módulo y la SandBox no está ejecutándose, se mostrará un mensaje en el que se indica que se puede iniciar la ejecución de Cuckoo en la opción 3 del menú principal.

Una vez que la SandBox se encuentra lista para que le sean asignadas tareas de análisis, como se muestra en la figura 4.9, se debe ingresar el nombre de la muestra que se desea analizar, y que no necesariamente tiene que ser un archivo obtenido de

los archivos descargados a través de B.A.S.E., ya que puede ser cualquier archivo que pudo ser obtenido a través de internet o cualquier otro medio y pueda ser dudoso.

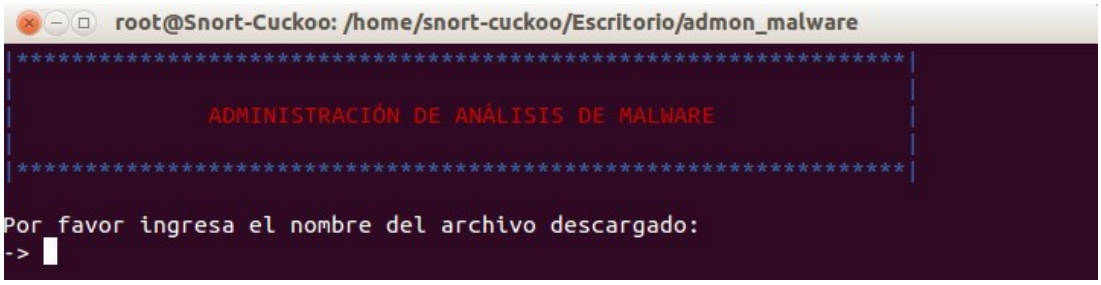


Figura 4.9: Nombre de la muestra a analizar. Fuente: Captura propia.

Este módulo puede ser empleado para comprobar el comportamiento de archivos ejecutables en los que se tenga duda acerca de su comportamiento, antes de que sean instalados en un ambiente de producción. La única restricción es que los archivos que se requieran analizar, deberán encontrarse en la carpeta Descargas.

Una vez que la muestra ha sido enviada para su análisis, como se observa en la figura 4.10, el programa notifica que se asignó una nueva tarea con su respectivo identificador, así como el nombre y tipo de la muestra; en este caso un archivo ejecutable llamado DoomJuice.exe.

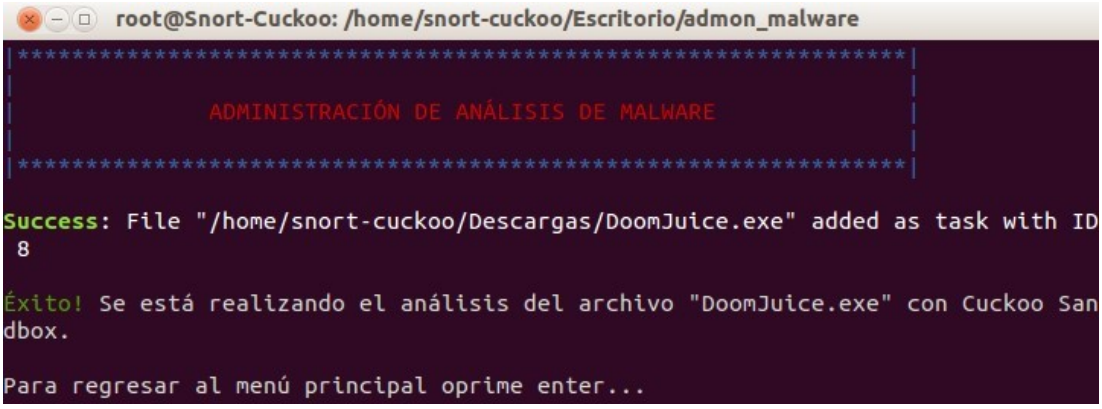


Figura 4.10: Análisis de un archivo ejecutable. Fuente: Captura propia.

Este módulo realiza este proceso a través de la función llamada *analisis_archivo_descargado()*, en la que se puede observar su estructura en la página 219 del apéndice B.

4.1.5. Módulo para analizar el repositorio de reportes de malware generados por Cuckoo SandBox

El módulo que a continuación se describe permite analizar un repositorio de reportes de muestras previamente analizadas. La función *analisis_repositorio()*, que se encuentra en la página 220 del apéndice B, es la que analiza cada uno de los reportes con formato html generados por Cuckoo SandBox. Para esta tarea, la función invoca la ejecución del archivo *analisis_reportes_cuckoo* el cual obtiene las características más relevantes de las muestras analizadas por Cuckoo. Estas características incluyen el directorio analizado, la fecha actual del análisis del repositorio de Cuckoo, el nombre de la muestra, la función hash que permite identificar unívocamente la muestra, el porcentaje de antivirus que detectaron la muestra como maliciosa y finalmente la reputación que los usuarios de la página de Virus Total le han asignado a la muestra (el rango de la reputación es de -100 a 100, por lo que entre más negativa sea la reputación, mayor es el número de usuarios que lo habrán clasificado como malicioso).

En la figura 4.11 se muestra dicho proceso:

```
|*****************************************************************|
|                                                                 |
|              ADMINISTRACIÓN DE ANÁLISIS DE MALWARE              |
|                                                                 |
|*****************************************************************|

Directorio 1 analizado
La fecha es: dom may 15 10:13:41 CDT 2016
El nombre de la muestra analizada es: w.exe
El SHA256 de la muestra analizada es: 03ccffd2630dad73f8c36198c9868515eec77e402d
2ee03bcb1a07f80cfd91cb
El índice de detección de antivirus es: 72.72727272727272727200%
La reputación del archivo w.exe es de: -100


Directorio 2 analizado
La fecha es: dom may 15 10:13:44 CDT 2016
El nombre de la muestra analizada es: tup27nyt.oiasdgfr.pw
El SHA256 de la muestra analizada es: Sin SHA256
El índice de detección de antivirus es: 1.587301587301587301000%
La reputación de la URL tup27nyt.oiasdgfr.pw es de: 0


Directorio 3 analizado
La fecha es: dom may 15 10:13:46 CDT 2016
El nombre de la muestra analizada es: factorygood.net
El SHA256 de la muestra analizada es: Sin SHA256
El índice de detección de antivirus es: 7.936507936507936507000%
La reputación de la URL factorygood.net es de: -55
```

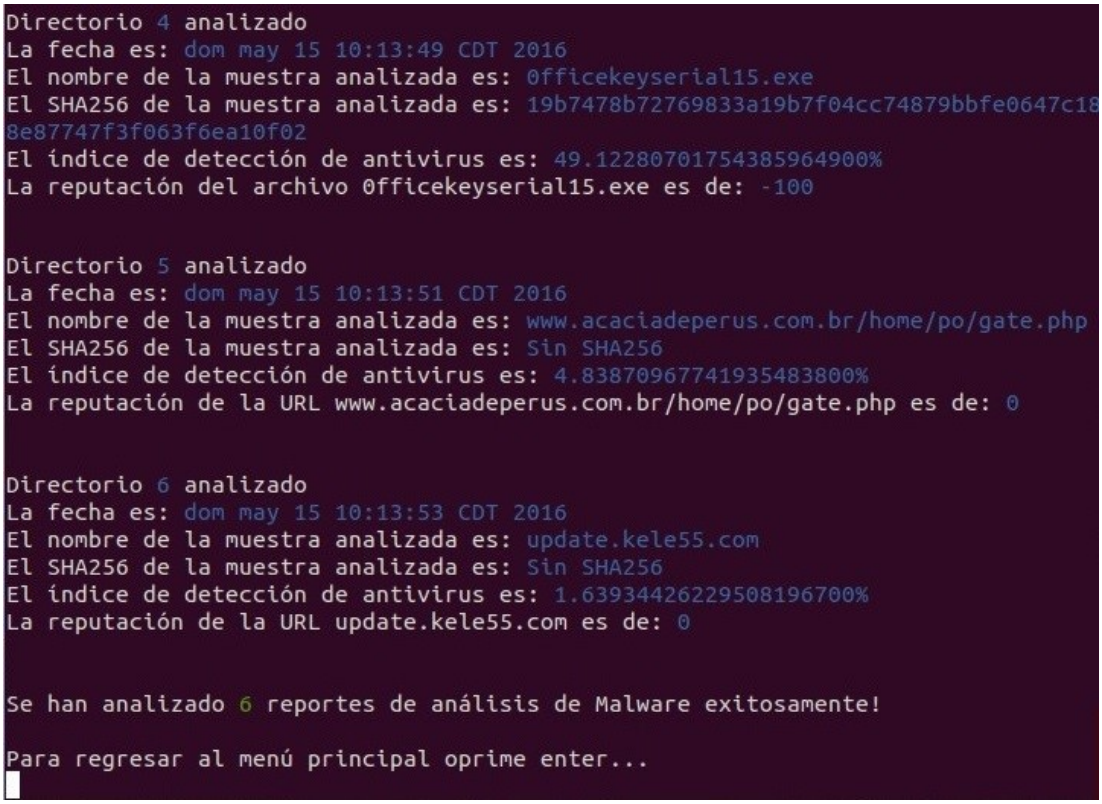


Figura 4.11: Proceso de análisis del repositorio de reportes de Cuckoo SandBox. Fuente: Captura propia.

Al mismo tiempo que la información obtenida del repositorio de reportes de Cuckoo es mostrada en pantalla, también se almacena con un formato específico en un archivo ubicado en el Escritorio llamado *Información_Muestras.txt*. Este proceso es realizado, para que la información pueda ser consultada a través de la página web, la cual fue diseñada para la fácil comprensión de las características más relevantes de cada una de las muestras analizadas por Cuckoo SandBox. Esta funcionalidad se explicará en la sección 4.2.

Cuando ha terminado el proceso, el usuario regresará al menú principal.

Para más detalles del archivo *analisis_reportes_cuckoo* consultar la página 225 del apéndice B.

4.1.6. Módulo para habilitar el servicio web de Cuckoo SandBox

Cuckoo también cuenta con una interfaz gráfica para poder administrar todos los reportes de las muestras y la revisión de estos informes desde una interfaz web. También

desde aquí se pueden enviar muestras para su análisis, de la misma manera en que se realiza desde el módulo 4.

Este módulo permite comprobar que el servicio web de Cuckoo esté ejecutándose, y en caso de no hacerlo, permite activar el servicio para la interfaz gráfica. Para ello, la función *web_cuckoo* comprueba sí ya existen los procesos del servicio de la interfaz gráfica.

Cuando el usuario elige este módulo, el programa determina sí el servicio web de Cuckoo está ejecutándose. Sí este servicio de Cuckoo SandBox se encuentra ejecutándose, el programa devolverá el identificador de los procesos correspondientes a la interfaz web. De acuerdo con lo descrito anteriormente, en la figura 4.12 se muestran los PID's del servicio web de Cuckoo.

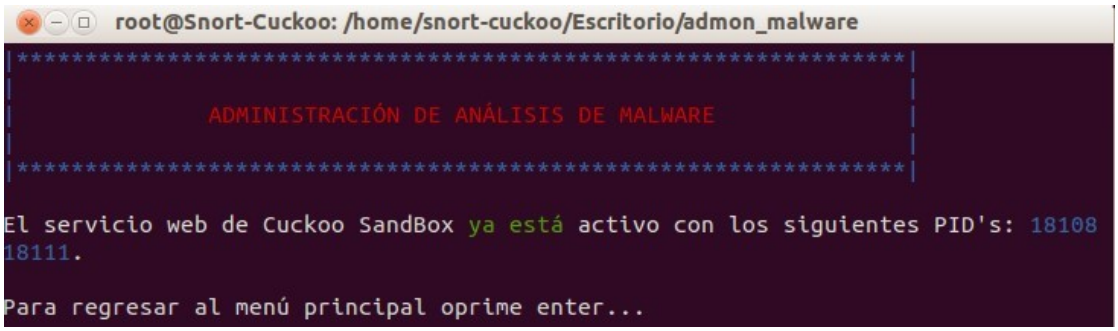


Figura 4.12: Servicio web de Cuckoo SandBox en ejecución. Fuente: Captura propia.

Sí el servicio web no está activo, este módulo permite notificarle al usuario esto, por lo que se podrá activar el servicio como se muestra a continuación en la figura 4.13:

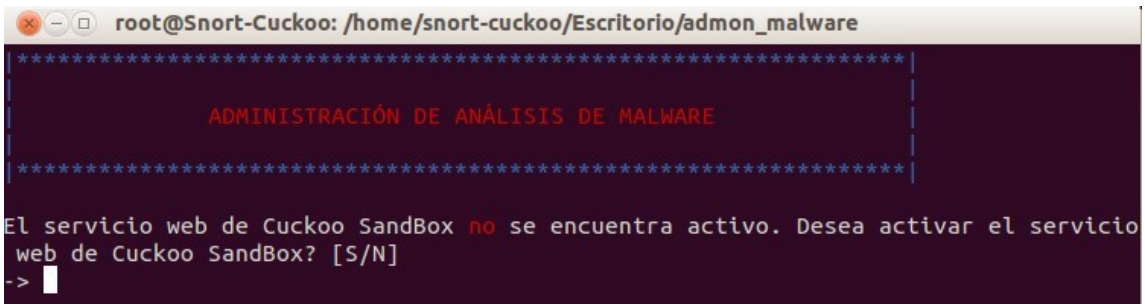


Figura 4.13: Notificación de que el servicio web de Cuckoo SandBox no está actualmente en ejecución. Fuente: Captura propia.

Cuando el usuario elige habilitar el servicio web de Cuckoo SandBox, aparece un mensaje de confirmación que dicho servicio está activo y que puede ser accedido a éste

desde cualquier navegador web, a través del puerto 8080. Esto se muestra a continuación en la figura 4.14:

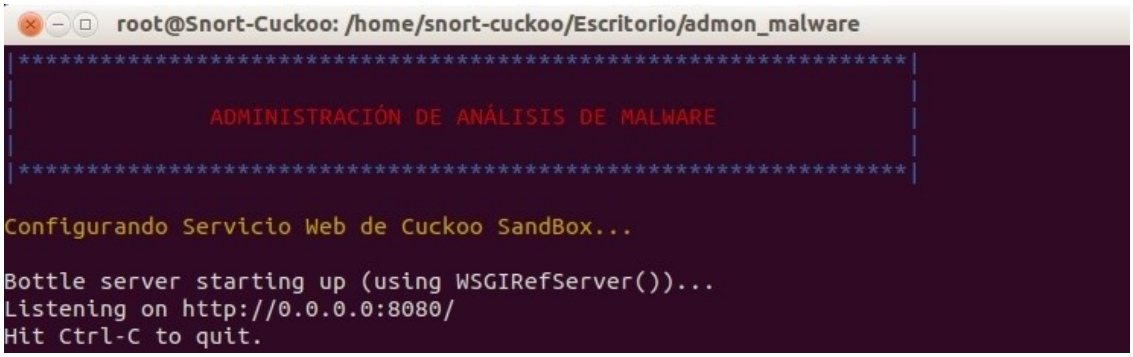


Figura 4.14: Servicio web de Cuckoo ejecutándose. Fuente: Captura propia.

El código fuente completo de este programa puede ser consultado en el apéndice B.

4.2. Sistema de Consulta de Malware

Para facilitar la revisión del impacto de las probables muestras obtenidas a través de los archivos descargados desde el IDS y las muestras analizadas independientemente, se desarrolló una página web, en la que se presentan de una manera ordenada y organizada los datos más relevantes de éstos.

Para establecer un nivel mínimo de seguridad para el sistema de consulta de malware, se creó una nueva base de datos en MySQL para permitir la autenticación de usuarios autorizados. Los pasos se describen a continuación:

Accedemos a MySQL a través del terminal de Ubuntu de la siguiente manera:

```
# mysql -u root -p
Enter password: <Password root de MySQL>
```

Una vez que se ingresó con el usuario *root* del manejador de base de datos MySQL, se creó la base de datos *usuario_web*.

```
mysql> create database usuario_web;
```

Se selecciona la base de datos anterior y se crea una tabla llamada *usuario* con sus respectivas columnas de la siguiente manera:

```
mysql> use usuario_web;
mysql> CREATE TABLE usuario (id int(1), nombre VARCHAR(20), password BLOB
    (40), visit_counter int(1));
```

Una vez creada la tabla *usuario*, se insertan los valores en la tabla que tendrá por defecto el usuario administrador del sistema de consulta de malware:

```
mysql> INSERT INTO usuario VALUES ('1', '<nombre_de_usuario>', AES_ENCRYPT
    ('<password_usuario>', 'key_aes'), '0');
```

Finalmente se validó la información que contiene la tabla *usuario* a través del siguiente query:

```
mysql> select * from usuario;
```

La información contenida en la tabla *usuario*, es como la que se muestra en la figura 4.15:

```
+------+--------+------------------+---------------+
| id   | nombre | password         | visit_counter |
+------+--------+------------------+---------------+
|    1 | admin  | ▯▯▯▯▯▯▯8{▯▯q     |             1 |
+------+--------+------------------+---------------+
1 row in set (0.03 sec)
```

Figura 4.15: Contenido de la tabla usuario. Fuente: Captura propia.

El sistema de consulta de malware está desarrollado con las tecnologías PHP, JavaScript, jQuery y Bootstrap; el cual está compuesto por los siguientes archivos, los cuales se explican a continuación:

- **index.php:** Archivo que permite autenticar las credenciales de un usuario con las que se encuentran en la base de datos *usuario_web* del servidor.
- **validacion.php:** Archivo que es invocado cuando las credenciales del usuario son incorrectas. Es desplegado un mensaje de aviso de error de autenticación.
- **principal.php:** Página que es desplegada una vez que el usuario ha logrado autenticarse de manera exitosa. En ella se muestra la información más relevante de las muestras analizadas con la SandBox.

- **reiniciar_contador.php:** Archivo que es invocado una vez que el usuario ha decidido reestablecer el número de ingresos al sistema.
- **sendmail.php:** Archivo que tiene la función de realizar el proceso de envío de correo electrónico con el reporte de la muestra seleccionada.
- **sesion.class.php:** Archivo que gestiona las sesiones.
- **cerrarsession.php:** Archivo que tiene la función de destruir toda la información registrada de una sesión.

Como se mencionó anteriormente, el módulo 5 del programa realizado en shell genera un archivo de texto, para que la información pueda ser consultada a través de la página web, para una mayor comprensión de las características de cada muestra analizada en la SandBox. A continuación se explicará el detalle de la página web.

Como se observa en la figura 4.16, en la página de autenticación el usuario deberá ingresar sus credenciales para poder tener acceso al sistema de consulta de malware. Adicionalmente en esta página de inicio, el usuario indicará sí la consulta es realizada debido un incidente de seguridad. Esto es de ayuda para llevar un registro de los incidentes relacionados con algún malware o archivo malicioso detectado en algún activo.

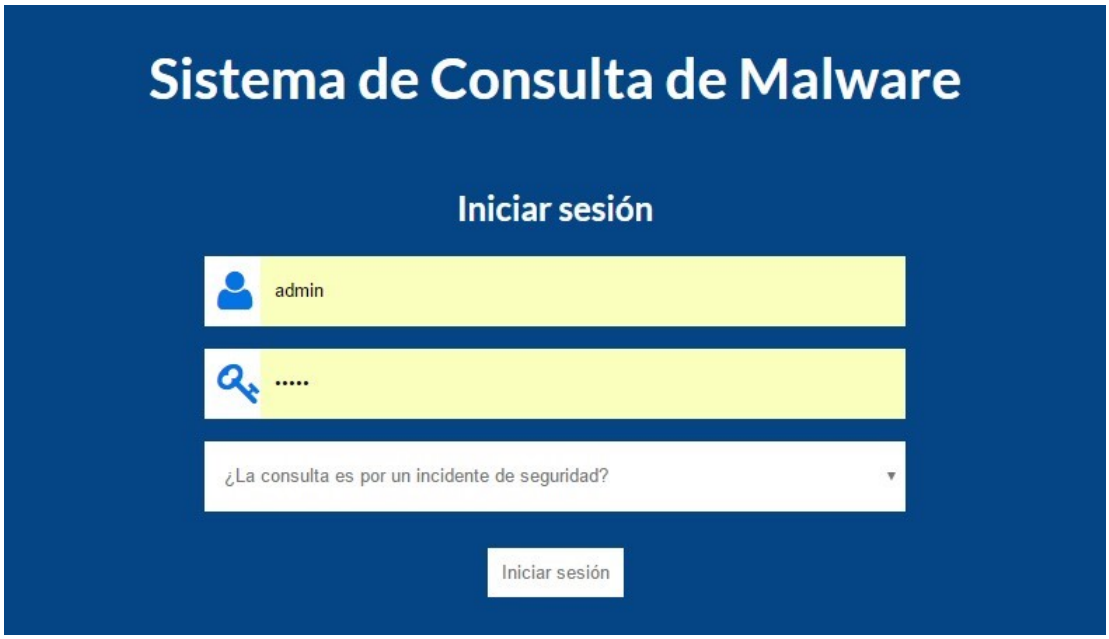


Figura 4.16: Página de autenticación del sistema de consulta de malware. Fuente: Captura propia.

Como lo muestra la figura 4.17, sí el usuario ingresa erróneamente sus credenciales, será desplegado un mensaje como se muestra a continuación; en el cual le es negado el acceso y deberá intentar nuevamente autenticarse al sistema.

Figura 4.17: Mensaje de fallo de autenticación. Fuente: Captura propia.

Una vez que el usuario logra autenticarse, obtiene acceso al sistema de consulta de malware. Es importante validar que exista el archivo *Información_Muestras.txt* y su nombre sea el correcto, para que pueda ser leído por la página web, y los datos de este puedan ser desplegados en la página web.

Sí es la primera vez que se ingresa al sistema, se debe ejecutar el módulo 5 del programa de shell, para que se genere el archivo mencionado anteriormente y pueda ser consumida la información de éste por la página web.

Sí el archivo no existe o su nombre es diferente al mencionado anteriormente, le será notificado al usuario como se aprecia en la figura 4.18:

Figura 4.18: Mensaje de error de lectura del archivo Información_Muestras.txt. Fuente: Captura propia.

Cuando el archivo existe y la autenticación es exitosa, el usuario tendrá acceso al sistema de consulta de malware, en la que es desplegada la información más relevante y que caracteriza a cada una de las muestras analizadas. Por orden de columnas la información desplegada es la siguiente:

- **ID:** Permite identificar el número de la muestra analizada.
- **Fecha de análisis:** Fecha en la que se realizó el análisis del repositorio de reportes de Cuckoo.
- **Nombre muestra:** Nombre de la muestra analizada.
- **SHA256:** Suma de verificación compuesta por 64 números hexadecimales, que permite identificar la muestra.
- **Tasa de Detección de Antivirus:** Porcentaje de antivirus disponibles en la página de Virus Total que detectaron la muestra como una potencial amenaza.
- **Reputación Virus Total (-100 a 100):** Calificación otorgada por los usuarios de Virus Total, en la que determinan que la muestra es identificada como maliciosa.
- **Reporte de Cuckoo:** Hipervínculo que conduce al reporte del análisis de la muestra, generado por Cuckoo SandBox.
- **Enviar reporte por e-mail:** Permite enviar el reporte de análisis de la muestra por correo electrónico.

A continuación, en la figura 4.19, se muestra como es desplegada la información del archivo *Información_Muestras.txt* en la página web:

ID	Fecha de análisis	Nombre muestra	SHA256	Tasa de Detección de Antivirus (%)	Reputación Virus Total (-100 a 100)	Reporte de Cuckoo	Enviar reporte por e-mail
1	dom may 15 10:50:08 CDT 2016	w.exe	03ccffd2630dad73f8c36198c9868515eec77e402d2ee03bcb1a07f80cfd91cb	72.72%	-100	Ver	Enviar
2	dom may 15 10:50:10 CDT 2016	tup27nyt.oiasdgfr.pw	Sin SHA256	1.587%	0	Ver	Enviar
3	dom may 15 10:50:12 CDT 2016	factorygood.net	Sin SHA256	7.936%	-55	Ver	Enviar
4	dom may 15 10:50:15 CDT 2016	0fficekeyserial15.exe	19b7478b72769833a19b7f04cc74879bbfe0647c188e87747f3f063f6ea10f02	49.12%	-100	Ver	Enviar
5	dom may 15 10:50:18 CDT 2016	www.acaciadeperus.com.br/home/po/gate.php	Sin SHA256	4.838%	0	Ver	Enviar
6	dom may 15 10:50:19 CDT 2016	update.kele55.com	Sin SHA256	1.639%	0	Ver	Enviar

Figura 4.19: Despliegue de información en el sistema de consulta de malware. Fuente: Captura propia.

Como se observa en la figura 4.20, la parte superior derecha de la página web del sistema de consulta de malware contiene un grupo de opciones. A continuación se describe la funcionalidad de estos:

- **Ingresar al repositorio de análisis de Cuckoo:** Permite acceder a la interfaz gráfica de Cuckoo SandBox en la que se administran las tareas de análisis de Cuckoo SandBox.
- **Reiniciar contador de Consultas de Malware:** Esta opción reinicia el contador de ingresos al sistema de consulta de malware.
- **Cerrar Sesión:** Permite cerrar sesión y salir del sistema.

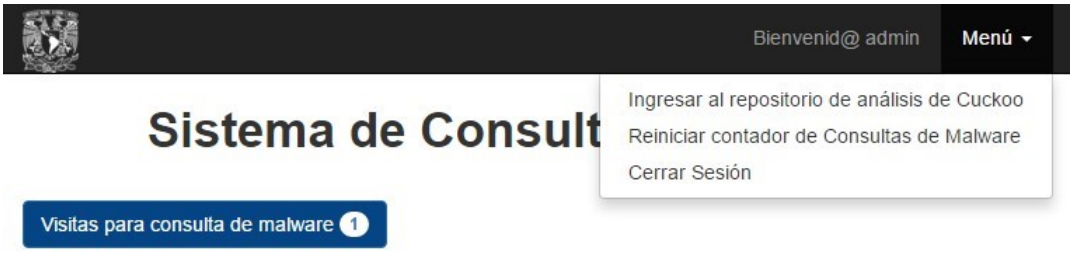


Figura 4.20: Menú de la página web. Fuente: Captura propia.

Cuando se desea compartir vía e-mail el reporte realizado por Cuckoo SandBox, respecto a una muestra en específico, se debe seleccionar la opción "Enviar" de la última columna que aparece en la página web. Posterior a esto, como se aprecia a continuación en la figura 4.21, se despliega una ventana modal, en la que el usuario debe ingresar el e-mail del destinatario.

Figura 4.21: Envío de correo con el reporte de la muestra analizada. Fuente: Captura propia.

Para el envío exitoso del e-mail, debe ser ingresado un correo electrónico válido. Sí el correo electrónico no es válido, el sistema desplegará el siguiente mensaje mostrado en la figura 4.22:

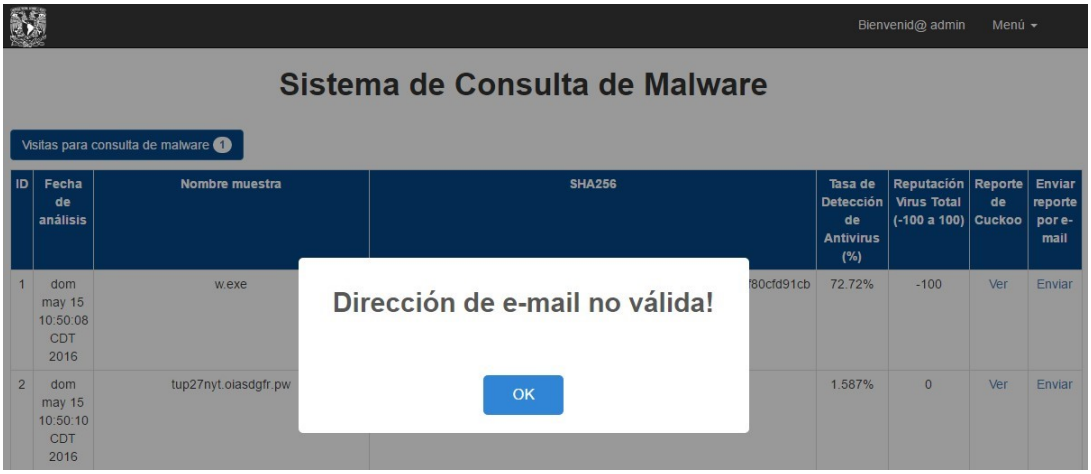


Figura 4.22: Dirección de e-mail no válida. Fuente: Captura propia.

Cuando es ingresado un correo electrónico válido de destinatario, el sistema informará que ha sido enviado exitosamente el reporte de análisis de la muestra, indicando el nombre de la muestra correspondiente. A continuación, en la figura 4.23, es mostrado el proceso descrito anteriormente. Es importante mencionar que el tiempo que tarde en enviarse el correo dependerá del tamaño del reporte de Cuckoo, así como la velocidad de subida con la que cuente el servidor.

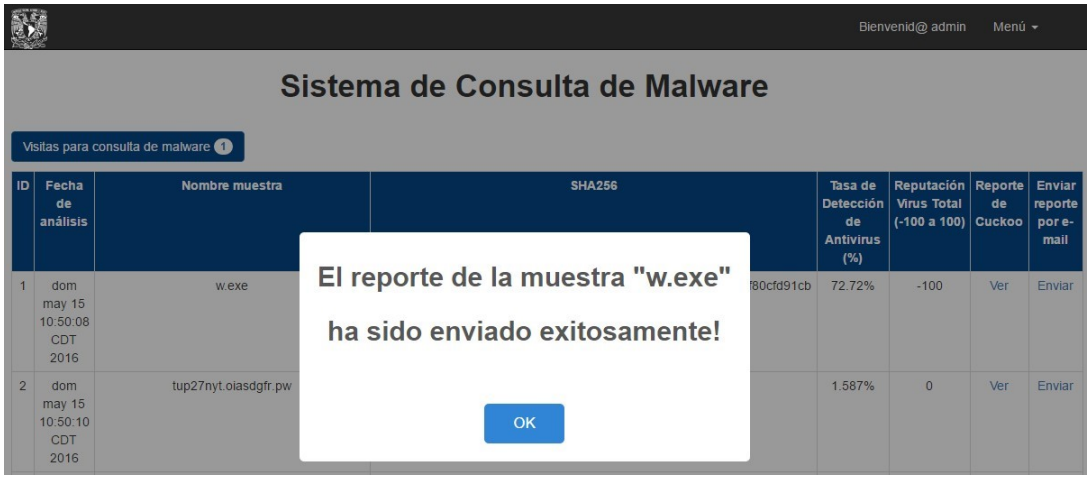


Figura 4.23: Envío exitoso de correo electrónico. Fuente: Captura propia.

Una vez que ha sido confirmado el envío exitoso del correo electrónico desde el sistema de consulta de malware, se procede a revisar la bandeja de entrada del destinatario, para la descarga y revisión del reporte de la muestra analizada con Cuckoo SandBox. La figura 4.24 muestra la recepción del correo electrónico por parte del destinatario.

El reporte es comprimido en un archivo con extensión .zip para facilitar el envío a través del correo electrónico. El usuario destinatario, debe descomprimir el archivo *Reporte.zip* para poder tener acceso al reporte realizado por Cuckoo SandBox. Por otro lado, para revisión del reporte, se recomienda hacerlo con un navegador web moderno como Google Chrome, Mozilla Firefox, Safari, entre otros; para su correcto despliegue y visualización.

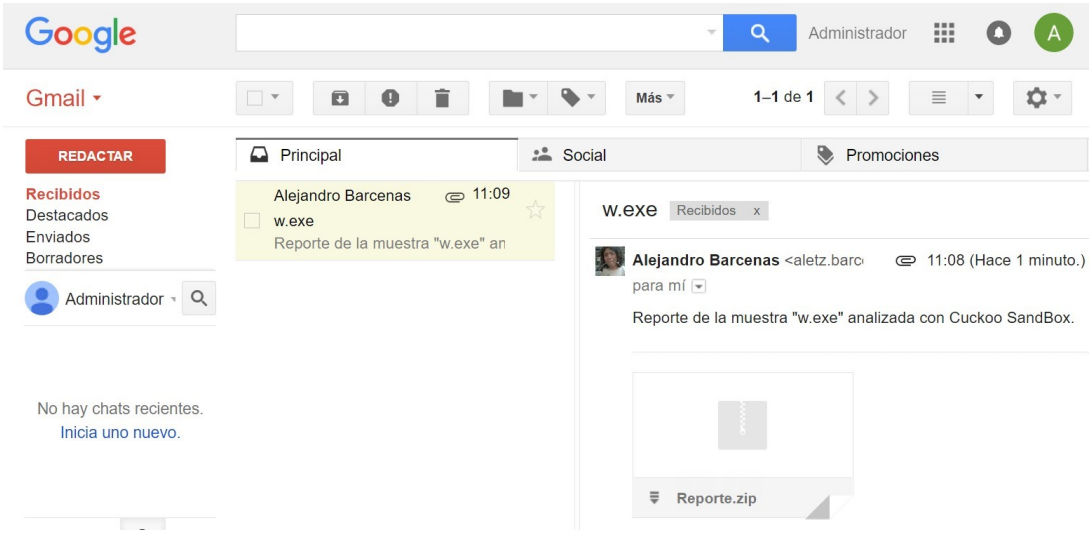


Figura 4.24: Correo electrónico con el reporte de la muestra recibido. Fuente: Captura propia.

El código fuente de la página web, puede ser consultado en el apéndice C.

Capítulo 5

Conclusiones

Finalizada la instalación del Sistema de Detección de Intrusos en el servidor (ver capítulo 2), se realizaron pruebas de conectividad para comprobar el correcto funcionamiento de dicho sistema en la intranet de la SEDESA, resultando contundentes. Para ello fue necesario establecer un port mirroring en el switch core, en el que todo el tráfico saliente del enlace troncal hacia el enlace WAN, fue capturado y enviado hacia el port mirroring específico donde se encontraba el IDS.

Una vez que se obtuvo la evidencia por parte del IDS, se implementó en el mismo servidor la SandBox. Inicialmente se realizaron pruebas manuales para el análisis de los eventos generados por Snort, pero este proceso fue tedioso y tardado, y más sí era necesario saber con prontitud su comportamiento e impacto que tendría sobre los activos de información; sin dejar de lado que para un usuario con pobres o nulos conocimientos sobre administración Linux podría ser muy complicado su correcta operación. Por lo anterior, se pensó en la realización de un programa para automatizar el proceso de recolección de evidencia y automatización del análisis.

El programa para automatizar el análisis de las alertas capturadas por el IDS a través de Cuckoo SandBox, surgió de la necesidad de realizar este proceso de una manera práctica, ágil y eficiente; ya que con la ejecución de dicho programa, se tiene acceso a muchas funcionalidades adicionales que permiten administrar y operar de una manera sencilla este sistema. Como se explicó anteriormente (ver capítulo 4), este programa es capaz de analizar archivos capturados por el IDS en busca de muestras que puedan ser enviadas a Cuckoo para su análisis, además de tener la capacidad de comprobar el estado y el servicio web de la SandBox, por mencionar algunos.

Para poder tener un panorama general de los análisis de las amenazas, se desarrolló una página web principalmente en PHP, en la cual son desplegados de manera sencilla y ordenada los datos más relevantes derivados del análisis con la SandBox. Por ejemplo, en ella es capaz de visualizar el nombre de la muestra analizada, así como su respectiva suma de verificación (SHA-256) permitiendo identificar unívocamente a ésta.

Finalmente, son mostrados dos valores que proporcionan la evidencia necesaria para identificar a la muestra como altamente potencial: la tasa de detección de antivirus y la reputación de Virus Total. La página también cuenta con un módulo para poder compartir mediante correo electrónico el reporte de análisis de la muestra.

Por lo anterior, la página web es una herramienta vital dentro del sistema, ya que permite al administrador del sistema observar y determinar de una manera sencilla el impacto negativo de la muestra analizada, descartando falsos positivos detectados por el IDS.

Cuckoo SandBox es un servicio que pertenece al proyecto Honeynet, el cual, se dedica a la investigación de los últimos ataques de piratas informáticos y al desarrollo de herramientas de seguridad de código abierto para mejorar la seguridad en Internet. Este proyecto está formado por voluntarios de todo el mundo, que han contribuido a luchar contra la propagación de malware.

Durante la implementación de la SandBox, surgieron problemas de incompatibilidad con las librerías requeridas por Cuckoo y las del sistema operativo original. Dicho problema fue detectado en *Debian 6 Squeeze*. Después de investigar en diferentes foros de discusión de Linux, se determinó que las librerías y paquetes basados en Python que necesita Cuckoo para su implementación son incompatibles con dicha versión de sistema operativo, ya que a su vez dependía de otras librerías y paquetes desarrollados por terceros que no siempre resultaba exitosa su compilación y mucho menos su instalación.

Por lo anterior se concluye, que para *Debian 6 Squeeze*, por el momento, la implementación de Cuckoo SandBox no es compatible y funcional; por lo que se descartó la posibilidad de que éste fuese el sistema operativo anfitrión del sistema.

El siguiente sistema operativo que se empleó para realizar la implementación de ambos sistemas fue *Ubuntu 12.04 Precise Pangolin*. La implementación del IDS y de la SandBox se realizó sin mayor contratiempo, resultando exitosa.

Inicialmente el proceso de implementación de ambos sistemas se realizó en un ambiente virtual sobre *VMware Workstation 10*. La puesta en funcionamiento del IDS en un ambiente virtual resultó exitoso, ya que se configuró una red de pruebas, en la que el IDS monitoreaba y emitía alertas por el tráfico entre los diversos equipos virtuales que se encontraban en el mismo segmento. Por otro lado, al realizar las pruebas de análisis de muestras en Cuckoo SandBox no pudo ser exitoso, ya que para el análisis de muestras necesita realizarse sobre la máquina huésped, la cual debe ser virtual. La versión *10* de *VMware Workstation* no es capaz de ejecutar una máquina huésped virtual sobre una máquina anfitriona virtualizada, por lo que se puso en marcha la implementación de ambos sistemas en un equipo físico, resultando exitosa.

Durante el tiempo que estuvo el IDS en la red de producción de la SEDESA no sólo se lograron identificar alertas con actividades relacionadas al malware, sino que se detectaron otros eventos, tales como ataques de DoS a páginas web nacionales y extranjeras, así como escaneos de segmentos y equipos de red, entre otros.

Como se planteó al inicio de este proyecto, el objetivo principal fue desarrollar e implementar una SandBox sobre un sistema operativo Linux, el cual tiene la función de analizar los eventos que son alertados por el IDS, y a su vez poder determinar el nivel de criticidad e impacto que pueden llegar a tener sobre los activos de información e incluso determinar sí los eventos generados por el IDS son falsos positivos.

Para poder brindar dicho servicio, ambos sistemas se pusieron en marcha en una infraestructura de red funcional y en producción, lo que hizo posible que se pudiera capturar cualquier tipo de actividad anómala, la cual se almacenó en bitácoras y en una base de datos para su posterior análisis mediante Cuckoo SandBox.

Durante el desarrollo de este proyecto mejoré mis habilidades sobre la administración de sistemas Linux, al mismo tiempo que aprendí las diferentes arquitecturas en que puede funcionar un Sistema de Detección de Intrusos. Por otro lado, también aprendí sobre las técnicas, herramientas y programas que emplean los piratas informáticos y como éstas han ido evolucionando, haciéndose cada vez más sofisticadas.

Es importante mencionar que el desarrollo e implementación de este proyecto es una versión mejorada y adaptada del proyecto Honeynet, ya que a diferencia del original, éste emplea otro sistema (IDS) del cual obtiene información acerca del tráfico de red en una organización, para posteriormente con ayuda del programa desarrollado en shell de Unix, realizar el análisis del tráfico de una manera automatizada en la SandBox.

Durante el desarrollo del proyecto profundicé sobre la administración de Sistemas Operativos, específicamente sobre Linux y la manera en que un script de Unix puede simplificar y automatizar los procesos. Por otro lado, este proyecto me permitió identificar las tres principales arquitecturas de un *IDS*, así como sus alcances y las acciones que deciden cuál de estas diferentes arquitecturas es mejor implantarse dentro de una organización. Adicionalmente, este proyecto me permitió aprender sobre los problemas de seguridad actuales. Lo anterior reflejó en mí, un crecimiento profesional en mí carrera en el ámbito de las redes de datos y como especialista en la seguridad de la información.

La realización de este proyecto responde a los esfuerzos de formación y generaración de recursos humanos de la Universidad Nacional Autónoma de México y de la Facultad de Ingeniería, capaces de responder y solucionar los problemas y necesidades actuales de la sociedad.

La formación que me brindó la Facultad de Ingeniería me permitió obtener el conoci-

miento teórico, técnico y las habilidades necesarias para afrontar y aportar soluciones a proyectos relacionados en el ámbito de las redes de datos y seguridad de la información. Lo anterior fue posible gracias a materias como redes de datos, seguridad informática I y II, así como el programa de formación de becarios de la Unidad de Servicios de Cómputo Académico de la Facultad de Ingeniería y a la currícula CCNA de Cisco que es impartida en esta Facultad.

Para finalizar, el objetivo de esta tesis, es servir como guía a usuarios u organizaciones que necesiten un sistema de bajo costo para la detección y análisis de actividades maliciosas, que atente contra los activos de información, así como la información que reside en ellos; para poder actuar con prontitud ante una amenaza y lograr la contención de ésta, a fin de evitar su propagación para su posterior erradicación total de los activos de información infectados.

Glosario

A

adodb (p. 54) • Capa de abstracción de bases de datos para PHP, que pemite a los programadores desarrollar aplicaciones web con características de portabilidad.

archivo CPL (p. 36) • La extensión CPL hace referencia a un conjunto de archivos de sistema asociados con el panel de control de Windows, que contiene controladores de red y sonido que emplean los sistemas operativos Windows.

archivo DLL (p. 36) • Un archivo DLL es una biblioteca que contiene código y datos que pueden ser utilizados por más de un programa al mismo tiempo en los sistemas Windows, lo que permiten que estos se carguen y ejecuten más rápido y necesiten menos espacio en disco en el equipo.

AWK (p. 152) • AWK es un lenguaje de programación diseñado para el procesamiento de datos basados en texto, el cual resulta apropiado para extraer datos individuales. La potencia de este lenguaje de programación radica en el uso extensivo de expresiones regulares para la selección de los fragmentos de información apropiados.

B

B.A.S.E. (pp. 51, 53, 54, 55, 56, 153, 154) • Basic Analysis and Security Engine, es un motor de análisis basado en PHP, que busca y procesa una base de datos de eventos de seguridad generados por un IDS.

Barnyard (p. 50) • Barnyard es un intérprete, de código abierto, de archivos de salida en formato unified2, que permite registrar las alertas en una base de datos.

Bootstrap (pp. 164, 235) • Conjunto de herramientas de Twitter de código abierto que simplifica el proceso de creación del diseño de las páginas web.

C

Cisco Systems (p. 16) • Compañía líder en el área de TI a nivel mundial, que se encarga de la fabricación, venta, mantenimiento y consultoría de equipos de telecomunicaciones.

classful (p. 16) • Protocolo que no incluye la máscara de subred en sus actualizaciones.

classless (p. 16) • Protocolo que incluye la máscara de subred en sus actualizaciones.

clave GPG (pp. 41, 42) • GNU Privacy Guard o GPG; es una herramienta de cifrado y firmas digitales, para el envío de datos de forma segura.

Cloud Computing (p. 29) • Conjunto compartido de recursos físicos y virtuales de cómputo, para ofrecer servicios en demanda a través de Internet.

convergencia (p. 16) • Estado en el que las tablas de enrutamientos de los routers se encuentran en un estado de uniformidad.

CRC32 (pp. 128, 140) • Cyclic Redundancy Check, es un código para la detección de errores y verificar la integridad de los datos.

D

drive-by-download (p. 135) • También conocido como drive-by-exploit es una técnica para la infección y propagación de malware en equipos informáticos, con el solo hecho de acceder a determinado sitio web.

E

e-learning (p. 2) • Cursos que son enviados utilizando recursos de red o a través de Internet.

enlace troncal (pp. 76, 173) • Un enlace troncal es un enlace punto a punto entre dos dispositivos de red, que transportan más de una VLAN.

exclusión mutua (pp. 130, 135, 146, 147) • Método empleado en el que un proceso evita el uso de recursos simultáneos, excluyendo temporalmente a todos los demás procesos, para usar un recurso compartido en común.

F

falso positivo (pp. 77, 83, 151, 174, 175) • Falla o error de cualquier software de detección de virus, en el que informa que un archivo se encuentra infectado, cuando en realidad el archivo está libre de infecciones de virus informáticos.

Fast Pattern Matcher (pp. 99, 101, 102, 104, 105, 107, 108, 110, 111) • El Fast Pattern Matcher se emplea para seleccionar sólo aquellas reglas que coincidieron usando el contenido de la regla, sí y sólo sí, el contenido fue encontrado en la regla.

función hash (pp. 128, 140, 160) • Una función hash es una operación matemática que se realiza sobre cualquier conjunto de datos de cualquier longitud. La salida de dicha operación es una huella digital de tamaño fijo para ese conjunto de datos que se les aplicó la función hash.

fuzzy hashing (pp. 63, 128, 140) • Técnica que divide un archivo en cantidades iguales de bytes y en las que por cada grupo calcula un hash. A partir de todos los hash obtenidos, se calcula un hash final, que representará el total del archivo. Es este hash el que compara con otros para obtener la similitud con otros archivos. Los algoritmos más conocidos son Ssdeep y Sdhash.

G

gateway interior (p. 16) • Protocolos usados dentro de un solo sistema autónomo.

Google Summer of Code (pp. 37, 38) • Programa anual, en el que la compañía Google gratifica a los participantes que puedan completar el desarrollo de un proyecto de programación de software libre.

GREP (p. 152) • GREP es una herramienta de la línea de comandos desarrollada para ser utilizada en los sistemas UNIX. Ésta, toma un patrón inicial para realizar la búsqueda en un archivo y mostrar las líneas que coincidan con dicho patrón.

I

ingeniería social (p. 25) • Técnica que emplean algunos individuos para obtener información, acceso o privilegios en los sistemas de información mediante engaños a usuarios legítimos; para realizar algún daño.

internetwork (p. 3) • Malla global de redes interconectadas. La internetwork más conocida, utilizada y accedida por el público en general es Internet.

ISP (p. 12) • Un Proveedor de Servicios de Internet, brinda conexiones de Internet a sus clientes a través de diferentes tecnologías.

J

JavaScript (pp. 164, 235) • Lenguaje de programación, que se ejecuta del lado del cliente para permitir efectos atractivos y dinámicos en las páginas web.

jQuery (pp. 164, 235) • Biblioteca de JavaScript que permite simplificar el desarrollo de páginas web, la animación, y la gestión de eventos.

L

llamada al sistema (p. 82) • Una llamada al sistema es un mecanismo que realiza una aplicación para solicitar un servicio al kernel del sistema operativo.

M

métrica (p. 16) • Valor utilizado por los protocolos de enrutamiento para asignar costos a fin de alcanzar las redes remotas. La métrica se utiliza para determinar qué ruta tiene mayor preferencia cuando existen múltiples rutas hacia la misma red remota.

multiplexación (p. 8) • Es la combinación de dos o más canales de comunicación sobre un mismo medio de transmisión.

P

PHP (pp. 36, 43, 52, 155, 164, 173) • Lenguaje popular de programación de código abierto, diseñado para el desarrollo web y que puede ser embebido dentro de páginas HTML.

podcasts (p. 2) • Distribución de archivos multimedia (audio, video, texto y notas) que pueden descargarse desde Internet mediante una previa suscripción, que pueden reproducirse posteriormente en una computadora o un reproductor digital.

R

Rapid7 (p. 37) • Proveedor de seguridad de la información y soluciones de análisis, que permite a las organizaciones implementar un enfoque activo basada en un análisis de seguridad cibernética.

S

SED (p. 152) • SED es una potente herramienta de la línea de comandos para los sistemas UNIX, la cual permite modificar el contenido de las diferentes líneas de un archivo, con base a una serie de comandos o expresiones regulares definidos.

sistema autónomo (p. 16) • Grupo de redes bajo el control administrativo de una única entidad, que presenta una política de enrutamiento propia e independiente.

T

token (p. 26) • Dispositivo electrónico que facilita el proceso de autenticación de un usuario para poder tener acceso a un sistema informático.

U

URI (pp. 91, 99, 104, 114) • Son las siglas de Uniform Resource Identifier, la cual es una cadena corta de caracteres que sirve para identificar recursos en internet.

V

VirtualBox (pp. 66, 67, 71, 72, 74, 79) • VirtualBox es una software de virtualización de sistemas operativos, en el que podremos crear sistemas invitados/huéspedes.

VLSM (p. 16) • Máscara de Subred de Longitud Variable.

volcado de memoria (pp. 36, 74) • Un volcado de memoria es el proceso en el que el contenido de la memoria es desplegado y almacenado en caso de que un sistema o aplicación colapse. Este proceso permite a los desarrolladores de software y administradores de sistemas, diagnosticar, identificar y resolver el problema que condujo al fallo del sistema o aplicación.

Bibliografía

[1] Automated malware analysis. https://www.cuckoosandbox.org/. Consultado en agosto de 2014.

[2] Cisco certified network associate routing and switching currícula versión 5. www.netacad.com. Consultado en febrero de 2016.

[3] Create beautiful looking css registration forms. http://codeconvey.com/html-css-registration-forms/. Consultado en marzo de 2016.

[4] The honeynet project. http://www.honeynet.org/project. Consultado en enero y febrero de 2014.

[5] Introducción al conjunto de protocolos tcp/ip. https://docs.oracle.com/cd/E19957-01/820-2981/6nei0r0r9/index.html. Consultado en febrero y marzo de 2016.

[6] Oracle vm virtualbox. https://www.virtualbox.org/. Consultado en agosto de 2014.

[7] Snort - network intrusion detection and prevention system. https://www.snort.org/. Consultado en mayo y junio de 2014.

[8] Snort users manual. http://manual.snort.org/. Consultado en mayo y junio de 2014.

[9] (2013). Iso/iec 27002:2013. In *Information technology — Security techniques — Code of practice for information security controls*, pages 30–38. International Organizational for Standardization, 2 edition.

[10] Alder, R., Babbin, J., Beale, J., Doxtater, A., Foster, J. C., Kohlenberg, T., and Rash, M. (2004). *Snort 2.1 Intrusion Detection Second Edition*, chapter Intrusion Detection Systems, Introducing Snort 2.1, pages 2, 9–14, 20–23, 25–27, 67–69. Syngress Publishing, Inc, 2 edition.

[11] Alder, R., Carter, D. E. F., Foster, J. C., Jonkman, M., Marty, R., and Seagren, E. (2007). *Snort IDS and IPS Toolkit*, chapter Intrusion Detection Systems, pages 8–11. Syngress Publishing, Inc.

[12] Beale, J., Foster, J. C., Posluns, J., and Caswell, B. (2003). *Snort 2.0 Intrusion Detection*, chapter Intrusion Detection Systems, Advanced Snort, pages 4–6, 478. Syngress Publishing, Inc.

[13] Galindo, C. J. (2009). *Diseño y optimización de un Sistema de Detección de Intrusos Híbrido*, chapter Snort híbrido, pages 71–75. Editorial Universidad de Almería.

[14] https://cryptome.org (2013). *Fundamental Security Concepts*, chapter Fundamental Security Concepts, pages 4–11, 18–20. Cryptome. Consultado en internet en febrero de 2016.

[15] Huerta, A. V. (2012). *Seguridad en UNIX y redes*, chapter Introducción y conceptos previos, pages 2–10. GNU Free Documentation License. Versión 2.1.

[16] López, J. G. (2009). *Optimización de Sistemas de Detección de Intrusos en Red, utilizando técnicas computacionales avanzadas*, chapter Snort, pages 27–35. Editorial Universidad de Almería.

[17] Stallings, W. (2000). *Comunicaciones y Redes de Computadoras*, chapter Introducción, Tecnologías LAN, pages 17–19, 403–407. Prentice-Hall, 6 edition.

[18] Tanenbaum, A. S. (2003). *Redes de computadoras*, chapter Introducción, pages 3–5, 37–44. Pearson Educacion, 4 edition.

[19] Whitman, M. E. and Mattord, H. J. (2011). *Principles of Information Security*, chapter Introduction to Information Security, The Need for Security, Risk Management, Security Technology: Firewalls and VPNs, pages 1–13, 67, 140–141, 147, 149, 164, 246–247, 447. Cengage Learning, 4 edition.

Apéndice A

Archivos de configuración de Cuckoo SandBox

En este apartado se establecen los archivos de configuración de Cuckoo SandBox, mencionados en la sección 2.2.6 de esta tesis.

Archivo auxiliary.conf

```
[sniffer]
# Enable or disable the use of an external sniffer (tcpdump) [yes/no].
enabled = yes

# Specify the path to your local installation of tcpdump. Make sure this
# path is correct.
tcpdump = /usr/sbin/tcpdump

# Specify the network interface name on which tcpdump should monitor the
# traffic. Make sure the interface is active.
interface = vboxnet0

# Specify a Berkeley packet filter to pass to tcpdump.
# bpf = not arp
```

Archivo cuckoo.conf

```
[cuckoo]
# Enable or disable startup version check. When enabled, Cuckoo will
    connect
# to a remote location to verify whether the running version is the latest
# one available.
version_check = on

# If turned on, Cuckoo will delete the original file after its analysis
# has been completed.
delete_original = off

# If turned on, Cuckoo will delete the copy of the original file in the
# local binaries repository after the analysis has finished. (On *nix this
# will also invalidate the file called "binary" in each analysis directory
    ,
# as this is a symlink.)
delete_bin_copy = off

# Specify the name of the machinery module to use, this module will
# define the interaction between Cuckoo and your virtualization software
# of choice.
machinery = virtualbox

# Enable creation of memory dump of the analysis machine before shutting
# down. Even if turned off, this functionality can also be enabled at
# submission. Currently available for: VirtualBox and libvirt modules (KVM
    ).
memory_dump = off

# Enable automatically re-schedule of "broken" tasks each startup.
# Each task found in status "processing" is re-queued for analysis.
reschedule = off
```

```
# Enable processing of results within the main cuckoo process.
# This is the default behavior but can be switched off for setups that
#  require high stability and process the results in a separate task.
process_results = on

# Limit the amount of analysis jobs a Cuckoo process goes through.
# This can be used together with a watchdog to mitigate risk of memory
    leaks.
max_analysis_count = 0

# Minimum amount of free space (in MB) available before starting a new
    task.
# This tries to avoid failing an analysis because the reports can't be
    written
# due out-of-diskspace errors. Setting this value to 0 disables the check.
# (Note: this feature is currently not supported under Windows.)
freespace = 64

# Temporary directory containing the files uploaded through Cuckoo
    interfaces
# (web.py, api.py, Django web interface).
tmppath = /tmp

[resultserver]
# The Result Server is used to receive in real time the behavioral logs
# produced by the analyzer.
# Specify the IP address of the host. The analysis machines should be able
# to contact the host through such address, so make sure it's valid.
# NOTE: if you set resultserver IP to 0.0.0.0 you have to set the option
# `resultserver_ip` for all your virtual machines in machinery
    configuration.
ip = 192.168.56.1

# Specify a port number to bind the result server on.
port = 2042
```

```
# Should the server write the legacy CSV format?
# (if you have any custom processing on those, switch this on)
store_csvs = off

# Maximum size of uploaded files from VM (screenshots, dropped files, log)
# The value is expressed in bytes, by default 10Mb.
upload_max_size = 10485760

[processing]
# Set the maximum size of analysis's generated files to process.
# This is used to avoid the processing of big files which can bring memory
    leak.
# The value is expressed in bytes, by default 100Mb.
analysis_size_limit = 104857600

# Enable or disable DNS lookups.
resolve_dns = on

[database]
# Specify the database connection string.
# Examples, see documentation for more:
# sqlite:///foo.db
# postgresql://foo:bar@localhost:5432/mydatabase
# mysql://foo:bar@localhost/mydatabase
# If empty, default is a SQLite in  db/cuckoo.db.
connection = mysql://snort:snort@localhost/cuckoo

# Database connection timeout in seconds.
# If empty, default is set to 60 seconds.
timeout =

[timeouts]
# Set the default analysis timeout expressed in seconds. This value will
    be
```

```
# used to define after how many seconds the analysis will terminate unless
# otherwise specified at submission.
default = 120

# Set the critical timeout expressed in seconds. After this timeout is hit
# Cuckoo will consider the analysis failed and it will shutdown the
    machine
# no matter what. When this happens the analysis results will most likely
# be lost. Make sure to have a critical timeout greater than the
# default timeout.
critical = 600

# Maximum time to wait for virtual machine status change. For example when
# shutting down a vm. Default is 300 seconds.
vm_state = 300
```

Archivo memory.conf

```
# Volatility configuration

# Basic settings
[basic]
# Profile to avoid wasting time identifying it
guest_profile = WinXPSP2x86
# Delete memory dump after volatility processing.
delete_memdump = no

# List of available modules
# enabled: enable this module
# filter: use filters to remove benign system data from the logs
# Filters are defined in the mask section at below

# Scans for hidden/injected code and dlls
# http://code.google.com/p/volatility/wiki/CommandReferenceMal23#malfind
[malfind]
enabled = yes
filter = on

# Lists hooked api in user mode and kernel space
# Expect it to be very slow when enabled
# http://code.google.com/p/volatility/wiki/CommandReferenceMal23#apihooks
[apihooks]
enabled = no
filter = on
# Lists official processes. Does not detect hidden processes
# http://code.google.com/p/volatility/wiki/CommandReference23#pslist
[pslist]
enabled = yes
filter = off

# Lists hidden processes. Uses several tricks to identify them
```

```
# http://code.google.com/p/volatility/wiki/CommandReferenceMal23#psxview
[psxview]
enabled = yes
filter = off

# Show callbacks
# http://code.google.com/p/volatility/wiki/CommandReferenceMal23#callbacks
[callbacks]
enabled = yes
filter = off

# Show idt
# http://code.google.com/p/volatility/wiki/CommandReferenceMal23#idt
[idt]
enabled = yes
filter = off

# Show timers
# http://code.google.com/p/volatility/wiki/CommandReferenceMal23#timers
[timers]
enabled = yes
filter = off

# Show messagehooks
# Expect it to be very slow when enabled
# http://code.google.com/p/volatility/wiki/CommandReferenceGui23#
    messagehooks
[messagehooks]
enabled = no
filter = off

# Show sids
# http://code.google.com/p/volatility/wiki/CommandReference23#getsids
[getsids]
enabled = yes
```

```
filter = off

# Show privileges
# http://code.google.com/p/volatility/wiki/CommandReference23#privs
[privs]
enabled = yes
filter = off

# Display processes' loaded DLLs- Does not display hidden DLLs
# http://code.google.com/p/volatility/wiki/CommandReference23#dlllist
[dlllist]
enabled = yes
filter = on

# List open handles of processes
# http://code.google.com/p/volatility/wiki/CommandReference23#handles
[handles]
enabled = yes
filter = on

# Displays processes' loaded DLLs - Even hidden one (unlinked from PEB
    linked list)
# http://code.google.com/p/volatility/wiki/CommandReferenceMal23#
    ldrmodules
[ldrmodules]
enabled = yes
filter = on

# Scan for Mutexes (whole system)
# http://code.google.com/p/volatility/wiki/CommandReference23#mutantscan
[mutantscan]
enabled = yes
filter = on

# List devices and drivers
```

```
# http://code.google.com/p/volatility/wiki/CommandReferenceMal23#
    devicetree
[devicetree]
enabled = yes
filter = on

# Scan for services
# http://code.google.com/p/volatility/wiki/CommandReferenceMal23#svcscan
[svcscan]
enabled = yes
filter = on

# Scan for kernel drivers (includes hidden, unloaded)
# http://code.google.com/p/volatility/wiki/CommandReference23#modscan
[modscan]
enabled = yes
filter = on

# Masks. Data that should not be logged
# Just get this information from your plain VM Snapshot (without running
    malware)
# This will filter out unwanted information in the logs
[mask]
enabled = no
pid_generic =
```

Archivo processing.conf

```
# Enable or disable the available processing modules [on/off].
# If you add a custom processing module to your Cuckoo setup, you have to
    add
# a dedicated entry in this file, or it won't be executed.
# You can also add additional options under the section of your module and
# they will be available in your Python class.

[analysisinfo]
enabled = yes

[behavior]
enabled = yes

[debug]
enabled = yes

[dropped]
enabled = yes

[memory]
enabled = no

[network]
enabled = yes
[static]
enabled = yes

[strings]
enabled = yes

[targetinfo]
enabled = yes
```

```
[virustotal]
enabled = yes
# Add your VirusTotal API key here. The default API key, kindly provided
# by the VirusTotal team, should enable you with a sufficient throughput
# and while being shared with all our users, it shouldn't affect your use.
key = a0283a2c3d55728300d064874239b5346fb991317e8449fe43c902879d758088
```

Archivo reporting.conf

```
# Enable or disable the available reporting modules [on/off].
# If you add a custom reporting module to your Cuckoo setup, you have to
    add
# a dedicated entry in this file, or it won't be executed.
# You can also add additional options under the section of your module and
# they will be available in your Python class.

[jsondump]
enabled = yes

[reporthtml]
enabled = yes

[mmdef]
enabled = no

[maec40]
enabled = no
mode = overview
processstree = true
output_handles = false
static = true
strings = true
virustotal = true
[mongodb]
enabled = no
host = 127.0.0.1
port = 27017

[hpfclient]
enabled = no
host =
port = 10000
```

```
ident =
secret =
channel =
```

Archivo virtualbox.conf

```
[virtualbox]
# Specify which VirtualBox mode you want to run your machines on.
# Can be "gui", "sdl" or "headless". Refer to VirtualBox's official
# documentation to understand the differences.
mode = gui

# Path to the local installation of the VBoxManage utility.
path = /usr/bin/VBoxManage

# Specify a comma-separated list of available machines to be used. For
    each
# specified ID you have to define a dedicated section containing the
    details
# on the respective machine. (E.g. cuckoo1,cuckoo2,cuckoo3)
machines = cuckoo1

[cuckoo1]
# Specify the label name of the current machine as specified in your
# VirtualBox configuration.
label = cuckoo1

# Specify the operating system platform used by current machine
# [windows/darwin/linux].
platform = windows

# Specify the IP address of the current virtual machine. Make sure that
    the
# IP address is valid and that the host machine is able to reach it. If
    not,
# the analysis will fail.
ip = 192.168.56.101
```

```
# (Optional) Specify the snapshot name to use. If you do not specify a
    snapshot
# name, the VirtualBox MachineManager will use the current snapshot.
# Example (Snapshot1 is the snapshot name):
# snapshot = Snapshot1

# (Optional) Specify the name of the network interface that should be used
# when dumping network traffic from this machine with tcpdump. If
    specified,
# overrides the default interface specified in cuckoo.conf
# Example (vboxnet0 is the interface name):
# interface = vboxnet0

# (Optional) Specify the IP of the Result Server, as your virtual machine
    sees it.
# The Result Server will always bind to the address and port specified in
    cuckoo.conf,
# however you could set up your virtual network to use NAT/PAT, so you can
     specify here
# the IP address for the Result Server as your machine sees it. If you don
    't specify an
# address here, the machine will use the default value from cuckoo.conf.
# NOTE: if you set this option you have to set result server IP to 0.0.0.0
     in cuckoo.conf.
# Example:
# resultserver_ip = 192.168.56.1

# (Optional) Specify the port for the Result Server, as your virtual
    machine sees it.
# The Result Server will always bind to the address and port specified in
    cuckoo.conf,
# however you could set up your virtual network to use NAT/PAT, so you can
     specify here
# the port for the Result Server as your machine sees it. If you don't
    specify a port
```

```
# here, the machine will use the default value from cuckoo.conf.
# Example:
# resultserver_port = 2042

# (Optional) Set your own tags. These are comma separated and help to
    identify
# specific VMs. You can run samples on VMs with tag you require.
# tags = windows_xp_sp3,32_bit,acrobat_reader_6
```

Apéndice B

Código fuente para la automatización del análisis de una muestra con Cuckoo SandBox y Administración de eventos del IDS Snort.

En este apartado es presentado el código fuente completo del programa realizado en Shell de Unix, para la automatización del proceso de análisis de un evento, mencionado en la sección 4.1. Es importante mencionar que dichos archivos que componen este programa, deben ser colocados en un directorio llamado *admon_malware*, el cual debe estar ubicado en el Escritorio de Ubuntu. Únicamente se le debe otorgar permiso de ejecución al archivo *admon_malware*, ya que durante la ejecución a los demás archivos, le son aignados estos permisos, y al término de la ejecución son retirados estos permisos. Las variables de entorno declaradas tanto en el archivo *admon_malware* y *analisis_reportes_cuckoo* deben ser modificados de acuerdo al nombre del usuario del sistema operativo en el que se desee implementar (para esta tesis el nombre de usuario fue *snort-cuckoo*).

Archivo admon_malware

```
#!/bin/bash

# CAPTURA LAS SEÑALES DEL TECLADO "Ctrl-C y Ctrl-Z" PARA EVITAR QUE EL
    USUARIO TERMINE ABRUPTAMENTE LA EJECUCIÓN DEL PROGRAMA
```

```
trap -- '' SIGTSTP
trap -- '' SIGINT

# DECLARACIÓN DE VARIABLES DE ENTORNO

export ADMON_MALWARE=/home/snort-cuckoo/Escritorio/admon_malware
export CUCKOO=/opt/cuckoo
export DESCARGAS=/home/snort-cuckoo/Descargas/
export TEMP_DESCARGAS=/home/snort-cuckoo/Descargas/temp_descarga
export CUCKOO_UTILS=/opt/cuckoo/utils
export MUESTRAS_MALWARE=/home/snort-cuckoo/Descargas/muestras_malware

# FUNCIÓN QUE DESPLIGA EL TÍTULO DEL PROGRAMA

admon_analisis_malware(){
    clear
    echo "|*************************************|"
    echo "|                                     |"
    echo "|  ADMINISTRACIÓN DE ANÁLISIS DE MALWARE|"
    echo "|                                     |"
    echo -e "|*************************************|\n"
}

# FUNCIÓN QUE DESPLIGA LOS MÓDULOS QUE COMPONEN EL PROGRAMA PARA EL
    PROCESO DE AUTOMATIZACIÓN DE LAS MUESTRAS DE MALWARE CON CUCKOO
    SANDBOX

menu_principal(){
    admon_analisis_malware
    echo -e "Por favor seleccione una opción:\n"
    echo "1) Análisis de archivos descargados del IDS Snort."
    echo "2) Borrar Base de Datos del IDS Snort."
    echo "3) Comprobar estado de Cuckoo SandBox."
    echo "4) Analizar archivo descargado con Cuckoo SandBox."
```

```
    echo "5) Analizar repositorio de reportes de malware de Cuckoo SandBox.
        "
    echo "6) Activar servicio web de Cuckoo SandBox"
    echo "7) Salir."
    echo -e -n "\n-> "
}

# FUNCIÓN QUE DETERMINA LA EJECUCIÓN DE LOS DIFERENTES MÓDULOS CONTENIDOS
    EN EL PROGRAMA.

main(){
    menu_principal
    read answer
    case $answer in
        1)
            archivos_ids
        ;;

        2)
            borrar_BD_Snort
        ;;

        3)
            status_cuckoo
        ;;

        4)
            analisis_archivo_descargado
        ;;

        5)
            analisis_repositorio
        ;;
```

```
        6)
            web_cuckoo
        ;;

        7)
            kill_script
        ;;

        *)
            opcion_no_valida
        ;;
    esac
}

# FUNCIÓN QUE PERMITE ANALIZAR EL PAYLOAD DE UN PAQUETE O UN ARCHIVO PCAP,
    PARA LA OBTENCIÓN DE UN ARCHIVO PARA SU POSTERIOR ANÁLISIS CON CUCKOO
    SANDBOX

archivos_ids(){
    admon_analisis_malware
    echo -e "Por favor seleccione una opción:\n"
    echo "1) Analizar el Payload de un paquete."
    echo "2) Analizar archivo con formato pcap."
    echo "3) Regresar al menú principal."
    echo -e -n "\n-> "
    read opcion
    case $opcion in
        1)
            payload_paquetes
        ;;

        2)
            archivos_pcap
        ;;
```

```
        3)
            main
        ;;

        *)
            admon_analisis_malware
            echo -e "ERROR! Opción no válida!\n"
            echo "Para regresar oprime enter..."
            read enter
            if [ "$enter" != " " ]; then
                archivos_ids
            fi
        ;;
    esac
}

# FUNCIÓN QUE OBTIENE UN ARCHIVO QUE SE ENCUENTRA EN EL PAYLOAD DE UN
    PAQUETE

payload_paquetes(){
    admon_analisis_malware
    echo "Por favor ingresa el nombre del payload a analizar (*.bin):"
    echo -e -n "-> "
    read payload
    if [ -f $DESCARGAS$payload ]; then
        admon_analisis_malware
        echo "Obteniendo información del archivo \"$payload\"..."
        cd $DESCARGAS
        host=`egrep -ih '(Host.*)' $DESCARGAS$payload | sed 's/Host.* //g' |
             sed 's/.$//'`get=`egrep -ih '(^GET.*)' $DESCARGAS$payload | sed
             's/GET //g' | sed 's/HTTP.*//g' | sed 's/^[ \t]*//;s/[ \t]*$
            //'`
        downloaded_file=`echo $get | awk -F "/" '{print $NF}'`
        URL=`echo $host$get`
```

```
status_url=`curl -Is --connect-timeout 2 $URL | head -n 1 | cut -d "
    " -f 2`
if [ "$status_url" == "200" ]; then
    if [ -d $MUESTRAS_MALWARE ]; then
        cd $MUESTRAS_MALWARE
        wget -erobots=off -o temp_descarga $URL &> /dev/null
        if [ -f $MUESTRAS_MALWARE/temp_descarga ]; then
            mv temp_descarga $DESCARGAS
        fi
        admon_analisis_malware
        echo "Se descargó el archivo \"$downloaded_file\" en:
            $MUESTRAS_MALWARE"
        sleep 5s
        analyses_payload
    else
        mkdir $MUESTRAS_MALWARE && cd $MUESTRAS_MALWARE
        wget -erobots=off -o temp_descarga $URL &> /dev/null
        if [ -f $MUESTRAS_MALWARE/temp_descarga ]; then
            mv temp_descarga $DESCARGAS
        fi
        admon_analisis_malware
        echo "Se descargó el archivo \"$get\" en: $MUESTRAS_MALWARE"
        sleep 5s
        analyses_payload
    fi
else
    admon_analisis_malware
    echo "ERROR! El Archivo \"$get\" no se encuentra para su descarga
        !"
    echo -e "\nPara regresar al menú principal oprime enter..."
    read enter
    if [ "$enter" != "" ]; then
        main
    fi
fi
```

```
    else
        admon_analisis_malware
        echo "ERROR! El archivo \"$payload\" no existe!"
        sleep 2s
        payload_paquetes
    fi
}

# FUNCIÓN QUE ES INVOCADA UNA VEZ QUE SE DESCARGÓ EL ARCHIVO CONTENIDO EN
    EL PAYLOAD DE UN PAQUETE. ESTA FUNCIÓN PERMITE ENVIAR LA MUESTRA
    OBTENIDA PARA SU ANÁLISIS A LA SANDBOX

analyses_payload(){
    admon_analisis_malware
    echo "Analizar el archivo ahora con Cuckoo SandBox? [S/N]"
    echo -e -n "-> "
    read respuesta
    case $respuesta in
        s|S)
            obtiene_nombre_archivo_descargado
        ;;

        n|N)
            reanalizar_payload
        ;;

        *)
            admon_analisis_malware
            echo "ERROR! Opción no válida!"
            echo -e "\nPara regresar oprime enter..."
            read enter
            if [ "$enter" != "" ]; then
                analyses
            fi
        ;;
```

```
    esac
}

# FUNCIÓN QUE PERMITE REPETIR EL PROCESO ANTERIOR PARA LA OBTENCIÓN DE UN
    ARCHIVO DESDE EL PAYLOAD DE UN PAQUETE

reanalizar_payload(){
    admon_analisis_malware
    echo "Desea analizar otro archivo (*.bin)? [S/N]"
    echo -e -n "-> "
    read respuesta
    case $respuesta in
        s|S)
            borra_temp_descargas
            payload_paquetes
        ;;

        n|N)
            admon_analisis_malware
            borra_temp_descargas
            echo "Para regresar al menú principal oprime enter..."
            read enter
            if [ "$enter" != "" ]; then
                main
            fi
        ;;

        *)
            admon_analisis_malware
            echo "ERROR! Opción no válida!"
            echo -e "\nPara regresar oprime enter..."
            read enter
            if [ "$enter" != "" ]; then
                reanalizar_payload
            fi
```

```
        ;;
    esac
}

# FUNCIÓN QUE OBTIENE UN ARCHIVO QUE SE ENCUENTRA EN UN ARCHIVO CON
    EXTENSIÓN PCAP

archivos_pcap(){
    admon_analisis_malware
    echo "Por favor ingresa el nombre del archivo pcap a analizar (*.pcap):
        "
    echo -e -n "-> "
    read pcap
    if [ -f $DESCARGAS$pcap ]; then
        admon_analisis_malware
        echo "Obteniendo información del archivo \"$pcap\"..."
        cd $DESCARGAS
        tcptrace -xHTTP $DESCARGAS$pcap &> /dev/null
        file_xpl=`ls -1 *.xpl | cut -d " " -f 10`
        file_times=`ls -1 *.times | cut -d " " -f 10`
        output_file=`ls -1 *.dat | cut -d " " -f 10`
        if [ -f $file_xpl ] && [ -f $file_times ]; then
            rm *.xpl *.times
        fi
        if [ -f $DESCARGAS$output_file ]; then
            cd $DESCARGAS
            host=`egrep -ih '(Host.*)' $DESCARGAS$output_file | sed 's/Host.*
                //g' | sed 's/.$//'`
            get=`egrep -ih '(^GET.*)' $DESCARGAS$output_file | sed 's/GET //g
                ' | sed 's/HTTP.*//g' | sed 's/^[ \t]*//;s/[ \t]*$//'`
            downloaded_file=`echo $get | awk -F "/" '{print $NF}'`
            URL=`echo $host$get`
            status_url=`curl -Is --connect-timeout 2 $URL | head -n 1 | cut -
                d " " -f 2`
            rm $output_file
```

```
        if [ "$status_url" == "200" ]; then
            if [ -d $MUESTRAS_MALWARE ]; then
                cd $MUESTRAS_MALWARE
                wget -erobots=off -o temp_descarga $URL &> /dev/null
                if [ -f $MUESTRAS_MALWARE/temp_descarga ]; then
                    mv temp_descarga $DESCARGAS
                fi
                admon_analisis_malware
                echo "Se descargó el archivo \"$downloaded_file\" en:
                    $MUESTRAS_MALWARE"
                sleep 5s
                analyses_pcap
            else
                mkdir $MUESTRAS_MALWARE && cd $MUESTRAS_MALWARE
                wget -erobots=off -o temp_descarga $URL &> /dev/null
                if [ -f $MUESTRAS_MALWARE/temp_descarga ]; then
                    mv temp_descarga $DESCARGAS
                fi
                admon_analisis_malware
                echo "Se descargó el archivo \"$get\" en: $MUESTRAS_MALWARE
                    "
                sleep 5s
                analyses_pcap
            fi
        else
            admon_analisis_malware
            echo "ERROR! El archivo \"$get\" no se encuentra para su
                descarga!"
            echo -e "\nPara regresar al menú principal oprime enter..."
            read enter
            if [ "$enter" != "" ]; then
                main
            fi
        fi
    else
```

```
            admon_analisis_malware
            echo "ERROR! Hubo un problema al procesar el archivo \"$pcap\""
            sleep 2s
            archivos_pcap
        fi
    else
        admon_analisis_malware
        echo "ERROR! El archivo \"$pcap\" no existe!"
        sleep 2s
        archivos_pcap
    fi
}

# FUNCIÓN QUE ES INVOCADA UNA VEZ QUE SE DESCARGÓ EL ARCHIVO CONTENIDO EN
    EL ARCHIVO .PCAP. ESTA FUNCIÓN PERMITE ENVIAR LA MUESTRA OBTENIDA A
    CUCKOO SANDBOX PARA SU ANÁLISIS

analyses_pcap(){
    admon_analisis_malware
    echo "Analizar el archivo ahora con Cuckoo SandBox? [S/N]"
    echo -e -n "-> "
    read respuesta
    case $respuesta in
        s|S)
            obtiene_nombre_archivo_descargado
        ;;

        n|N)
            reanalizar_pcap
        ;;

        *)
            admon_analisis_malware
            echo "ERROR! Opción no válida!"
            echo -e "\nPara regresar oprime enter..."
```

```
            read enter
            if [ "$enter" != "" ]; then
               analyses_pcap
            fi
        ;;
    esac
}

# FUNCIÓN QUE PERMITE REPETIR EL PROCESO ANTERIOR PARA LA OBTENCIÓN DE UN
    ARCHIVO DESDE UN ARCHIVO CON LA EXTENSIÓN PCAP

reanalizar_pcap(){
    admon_analisis_malware
    echo "Desea analizar otro archivo (*.pcap)? [S/N]"
    echo -e -n "-> "
    read respuesta
    case $respuesta in
        s|S)
            borra_temp_descargas
            archivos_pcap
        ;;

        n|N)
            admon_analisis_malware
            borra_temp_descargas
            echo "Para regresar al menú principal oprime enter..."
            read enter
            if [ "$enter" != "" ]; then
               main
            fi
        ;;

        *)
            admon_analisis_malware
            echo "ERROR! Opción no válida!"
```

```
            echo -e "\nPara regresar oprime enter..."
            read enter
            if [ "$enter" != "" ]; then
               reanalizar_pcap
            fi
         ;;
      esac
}

# FUNCIÓN QUE PERMITE OBTENER EL NOMBRE DEL ARCHIVO DESCARGADO DESDE EL
    PAYLOAD DE UN PAQUETE O DESDE UN ARCHIVO CON PCAP. UNA VEZ QUE SE
    OBTUVO EL NOMBRE DEL ARCHIVO DESCARGADO, SE LE PROPORCIONA ESTA
    INFORMACIÓN A LA SANDBOX PARA PODER REALIZAR EL ANÁLISIS

obtiene_nombre_archivo_descargado(){
   if [ -f $TEMP_DESCARGAS ]; then
      cd $DESCARGAS
      name_file_download=`egrep -ih '(Grabando a:*)' temp_descarga | sed '
          s/Grabando a: //g' | sed 's/^["]*//;s/[ \t]*$//' | sed 's/["]*$
          //;s/[ \t]*$//'`
      admon_analisis_malware
      pid_cuckoo=`ps -ef | grep -v grep | grep cuckoo.py | awk '{print $2
          }'`
      borra_temp_descargas
      if [ "$pid_cuckoo" != "" ]; then
         if [ -f $MUESTRAS_MALWARE/$name_file_download ]; then
            admon_analisis_malware
            cd $CUCKOO_UTILS && ./submit.py $MUESTRAS_MALWARE/
                $name_file_download
            echo -e "\nÉxito! Se está realizando el análisis del archivo \
                "$name_file_download\" con Cuckoo Sandbox."
            echo -e "\nPara regresar al menú principal oprime enter..."
            read enter
            if [ "$enter" != "" ]; then
               main
```

```
            fi
        else
            admon_analisis_malware
            echo "Error! El nombre de archivo no existe"
            echo -e "\nPara regresar al menú principal oprime enter..."
            read enter
            if [ "$enter" != "" ]; then
                main
            fi
        fi
    else
        echo -e "Error!\n\nNo se encuentra activo Cuckoo SandBox. Puedes
            activar Cuckoo SandBox en la opción 3 del menú principal."
        echo -e "\nPara regresar al menú principal oprime enter..."
        read enter
        if [ "$enter" != "" ]; then
            main
        fi
    fi
else
    admon_analisis_malware
    echo "No existen descargas por analizar"
    sleep 2s
    main
fi
}

# FUNCIÓN QUE PERMITE ELIMINAR EL ARCHIVO LOG DE LA DESCARGA DE LA MUESTRA
    , EL CUAL FUE GENERADO A PARTIR DEL ANÁLISIS DEL PAYLOAD DE UN PAQUETE
     Y EL ARCHIVO PCAP

borra_temp_descargas(){
    if [ -f $TEMP_DESCARGAS ]; then
        rm $TEMP_DESCARGAS
    fi
```

```
}

# FUNCIÓN QUE DESPLIEGA EL MENSAJE DE CONFIRMACIÓN PARA BORRAR LOS DATOS
    DE LA BASE DE DATOS DE SNORT

borrar_BD(){
    admon_analisis_malware
    echo -e "W A R N I N G! \n\nDesea borrar la Base de Datos del IDS Snort
        ? [S/N]"
    echo -e -n "-> "
}

# FUNCIÓN QUE DETERMINA SÍ ES REALIZADO EL BORRADO DE LOS DATOS DE LAS
    TABLAS DE LA BASE DE DATOS DE SNORT, O SE REGRESA AL MENÚ PRINCIPAL

borrar_BD_Snort(){
    borrar_BD
    read borrar
    case $borrar in
        s|S)
            borrar_base_de_datos
        ;;

        n|N)
            admon_analisis_malware
            echo "Para regresar al menú principal oprime enter..."
            read enter
            if [ "$enter" != " " ]; then
                main
            fi
        ;;

        *)
            admon_analisis_malware
            echo "ERROR! Opción no válida!"
```

```
            echo -e "\nPara regresar oprime enter..."
            read enter
            if [ "$enter" != "" ]; then
               borrar_BD_Snort
            fi
        ;;
    esac
}

# FUNCIÓN QUE SE INVOCA PARA ELIMINAR LA INFORMACIÓN DE LAS TABLAS DE LA
    BASE DE DATOS Y ELIMINAR TODOS LOS ARCHIVOS GENERADOS POR LAS ALERTAS
    DEL IDS SNORT

borrar_base_de_datos(){
    admon_analisis_malware
    pid_eliminar_tablas_ids =`ps -ef | grep -v grep | grep
        eliminar_tablas_ids.php | awk '{print $2}'`
    if [ -f $ADMON_MALWARE/eliminar_eventos_ids ] && [ -f ADMON_MALWARE/
        eliminar_tablas_ids.php ]; then
        if [ "$pid_eliminar_tablas_ids" == "" ]; then
            echo -e "Borrando tablas de la Base de Datos del IDS Snort...\n"
            cd $ADMON_MALWARE
            chmod 755 eliminar_eventos_ids && ./eliminar_eventos_ids
            wait $ pid_eliminar_tablas_ids
            chmod 644 eliminar_eventos_ids
            echo -e "\n\nAcción ejecutada exitosamente."
            echo -e "\nPara regresar al menú principal oprime enter..."
            read enter
            if [ "$enter" != "" ]; then
               main
            fi
        fi
    else
```

```
        echo "Error! Revise que se encuentren los siguientes archivos:
            eliminar_eventos_ids y eliminar_tablas_ids.php en la ruta
            $ADMON_MALWARE."
        echo -e "\nPara regresar al menú principal oprime enter..."
        read enter
        if [ "$enter" != "" ]; then
            main
        fi
    fi
}

# FUNCIÓN QUE PERMITE COMPROBAR SI LA SANDBOX SE ENCUENTRA EN EJECUCIÓN.
    EN CASO DE NO ESTAR EJECUTÁNDOSE EL USUARIO PODRÁ HABILITAR LA SANDBOX
     PARA QUE PUEDAN ENVIARSE MUESTRAS PARA SU ANÁLISIS

status_cuckoo(){
    admon_analisis_malware
    pid_cuckoo=`ps -ef | grep -v grep | grep cuckoo.py | awk '{print $2}'`
    if [ "$pid_cuckoo" == "" ]; then
        echo "Cuckoo no se encuentra activo. Desea activar Cuckoo? [S/N]"
        echo -e -n "-> "
        read activar
        case $activar in
            s|S)
                admon_analisis_malware
                echo "Configurando interfaz vboxnet0..."
                ifup vboxnet0 &> /dev/null
                ifconfig vboxnet0 192.168.56.1 &> /dev/null
                sleep 1s
                echo -e "\nConfigurando Máquina Virtual \"cuckoo1\"..."
                VBoxManage startvm "cuckoo1"
                sleep 1s
                cd $CUCKOO && ./cuckoo.py
                sleep 2s
                if [ "$pid_cuckoo" == "" ]; then
```

```
                reintentar_cuckoo
            fi
        ;;

        n|N)
            admon_analisis_malware
            echo "Para regresar al menú principal oprime enter..."
            read enter
            if [ "$enter" != "" ]; then
                main
            fi
        ;;

        *)
            status_cuckoo
        ;;
    esac
  else
    echo "Cuckoo ya esta activo con el PID = $pid_cuckoo."
    echo -e "\nPara regresar al menú principal oprime enter..."
    read enter
    if [ "$enter" != "" ]; then
        main
    fi
  fi
}

# FUNCIÓN QUE PERMITE HABILITAR EL SERVICIO DE CUCKO SANDBOX, EN CASO DE
    QUE AL REALIZARLO ANTERIORMENTE NO HAYA SIDO EXITOSO

reintentar_cuckoo(){
  admon_analisis_malware
  echo "ERROR! Falló al intentar activar el servicio de Cuckoo SandBox."
  echo -e "\nIntentar de nuevo? [S/N]"
  echo -e -n "-> "
```

```
    read respuesta
    case $respuesta in
        s|S)
            status_cuckoo
        ;;

        n|N)
            admon_analisis_malware
            echo "Para regresar al menú principal oprime enter..."
            read enter
            if [ "$enter" != "" ]; then
                main
            fi
        ;;

        *)
            admon_analisis_malware
            echo "ERROR! Opción no válida!"
            echo -e "\nPara regresar oprime enter..."
            read enter
            if [ "$enter" != "" ]; then
                reintentar_cuckoo
            fi
        ;;
    esac
}

# FUNCIÓN QUE PERMITE ENVIAR A CUCKOO SANDBOX EL ENVÍO DE UNA MUESTRA O
    ARCHIVO SOSPECHOSO

analisis_archivo_descargado(){
    admon_analisis_malware
    pid_cuckoo=`ps -ef | grep -v grep | grep cuckoo.py | awk '{print $2}'`
    if [ "$pid_cuckoo" != "" ]; then
        echo "Por favor ingresa el nombre del archivo descargado:"
```

```
        echo -e -n "-> "
        read descargado
        if [ -f $DESCARGAS$descargado ]; then
            admon_analisis_malware
            cd $CUCKOO_UTILS && ./submit.py $DESCARGAS$descargado
            echo -e "\nÉxito! Se está realizando el análisis del archivo \"
                $descargado\" con Cuckoo Sandbox."
            echo -e "\nPara regresar al menú principal oprime enter..."
            read enter
            if [ "$enter" != "" ]; then
                main
            fi
        else
            admon_analisis_malware
            echo "Error! El archivo \"$descargado\" no existe!"
            sleep 2s
            analisis_archivo_descargado
        fi
    else
        echo -e "Error!\n\nNo se encuentra activo Cuckoo SandBox. Puedes
            activar Cuckoo SandBox en la opción 3 del menú principal."
        echo -e "\nPara regresar al menú principal oprime enter..."
        read enter
        if [ "$enter" != "" ]; then
            main
        fi
    fi
}

# FUNCIÓN QUE INICIA EL PROCESO DE ANALIZAR EL REPOSITORIO DE REPORTES DE
    CUCKOO SANDBOX

analisis_repositorio(){
    if [ -f $ADMON_MALWARE/analisis_reportes_cuckoo ]; then
        ejecuta_analisis_repositorio
```

```
        echo -e "\nPara regresar al menú principal oprime enter..."
        read enter
        if [ "$enter" != "" ]; then
            main
        fi
    else
        admon_analisis_malware
        echo -e "\nNo es posible realizar el análisis de los repositorios."
        echo -e "\nPor favor revise que el script \"analisis_reportes_cuckoo
            \" se encuentra en la siguiente ruta: $ADMON_MALWARE"
        echo -e "\nPara regresar al menú principal oprime enter..."
        read enter
        if [ "$enter" != "" ]; then
            main
        fi
    fi
}

# FUNCIÓN QUE EJECUTA EL ARCHIVO admon_analisis_malware PARA EL PROCESO DE
    ANÁLISIS DEL REPOSITORIO DE REPORTES DE MUESTRAS PREVIAMENTE
    ANALIZADAS

ejecuta_analisis_repositorio(){
    admon_analisis_malware
    cd $ADMON_MALWARE
    chmod 755 analisis_reportes_cuckoo && ./analisis_reportes_cuckoo
    chmod 644 analisis_reportes_cuckoo
}

# FUNCIÓN QUE PERMITE COMPROBAR SÍ EL SERVICIO WEB DE CUCKOO SE ENCUENTRA
    EN EJECUCIÓN. EN CASO DE NO ESTAR EJECUTÁNDOSE, EL USUARIO PODRÁ
    HABILITAR EL SERVICIO WEB DE CUCKOO

web_cuckoo(){
    admon_analisis_malware
```

```
pid_web_cuckoo=`ps -ef | grep -v grep | grep web.py | awk '{print $2}'`
if [ "$pid_web_cuckoo" == "" ]; then
    echo "El servicio web de Cuckoo SandBox no se encuentra activo.
        Desea activar el servicio web de Cuckoo SandBox? [S/N]"
    echo -e -n "-> "
    read activar
    case $activar in
        s|S)
            admon_analisis_malware
            echo -e "Configurando Servicio Web de Cuckoo SandBox...\n"
            cd $CUCKOO_UTILS && ./web.py
            sleep 1s
            if [ "$pid_web_cuckoo" == "" ]; then
                reintentar_web_cuckoo
            fi
        ;;

        n|N)
            admon_analisis_malware
            echo "Para regresar al menú principal oprime enter..."
            read enter
            if [ "$enter" != " " ]; then
                main
            fi
        ;;

        *)
            web_cuckoo
        ;;
    esac
else
    echo "El servicio web de Cuckoo SandBox ya está activo con los
        siguientes PID's: $pid_web_cuckoo."
    echo -e "\nPara regresar al menú principal oprime enter..."
    read enter
```

```
        if [ "$enter" != "␣" ]; then
            main
        fi
    fi
}

# FUNCIÓN QUE PERMITE HABILITAR EL SERVICIO WEB DE CUCKO SANDBOX, EN CASO
    DE QUE AL REALIZARLO ANTERIORMENTE NO HAYA SIDO EXITOSO

reintentar_web_cuckoo(){
    admon_analisis_malware
    echo "ERROR!␣Falló␣al␣intentar␣activar␣el␣servicio␣web␣de␣Cuckoo␣
        SandBox."
    echo -e "\nIntentar␣de␣nuevo?␣[S/N]"
    echo -e -n "->␣"
    read respuesta
    case $respuesta in
        s|S)
            web_cuckoo
        ;;

        n|N)
            admon_analisis_malware
            echo "Para␣regresar␣al␣menú␣principal␣oprime␣enter..."
            read enter
            if [ "$enter" != "␣" ]; then
                main
            fi
        ;;

        *)
            admon_analisis_malware
            echo "ERROR!␣Opción␣no␣válida!"
            echo -e "\nPara␣regresar␣oprime␣enter..."
            read enter
```

```
            if [ "$enter" != "␣" ]; then
                reintentar_web_cuckoo
            fi
        ;;
    esac
}

# FUNCIÓN QUE PERMITE TERMINAR LA EJECUCIÓN DEL PROGRAMA

kill_script(){
    kill $(pidof ./admon_malware) 2> /dev/null
    clear
    return 0
}

# FUNCIÓN QUE DEVUELVE UN MENSAJE DE OPCIÓN INVÁLIDA, CUANDO ES
    SELECCIONADO UN MÓDULO FUERA DEL ALCANCE

opcion_no_valida(){
    admon_analisis_malware
    echo "ERROR!␣Opción␣no␣válida!"
    echo -e "\nPara␣regresar␣oprime␣enter..."
    read enter
    if [ "$enter" != "␣" ]; then
        main
    fi
}

main
```

Archivo analisis_reportes_cuckoo

```
#!/bin/bash

# DECLARACIÓN DE VARIABLES DE ENTORNO

export ADMON_MALWARE=/home/snort-cuckoo/Escritorio/admon_malware
export ESCRITORIO=/home/snort-cuckoo/Escritorio
export ANALYSES=/opt/cuckoo/storage/analyses
export WEB_CUCKOO=/var/www/base/web_cuckoo

# FUNCIÓN QUE PERMITE OBTENER LA FECHA ACTUAL DEL SISTEMA

fecha(){
    fecha=`date`
    echo "La fecha es: $fecha"
}

# FUNCIÓN QUE PERMITE OBTENER EL NOMBRE DE LA MUESTRA ANALIZADA, DESDE UN
    REPORTE GENERADO POR CUCKOO SANDBOX

nombre_muestra(){
    nombre=`awk '/<span class="mono">/' report.html`
    nombre=`echo $nombre | cut -d ">" -f 3 | cut -d "<" -f 1`
    echo "El nombre de la muestra analizada es: $nombre"
}


# FUNCIÓN QUE PERMITE DETERMINAR SÍ EL REPORTE ANALIZADO CON CUCKOO
    SANDBOX CORRESPONDE A UN ARCHIVO O A UNA URL

reputacion_vt(){
    valida=`awk '/<h4>/ {print}' report.html`
    valida=`echo $valida | cut -d ">" -f 2 | cut -d "<" -f 1`
}
```

```
# FUNCIÓN QUE PERMITE OBTENER LA FUNCIÓN HASH SHA256 DE LA MUESTRA
    ANALIZADA

sha256_archivo(){
    SHA256=`awk 'NR==405' report.html`
    SHA256=`echo $SHA256 | cut -d ">" -f 3 | cut -d "<" -f 1`
    echo "El SHA256 de la muestra analizada es: $SHA256"
}

# PARA EL ANÁLISIS DE URL's DE CUCKOO SANDBOX, NO EXISTE UNA FUNCIÓN HASH,
    POR LO QUE DESPLIEGA EL MENSAJE QUE NO CUENTA CON DICHA FUNCIÓN HASH

url_sha256(){
    SHA256="Sin SHA256"
    echo "El SHA256 de la muestra analizada es: $SHA256"
}

# FUNCIÓN QUE OBTIENE EL PORCENTAJE DE ANTIVIRUS QUE DETECTARON LA MUESTRA
    COMO UNA AMENAZA.

detection_rate(){
    rate=`awk '/Detection Rate:/ {print}' report.html`
    rate=`echo $rate | cut -d ":" -f 2 | cut -d " " -f 2`
    numerador=`echo $rate | cut -d "/" -f 1`
    denominador=`echo $rate | cut -d "/" -f 2`
    division=$(echo "$numerador/$denominador" | bc -l)
    porcentaje=$(echo "$division*100" | bc -l)
    echo "El índice de detección de antivirus es: $porcentaje""%
}

# FUNCIÓN QUE PERMITE OBTENER LA REPUTACIÓN DE VIRUS TOTAL DE UN ARCHIVO
    ANALIZADO, DESDE UN REPORTE GENERADO POR CUCKOO SANDBOX

reputacion_archivo(){
    cd $ADMON_MALWARE
```

```
    virus_total_archivo=https://www.virustotal.com/es/file/$SHA256/votes-
        resume/
    status_archivo=`curl -Is --connect-timeout 3 $virus_total_archivo |
        head -n 1 | cut -d " " -f 2`
    if [ "$status_archivo" == "200" ]; then
        wget -erobots=off $virus_total_archivo &> /dev/null
        reputacion=`cut -d "\" -f 9 index.html | cut -d "e" -f 6 | cut -d "
            " -f 2`
        echo -e "La reputación del archivo $nombre es de: $reputacion\n"
        if [ -f index.html ]; then
            rm index.html
        fi
    else
        echo -e "Página de Virus Total no encontrada!\n"
    fi
}

# FUNCIÓN QUE PERMITE OBTENER LA REPUTACIÓN DE VIRUS TOTAL DE UNA URL
    ANALIZADA, DESDE UN REPORTE GENERADO POR CUCKOO SANDBOX

reputacion_URL(){
    rurl=`awk '/https/ {print}' report.html`
    rurl=`echo $rurl | cut -d "=" -f 2 | cut -d ">" -f 1`
    protocolo=`echo $rurl | cut -d "/" -f 1`
    dominio=`echo $rurl | cut -d "/" -f 3`
    directorio_1=`echo $rurl | cut -d "/" -f 4`
    directorio_2=`echo $rurl | cut -d "/" -f 5`
    virus_total_url=$(echo "$protocolo//$dominio/es/$directorio_1/
        $directorio_2/votes-resume")
    status_url=`curl -Is --connect-timeout 3 $virus_total_url | head -n 1 |
         cut -d " " -f 2`
    if [ "$status_url" == "301" ]; then
        cd $ADMON_MALWARE
        wget -erobots=off $virus_total_url &> /dev/null
```

```
        reputacion=`cut -d "=" -f 4 votes-resume | cut -d "e" -f 5 | cut -d
            " " -f 2`
        echo -e "La reputación de la URL $nombre es de: $reputacion\n"
        if [ -f votes-resume ]; then
           rm votes-resume
        fi
    else
        echo -e "Página de Virus Total no encontrada!\n"
    fi
}

# REALIZA ITERATIVAMENTE LA OBTENCIÓN DE INFORMACIÓN POR CADA UNO DE LOS
    REPORTES ENCONTRADOS EN EL REPOSITORIO

if [ -d $ANALYSES ]; then
   NUM=0
    cd $ANALYSES
    analyses=$(ls -1 | wc -l)
    for ((i=1; i<=analyses; i++))
    do
        if [ -f $ANALYSES/$i/reports/report.html ]; then
            ruta_cuckoo=$ANALYSES/$i/reports/report.html
            path_web_cuckoo=$WEB_CUCKOO/$i
            file_web_cuckoo=$WEB_CUCKOO/$i/report.html
            reports_cuckoo=/web_cuckoo/$i/report.html
            if [ -d $WEB_CUCKOO ]; then
                if [ ! -d $path_web_cuckoo ]; then
                    mkdir $path_web_cuckoo
                    cp $ruta_cuckoo $path_web_cuckoo
                else
                    if [ ! -f $file_web_cuckoo ]; then
                        cp $ruta_cuckoo $path_web_cuckoo
                    fi
                fi
            else
```

```
        mkdir $WEB_CUCKOO
        mkdir $path_web_cuckoo
        cp $ruta_cuckoo $path_web_cuckoo
    fi
    cd $ANALYSES/$i/reports
    echo "Directorio $i analizado"

    # CUENTA EL NÚMERO DE DIRECTORIOS ANALIZADOS
    NUM=`expr $NUM + 1`
    NEW_NUM=`echo ${NUM}`

    # SE LLAMAN LAS FUNCIONES ANTERIORES
    fecha
    nombre_muestra
    reputacion_vt

    # SE OBTIENE EL SHA256 DE LA MUESTRA ANALIZADA
    if [ "$valida" == "File Details" ];then
        sha256_archivo
    else
        url_sha256
    fi

    # ÍNDICE DE DETECCIÓN DE ANTIVIRUS
    detection_rate

    # SE DETERMINA SÍ ES UN ARCHIVO O UNA URL, PARA OBTENER SU
        RESPECTIVA REPUTACIÓN

    if [ "$valida" == "File Details" ];then
        reputacion_archivo
    else
        reputacion_URL
    fi
```

```
            # SALIDA QUE ES REDIRECCIONADA AL ARCHIVO Informacion_Muestras.
                txt
            echo $fecha!$nombre!$SHA256!$porcentaje%!$reputacion!
            $reports_cuckoo >> $ESCRITORIO/Informacion_Muestras.txt
        fi
    done
    echo "Se han analizado $NEW_NUM reportes de análisis de malware
        exitosamente!"
else
    echo "Error: No existen reportes de Cuckoo por analizar"
fi
```

Archivo eliminar_eventos_ids

```
#!/bin/sh

# DECLARACIÓN DE VARIABLES DE ENTORNO

export ADMON_MALWARE=/home/snort-cuckoo/Escritorio/admon_malware/
eliminar_tablas_ids.php
export SNORT=/var/log/snort/alert
export BARNYARD=/var/log/barnyard2

# DETERMINA LA EXISTENCIA DEL ARCHIVO eliminar_tablas_ids.php PARA
    POSTERIORMENTE ELIMINAR LOS DATOS DE LAS TABLAS DE LA BASE DE DATOS
    DEL IDS

if [ -f $ADMON_MALWARE ]; then
   /usr/bin/php -f $ADMON_MALWARE
fi

# REALIZA EL BORRADO DE LAS ALERTAS QUE EMPLEA BARNYARD

if [ -f $BARNYARD ]; then
   cat /dev/null > $BARNYARD/alert
fi

# ELIMINA TODOS LOS LOGS GENERADOS POR EL IDS SNORT

if [ -f $SNORT ]; then
   rm $SNORT/snort.*
fi
```

Archivo eliminar_tablas_ids.php

```
<?php

// DECLARACIÓN DE VARIABLES

$Icmphdr = 'icmphdr';
$Data = 'data';
$Iphdr = 'iphdr';
$Opt = 'opt';
$Signature = 'signature';
$Tcphdr = 'tcphdr';
$Event = 'event';
$Acid_event = 'acid_event';

// SE ESTABLECE LA CONEXIÓN CON MYSQL

$cnx = mysql_connect("localhost","root","MySQL");
if(!$cnx){
    die("\nNo se pudo conectar: ".mysql_error()."\n");
}

// SE REALIZA LA CONEXIÓN A LA BASE DE DATOS

$bd_sn = mysql_select_db("snort",$cnx);
if(!$bd_sn){
    die("\nError al seleccionar snort:".mysql_error()."\n");
}

// SE REALIZAN LAS CONSULTAS PARA EL BORRADO DE LAS TABLAS DE LA BASE DE
    DATOS DE SNORT

$qry1 = mysql_query("delete from $Icmphdr");
$qry2 = mysql_query("delete from $Data");
$qry3 = mysql_query("delete from $Iphdr");
$qry4 = mysql_query("delete from $Opt");
```

```
$qry5 = mysql_query("delete from $Signature");
$qry6 = mysql_query("delete from $Tcphdr");
$qry7 = mysql_query("delete from $Event");
$qry8 = mysql_query("delete from $Acid_event");

// VALIDACIÓN DEL CORRECTO BORRADO DE LOS DATOS DE LAS TABLAS DE LA BASE
    DE DATOS DEL IDS SNORT

if(!$qry1){
   echo "\nError en consulta ".mysql_error()."\n";
}else{
   echo "\nÉxito! Elementos Borrados de la tabla $Icmphdr: ".
        mysql_affected_rows()."\n";
}

if(!$qry2){
   echo "\nError en consulta ".mysql_error()."\n";
}else{
   echo "\nÉxito! Elementos Borrados de la tabla $Data: ".
        mysql_affected_rows()."\n";
}

if(!$qry3){
   echo "\nError en consulta ".mysql_error()."\n";
}else{
   echo "\nÉxito! Elementos Borrados de la tabla $Iphdr: ".
        mysql_affected_rows()."\n";
}

if(!$qry4){
   echo "\nError en consulta ".mysql_error()."\n";
}else{
   echo "\nÉxito! Elementos Borrados de la tabla $Opt: ".
        mysql_affected_rows()."\n";
}
```

```
if(!$qry5){
   echo "\nError en consulta ".mysql_error()."\n";
}else{
   echo "\nÉxito! Elementos Borrados de la tabla $Signature: ".
       mysql_affected_rows()."\n";
}

if(!$qry6){
   echo "\nError en consulta ".mysql_error()."\n";
}else{
   echo "\nÉxito! Elementos Borrados de la tabla $Tcphdr: ".
       mysql_affected_rows()."\n";
}

if(!$qry7){
   echo "\nError en consulta ".mysql_error()."\n";
}else{
   echo "\nÉxito! Elementos Borrados de la tabla $Event: ".
       mysql_affected_rows()."\n";
}

if(!$qry8){
   echo "\nError en consulta ".mysql_error()."\n";
}else{
   echo "\nÉxito! Elementos Borrados de la tabla $Acid_event: ".
       mysql_affected_rows()."\n";
}

// CIERRE DE CONEXIÓN A MYSQL

mysql_close($cnx);
?>
```

Apéndice C

Código fuente del Sistema de Consulta de Malware.

Este apartado contiene el código fuente y los archivos que componen la página web desarrollada con las tecnologías PHP, JavaScript, jQuery y Bootstrap. Cabe recordar que la principal función de este sistema web es presentar los valores más importantes de las amenzas, detectadas tras ser analizadas con Cuckoo SandBox. Estos valores son presentados al usuario de una manera clara y fácil de comprender.

Para el correcto despliegue de la página web, es necesario que los archivos que a continuación se describen, se encuentren en un directorio llamado *login* en la ruta */var/www/base*.

Para acceder al Sistema de Consulta de Malware, el usuario podrá emplear cualquier navegador web moderno, e ingresar la dirección IP del servidor para acceder a la página de autenticación de dicho sistema. Esto se describe de la siguiente manera: *http://<dirección_ip_servidor>/login/*.

Archivo index.php

```
<?php
    require_once("sesion.class.php");
    $sesion = new sesion();

    if(isset($_POST["iniciar"])){
        $usuario = $_POST["user"];
        $password = $_POST["pass"];
        $validate = validarUsuario($usuario,$password);
```

```
    if($validate["flag"] == true){
        $sesion -> set("user", $usuario, $validate["counter"]);
        header("location: principal.php");
    }else {
        header("location: validacion.php");
    }
}

function validarUsuario($usuario, $password){
    $conexion = mysqli_connect("localhost","root","<Password root MySQL>
        ","usuario_web");
    $sql = "SELECT * FROM usuario WHERE nombre='$usuario' AND password=
        AES_ENCRYPT('$password','key_aes')";
    $consulta = mysqli_query($conexion,$sql);
    $result = mysqli_num_rows($consulta);
    if($_POST['Incidente'] == '1' && $result > 0){
        while($obj = mysqli_fetch_object($consulta)){
            $counter = $obj -> visit_counter;
        }
        $counter = $counter + 1;
        $update_query = "UPDATE usuario SET visit_counter=".$counter." 
            WHERE nombre = '".$usuario."'";
        mysqli_query($conexion,$update_query);
    }else {
        while($obj = mysqli_fetch_object($consulta)){
            $counter = $obj -> visit_counter;
        }
    }
}

mysqli_free_result($conexion, $result);
mysqli_close($link);
if($result > 0){
    $response = array("flag" => true, "counter" => $counter);
    return $response;
```

```
    }else {
        $response = array("flag" => false);
        return $response;
    }
?>

<!DOCTYPE html>
<html lang="en" class="demo-1 no-js">
    <head>
        <meta charset="UTF-8" />
        <meta http-equiv="X-UA-Compatible" content="IE=edge">
        <meta name="viewport" content="width=device-width, initial-scale=1">
        <title>Sistema de Consulta de Malware</title>
        <meta name="description" content="Muestras de malware detectadas por
            un IDS y que han sido analizadas por Cuckoo SandBox." />
        <meta name="keywords" content="malware, ids, samples, worms." />
        <meta name="author" content="Alejandro Bárcenas"/>
        <link rel="stylesheet" type="text/css" href="css/demo.css" />
        <link rel="stylesheet" type="text/css" href="css/component.css" />
    </head>
    <body>
        <div class="container">
            <header class="ccheader">
                <h1><b>Sistema de Consulta de Malware</b></h1>
            </header>

            <div class="wrapper">
                <div class="cclogin icons">
                    <h2 align='center'>Iniciar sesion</h2>
                    <form method="post" action="">
                        <p>
                        <span class="cclogin-addon"><i class="fa fa-user fa-2x
                            fa-spin"></i></span>
                        <input type="text" name="user" value="" placeholder="
                            Nombre de usuario">
```

```
                    </p>
                    <p>
                    <span class="cclogin-addon"><i class="fa fa-key fa-2x fa
                        -spin"></i></span>
                    <input type="password" name="pass" value="" placeholder=
                        "Contraseña">
                    </p>
                    <p>
                    <select name="Incidente">
                       <option disabled="" selected="" value="0">La consulta
                            es por un incidente de seguridad?</option>
                       <option value="1">Si</option>
                       <option value="0">No</option>
                    </select>
                    </p>
                    <h2 align='center'>
                    <p><input type="submit" name="iniciar" align="center"
                        value="Iniciar sesion"></p>
                    </h2>
                </form>

                <?php
                    if (isset($_POST['submit'])){
                        $selected_value = $_POST['Incidente'];
                    }
                ?>
            </div>
        </div>
    </div>
  </body>
</html>
```

Archivo validacion.php

```
<!DOCTYPE html>
<html lang="en" class="demo-1 no-js">
   <head>
      <meta charset="UTF-8" />
      <meta http-equiv="X-UA-Compatible" content="IE=edge">
      <meta name="viewport" content="width=device-width, initial-scale=1">
      <title>Sistema de Registro de Malware</title>
      <meta name="author" content="Alejandro Bárcenas"/>
      <link rel="stylesheet" type="text/css" href="css/demo.css" />
      <link rel="stylesheet" type="text/css" href="css/component.css" />
   </head>
   <body>
      <div class="container">
         <header class="ccheader">
            <h1><b>Sistema de Registro de Malware</b></h1>
         </header>

         <div class="wrapper">
            <div class="cclogin icons">
               <h2 align='center'>El nombre de usuario o la contraseña que
                    ingresaste no coinciden con nuestros registros. <p>
                   Por favor, revisa e inténtalo de nuevo.
                  <form action='index.php' method=POST>
                     <p><br><input type="submit" value="Regresar"></p>
                  </form>
               </h2>
            </div>
         </div>
      </div>
   </body>
</html>
```

Archivo principal.php

```
<?php
    require_once("sesion.class.php");
    require_once("sesion.class.php");
    $sesion = new sesion();
    $usuario = $sesion->get("user");
    $cnx = mysqli_connect("localhost", "root", "<Password root MySQL>", "
        usuario_web");
    $counter_query = "SELECT visit_counter FROM usuario WHERE nombre='
        $usuario'";
    $consulta = mysqli_query($cnx, $counter_query);
    $result = mysqli_num_rows($consulta);
    if($result > 0){
        while($obj = mysqli_fetch_object($consulta))
            $counter = $obj -> visit_counter;
    }
    $_SESSION['counter'] = $counter;
    mysqli_close();
    if( $usuario == false ){
        header("Location: index.php");
    }else{
        ?>
        <!DOCTYPE html>
        <html lang="en" class="demo-1 no-js">
            <head>
                <meta charset="UTF-8" />
                <meta http-equiv="X-UA-Compatible" content="IE=edge">
                <meta name="viewport" content="width=device-width, initial-
                    scale=1">
                <title>Sistema de Consulta de Malware</title>
                <meta name="author" content="Alejandro Bárcenas"/>
                <link rel="stylesheet" type="text/css" href="css/demo.css" />
                <link rel="stylesheet" type="text/css" href="css/component.css
                    " />
```

```
        <link rel="stylesheet" type="text/css" href="assets/bootstrap/
            css/bootstrap.min.css" />
        <link rel="stylesheet" type="text/css" href="css/
            estilos_tablas.css">
        <link rel="stylesheet" type="text/css" href="assets/
            sweetalert2.css">
    </head>
    <body>
        <header>
        <!-- BOOTSTRAP NAVBAR -->
            <nav class="navbar navbar-inverse navbar-fixed-top">
                <div class="container">
                    <div class="navbar-header">
                        <button type="button" class="navbar-toggle
                            collapsed" data-toggle="collapse" data-target=
                            "#navbar" aria-expanded="false" aria-controls=
                            "navbar">
                            <span class="sr-only">Toggle navigation </span>
                            <span class="icon-bar"></span>
                            <span class="icon-bar"></span>
                            <span class="icon-bar"></span>
                        </button>
                        <a class="navbar-brand" href="#"><img src="images
                            /unam_white.png" style="width: 40px;margin-top
                            : -10px;"></a>
                    </div>
                    <div id="navbar" class="collapse navbar-collapse">
                        <ul class="nav navbar-nav navbar-right">
                        <li><a href="#">Bienvenid@ <?php echo $sesion ->
                            get("user"); ?></a></li>
                        <li class="dropdown">
                            <a href="#" class="dropdown-toggle" data-toggle
                                ="dropdown" role="button" aria-haspopup="
                                true" aria expanded="false"> Menú <span
                                class="caret"> </span></a>
```

```
                        <ul class="dropdown-menu">
                            <li><a href="http://<?php print $_SERVER['
                                SERVER_ADDR'];?>:8080/browse/page/1"
                                target="_blank">Ingresar al repositorio
                                de análisis de Cuckoo</a></li>
                            <li><a href="#" onclick="reiniciar_contador
                                ()">Reiniciar contador de Consultas de
                                Malware</a></li>
                            <input id="sesion_user" type="hidden" name="
                                sesion" value="<?php echo $sesion->get
                                ('user');?>" />
                            <li><a href="cerrarsesion.php">Cerrar Sesión
                                </a></li>
                        </ul>
                    </li>
                    </ul>
                </div><!--/.nav-collapse -->
            </div>
        </nav>
    </header>
    <!-- BOOTSTRAP NAVBAR -->
    <div class="container" style="margin-top: 40px;">
        <header class="ccheader">
            <h1><b>Sistema de Consulta de Malware</b></h1>
        </header>
        <?php
            $file = fopen("/home/snort-cuckoo/Escritorio/
                Informacion_Muestras.txt", "r");
            if (!$file){
```

```
                echo '<button class="btn btn-primary" type="button"
                    style="margin-bottom: 10px; background-color
                    :#054586;"> Visitas para consulta de malware <
                    span id="counter" class="badge">'.$_SESSION['
                    counter']. '</span></button><p align=center><br><
                    font color="red"><b>ERROR!</b></font></br></br>
                    No ha sido posible leer el archivo <b>"
                    Informacion Muestras.txt"</b>.</br> Favor de
                    revisar su nombre y sus permisos. </p>';
            }else {
                $loop = 0;
                echo '<button class="btn btn-primary" type="button"
                    style="margin-bottom: 10px; background-color
                    :#054586;"> Visitas para consulta de malware <
                    span id="counter" class="badge">'.$_SESSION['
                    counter']. '</span></button><table class="center
                    table table-condensed table-responsive table-
                    bordered"><tr class="panel-default" style="border
                    -color: #054586; background-color: #054586; color:
                    white;"><th>ID</th><th>Fecha de análisis </th><th>
                    Nombre muestra</th><th>SHA256</th><th>Tasa de
                    Detección de Antivirus (%)</th><th>Reputación
                    Virus Total (-100 a 100)</th><th>Reporte de
                    Cuckoo</th><th>Enviar reporte por e-mail</th></tr
                    >';
                $c = "'";
                while(!feof($file)){
                    $loop++;
                    $line = fgets($file);
                    $field[$loop] = explode ('!', $line);
                    $muestra = $c.trim($field[$loop][1]).$c;
                    $path = $c.trim($field[$loop][5]).$c;
                    if ($field[$loop][0] == '') {
                        break;
                    }
```

```
                        echo '<tr><td>'.$loop.'</td><td>'.$field[$loop
                            ][0].'</td><td>'.$field[$loop][1].'</td><td
                            >'.$field[$loop][2].'</td><td>'.substr(
                            $field[$loop][3],0,5).'%</td><td>'.$field[
                            $loop][4].'</td><td><a href="'.$field[$loop
                            ][5].'" target="_blank">Ver</a></td><td><a
                            href="#" onclick="mail_attachment('.$path.'
                            ,'.$muestra.');">Enviar</a></td></tr>';
                        $file++;
                    }
                    echo '</table>';
                fclose($file);
            }
        ?>
    </div>
    <footer>
        <script src="assets/jquery-2.2.1.min"></script>
        <script src="assets/bootstrap/js/bootstrap.min.js"></script
            >
        <script src="assets/sweetalert2.min.js"></script>
        <script charset="UTF-8">function mail_attachment(ruta,
            nombre_muestra){
            swal({
            title: 'Enviar correo al SysAdmin:',
            showCancelButton: true,
            confirmButtonText: 'Enviar',
            closeOnConfirm: false,
            html: '<p><input id="email" placeholder="e-mail"><input
                id="ruta" value="'+ ruta +'" type="hidden"><input id
                ="nombre_muestra" value="'+ nombre_muestra +'" type
                ="hidden"/>',
            allowOutsideClick: false
            },
            function(){
                var parametro1 = {
```

```
                "muestra" : $('#nombre_muestra').val(),
                "email" : $('#email').val(),
                "path" : $('#ruta').val()
            };

            $.ajax({
                data: parametro1,
                url: 'sendmail.php',
                type: 'POST',
                beforeSend: function () {
                },
                success: function (response) {
                swal(response)
                }
            });
        })
    }

    function reiniciar_contador(){
        var usuario_php = {
            "sesion_user" : $('#sesion_user').val()
        };

        $.ajax({
            data: usuario_php,
            url: 'reiniciar_contador.php',
            type: 'POST',
            }).done(function( msg ) {
            <?php
                $_SESSION['counter']=0;
            ?>
        location.reload();
        });
        }
    </script>
```

```
                </footer>
            </body>
        </html>
    <?php
    }
?>
```

Archivo reiniciar_contador.php

```
<?php
    $user_session = $_POST['sesion_user'];
    $conexion = mysqli_connect("localhost","root","<Password root MySQL>","
        usuario_web");
    $update_query = "UPDATE usuario SET visit_counter='0' WHERE nombre='".
        $user_session."'";
    mysqli_query($conexion,$update_query);
?>
```

Archivo sendmail.php

```
<?php
    $archivo_adjunto = $_POST['muestra'];
    $my_file = "report.html";
    $my_name = "<nombre>";
    $my_mail = "<correo desde donde se enviarán los correos electrónicos>";
    $my_replyto = "<correo hacia donde podrá ser respondido el correo
        electrónico entrante>";
    $my_subject = "$archivo_adjunto";
    $my_message = "Reporte de la muestra \"$archivo_adjunto\" analizada con
        Cuckoo SandBox.";
    $my_name_folder = 'Reporte.zip';
    $mailto = $_POST['email'];
    $link = $_POST['path'];
    $archive = $_SERVER['DOCUMENT_ROOT'].$link;
    $my_filename = 'report.html';

    mail_attachment($my_file, $mailto, $my_mail, $my_name, $my_replyto,
        $my_subject, $my_message, $my_name_folder, $archive, $my_filename,
        $archivo_adjunto);

    function mail_attachment($filename, $mailto, $from_mail, $from_name,
        $replyto, $subject, $message, $name_folder, $file, $filename,
        $attached_file) {
       if (filter_var($mailto, FILTER_VALIDATE_EMAIL)){
```

```
$zip = new ZipArchive;
if ($zip -> open($name_folder, ZipArchive::CREATE)!==TRUE){
   exit("Imposible abrir el archivo: $name_folder");
}else {
   $zip -> addFile($file, $filename);
   $zip -> close();
}

$file = $name_folder;
$file_size = filesize($file);
$handle = fopen($file, "r");
$content = fread($handle, $file_size);
fclose($handle);
$content = chunk_split(base64_encode($content));
$uid = md5(uniqid(time()));
$header = "From: ".$from_name." <".$from_mail.">\r\n";
$header .= "Reply-To: ".$replyto."\r\n";
$header .= "MIME-Version: 1.0\r\n";
$header .= "Content-Type: multipart/mixed; boundary=\"".$uid."\"\
    r\n\r\n";
$header .= "This is a multi-part message in MIME format.\r\n";
$header .= "--".$uid."\r\n";
$header .= "Content-type:text/plain; charset=iso-8859-1\r\n";
$header .= "Content-Transfer-Encoding: 7bit\r\n\r\n";
$header .= $message."\r\n\r\n";
$header .= "--".$uid."\r\n";
$header .= "Content-Type: application/octet-stream; name=\"".
    $file."\"\r\n";
$header .= "Content-Transfer-Encoding: base64\r\n";
$header .= "Content-Disposition: attachment; filename=\"".$file."
    \"\r\n\r\n";
$header .= $content."\r\n\r\n";
$header .= "--".$uid."--";
if (mail($mailto, $subject, "", $header)) {
```

```
            echo "El reporte de la muestra \"$attached_file\" ha sido
                enviado exitosamente!";
            unlink($file);
        }else {
            echo "ERROR! El correo no pudo ser enviado.";
        }
    }else {
        echo "Dirección de e-mail no válida!";
    }
}
?>
```

Archivo sesion.class.php

```
<?php
   class sesion {
      function_construct(){
         session_start ();
      }

      public function set($nombre, $valor, $contador){
         $_SESSION [$nombre] = $valor;
         $_SESSION ["counter"] = $contador;
      }

      public function get($nombre){
         if (isset ( $_SESSION [$nombre] )) {
            return $_SESSION [$nombre];
         }else {
            return false;
         }
      }

      public function elimina_variable($nombre) {
         unset ( $_SESSION [$nombre] );
      }

      public function termina_sesion() {
         $_SESSION = array();
         session_destroy ();
      }
   }
?>
```

Archivo cerrarsession.php

```
<?php
    require_once("sesion.class.php");
    $sesion = new sesion();
    $usuario = $sesion -> get("user");
    if($usuario == false){
        header("Location: index.php");
    }else {
        $usuario=$sesion -> get("user");
        $sesion -> termina_sesion();
        header("location: index.php");
    }
?>
```