ESTUDIO COMPARATIVO DE LA EFECTIVIDAD DE LOS SISTEMAS DE DETECCIÓN DE INTRUSOS

ÁLVARO ERNESTO ROBLES RINCÓN
DAMIÁN FERNANDO PINTO NIÑO

UNIVERSIDAD PONTIFICIA BOLIVARIANA
ESCUELA DE INGENIERÍA
FACULTAD DE INGENIERÍA INFORMÁTICA
BUCARAMANGA
2012

ESTUDIO COMPARATIVO DE LA EFECTIVIDAD DE LOS SISTEMAS DE DETECCIÓN DE INTRUSOS

ÁLVARO ERNESTO ROBLES RINCÓN
DAMIÁN FERNANDO PINTO NIÑO

PROYECTO DE GRADO

ANGÉLICA FLÓREZ ABRIL, MSc.
Directora

UNIVERSIDAD PONTIFICIA BOLIVARIANA
ESCUELA DE INGENIERÍA
FACULTAD DE INGENIERÍA INFORMÁTICA
BUCARAMANGA
2012

PAGINA DE AGRADECIMIENTOS

A mis padres por la paciencia y el apoyo que me han ofrecido durante todo el proceso que han sido mis estudios, a mi directora del proyecto la ingeniera Angélica Flórez Abril por toda la asesoría y paciencia recibida durante el proceso y desarrollo de este proyecto, a mi compañero de proyecto y a todas aquellas personas que de alguna u otra forma ayudaron en el desarrollo de la investigación, mi más sincero agradecimiento.

Álvaro Ernesto Robles Rincón.

Al culminar éste proyecto con esfuerzo y dedicación quiero agradecer a mi familia por el apoyo brindado, a mi directora de proyecto por ser la guía en éste largo camino, a mi compañero de proyecto, a los docentes de mi vida universitaria y a mis amigos.

Damián Fdo. Pinto Niño.

CONTENIDO

pág.

LISTA DE FIGURAS

pág.

LISTA DE GRÁFICAS

pág.

LISTA DE TABLAS

pág.

LISTA DE ANEXOS

pág.

RESUMEN GENERAL DE TRABAJO DE GRADO

TITULO: ESTUDIO COMPARATIVO DE LA EFECTIVIDAD DE LOS SISTEMAS DE DETECCIÓN DE INTRUSOS

AUTOR(ES): ÁLVARO ERNESTO ROBLES RINCÓN

DAMIÁN FERNANDO PINTO NIÑO

FACULTAD: Ingeniería Informática

DIRECTOR(A): ANGÉLICA FLÓREZ ABRIL, MSc.

RESUMEN

Este proyecto tiene como objetivo caracterizar las respuestas de un grupo de herramientas de detección de intrusos, permitiendo establecer cuáles presentan mejor rendimiento frente a diferentes acciones realizadas por medio de troyanos tipo *backdoor*. En el documento se presentan las herramientas de detección de intrusos, seleccionadas en base a unos requerimientos establecidos; de la misma manera se analiza la arquitectura interna de funcionamiento y atributos de cada una de las herramientas, junto con sus ventajas. Además, se presentan los criterios de selección tomados para la elección de los troyanos tipo *backdoor,* las características de cada uno de ellos y las diversas funciones que ofrecen. Conjuntamente se desarrollan actividades como la caracterización de los parámetros que permitirán efectuar la evaluación de cada una de las herramientas. El resultado final del proyecto es el análisis de cada una de las repuestas brindadas por las herramientas de detección de intrusos dentro de la ejecución de las pruebas, y el estudio de los parámetros de los diferentes troyanos, permitiendo identificar qué parámetros son más comunes o no a ser detectados por las herramientas de detección de intrusos.

PALABRAS CLAVES: Herramienta de detección de intrusos, troyanos *backdoor*, arquitectura, funciones, parámetros.

V° B° DIRECTOR DE TRABAJO DE GRADO

GENERAL SUMMARY OF DEGREE OF WORK

TITLE: COMPARATIVE STUDY OF THE EFFECTIVENESS OF INTRUSION DETECTION SYSTEMS

AUTHORS: ÁLVARO ERNESTO ROBLES RINCÓN

DAMIÁN FERNANDO PINTO NIÑO

FACULTY: Computing Engineering

DIRECTOR: ANGÉLICA FLÓREZ ABRIL, MSc.

ABSTRACT

This project aims to characterize the replies from a group of tools for intruders detection, allowing to establish which has a better efficiency against different actions performed by Trojans backdoors kind. The document presents the tools of intruders detection selected through requirements established; in the same way analyzing the inner architecture about operation and attributes from each tool, along with their benefits. Moreover, it presents the selection criteria taken to election of Trojans backdoor kind, their characteristics and several functions they offer. Jointly activities are developed as parameters characterization which will perform the evaluation of each tools. The end project result is the analysis of each replies submitted by intruders detection tools through performing tests and the study the parameters of different Trojans, enabling to identify which parameters are most common or not to be detected by detector intruders tools.

KEY WORDS: Intruders Detection Tool, backdoor Trojans, architecture, functions, parameters.

APPROVAL DIRECTOR GRADE WORK

INTRODUCCIÓN

La gran cantidad de riesgos informáticos como virus, troyanos, gusanos, y *malware* a los cuales se encuentra expuesta la información y/o datos vitales de los usuarios, hace necesario que el resguardo de ésta forme parte de las prioridades en las organizaciones, siendo necesario no solo la implementación de hardware que blinde la información, sino también la implementación de software que colabore en esta labor.

Las herramientas de detección de intrusos, son aplicaciones que poseen la versatilidad de funcionar tanto en grandes como en pequeños grupos de computadores, o en un único computador, permitiendo proteger la información almacenada dentro de ellos. El surgimiento de las herramientas de detección de intrusos (IDS - *Intrusion Detection System*) se basó en el intento de mitigar acciones efectuadas por terceros que aprovechan vulnerabilidades de los Sistemas Operativos (SO) y/o el uso de *malware* para acceder a computadores para extraer información y comprometer el SO.

La inseguridad informática evoluciona constantemente con métodos y formas de vulnerar las diversas herramientas de seguridad existentes, por ello, estas herramientas deben evolucionar acorde con las nuevas vulnerabilidades y amenazas. Los troyanos han surgido como uno de los *malware* utilizados por los ciberdelicuentes por la confiabilidad que ofrece y la versatilidad que posee.

La realización de este proyecto se basa en la comparación de algunas herramientas IDS seleccionadas, ante el uso de troyanos tipo *backdoor*, y de esta forma conocer la eficiencia que dichas herramientas brindan frente a esto tipos de *malware*.

1. SITUACIÓN PROBLEMA

Los IDS son una herramienta utilizada para la seguridad informática, pero poseen algunos problemas como: la generación de falsos positivos y los falsos negativos sobre la efectividad real de las mismas, razón por lo que algunas personas u organizaciones toman la decisión de no utilizarlos. Adicionalmente su efectividad también depende de la experiencia y conocimiento que posea el administrador sobre la herramienta IDS al momento de configurarla y de crear las reglas y patrones a detectar.

Con la utilización de datos sintéticos[1] se espera lograr conocer qué IDS es capaz de encontrar menos falsos negativos y poder reportar más ataques generados por actividades ilegitimas al emplear malware tipo troyanos, hallando posiblemente un IDS que sea capaz de detectar los patrones de una forma efectiva.

[1] Son datos producidos que se pueden aplicar a situaciones dadas y que no se pueden obtener por medición directa.

2. JUSTIFICACIÓN

Actualmente la seguridad informática está obteniendo importancia dentro de las organizaciones, por consiguiente necesita de herramientas idóneas para dar solución a los problemas informáticos que surgen a diario en las redes y computadores.

La seguridad informática posee ramas como la detecciones de patrones utilizando para ello herramientas como los IDS, a pesar de que estas herramientas han existido hace varios años, sus estudios han sido escasos, esto se evidencio al momento de realizar pesquisas por medio de motores búsqueda en internet y en bases datos ofrecidas en la universidad, quizás una de las razones sea por lo complejas que pueden ser al momento de instalarlas, configurarlas y administrarlas.

La cantidad de información que se almacena en discos duros y bases de datos y que se a su vez se intercambia a través de internet requiere el uso de nuevos y mejores métodos para proteger esa información, las herramientas IDS están resurgiendo con nuevos métodos para la detección de intrusos por medio de patrones como lo hacen las herramientas antivirus.

Estudios comparativos basados en la eficiencia que tienen los IDS para detectar ataques son pocos y/o escasos como se envileció en las pesquisas realizadas para conocer el estado del arte de estas herramientas, con este proyecto se desea mostrar la importancia que tienen los IDS para la seguridad informática, buscando además, abrir puertas para continuar investigando sobre cómo trabajan los IDS frente algunos ataques con troyanos.

3. OBJETIVOS

3.1. OBJETIVO GENERAL

Caracterizar las respuestas de un conjunto de herramientas IDS basado en *Host*, ante la incidencia de un grupo de troyanos tipo *backdoor*.

3.2. OBJETIVOS ESPECÍFICOS

- Definir los parámetros de caracterización para determinar la eficiencia de las herramientas IDS en la detección de ataques con troyanos.
- Comparar la forma de respuesta de diferentes tipos de IDS ante el impacto generado por un grupo de troyanos.
- Seleccionar el IDS apropiado a partir de los resultados obtenidos en la prueba de comparación.
- Encontrar los patrones comunes que tienen los troyanos detectados y no detectados por el IDS seleccionado.

4. MARCO TEÓRICO

4.1. ESTADO DEL ARTE DE LOS TROYANOS *BACKDOOR*

Los ataques informáticos efectuados por troyanos son los más utilizados y pueden generar preocupación en las empresas, las cuales podrían ser víctimas en cualquier momento por este tipo de *malware*. Estudios publicados en Evilfingers.com (portal web especializado en seguridad informática) han demostrado que aproximadamente el 80% de los ataques informáticos son ejecutados mediante troyanos [1]; el ataque más frecuente con troyanos es la obtención de información privada de cuentas de correo electrónico de clientes de alguna compañía; también se encuentran los casos en donde se utilizan los troyanos para conseguir contraseñas de computadores, de correo electrónico, y en el peor de los casos claves de cuentas bancarias.

En la actualidad existen personas que ofrecen troyanos como un producto comercial, estas personas brindan actualizaciones y/o mantenimiento al troyano. Los troyanos, al igual que cualquier programa, solo funcionan cuando son activados, en este caso por la víctima, la cual en la mayoría de los casos desconoce lo que está sucediendo, por esta razón el atacante se vale de cualquier método para que su ataque tenga éxito [2].

Los ataques con troyanos logran su objetivo debido a las diversas opciones que les permiten camuflarse. Un estudio reciente detectó una red de computadores *zombies*[2] que fue creada por medio de un troyano derivado de uno llamado *Zeus*,

[2] Zombie: Nombre otorgado a un computador que ha sido infectado por algún tipo de malware, sin conocimiento alguno por parte del usuario. [http://www.alegsa.com.ar/Dic/zombie.php]

éste infectó alrededor de 75.000 sistemas en 2.500 organizaciones en todo el mundo, comprometiendo información de vital importancia para las empresas y hogares en los cuales se encontró este troyano. El estudio reveló que este troyano era capaz de obtener información como credenciales de acceso corporativo, contraseñas de correo electrónico, de cuentas bancarias, de redes sociales, y alrededor de 2.000 archivos de certificados SSL: esta *botnet*[3] fue llamada *Kneber botnet* y fue encontrada a principios del 2010 [3].

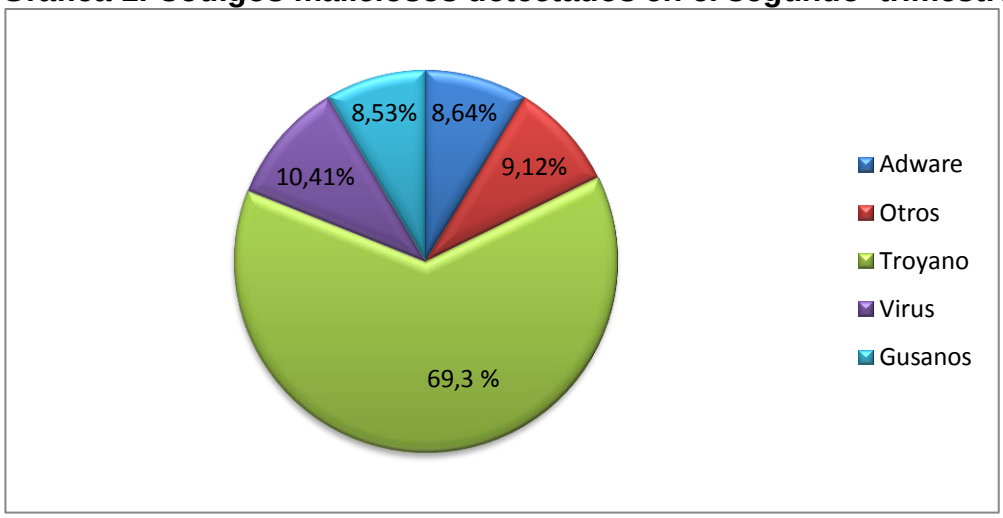
Según el informe trimestral de PandaLabs [4], en el primer trimestre del año 2011 los troyanos fueron el *malware* que más usaron los delincuentes informáticos gracias a sus diferentes opciones de camuflaje e información obtenida (ver Gráfica No. 1).

Algunos de los ataques informáticos más destacados en el año 2009, 2010, 2011 y 2012 fueron el *malware* Aurora, Mariposa y el virus de la policía. Aurora estaba enfocada en atacar a compañías multinacionales, era un ataque dirigido porque solo llegaba a ciertas compañías y solo a los directivos de las empresas o compañías que iban a atacar. Por otro lado, Aurora aprovechó una vulnerabilidad presentada en Internet Explorer [5] que permitía ejecutar códigos de forma remota. El código era un troyano y estaba programado para robar información vital de las empresas como información de propiedad intelectual y robo de cuentas de correo de personal administrativo de la empresa. Una conclusión a la que llegaron era que robaban cuentas de *gmail* sobre activistas de derechos humanos en China. Aurora realizaba conexiones cifradas por lo cual fue difícil detectarlo y también

[3] Botnet: Es un conjunto de computadores infectados por un programa diseñado para automatizar tareas que son controlados de manera remota desde un centro de comando. [http://www.eset.es/centro-de-alertas/diccionario-amenazas]

usaba DNS[4] dinámicos, a pesar de esto algunos de los servidores descubiertos se encontraban ubicados en Texas y Taiwán.

Gráfica 1. Códigos maliciosos detectados en el segundo trimestre de 2011



En el caso de Mariposa, es un *malware* que se propagó por gran parte del mundo creciendo de manera alarmante e infectando principalmente direcciones IP de países como la India, Brasil, México y Colombia (ver Gráfica No. 2) creando una gran *botnet* con computadores a lo largo del planeta. Mariposa fue descubierto por la empresa Canadiense *Defence Intelligence*, quien a su vez se asoció con otras compañías de seguridad como *Panda Security* con el propósito de desmantelar la *botnet* conformada por éste *malware*. La primera acción fue recoger toda la información que permitiera encontrar las subredes que manejaba y encontrar los paneles de control que poseía, y de esta manera observar el comportamiento que

[4] DNS: Sigla de *Domain Name System* en Español Sistema de Nombres de Dominio, sistema que permite relacionar direcciones de internet expresada en lenguaje natural con una dirección IP. [http://technet.microsoft.com/es-es/library/cc787920%28WS.10%29.aspx]

tenía Mariposa. Una de las funciones principales de éste *malware* era alquilar subredes controladas y ofrecer servicios de *spam*, además ofrecía servicios a ciberdelincuentes para robar cuentas de correo con sus respectivas contraseñas, número de tarjetas de crédito y sus contraseñas.

El modo de operación que tenía Mariposa era por medio de una conexión VPN (*Virtual Private Network*)[5], impidiendo descubrir la verdadera dirección IP que utilizaban. En el momento que *Defence Intelligence*, después de una larga investigación, tomó el control de todos los centros de comando que utilizaba Mariposa, el administrador de la *botnet* intentó recuperar el control sobre estos, pero era demasiado tarde y en su afán de recobrar el control sobre la *botnet* se conectó desde el computador de su hogar sin abrir la conexión VPN que le permitía permanecer anónimo, de esta manera se logró descubrir la dirección IP desde la cual se conectaba. El administrador de la *botnet* logró recuperar un centro de comando desde el cual realizó un ataque de denegación de servicio (DoS) a *Defence Intelligence*, como consecuencia dejó por horas sin servicio a este proveedor de servicios de internet (ISP)[6]. Después de recuperarse del ataque de DoS, los especialistas en seguridad informática lograron posesionarse de Mariposa y consiguieron ver que la red que manejaban era de más de 12 millones de equipos en los cuales se encontraba información de grandes e importantes compañías a nivel mundial (todo esto fue logrado con conocimientos

---

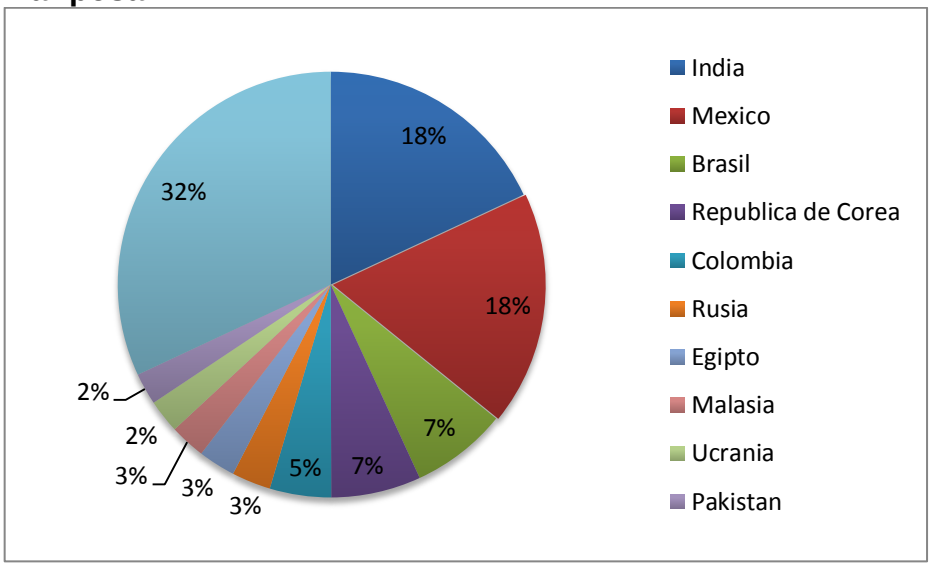
[5] VPN: Red Virtual Privada realizada dentro de una red o infraestructura pública permitiendo accesos y usos controlados de comunicación.

[http://www.cisco.com/web/about/ac123/ac147/archived_issues/ipj_1-1/what_is_a_vpn.html]

[6] Proveedor de Servicios de Internet (ISP): en ingles *Internet Service Provider*, empresa encargada de prestar el servicio de internet para empresas y hogares.

básicos de programación como de seguridad informática y a su vez de herramientas que encontraron en el mercado negro).

Gráfica 2. Porcentaje de direcciones IP por país comprometidas por el *bootnet* Mariposa.



Para el año 2012, específicamente en el primer trimestre, los cibercriminales han realizado ataques a entidades financieras ocasionándoles pérdidas de miles de dólares en robos y en reparaciones a las víctimas; sin embargo, el caso más relevante de los ataques presentados es el *malware* llamado "*virus de la policía*", este *malware* se propagó a través de internet en Europa, y una vez infectado, el computador víctima era re-direccionado a una página web que suplantaba la página web oficial de la policía dependiendo del país en que la víctima accedía a internet, los países afectados fueron España, Italia, Holanda y Alemania, entre otros. La página falsa advertía al usuario que "*ha sido detectado acceso a contenido ilegal desde ese ordenador*", y que el computador iba a ser bloqueado, para evitar el bloqueo del computador era necesario cancelar una suma de 100€. El *malware* realmente bloqueaba el computador víctima, haciendo difícil la eliminación del virus [6].

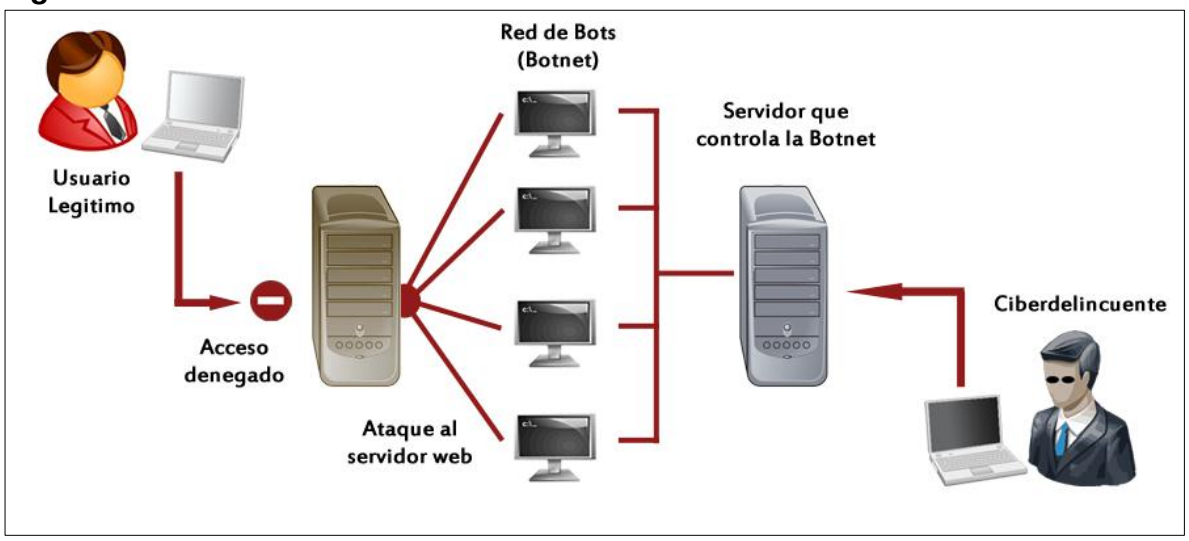
4.2. VULNERABILIDADES DE LOS COMPUTADORES FRENTE A LOS TROYANOS *BACKDOOR*

Las *Botnet* están empezando a generar gran preocupación dentro de las máximas autoridades informáticas y los desarrolladores de software; tal es así que grandes empresas como Microsoft y organismos federales como la FTC (*Federal Trade Commission*) han comenzado a intensificar su lucha contra las *botnet* mediante acciones legales logrando que se suspendan dominios relacionados con la administración de las mismas [7].

Un motivo de alerta en las grandes compañías mundiales son las *botnets* las cuales atacan vulnerabilidades encontradas en diferentes aplicaciones, y según datos publicados en viruslist.com, portal web dedicado a informar al público aspectos de seguridad y amenazas en Internet, empresas como Microsoft, Adobe y Sun Microsystems se encuentran dentro de las empresas con más aplicaciones vulnerables a éste tipo de amenazas. Las *Botnet* sacan provecho de estas vulnerabilidades y reclutan nuevos equipos sin que el usuario sepa que pertenece a una, son difíciles de detectar por las herramientas antivirus existentes, debido que trabajan de la mano con troyanos los cuales son activados por el mismo usuario abriendo puertos sin que el antivirus lo detecte como peligroso, y en la mayoría de los casos estos computadores terminan formando parte de una *botnet* [8].

El modus operandi de una *botnet* (ver figura 1) empieza con una persona "*Ciberdelincuente*" que desea infectar computadores; ésta persona es la encargada de realizar la ingeniería social y el desarrollo o la adquisición del código del troyano; el ciberdelincuente precisa de un servidor desde el cual va a efectuar ataques, recibir información recolectada y reclutar más computadores para la *botnet*. Otra forma de operar consiste en programar el troyano para que éste se pueda replicar solo, por diferentes medios como memorias USB o por medio del correo electrónico de la víctima.

Figura 1. Funcionamiento de una *Botnet*.



Con una *botnet* de gran tamaño se pueden obtener miles de contraseñas de correos electrónicos, claves de cuentas bancarias o realizar ataques de DoS a servidores. La *Kneber Botnet* es un ejemplo de los ataques que se pueden realizar con el objetivo de robar información, esta *botnet* se implementó con el troyano conocido como *Zeus* y por medio de ésta ingresaron a más de 2.500 empresas en 200 países [9]. Es necesario entender que las *botnet* no son un concepto nuevo en la seguridad informática, sino que por el contrario, existen desde tiempo atrás, simplemente que en la actualidad son tenidas en cuenta cada vez más por los ciberdelincuentes ya que ofrecen la posibilidad de conectarse con cualquier computador con acceso a Internet por medio de una deficiencia ya sea propia del sistema operativo o de una aplicación específica.

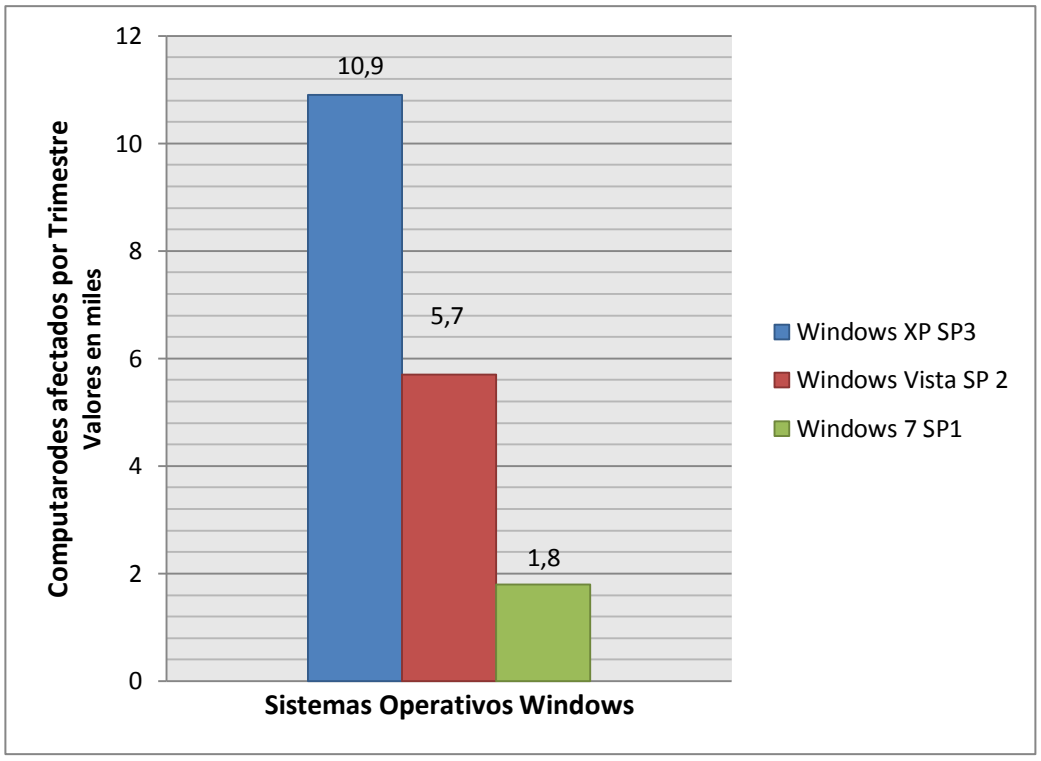
Tal vez la mayor vulnerabilidad que puede encontrar un troyano en un computador es la falta de conocimiento en seguridad informática que tenga el usuario, porque ningún código malicioso se activa solo, requiere de una persona que lo ejecute. Otro tipo de vulnerabilidad que poseen los computadores son los puertos abiertos, porque los troyanos requieren de un puerto por el cual se puedan comunicar y tener acceso a la información. Las empresas que producen antivirus son capaces de detectar códigos maliciosos, pero cada día nacen nuevas amenazas las cuales

en ocasiones son indetectables para los antivirus. También existen los ataques llamados "*día cero*" los cuales pueden causar grandes daños a empresas y hogares porque se aprovechan de vulnerabilidades que aún no son conocidas por los desarrolladores de software antivirus.

4.3. EMPLEO DE TROYANOS A LARGO PLAZO

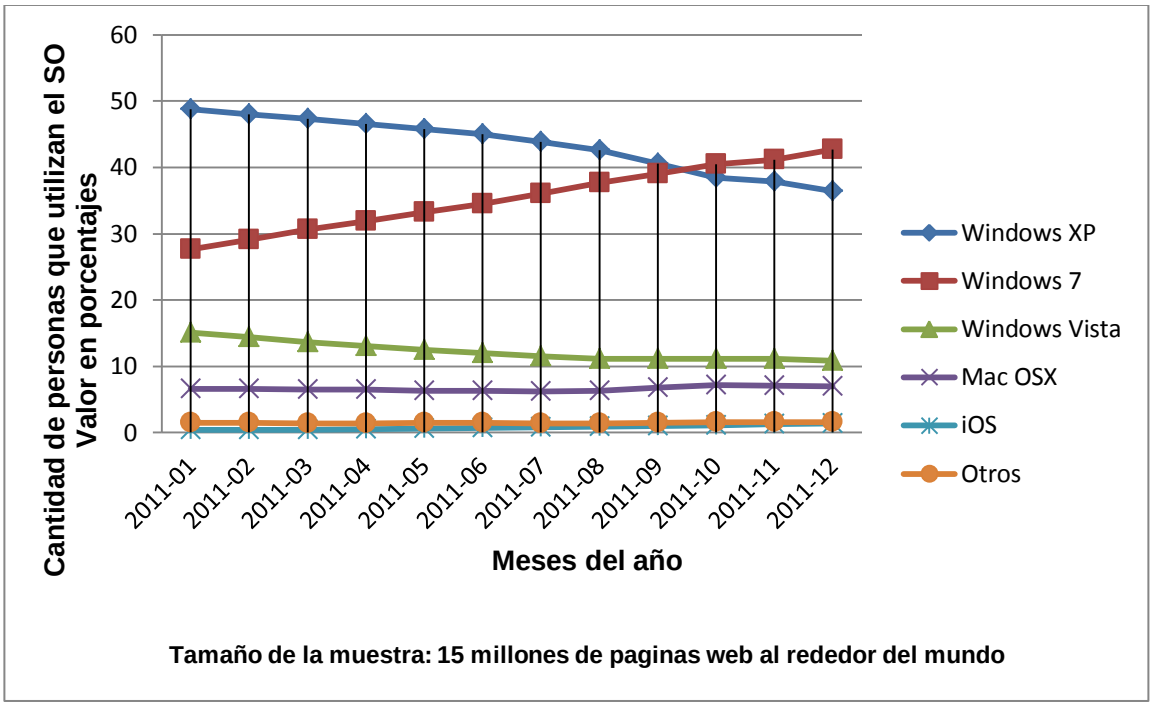
El uso de Internet cada día es más frecuente en una gran variedad de dispositivos, en la actualidad no solo los computadores tienen acceso a esta gran red, sino también teléfonos móviles y televisores, aumentando la cantidad de usuarios que navegan por Internet y que pueden ser víctimas de algún tipo de *malware*. Según estudios publicados por PandaLabs se ha demostrado que los ataques por medio de troyanos son los más usados por los ciberdelincuentes [4].

Gráfica 3. Tasa de infección trimestral por Sistema Operativo de las tres últimas versiones de Microsoft Windows en el año 2011



Como se muestra en la gráfica 3, con la nueva versión del SO de Microsoft, Windows 7, se ha logrado disminuir el número de ataques realizados a esta familia de SO [10]. En el año 2011, Windows 7 fue el SO más utilizado, seguido de Windows XP como lo demuestra *Statcounter* (portal web de estadísticas), lo que indica que a pesar de la alta vulnerabilidad que Windows XP presenta aún gran cantidad de usuarios lo utilizan (ver grafica 4) [11].

Gráfica 4. Top 5 de los sistemas operativos más utilizados en el año 2011



Anteriormente, lo que buscaba un ciberdelincuente era atacar a usuarios y a empresas ocasionándoles el mayor daño posible enfocado a software y/o hardware. La modalidad usada recientemente busca robar información importante de las compañías sin levantar sospechas brindando al ataque el ocultamiento deseado. Otro factor importante que se debe tener en cuenta es el uso de la Internet, los usuarios de esta gran red van creciendo rápidamente, razón por la

cual también se ha comenzado a comercializar a través de internet, las personas pueden comprar desde carros, casas, ropa, electrodomésticos, hasta pagar recibos o comprar boletas para cine. El uso a largo plazo de los troyanos está en atacar esa confianza que tienen los usuarios al momento de efectuar sus pagos y compras por Internet [12].

Un factor importante a tener en cuenta es lo que se está desarrollando para la web 2.0, que ofrece servicios como redes sociales, wikis, blogs, etc., esta red genera el ambiente apropiado para la evolución de los troyanos. El uso masivo de las redes sociales está llevando también a que los ciberdelincuentes estén desarrollando troyanos para obtener información de las cuentas de *Facebook*, *Twitter*, entre otras [13].

4.4. SISTEMAS DE DETECCIÓN DE INTRUSOS

Un Sistema de Detección de Intrusos (IDS), es una herramienta informática que permite detectar en tiempo real accesos no autorizados o ataques realizados a un computador o red dependiendo de la herramienta utilizada y de su configuración [14].

En procura de una mayor precisión de los datos a ser obtenidos para el desarrollo del proyecto y de esta manera poder efectuar una investigación con un buen criterio, se han definido como requerimientos de selección de las herramientas IDS a comparar los siguientes:

- Deben ser HIDS (*Host-Base Intrusion Detection System*). Sabiendo que las pruebas a realizar con los troyanos van a tener lugar en un computador específico, de este modo poder monitorizar los cambios que ocurran en él.

- Software Libre. Este proyecto está orientado a proteger un computador sin tener que invertir en costos elevados y aunque existe un grupo de IDS comerciales, también existen aplicaciones gratuitas de gran utilidad y funcionamiento con las cuales se puede trabajar.

- Basado o no en Snort. Snort es un IDS que posee reconocimiento a nivel mundial por ser funcional, por ser GPL[7], y porqué varios de los IDS que se encuentran en Internet están basados en él, pero sería interesante poder también trabajar con algunos que no sean basados en él para observar las diferencias.

4.5. HERRAMIENTAS IDS

Las herramientas que se optaron por implementar y que posee las características mencionadas…en la sección 4.4 … son:

4.5.1. OSSEC (Ossec Source Security) HIDS. Es una aplicación de código abierto, que realiza análisis de *logs*, políticas de monitorización, chequeo de integridad de archivos, detección de *rootkits* y posee respuesta activa en tiempo real, además de tener como gran ventaja que puede ser ejecutado en diversos SO [15]. Cuenta con el aval que le da el ser desarrollado por Trend Micro, una

[7] GPL (*General Public License*): En español Licencia Publica General, es una licencia que está dirigida a proteger la distribución, modificación y uso de software libre con el objetivo de protegerlo de apropiación que limite a los usuarios de esas opciones.

compañía japonesa de seguridad informática con más de 20 años de experiencia en el campo.

4.5.2. Sguil. Desarrollado por analistas de seguridad, maneja una interfaz gráfica amigable mostrando en tiempo real los eventos que van ocurriendo en el computador, está escrito en tcl/tk[8] y está orientado a eventos; además tiene la capacidad de poder trabajar en la mayoría de los sistemas operativos existentes y puede trabajar de la mano con Snort obteniendo mejores resultados [16].

4.5.3. Prelude. Es un IDS con capacidad de recoger todos los registros y/o eventos que ocurren en el computador y a su vez puede trabajar con Snort, Samhain, OSSEC, Auditd, entre otros; además, tiene la capacidad de trabajar como HIDS o NIDS[9]. Una característica resaltable de Prelude es que utiliza un formato único llamado IDMEF (*Intrusion Detection Message Exchange Format*) para la comunicación entre sus módulos, es un estándar creado por la IETF (*The Internet Engineering Task Force*) que ha tenido éxito en compañías Europeas importantes [17].

4.5.4. Snort. Es uno de los IDS libre más usado en el mundo porque brinda la facilidad de poder utilizarlo en la mayoría de sistemas operativos. Otra característica importante es que trabaja con licencias de GPL, cuenta con más de

---

[8] TCL/TK (Tool Command Language/Tool Kit) Es un lenguaje de programación dinámico, con una gran utilizado para programación de aplicaciones web y de escritorio, entre otros.

[9] NIDS: Sigla de *Network Intrusion Detection System*, en español Sistema de Detección de Intrusos de Red, Es una herramienta que se encarga de monitorizar el tráfico de información que circula por una red en búsqueda de accesos no autorizados. [http://studies.ac.upc.edu/FIB/CASO/seminaris/2q0304/M6.pdf]

400.000 usuarios registrados en sus bases de datos. Al ser un software con licencia GPL, brinda la posibilidad de que casas desarrolladoras estén constantemente revisando el software en busca de anomalías para informar y poder corregirlas, también posee una base de datos que se va actualizando constantemente. Snort puede trabajar como HIDS o NIDS y existen herramientas de terceros las cuales se pueden acoplar perfectamente para obtener un mejor funcionamiento [18].

4.6. TROYANOS

En el contexto de la documentación o estudios acerca de troyanos tipo *backdoor* más significativos, la información disponible es escasa, por este motivo se hace necesaria la utilización de este tipo de *malware* según experiencias encontradas en foros y páginas web especializadas en seguridad informática, tomando como referencia la popularidad de los troyanos en internet, y estableciendo popularidad como la cantidad de páginas web donde se encuentre información acerca de los troyanos con base a resultados obtenidos por medio de diferentes motores de búsqueda en internet (ver tabla 1), de esta manera se seleccionaron 4 troyanos tipo *backdoor* con diferente popularidad, logrando así diversidad de troyanos tipo *backdoor* para realizar las pruebas con los IDS.

4.6.1. Darkcomet RAT. Es un troyano tipo *backdoor* que posee entorno gráfico de fácil uso. Aunque es común encontrarlo clasificado como una herramienta de administración remota, su uso más frecuente es el de troyano y es detectado como tal por las herramientas antivirus, debido a las funciones que incorpora en él [19].

Tabla 1. Popularidad de troyanos por medio de motores de búsqueda en internet

Troyano	Palabras de búsqueda	Resultado de números de sitios con Google (Enero 2011)	Resultado de números de sitios con Altavista (Enero 2011)	Resultado de números de sitios con Bing (Enero 2011)
Darkcomet	troyano darkcomet	91.500	277	1.500
Poison ivy	troyano poison ivy	52.700	61.80	8.260
Spy-net	troyano spy-net	42.600	30.300	5.350
Bifrost	troyano bifrost	34.600	15.200	18.700

4.6.2. Poison Ivy. Es una herramienta de control remoto usada como troyano, posee características para implantarse en el sistema y pasar desapercibido por el antivirus. Desde el año 2008 no existen actualizaciones ni nuevas versiones oficiales, pero aún es utilizado, a pesar de ser detectado por los antivirus, debido a que con técnicas de cifrado y ocultación se puede camuflar dentro del computador [20].

4.6.3. Bifrost. Ésta herramienta, permite tener acceso remoto a un computador, posee funcionalidades de captura de datos, imagen y cambio de registros, su funcionamiento es similar a otras herramientas de éste tipo que manejan

arquitectura cliente/servidor. Los antivirus la reconocen como código malicioso e impiden su ejecución. Actualmente no tiene una página oficial para su descarga pero es posible encontrarlo y descargarlo por medio de internet en foros.

4.6.4. Spy-net. Troyano que permite el uso de equipos de forma remota, tiene capacidades similares a otros troyanos tipo *backdoor*, con la posibilidad de crear conexión directa con el servidor utilizando la arquitectura cliente/servidor; permite la creación de *plugins* para que el troyano se conecte directamente a un servidor FTP[10] o web: además, ofrece un entorno estéticamente amigable y moderno para el usuario.

4.7. ARQUITECTURA DE LOS TROYANOS *BACKDOOR*

La arquitectura de un troyano varía según el ataque que se desee realizar y/o según el tipo de troyano, pero la mayoría de los troyanos tienen una estructura base que consta de tres partes fundamentales:

- Módulo de seguridad
- Módulo de daño
- Módulo de comunicación

Estos módulos conforman la estructura del troyano y poseen subdivisiones con operaciones específicas.

[10] FTP: En ingles *File Transfer Protocol*, es un servicio utilizado para transferir archivos entre dos computadores.

4.7.1. Módulo de Seguridad. En éste módulo se encuentran las opciones que le permiten al troyano poder camuflarse y pasar desapercibido al usuario. Dependiendo de lo que el programador desee que el troyano realice, éste puede poseer la cualidad de desactivar el software de seguridad que el computador tenga como el *firewall* o el antivirus.

A su vez, éste módulo puede contener un mecanismo de autoprotección el cual se basa en permanecer el mayor tiempo posible en el equipo infectado sin ser detectado; una forma de lograr esto es por medio de polimorfismo[11], de esta forma puede engañar o confundir a la herramienta antivirus.

Así mismo, también se pueden encontrar mecanismos de actualizaciones del código del troyano o descargas que mejoraran el funcionamiento del troyano e inclusive que le agrega nuevas funcionalidades.

4.7.2. Módulo de Daño. En éste módulo se implementan las funciones de daño del troyano, estas funciones permiten la captura de datos, captura de contraseñas, captura de audio, captura de video, robo de archivos o documentos, envío de spam, deshabilitar las herramientas de seguridad y apertura de puertos, entre otras. Las funciones que el programador quiera agregarle al troyano también dependen del conocimiento que éste tenga acerca de la actividad a realizar.

[11] Polimorfismo: Técnica usada por algunos tipos de *malware* para cambiar algunas secciones de su código haciendo más compleja su detección.

4.7.3. Módulo de Comunicación. Para lograr los propósitos del troyano, éste necesita comunicarse para enviar la información que ha encontrado o recolectado de la víctima o víctimas, para ello el creador del troyano puede tener un servidor de correo el cual le permita enviar y recibir información, o solo obtener la información cuando la víctima y el ciberdelincuente se encuentren online; además, es importante tener una dirección IP oculta, de esta forma será difícil rastrearlo. Otro aspecto a tener en cuenta en el módulo de comunicación es el cifrado de la información ya que el programador del troyano debe evitar que los datos conseguidos viajen por Internet sin ser codificados permitiendo que cualquier persona pueda accederlos de manera fácil.

4.8. ANTIVIRUS

Las herramientas antivirus son aplicaciones que brindan seguridad para la protección de la confidencialidad y disponibilidad de la información almacenada en los computadores.

La funcionalidad de las herramientas antivirus está enfocada en tres aspectos:

- **Detección:** Una vez infectado el computador, se debe determinar la localización del código malicioso.
- **Identificación:** Determinar el tipo de código malicioso que ha infectado el computador.
- **Eliminación:** Después de identificar, se debe remover todos los rastros o huellas del código malicioso, evitando su propagación y restaurar el sistema a su estado original.

Teniendo en cuenta estos aspectos se han logrado identificar 4 generaciones de herramientas antivirus [21]:

- Primera generación: Escaneo simple
- Segunda generación: Métodos heurísticos e integridad
- Tercera generación: Activación de trampas de ejecución
- Cuarta generación: Protección completa

PassMark, una empresa desarrolladora de software australiana, realizó un estudio comparativo entre herramientas antivirus comerciales y libres.

El estudio consistió en utilizar métricas para observar el tiempo de respuesta del SO ante la carga de los procesos utilizados por el antivirus.

Las métricas que se emplearon fueron:

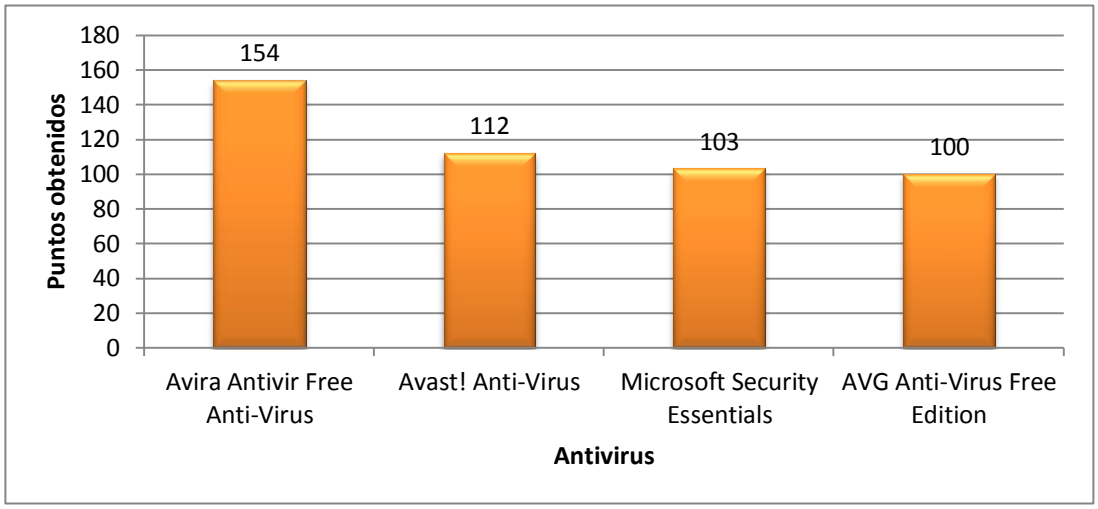
Tabla 2. Métricas utilizadas por *PassMark* para el estudio comparativo de antivirus realizado en el año 2010

Métrica	Descripción
Boot Time	Tiempo tomado por la herramienta antivirus para cargarse al iniciar el SO.
Scan Time	Tiempo requerido para escanear archivos en el computador. Para este caso se utilizaron 1.2 GB de información.
Scan Time of a Solid State Drive (SSD)	Tiempo que se demora en escanear la misma muestra de archivos en la prueba *Scan Time* utilizando un disco duro sólido.
User Interface Launch Time	Mediciones del tiempo que toma cargar la interfaz gráfica de usuario de la herramienta antivirus.
Memory Usage during System Idle	Cantidad de memoria RAM utilizada por la herramienta antivirus en estado de reposo y sin escanear. Adicionalmente reconoce los recursos que la

	herramienta antivirus utiliza del SO.
Browse Time	Tiempo que toma navegar por algunos sitios web con el antivirus analizando el computador.
Internet Explorer Launch Time	Mide el tiempo que demora en cargar Internet Explorer en memoria cache. Esta prueba se realiza dos veces y se saca un promedio observando si el antivirus genera alguna tipo de alerta o lentitud en la prueba.
Installation Size	Mide el tamaño total de instalación de algún programa y el tamaño total utilizado del disco duro.
Installation Time	Tiempo que se demora en instalar un programa con los valores por omisión.
Registry Key Count	Cantidad de llaves de registro que crea el antivirus después de ser instalado; entre menos llaves crea se obtiene mejor calificación.
File Copy, Move and Delete	Tiempo que demora en copiar y pegar diversos tipos de formatos en diferentes partes del disco duro.
Installation of Third Party Applications	Tiempo que se requiere para instalar y/o desinstalar software de terceros.
Network Throughput (previously named "Binary Download Test")	Tiempo que toma descargar diversos formatos de archivos como imágenes, videos, documentos de texto, etc., por medio del navegador de internet IE7 con el antivirus ejecutándose.
File Format Conversion	Tiempo requerido para convertir un archivo MP3 a WAV y el mismo MP3 a WMA.
File Compression and Decompression	Tiempo requerido para la compresión y descompresión de diversos formatos de archivos.
File Write, Open and Close	Tiempos de entrada y salida que toma para abrir y cerrar el mismo archivo varias veces.

Las pruebas anteriores se realizaron a antivirus comerciales y libres. En la gráfica 5 se pueden observar los resultados de las pruebas con las métricas mencionadas en la tabla 2 a los antivirus libres y cuáles obtuvieron el mejor resultado en las pruebas [22].

Gráfica 5. Puntuación obtenida en el estudio comparativo realizado por *PassMark* de los antivirus libres.



El estudio realizado por *PassMark* demostró que entre las herramientas antivirus libres, Avira se destacó, obteniendo resultados satisfactorios comparados con otras herramientas antivirus libres y comerciales evaluadas en el mismo estudio. Entre las pruebas obtenidas por Avira se destacan la creación mínima de llaves de registro que produce al SO en su instalación, el bajo consumo de memoria RAM cuando se encuentra en espera, el tiempo que necesita para inicializarse en el SO, el tiempo que toma instalar alguna aplicación de terceros con Avira analizando el proceso de instalación, entre otros.

PassMark demostró a través del estudio comparativo la efectividad que posee Avira como herramienta antivirus, con base a ese estudio y, en la necesidad de tener una herramienta antivirus en el proyecto, Avira será la herramienta antivirus seleccionada.

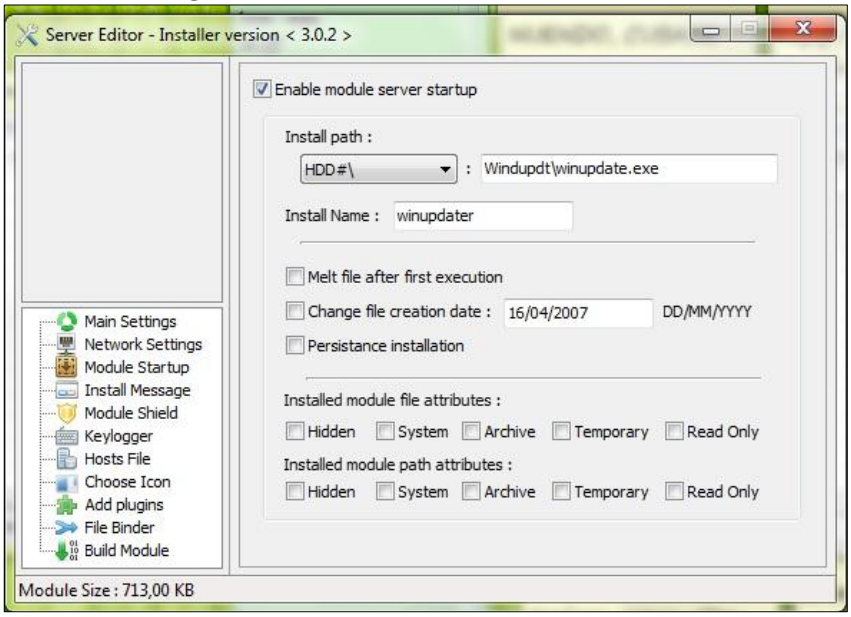
5. ANÁLISIS DE LAS HERRAMIENTAS

5.1. ARQUITECTURA Y FUNCIONES DE LOS TROYANOS

5.1.1. Darkcomet. Es un troyano de nueva generación con un aspecto más amigable y de fácil uso el cual tiene una gama de funcionalidades en los tres módulos de su arquitectura, permitiéndole al ciberdelincuente tener una herramienta con una interfaz gráfica intuitiva y de fácil uso, también permite la personalización del troyano, entre otras posibilidades.

Módulo de Seguridad. En éste módulo Darkcomet ofrece varias opciones para protegerse (ver figura 2), las cuales van desde escoger la ubicación en la cual se desea camuflar el troyano hasta la posibilidad de nombrarlo como un proceso normal del SO.

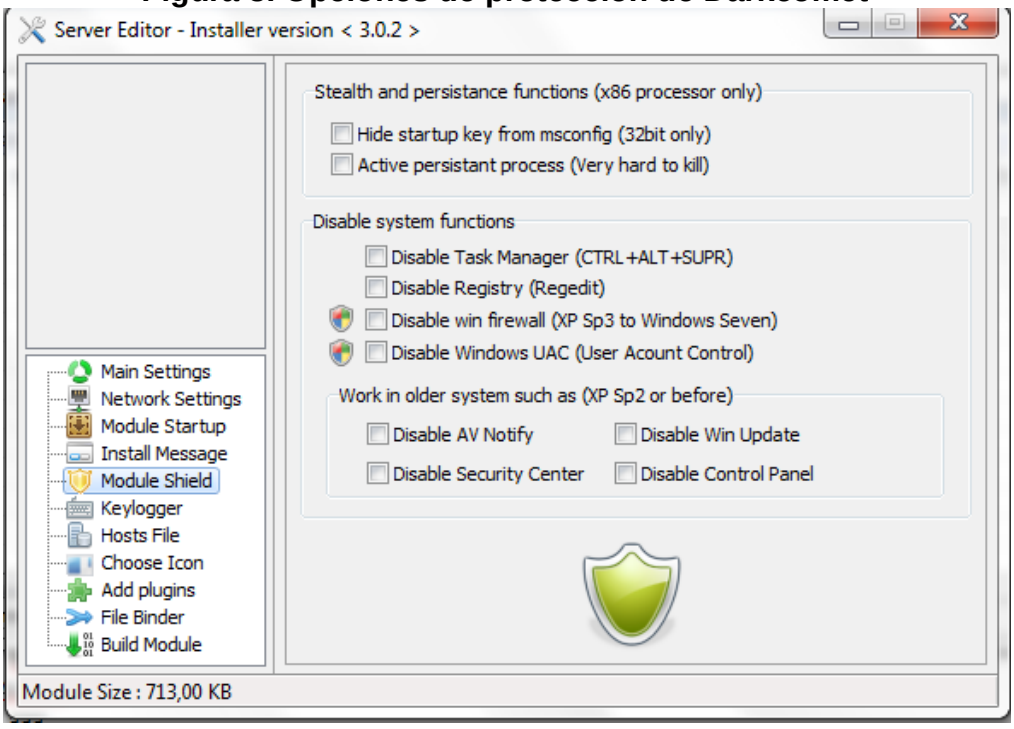
Figura 2. Panel de inicio Darkcomet



Permite ejecutarse como un programa de inicio del SO, de tal forma que, siempre que el usuario se encuentre trabajando en el computador, el ciberdelincuente desde el cliente pueda acceder a él sin ningún problema y además sin tener que volver a enviar el servidor a la víctima, de éste modo puede camuflarse como un servicio o proceso nativo del SO y así intentar eludir las posibles protecciones del computador.

La protección que ofrece Darkcomet permite configurar persistencia (ver figura 3) logrando mantenerse instalado en el SO aunque se intente eliminar, de esta forma asegura que pueda estar el mayor tiempo posible en él. También posee opciones para deshabilitar algunas características de seguridad del SO como el *firewall*, el editor de registros del computador, el administrador de tareas, y el control de cuentas de usuario. Como opciones adicionales puede también deshabilitar las actualizaciones del SO, el panel de control, entre otros.

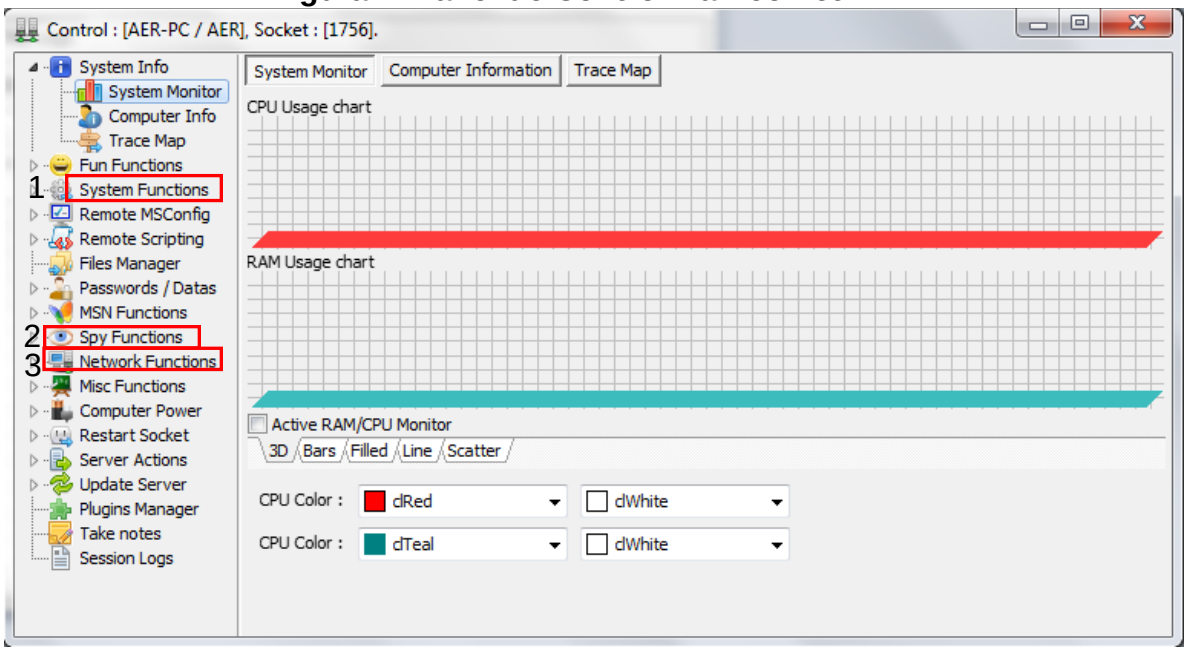
Figura 3. Opciones de protección de Darkcomet



Módulo de Daño. Permite configurar un *keylogger* el cual captura los datos digitados por la víctima, para realizar esto es necesario habilitar la opción en el panel de control del troyano. La información puede ser enviada a un servidor ftp, de lo contrario será enviado al ciberdelincuente en un correo.

En el momento que se tenga instalado el servidor en el computador víctima, éste proporciona opciones que permiten conocer vulnerabilidades y tener control sobre la víctima, estas se pueden ver en la opción *System Functions* (ver figura 4), desde donde se pueden controlar todos los procesos del computador víctima (1), registrar el servidor como servicio del sistema, abrir y utilizar la consola de Windows y realizar cambios por medio de comandos al sistema, además, desinstalar aplicaciones y cambiar privilegios.

Figura 4. Panel de Control Darkcomet RAT



Además, se puede encontrar la opción de transferir archivos del servidor al cliente y viceversa. Otra característica es el *Spy Functions* (2) en el cual se encuentran las opciones de capturar imágenes del computador víctima, capturar imágenes de

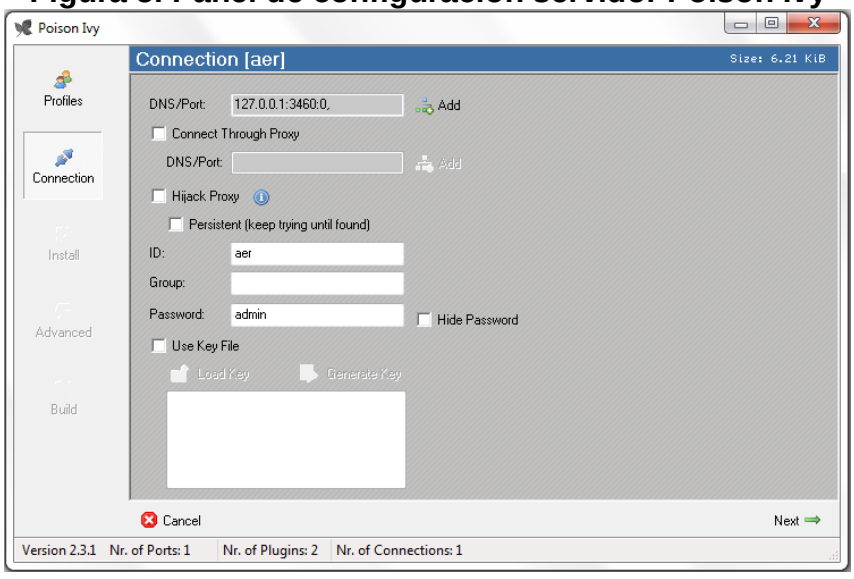
la cámara web, activación de un *keylogger* y captura de audio, función solamente habilitada cuando el cliente y la víctima estén conectados.

Módulo de Comunicación. Se encuentra contenido en la opción *Network Functions* (3) y se utiliza para enviar información y/o para realizar actualizaciones del troyano, además se puede conocer qué vulnerabilidades puede tener el computador revisando los puertos que tiene abiertos, conocer qué computadores conforman o hacen parte de la red LAN donde se encuentra el servidor.

5.1.2. Poison Ivy. Éste troyano creado en el año 2006 con actualizaciones hasta el año 2008 aún en el 2012 sigue siendo funcional. Su entorno gráfico es simple y fácil para el usuario, su arquitectura es cliente/servidor.

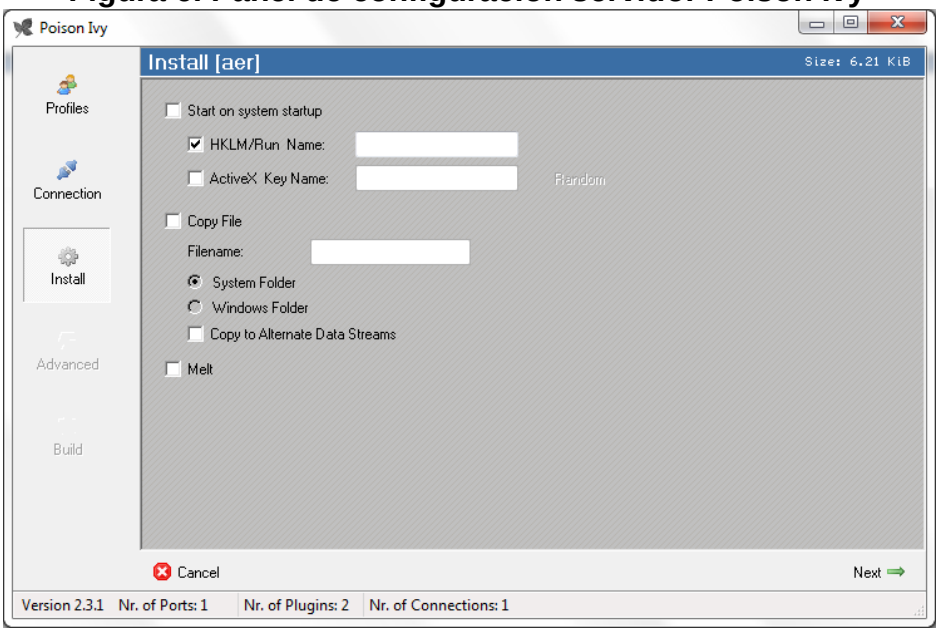
Módulo de Seguridad. Ofrece diversas opciones de configuración (ver figura 5) para la conexión, protección y acceso del troyano, una de ellas es establecer una contraseña que garantice el uso del servidor por parte del ciberdelincuente.

Figura 5. Panel de configuración servidor Poison Ivy



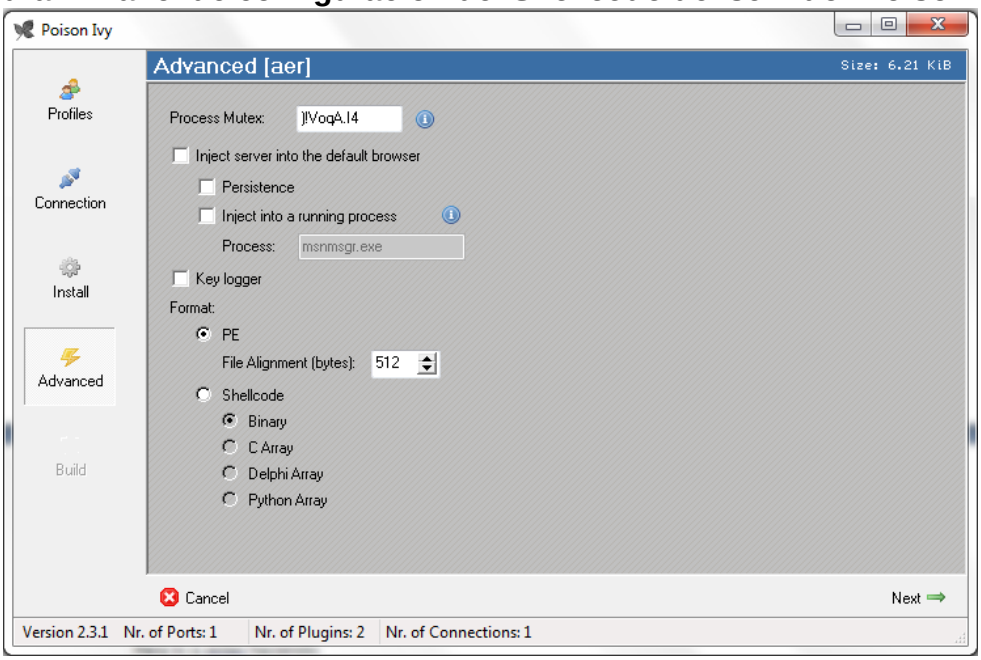
Otra característica de configuración importante de Poison es que para poder camuflarse posee opciones como (ver figura 6): ubicarse en el arranque del SO, en la carpeta Windows y/o también en las llaves de registro del sistema, de tal manera que cuando el SO se inicie, el troyano también lo haga sin que la víctima se entere. De igual forma, para no dejar rastros después de ejecutado, se encuentra la opción *melt* la cual permite desaparecer el instalador del servidor después de ser ejecutado por primera vez.

Figura 6. Panel de configuración servidor Poison Ivy



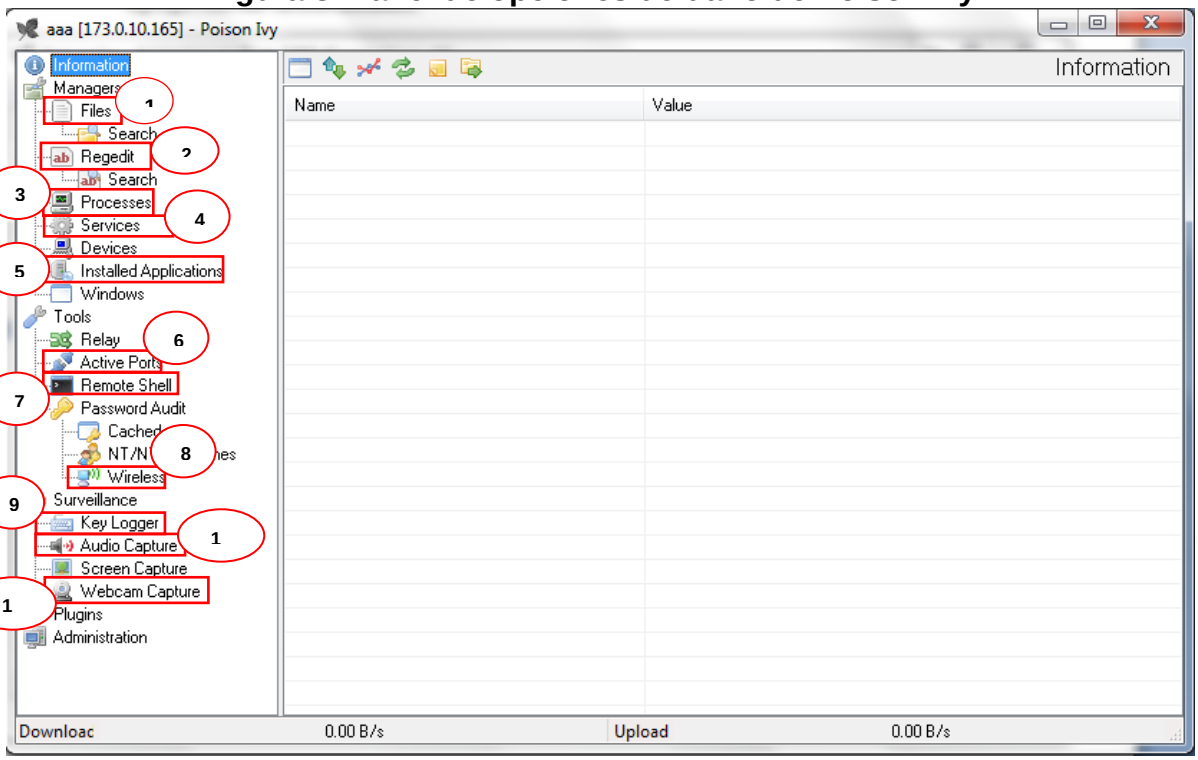
Poison Ivy define una función que le permite pasar como proceso legítimo, como el de Internet Explorer, Messenger, Firefox, o cualquiera que sea utilizado normalmente por los usuarios, logrando engañar a la víctima al momento de revisar las aplicaciones que están ejecutándose, evitando que sea eliminado de manera fácil. No obstante, también tiene la opción de escoger el tipo de *shellcode* que va a tener el troyano (ver figura 7), aunque la opción seleccionada por omisión es Binario, el ciberdelincuente puede seleccionar entre algunos lenguajes como C, Delphi y Phyton.

Figura 7. Panel de configuración del *Shellcode* del servidor Poison Ivy



Módulo de Daño. Éste módulo proporciona diversas opciones (ver figura 8) como: descargar archivos (1), modificar el editor de registros (2), ofrece la opción de revisar procesos que la víctima tiene (3), los servicios que está utilizando (4), eliminar algún software instalado (5), entre otros. Por otra parte, tiene las opciones de mirar los puertos abiertos del sistema para poder utilizarlos (6), trabajar con la consola de *Windows* (7) sin que la víctima lo advierta, mirar las contraseñas que el sistema maneja, como las de las cuentas de usuarios y hasta las contraseñas de la red *WiFi* (8). Así mismo, como espía, éste troyano ofrece opciones básicas de otros troyanos, permite mirar lo que el usuario está haciendo, permite poder controlar el *mouse* y el teclado. Ofrece un *keylogger* (9) que solo funciona si ambas partes están conectados. También posee la opción capturar el sonido por medio del micrófono (10) y capturar imágenes de la cámara web (11) si la víctima posee.

Figura 8. Panel de opciones de daño de Poison Ivy



Módulo de comunicación. Poison Ivy no posee un módulo específico de comunicación como otros troyanos, pero tiene la opción de mirar los puertos que tiene abiertos el computador y de esta forma establecer algún tipo de comunicación entre los dos. También permite compartir el servidor, es decir, que otro cliente pueda conectarse a él.

5.1.3. Bifrost. Su funcionalidad es similar a la de los demás troyanos y hoy en día no se consiguen más actualizaciones.

Módulo de seguridad. Este troyano, al igual que los analizados anteriormente, posee la cualidad de camuflarse como un proceso nativo o hacerse pasar por otro, y se debe configurar en el momento que se esté creando el servidor. También

presenta la opción de asignarle una contraseña (ver figura 9) para que solo se conecte con el ciberdelincuente.

Figura 9. Configuración puertos y contraseña *Bifrost*

Al momento de configurar el servidor se encuentran opciones como (ver figura 10): ocultar el instalador cuando éste se ejecute, configurar la carpeta donde se va a instalar el servidor para que éste tenga persistencia en el computador afectado y asignar el nombre de un proceso cualquiera que sea de común uso del SO.

Como forma de protección ofrece tres modos de sigilo (ver figura 11): *visible, cautious, agressive*; el primero no ofrece ningún tipo de sigilo, el segundo ofrece solo lo necesario para pasar desapercibido y el último es el más completo, aunque puede comprometer la funcionalidad del troyano.

También se puede cambiar la fecha al archivo para intentar confundir las herramientas antivirus y/o usuario del computador que encuentre el archivo como sospechoso. El troyano también posee la función de *delayed connection* la cual permite colocar una fecha específica para realizar las conexiones al servidor.

Figura 10. Configuración servidor *Bifrost*

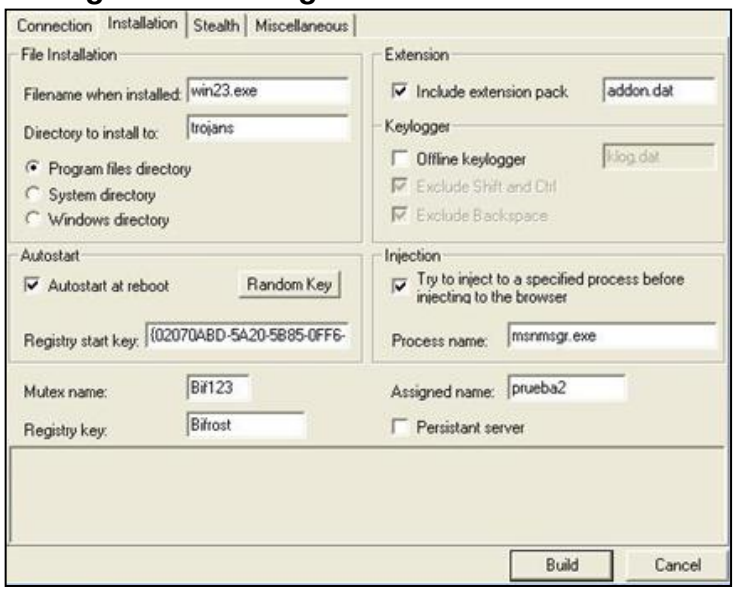
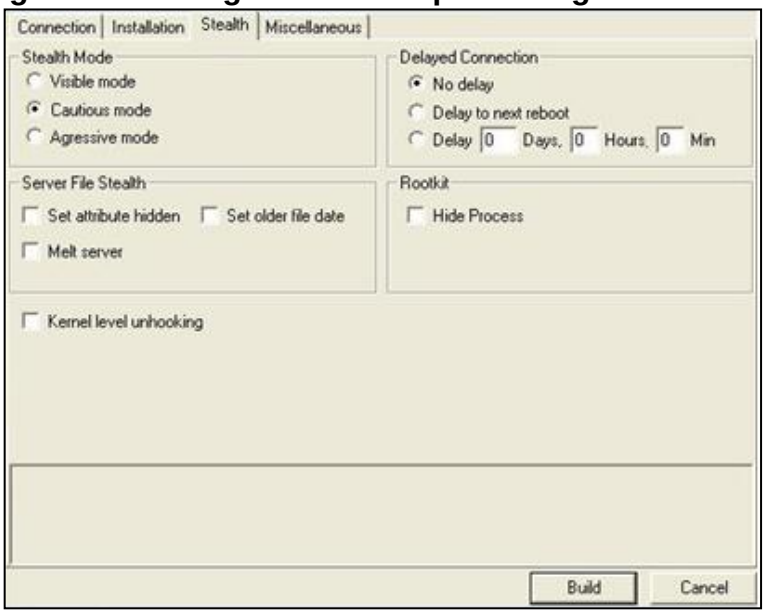


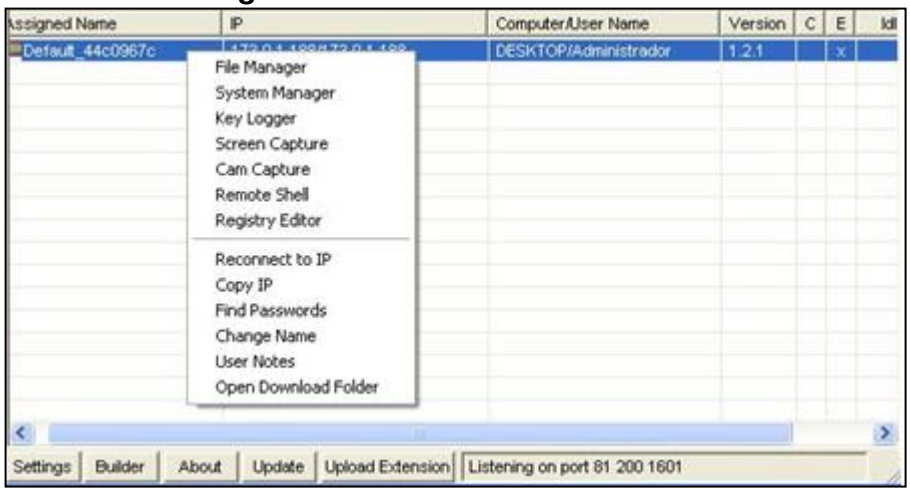
Figura 11. Configuración de tipos de sigilo de *Bifrost*



Módulo de daño. *Bifrost* brinda opciones tales como (ver figura 12): verificar los procesos que está ejecutando el SO, transferencia de archivos desde el cliente al servidor y del servidor al cliente, sustraer información, agregar programas o archivos que afecten el correcto funcionamiento del SO.

Posee un *keylogger* que puede trabajar *offline* sin necesidad de tener un servidor para que envíe la información a éste, sencillamente guarda lo escrito en el mismo computador para que cuando se realice la conexión al servidor del troyano pueda ser descargado, asimismo tiene la opción de *keylogger* online para capturar lo que el usuario está digitando en tiempo real.

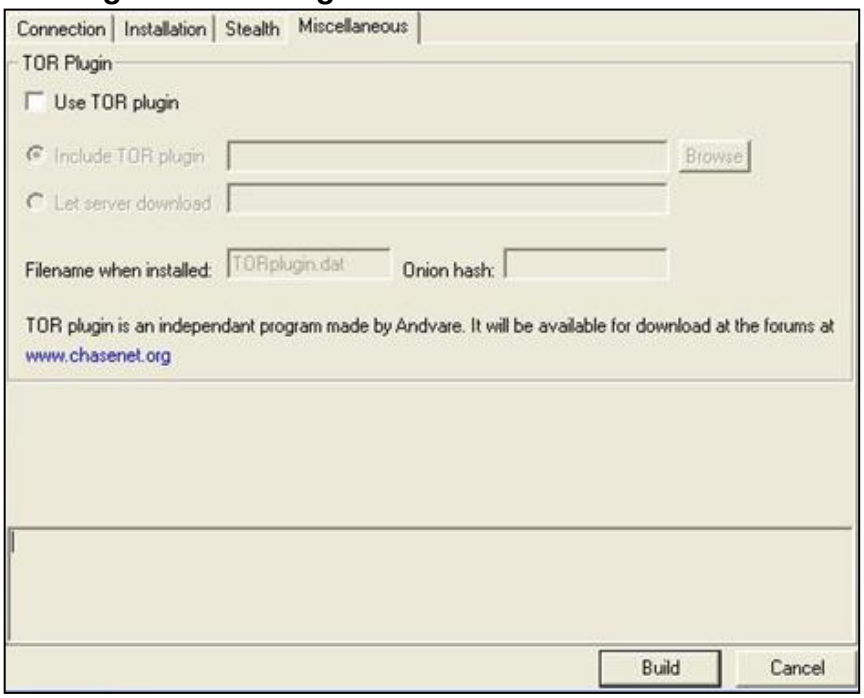
Figura 12. Interfaz cliente *Bifrost*



Este troyano incorpora capturas de pantalla en tiempo real para espiar lo que la otra persona está realizando en el computador donde está instalado el servidor, pero solo puede observar, no puede tomar el control de él; por éste motivo incorpora un *Remote Shell* para poder manipular el SO de forma imperceptible al usuario.

Módulo comunicación. Bifrost no cuenta con un módulo de comunicación como el de los otros troyanos que buscan alguna vulnerabilidad del SO para realizar conexiones estables. Por esto, él utiliza conexiones TOR (*The Onion Router*) (ver figura 13), la cual es una herramienta que permite realizar conexiones de manera anónima a través de internet, y en sistemas operativos como Microsoft Windows y Apple Mac OS [23].

Figura 13. Configuración conexión TOR *Bifrost*



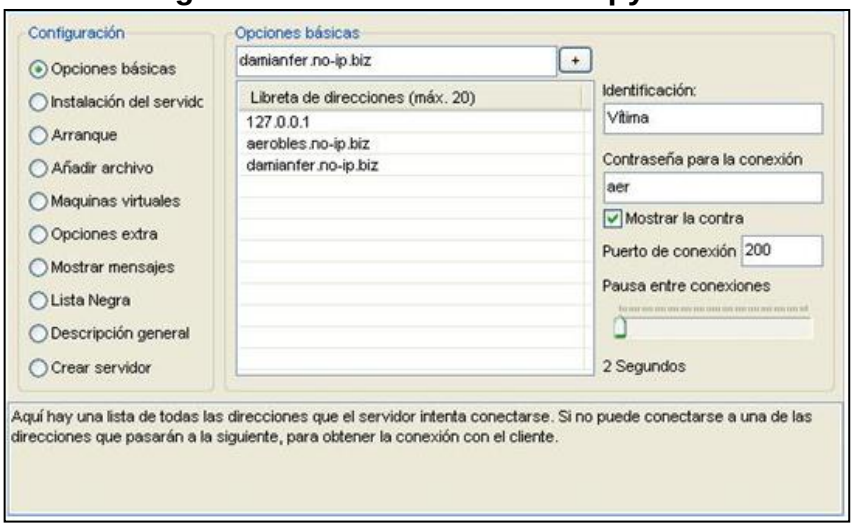
5.1.4. Spy-net. Es de los troyanos relativamente nuevos, maneja una interfaz intuitiva y amigable, es estable, adicionalmente el archivo ejecutable del servidor no sobrepasa 90 Kilobytes ayudando de esta forma a pasar inadvertido.

Módulo de seguridad. Este troyano en el momento de configurarlo exige colocar una contraseña para sus conexiones con el cliente o los clientes. Como método de seguridad permite las opciones de crear una carpeta en el SO para que el instalador del servidor pueda iniciarse las veces que sean necesarias. Al momento de instalarlo se puede configurar para que trabaje camuflado o no; además, proporciona la opción de modificar las llaves de registro del SO víctima e iniciarse como un servicio del sistema.

Al momento de crear el servidor (ver figura 14), y después de ser creado existen formas para mantener el troyano más tiempo en el computador, como cambiar de nombre el archivo, camuflarlo en otro programa, observar los puertos activos para

realizar conexiones como método de contingencia o *backup*, y eliminar aplicaciones como el antivirus.

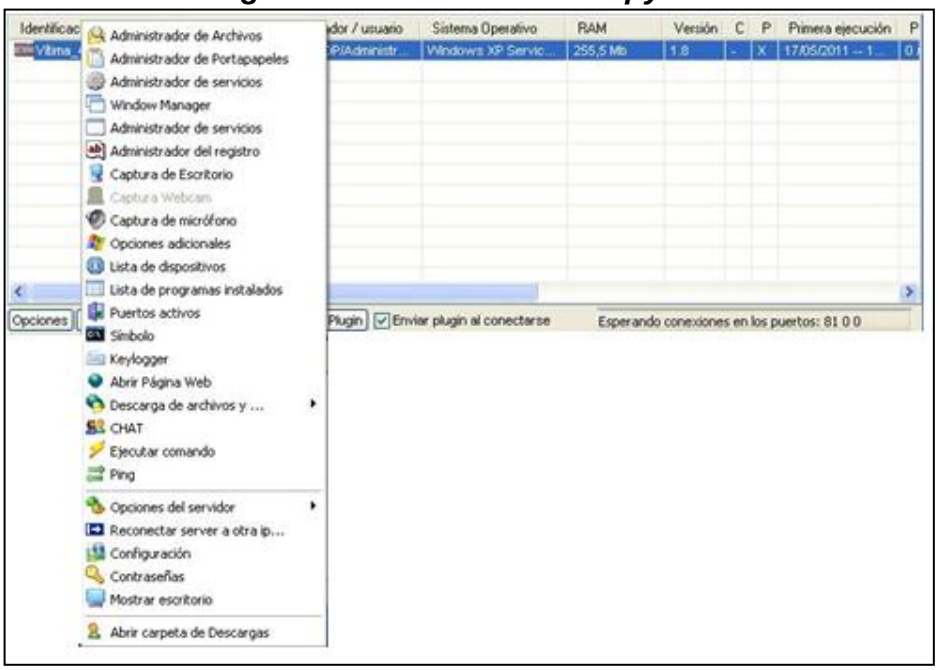
Figura 14. Creación servidor Spy-Net



Módulo de daño. *Spy-Net* se destaca por ser rápido y estable, además de ofrecer una gran cantidad de opciones.

Spy-Net ofrece al cliente (ver figura 15) las opciones: observar la lista de dispositivos que están instalados en el computador, administrar llaves de registros para agregar o quitar programas al iniciar el SO, manejar el *Shell* de Windows de forma visual u oculta a la víctima, controlar los servicios, capturar sonido y video de la víctima; permite descargar o subir archivos, también posee un *keylogger* que se puede configurar para que trabaje mientras la víctima y el ciberdelincuente estén o no conectados.

Figura 15. Interfaz cliente *Spy-Net*



Módulo de comunicación. Permite actualizar el servidor, restablecer conexiones, desconectarse, desinstalar el servidor, conectarse a otra dirección IP para así utilizar más de un cliente y observar qué puertos se encuentran abiertos.

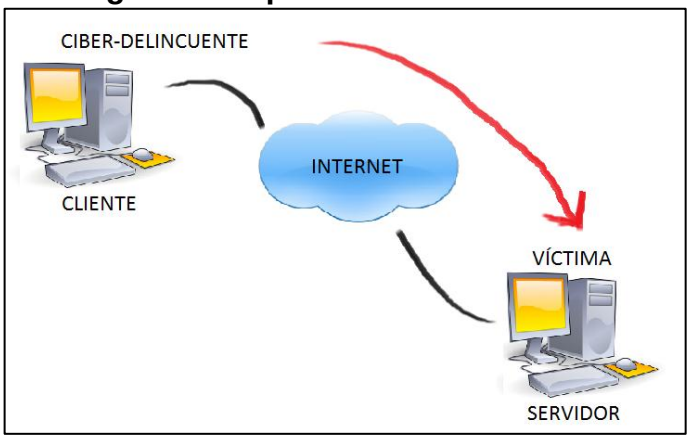
Figura 16. Opciones de comunicación entre el cliente y el servidor de Spy-Net

5.2. ARQUITECTURA DE CONEXIÓN DE LOS TROYANOS

La arquitectura que define el troyano tipo *backdoor* es cliente/servidor. Los troyanos definen dos tipos de conexión: directa o inversa.

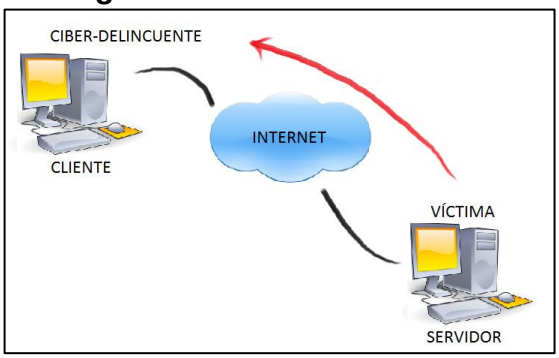
5.2.1. Conexión directa. Es una forma de conexión del cliente al servidor (ver figura 17); aunque es básica en su funcionamiento es implementada en troyanos antiguos, tiene como desventaja ser restringida por los firewall que protegen las redes haciendo más difícil la comunicación entre el cliente y el servidor.

Figura 17. Tipo de conexión directa



5.2.2. Conexión inversa. En este tipo de conexión es el servidor quien contacta al cliente (ver figura 18). Su principal ventaja es evadir el control que hace el *firewall* debido a que la mayoría de los *firewall* simplemente analizan los paquetes de información que entran a una red, y muy pocos analizan los paquetes que salen de la red [24].

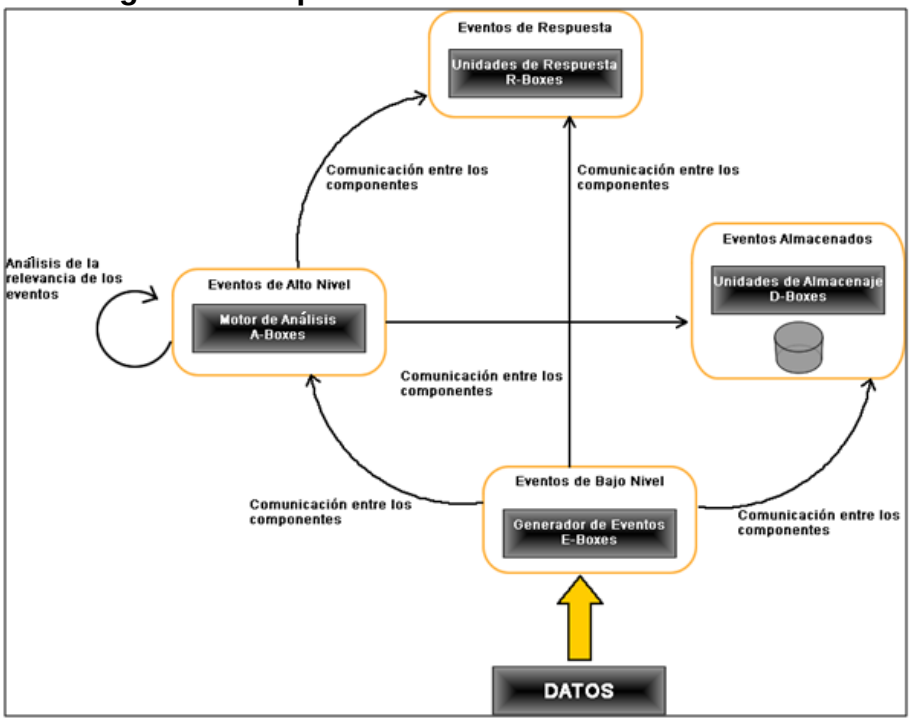
Figura 18. Conexión inversa



5.3. ARQUITECTURA DE LOS IDS

Aunque no existe una arquitectura universal de los IDS, la CIDF (*Common Intrusion Detection Framework*) estableció un modelo en el cual por medio de componentes llamados *boxes* simula las interacciones que normalmente realizaría (ver figura 19) [25]:

Figura 19. Arquitectura de los IDS. Modelo CIDF



Generador de Eventos o *E-Boxes*. Son los sensores del IDS, están encargados de recopilar la información que está pasando por la red o el *host*, generando eventos de los datos obtenidos.

Motor de Análisis o *A-Boxes*. Es uno de los componentes claves para el IDS, es el núcleo del mismo, se encarga de tomar los datos generados por el *E-boxes* y generar nuevos eventos. Los *A-Boxes* se pueden clasificar según el tipo de detección, por ejemplo, sistemas estadísticos de *profiling*[12], reconocedores de patrones, entre otros, por ende se requiere estar en constante desarrollo y actualización de los *A-Boxes* para lograr un buen funcionamiento del mismo.

Unidades de Almacenaje o *D-Boxes*. En este caso se toman los eventos generados por *A-boxes y E-boxes*, obteniendo una base de datos que contenga las inferencias del motor de análisis. En consecuencia, es un componente importante a la hora de aplicar ingeniería forense o minería de datos.

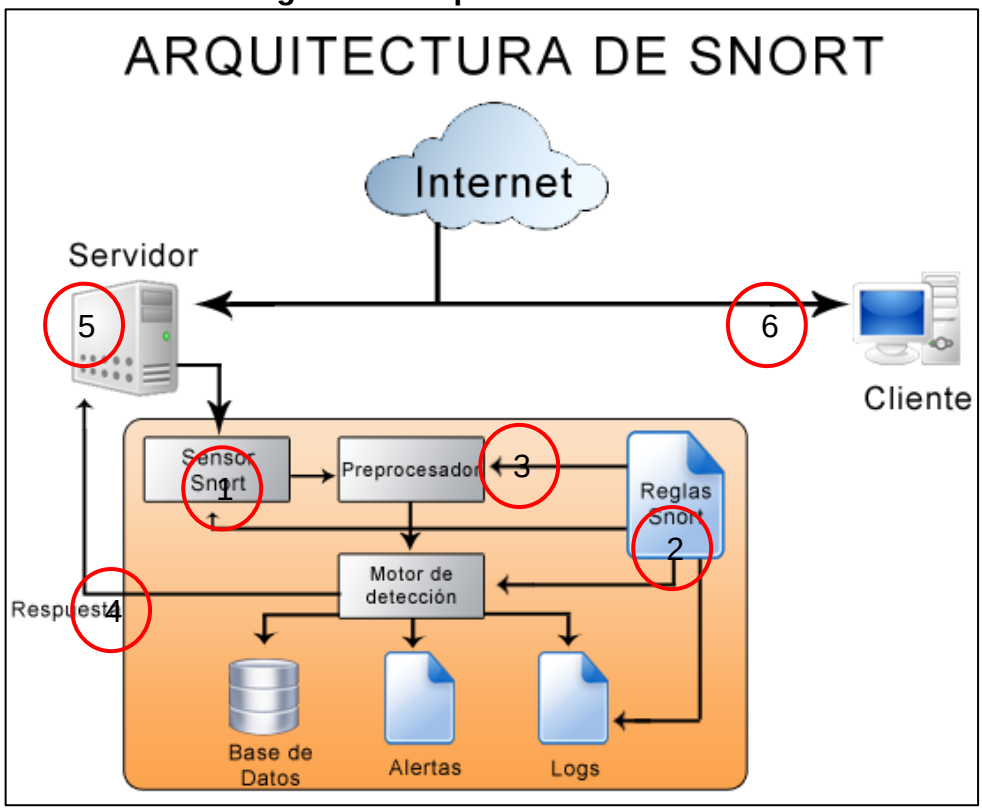
Unidades de Respuesta o *R-Boxes*. En esta unidad se analizan los datos obtenidos para generar una respuesta según el evento encontrado, ya sea por unas firmas o por acciones detectadas como maliciosas.

5.3.1. Arquitectura de Snort. Entre las herramientas IDS, Snort es mundialmente conocida, dentro de sus cualidades ofrece diversas formas de configuración siendo así personalizable. En su arquitectura (ver figura 20) posee sensor o sensores (1), los cuales trabajarán por medio de reglas (2) y de preprocesadores

[12] *Profiling*: Perfiles creados a partir de datos estadísticos de comportamientos en detecciones de *malware*. [http://dgonzalez.net/pub/ids/IDS_v1.0.pdf].

dinámicos[13] (3) para realizar análisis y lanzar las alertas o respuestas (4) pertinentes según las reglas. El servidor (5) puede estar en el mismo computador o en uno externo para que trabaje junto con el sensor, y un cliente (6) al que se le monitorizará el tráfico que entre o salga de él, al igual que monitorizará los cambios que han surgido en archivos estipulados para ser analizados.

Figura 20. Arquitectura de Snort



[13] Preprocesadores Dinámicos: *Dynamic Preprocessor* en Inglés, es un módulo de Snort que permite cargar reglas y opciones de detección antes que éste inicie la monitorización del computador o de la red LAN. Esta opción está disponible a partir de la versión Snort 2.6 en adelante.

Para su funcionamiento Snort trabaja de la siguiente forma [26]:

1) Establece variables de red. Coloca las direcciones IP con las cuales se conectará el servidor; por ejemplo: con un gestor de bases de datos, un servidor de correo, un servidor *web*, etc., y la dirección IP del computador o la red LAN que se monitorizará.

2) Configura el decodificador. Determina los protocolos que son usados para el análisis de los segmentos TCP, IP, entre otros. El decodificador no solo revisa la cabecera de los segmentos, también si tiene o no anomalías y dependiendo de la configuración de *Snort* este lanzará una respuesta.

3) Configura la base del motor de detección. Revisa los eventos por paquetes, y los clasifica para mostrarlos según la configuración de los eventos; aunque esta opción viene configurada por omisión, Snort permite que el administrador controle totalmente esta opción y así cambiar la prioridad de los eventos por paquetes.

4) Configura preprocesadores. Snort permite la creación y/o configuración de preprocesadores dinámicos para facilitar el trabajo de monitorización. Estos preprocesadores deben ser ubicados en una carpeta específica que posee Snort.

5) Configura las rutas de las librerías dinámicas. Establece las rutas en las cuales se encuentran las carpetas con las librerías de preprocesadores dinámicos o preprocesadores, que serán cargados antes de iniciarse la monitorización.

6) Configura *plugins* de salida. Permite modificar la forma como la información es presentada en Snort, las alertas, el tipo de archivo que se genere para estas, además, si se tiene buen conocimiento de programación se puede hacer un *plugin* propio para que funcione con alguna aplicación o herramienta específica.

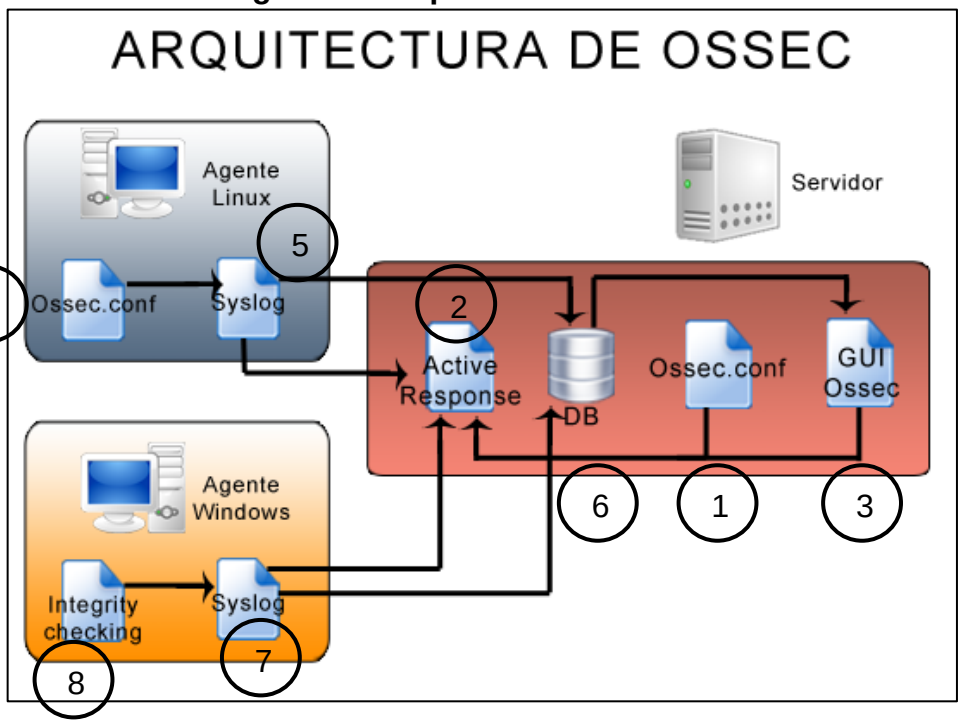
7) Personalización de reglas**.** Editar y/o crear reglas según las necesidades del administrador haciendo más sencilla la monitorización del computador, además de permitir seleccionar un grupo de reglas con las que se desean trabajar.

5.3.2. Arquitectura de OSSEC. Desarrollado por *Trend Micro* (compañía Japonesa especializada en seguridad), OSSEC funciona como HIDS con arquitectura basada en el modelo establecido por la CIDF. La arquitectura se compone de un servidor, un sensor o sensores y un cliente, el cual es conocido como agente (ver figura 21). El servidor debe operar bajo ambiente Linux mientras que el agente trabaja en plataformas Linux y/o Windows.

Como se muestra en la figura 21, el servidor está compuesto por un archivo llamado *OSSEC.conf* (1) donde se almacena la configuración que se le haya realizado. Dependiendo de esas configuraciones el *Active Response* (2) emitirá las alertas que podrán ser observadas en la interfaz gráfica de OSSEC (GUI OSSEC) (3). Para el cliente de OSSEC en Linux también se tendrá un archivo *OSSEC.conf* (4) con las configuraciones, un *syslog* (5) el cual guardará todos los sucesos del SO en la Base de Datos (6) y el *Active Response* que será el encargado de correlacionar el mensaje para definir si es una alerta o no.

OSSEC tiene un cliente para Windows que trabaja de forma similar a los de Linux, con la diferencia que este cliente realiza un análisis al syslog (7) del SO, además posee un *Integrity Cheking* (8) el cual verifica la integridad de archivos importantes para el SO como la llaves de registro.

Figura 21. Arquitectura de OSSEC



OSSEC posee un grupo de opciones de configuración y características que hacen más fácil la tarea de monitorización para el administrador, tales como [27]:

- Notificaciones al correo electrónico. Una vez ocurrido un evento categorizado como alerta será enviado un correo electrónico al administrador, con la información relevante sobre lo acontecido.
- Integridad del sistema. OSSEC escanea de manera constante un grupo de rutas y archivos definidos en la configuración inicial a modo de conocer si el sistema está funcionando en buenas condiciones, de lo contrario emitirá una alerta.
- Ataques *Rootkit*. Emitirá una alerta si encuentra que el sistema está siendo atacado por un *Rootkit*, indicando en dicha alerta la ubicación dentro del computador y mostrando los cambios o daños si los ha realizado.
- Respuestas activas. OSSEC permite almacenar direcciones IP en una tabla, evitando que estas direcciones se conecten tanto con el servidor como el

cliente, de esta manera se evitan ataques de cualquier tipo de *malware* al computador.

- Direcciones IP blancas. Al contrario de la opción respuestas activas, OSSEC permite crear una tabla con direcciones IP que no serán analizadas por él, permitiendo realizar conexiones con el servidor y/o cliente sin contratiempos.
- *Syslog* remoto. Esta opción permite al administrador el acceso al archivo *syslog* del IDS desde un equipo remoto y poder observar los eventos ocurridos.

5.3.3. Arquitectura de Sguil. Es un software de monitorización de redes que puede ser implementado como HIDS en combinación con algunos módulos y reglas de Snort. Sguil posee una interfaz gráfica que brinda acceso a eventos en tiempo real, análisis de datos y captura de paquetes, además posee la ventaja de estar programado en Tcl/tk[14] y esto le brinda la posibilidad de ser ejecutado en los diferentes sistemas operativos que lo soportan (Linux, Solaris, MacOS, BSD y Windows) [28].

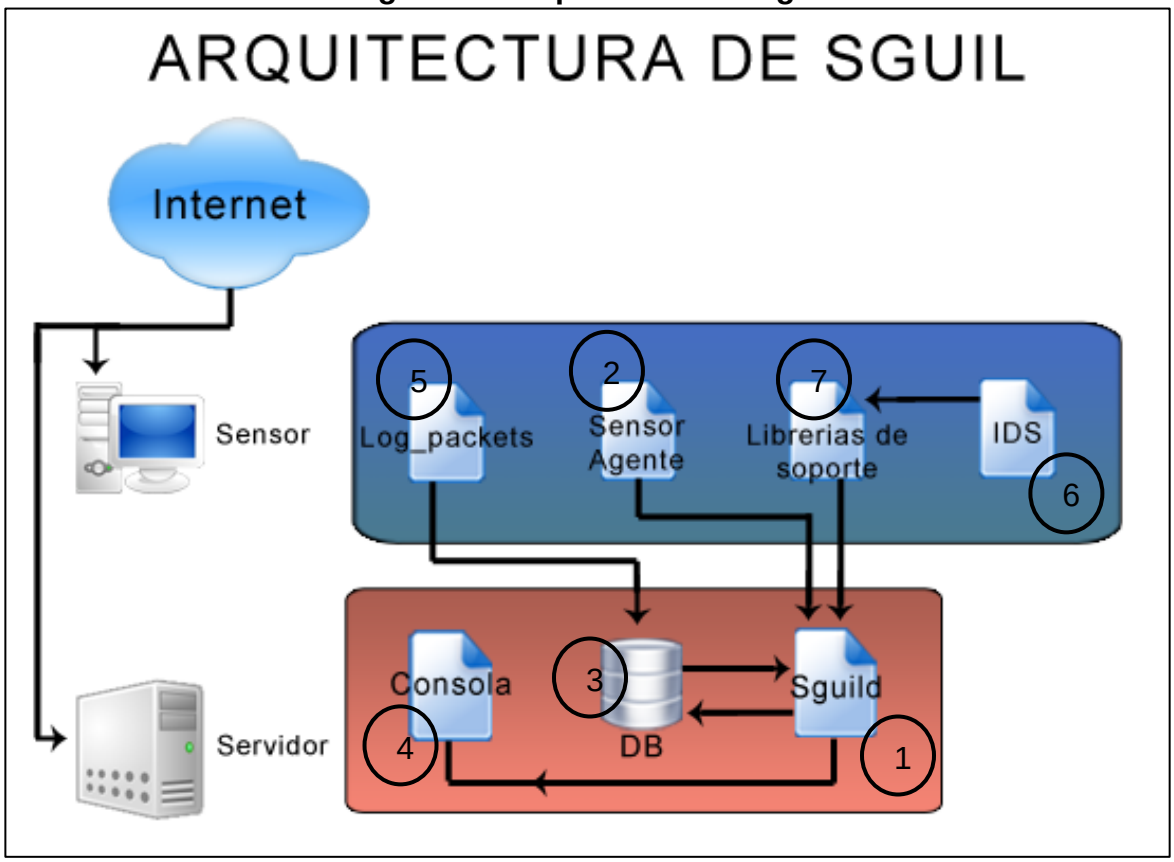
La arquitectura de Sguil está compuesta por un servidor y un sensor o sensores (ver figura 22), el servidor cuenta con un módulo llamado *Sguild* (1) el cual es el encargado de recibir toda la información enviada por los sensores (2) para correlacionarla y emitir las alertas, a su vez almacenarla en la base de datos (3), las alertas pueden ser vistas en la Consola (4) que es la interfaz gráfica para la administración de las alertas en Sguil.

---

[14] Tcl/tk: (*Tool Command Lenguage / Tool Kit*), en Español Lenguaje de Herramientas de Comando, es un lenguaje de programación potente y dinámico que permite el desarrollo de aplicaciones web y de escritorio de manera sencilla.

El sensor posee *Log_packets* (5) en el cual guarda todo lo ocurrido en el SO y lo almacena en la base de datos, cuenta también con un Sensor Agente (2) el cual toma todo lo que considera como alerta para enviarlo a *Sguild*. Sguil posee compatibilidad con otros IDS (6), para el correcto funcionamiento de ellos se requiere instalar librerías (7) adicionales.

Figura 22. Arquitectura de Sguil



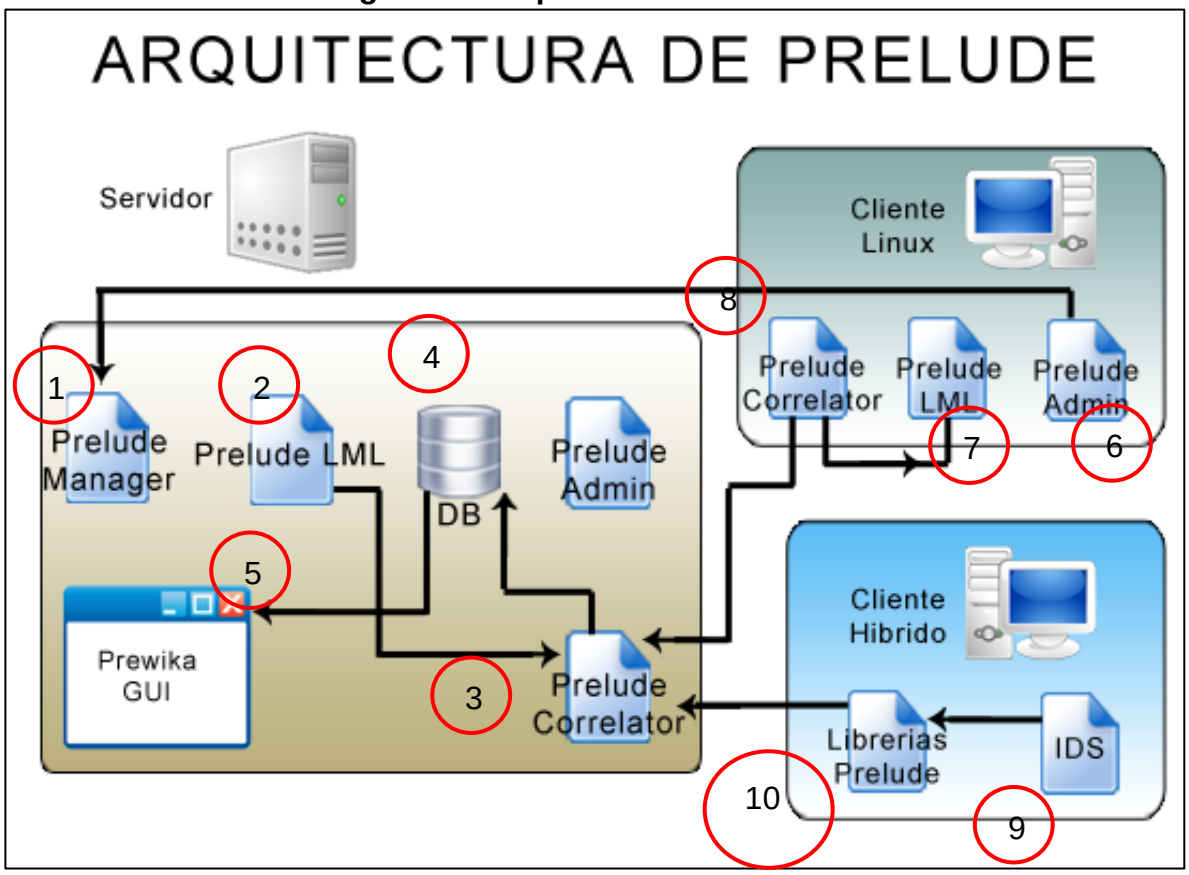
5.3.4. Arquitectura de Prelude. Aunque sea definido como un IDS hibrido, Prelude presenta un rendimiento estable como HIDS, ya que posee la cualidad de acoplar su funcionamiento a otros IDS permitiéndole trabajar en diferentes SO (Linux, Windows, BSD, Solaris).

La arquitectura que posee Prelude es cliente/servidor (ver figura 23). El servidor cuenta con *Prelude Manager* (1) para administrar la eliminación o incorporación de los clientes a la red, cuenta también con *Prelude-lml* (2) el cual es un sensor que trabaja conjuntamente con *Prelude Correlator* (3) encargado de entender todo lo que recibe y de almacenarlo en la base de datos (4) del servidor para posteriormente poder ver los cambios a través de *Prewikka* (5) que es el entorno gráfico para la administración de las alertas de Prelude.

El cliente en Linux cuenta con *Prelude Admin* (6) encargado de registrar el cliente simultáneamente con el *Prelude Manager* (1) del servidor, al igual que el servidor el cliente posee *Prelude-lml* (7) como sensor que emite las alertas y *Prelude Correlator* (8) para interpretarlas, y enviarlas al *Prelude Correlator* (3) del servidor.

Para los clientes que no hacen parte de Prelude existen dos formas de configuración: la primera es que el IDS (9) que se va a instalar tenga compatibilidad nativa con Prelude y solo se necesite realizar algunas configuraciones específicas; la segunda forma es que no tenga compatibilidad nativa con Prelude, para ello se requiere descargar las librerías (10) y soportes específicos para ese IDS.

Figura 23. Arquitectura de Prelude



5.4. INSTALACIONES Y CONFIGURACIONES

5.4.1. Instalación de Snort en Windows. Para que Snort pueda funcionar en Windows requiere la instalación de *Winpcap*, librería que permite a la tarjeta de red trabajar de manera promiscua, y así, poder realizar la captura de paquetes.

También es necesario crear una cuenta en la página oficial de Snort, y descargar la versión *user* con sus respectivas reglas. Debido a que no es obligatorio tener

una cuenta en Snort para realizar algunas descargas, si se debe tener en cuenta descargar las reglas correspondientes a la versión de Snort que se tenga, esto se debe a que puede generar conflicto en el momento de instalación de las mismas (Anexo A).

Para que Snort trabaje en entrono gráfico requiere de: un servidor web, PHP, ADODB, PHPlot y ACID.

- **ACID** (*Analysis Console for Intrusion Database*): Analiza y procesa eventos de las bases de datos con un motor basado en PHP para recolectar y procesar información generada y almacenada por el IDS, el *firewall* y las herramientas de monitorización de redes.

  La función de ACID en los IDS se basa en recolectar información de la base de datos, para decodificarla y mostrar datos relevantes de los eventos que el IDS haya detectado como firmas, tiempo de detección, rastros de los eventos como direcciones IP origen y destino, puertos utilizados y también permite manipulación de alertas para eliminar los falsos positivos [29].

- **PHPlot:** Es un proyecto realizado en PHP, lenguaje que permite realizar páginas web con conexiones a bases de datos, formularios de datos y aplicaciones de servidor, etc. PHPlot efectúa la conexión con la base de datos y toma la información que necesite de ellas para realizar gráficas utilizando la librería GD de PHP la cual permite crear líneas, rectángulos, formas elementales, entre otros. Una de las ventajas al utilizar PHPlot es que todo se configura e instala en el servidor, de esta forma no existen problemas de compatibilidad para los usuarios, solo se necesita un navegador web [30].

- **ADODB:** Es una librería de PHP para la abstracción de información en bases de datos, aunque existe una versión para Phyton, la versión para PHP tiene

soporte para la mayoría de los gestores de bases de datos. Algunas aplicaciones y proyectos que se encuentran utilizando esta librería son: *Xaraya*, *PHPWiki, Mambo, TrikiWiki*, entre otras. Uno de sus puntos a favor es la velocidad con que puede extraer información de la base de datos, su portabilidad, su manejo, la creación de esquemas y su facilidad para aprender a utilizarlo [31] .

- **PHP:** Es un lenguaje basado en *scripts* el cual puede ser insertado en páginas HTML. A pesar de tener características propias, gran parte de su sintaxis proviene de los lenguajes de programación C, Java y Perl; su gran cualidad es desarrollar páginas web de forma rápida y dinámica [32].

- **Servidor web:** Un servidor web se requiere porque en él se alojará la página web, los códigos y herramientas que se necesitan para que Snort pueda funcionar correctamente. Allí se encontrará diversa información, como conexiones con la base de datos, si se tiene, y otro tipo de conexiones, configurar e instalar elementos de compatibilidad del lenguaje de programación para la página web, por ejemplo PHP y cualquier otro archivo o herramienta que se necesite para el correcto funcionamiento del mismo. El servidor web servirá para centralizar la información recolectada y poder tener acceso a través de internet desde cualquier lugar.

Para llevar a cabo la instalación del servidor web en el proyecto se utilizó el servidor Apache por la compatibilidad que posee con las herramientas que se necesitan para la instalación del IDS.

Algunos aspectos a tener en cuenta en la configuración de Snort son: el archivo *snort.conf* el cual contiene la variable "*var HOME_NET*" donde debe colocarse la dirección IP del computador que se desea analizar; tener un gestor de Base de Datos, para éste proyecto se utilizará MySql; además, se necesitará el usuario y la

contraseña, ya que se requiere en la línea "*output database*" para que se pueda conectar a la base de datos. Por último, las líneas que contengan al inicio el carácter de numeral *"#"* serán omitidas por Snort.

Snort se puede configurar para trabajar como un servicio de Windows, para llevar a cabo esto proceso se necesita abrir un símbolo del sistema o MS-DOS como administrador, ubicarse en el directorio donde se encuentra Snort y acceder a la carpeta llamada "*bin*", desde allí digitar:

```
snort/SERVICE/INSTALL –de-c {ruta completa donde se encuentra el archivo snort.conf} –l snort\log
```

Para saber qué adaptador de red utilizará Snort, de nuevo en el símbolo del sistema y desde la carpeta "*bin*" de Snort digitar "*snort –W*", de esta forma ver qué número tiene el adaptador a monitorear. El comando para que monitorice la red con el adaptador es "*snort –v –i#"* siendo *"#"* el número del adaptador de red.

Para que almacene correctamente el escaneo de puertos se requiere modificar el esquema que tiene por omisión Snort para MySql. Se debe buscar el archivo "*create_mysql*" en el directorio *snort**schemas* y con un editor de texto buscar y modificar la línea:

```
sig_class_id INT UNSIGNED NOT NULL, y cambiarla por sig_class_id INT UNSIGNED
```

Después de realizar esto, desde el símbolo del sistema ingresar al directorio donde se instaló MySql y acceder a la carpeta "*bin*", desde allí digitar:

```
mysql -u root -p -D snort < c:\Snort\schemas\create_mysql
```

De esta forma serán creados todos los esquemas necesarios para que Snort pueda almacenar en la base de datos todos los eventos; si Snort se instaló en una ruta diferente, es necesario remplazarla en los comandos de ejemplo.

5.4.2. Instalación de Snort en Linux. Para la implementación de Snort en la distribución Debian *Squeeze* se creó la guía que se encuentra en el anexo B.

Es importante descargar todos los paquetes/librerías mencionadas en el anexo B, logrando así el correcto funcionamiento de Snort con la distribución Debian *Squeeze*. Para la descarga de los paquetes/librerías se recomienda utilizar los repositorios de Debian utilizando el comando "*apt-get install*". En el caso de no encontrar o no poder instalar el paquete/librería por medio de los repositorios, se debe buscar en la página oficial de Debian e instalarlo manualmente.

Al igual que Snort en Windows, se requiere de un gestor de base de datos para almacenar lo generado por el IDS, posterior a la instalación de la base de datos, debe ser configurado el archivo *snort.conf* con el usuario y la contraseña respectiva para poder establecer la conexión con Snort.

5.4.3. Instalación de OSSEC. Para la instalación de OSSEC se utilizaron las guías de los anexos C y D. Se recomienda tener cuenta que el servidor para este IDS debe ser instalado en Linux y el agente puede ser instalado en Linux o en Windows. En el momento de instalar el servidor se utilizó la distribución Debian *Squeeze*, la cual cuenta con las librerías necesarias para desempaquetar y poder ejecutar el comando *make* en el SO. En el caso de no contar con éste comando, basta con realizar un "*apt-get install automake*".

Las guías realizadas para la instalación de OSSEC son bastante completas, los errores que se podrían cometer son pequeños pero no dejan de ser importantes. Se recomienda en el momento de extraer la llave para el cliente de OSSEC verificar que se copió de manera correcta la llave, también revisar la dirección IP del servidor y del cliente o los clientes. Se debe tener en cuenta el levantar los servicios en Linux para el servidor y en Windows reiniciar el cliente para que tomen efecto los cambios realizados.

Al momento de poner en funcionamiento el servidor puede surgir un error "*OSSEC analysisd: Testing rules failed. Configuration error*.". Para solucionar este problema se requiere hacer un enlace simbólico para OSSEC desde una terminal con el comando:

```
ln -s /var/ossec/bin/ossec-logtest/ var/ossec/ossec logtest-
```

Así mismo, si se necesita mostrar los datos a través de una interfaz web, se necesitará de un servidor web. En la página oficial de OSSEC existe documentación donde se explica cómo se puede configurar OSSEC con el servidor web Apache y con *Lighttpd*. La instalación y las configuraciones realizadas para la interfaz web se encuentran en el anexo H.

5.4.4. Instalación de Prelude. En la instalación de Prelude se utilizó la guía del anexo E. Al instalar Prelude se encontró que efectivamente podía trabajar de forma hibrida, esto quiere decir como un HIDS y NIDS, pero Prelude no maneja un cliente/agente propio para que trabaje como HIDS, por lo que se utilizará el agente/cliente de OSSEC recordando que para que funcione el agente/cliente requiere tener instalado el servidor de OSSEC.

La guía de instalación es bastante completa y paso a paso, algunas de las recomendaciones en el momento de instalar OSSEC es que se debe instalar en modo servidor y, además se debe configurar con soporte para Prelude antes de ser instalado, solo de esta forma el *Prelude_manager* podrá reconocer a OSSEC. El servidor de OSSEC se puede instalar en el mismo computador donde se encuentra Prelude, de lo contrario, bastará tener en cuenta la dirección IP del servidor y el cliente en el momento de configuración de Prelude y OSSEC. Además, también se logró instalar Snort para que trabajara conjuntamente con Prelude debido a que, al igual que OSSEC, manejan el formato IDMEF para comunicarse.

5.4.5. Instalación de Sguil. En el caso de la instalación de Sguil los manuales que se encontraron en Internet no funcionaron con la distribución de Linux "*Debian Squeeze*" que es la utilizada en el proyecto para la instalación de las herramientas IDS. La documentación que se encontró es para la distribución *Redhat*, el cual utiliza la compilación por medio de los comandos *make* y *makefile*; aunque el comando *make* de *Debian* realiza las mismas funciones que se ejecutarían al compilar con los comandos *make* y *makefile* de la distribución *RedHat*, dicho comando *make* de *Debian* no compiló correctamente; además, algunos paquetes/librerías no están disponibles para Debian y/o algunos no son compatibles. No obstante, se realizaron búsquedas para dar solución a éste inconveniente logrando encontrar la distribución de Linux llamada *Security Onion* orientada a trabajar únicamente con IDS, por lo tanto se instaló el servidor del IDS en dicha distribución. *Security Onion* es una distribución de Linux basada en Xubuntu que ofrece la posibilidad de interactuar con varios IDS preconfigurados como Snort, Sguil, entre otros, evitando así conflictos de compatibilidad y errores de configuración [33].

Para la instalación del servidor de Sguil en *Security Onion* se utilizó parte de la información que se encontró en el sitio oficial de *security Onion*, tomando como base esa información se creó la guía de instalación que se encuentra en el anexo F.

Para la instalación del cliente en Windows se utilizó la guía que se encuentra en el anexo G. Para que funcione correctamente se requiere instalar en el computador el software *Active TCL* con la versión 8.4 debido a que versiones anteriores generan conflictos con Sguil.

6. CARACTERIZACIÓN

6.1. CARACTERIZACIÓN DE LAS REGLAS DE LOS IDS A EMPLEAR

Las reglas para un IDS son el conjunto de instrucciones que le indican qué tipo de acciones deben ser consideradas como intrusiones o accesos no autorizados al computador. Cada IDS determina cuál debe ser el protocolo a seguir según sus reglas para generar una alerta, incidiendo en la forma en que se caracterizan los IDS.

6.1.1. Identificación de reglas en Snort para *host*. La manera en que Snort envía información sobre lo ocurrido en el SO es por medio de alertas, las cuales son enviadas desde el agente al servidor. Aunque el servidor y el agente/sensor pueden estar en el mismo computador no siempre será así, y requiere de configuraciones adicionales para su funcionamiento por separado. En el momento de analizar un *host* Snort puede hacerlo por medio de la revisión del *syslog* del computador monitoreado.

***Syslog*.** Es un estándar para el registro de mensajes o eventos del sistema, estos mensajes o eventos son enviados desde el *syslog* a un servidor como Snort. Estos mensajes pueden ser trasmitidos por medio del protocolo UDP o TCP, pese a que puede funcionar con cualquiera de los dos. El protocolo TCP brinda mayor confiabilidad al pedir confirmación del mensaje recibido.

Syslog también puede recoger información del protocolo de gestión de redes SNMP (*Simple Network Management Protocol*) en una red o de algún agente que este monitorizando uno o más dispositivos.

Syslog toma los eventos ocurridos en el SO almacenándolo por lo general en la ruta */var/log/syslog*. Los eventos que puede registrar *syslog* son errores de

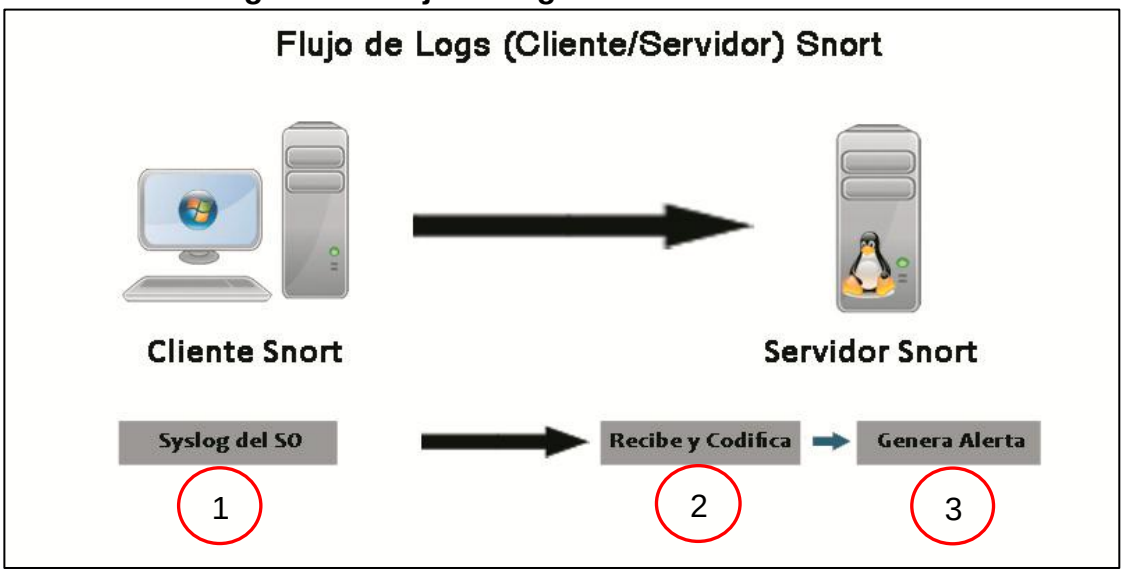
configuración del SO, de instalaciones hechas por los usuarios, de hardware, de autorizaciones, alerta sobre registros de usuarios y a su vez estos tienen código de alerta el cual se le asigna al mensaje (ver tabla 3) [34].

Tabla 3. Niveles de alerta *syslog* del Snort

Número de Alerta	Descripción
0	Se clasifica como emergencia enunciando que el SO está inutilizable.
1	Indica una alerta informando al administrador que debe tomar medidas sobre ese error o alerta.
2	Informa que algo en el SO posee o tiene algo crítico.
3	Informa sobre errores del SO o sus componentes.
4	Alerta de peligro o condiciones de peligro.
5	Noticias sobre el estado del computador monitoreado.
6	Mensajes informativos del SO.
7	Depuración de mensajes de bajo nivel.

Como se observa en la figura 24, el cliente se encuentra en el computador que va a ser analizado, y la forma de tomar los eventos que se encuentran en el SO es por medio de los registros de sucesos generados en el *syslog*, así que el cliente se encarga de tomar los eventos del *syslog* (1), después realiza el envío al servidor el cual se encargará de recibir los mensajes y codificarlos (2) de tal manera que se genere un mensaje de alerta al administrador del servidor (3).

Figura 24. Flujo de registros de sucesos en Snort



6.1.2. Identificación de reglas en OSSEC para *host* Las reglas para OSSEC están conformadas por reglas atómicas y reglas compuestas, para ambos casos el método o protocolo que se utiliza para enviar el mensaje cuando se genera una alerta es el mismo, OSSEC puede utilizar el protocolo UDP o el protocolo o TCP. Por omisión, OSSEC trabajará con el protocolo TCP debido a la ventaja que éste protocolo ofrece en la confirmación del mensaje emitido entre el servidor y el agente [35].

- **Reglas Atómicas**: Generan alertas enviadas al servidor sin ser analizadas por el preprocesador del IDS, de esta forma se obtendrá una alerta por cada evento que esté relacionado con una regla que se tenga configurada; por ejemplo, intentar acceder como *root* al SO sin serlo, en este caso se enviarán el mismo número de alertas de acuerdo al número de intentos de ingresar al SO, como consecuencia se recibirá una cantidad considerable e innecesaria de alertas del mismo tipo al administrador del IDS.

- **Reglas Compuestas**: Como su nombre lo indica generan las alertas compuestas. Las reglas compuestas están diseñadas para trabajar junto con otras reglas del IDS, brindando mayor confiabilidad en el tipo de alerta recibida. Si se retoma el ejemplo anterior al intentar ingresar al SO como *root* o intentar obtener esos privilegios sin serlo, con las alertas compuestas se enviará un mensaje de ataque y no un simple mensaje de alerta, debido a que reiteradamente se está intentando acceder al SO, vulnerando uno de los pilares de la seguridad informática como lo es la confidencialidad.

OSSEC maneja niveles de alertas tanto para las reglas atómicas como para las compuestas, estos niveles toman valores entre 0 y 15, siendo 0 el valor menos significativo y 15 el valor más significativo con el que se puede representar una alerta (ver tabla 4) .

Tabla 4. Niveles de las alertas asignados para las reglas en OSSEC

Alertas	Descripción
Nivel 0	Las alertas que se encuentren en el nivel 0 serán ignoradas debido a que no tiene relevancia en el sistema analizado.
Nivel 2	Alertas sobre el estado del sistema, de poca relevancia para el IDS.
Nivel 3	Indica los intentos de conexiones exitosas que ocurran en los sistemas monitorizados.
Nivel 4	Alerta sobre alguna mala configuración de dispositivos o de software, de poca relevancia para el IDS.

Nivel 5	Errores generados por usuarios al introducir contraseñas inválidas o denegaciones de accesos a los usuarios, de poca relevancia para el IDS.
Nivel 6	Alertas generadas por posibles intentos de un gusano o un virus que desean ingresar al sistema, también pueden ser confundidos por acciones del mismo IDS. Es una alerta de ataque de poca relevancia.
Nivel 9	Intentos de acceder como *root* al SO y/u obtener permisos como *root*. Es de poca relevancia si no es persistente; si existe persistencia en los intentos será relevante.
Nivel 10	Errores de contraseñas con usuarios. Alerta cuando los intentos son reiterados y/o continuamente erróneos.
Nivel 12	Alerta cambios o errores en el *kernel*, SO, software, etc. Es de alta relevancia debido a que puede ser consecuencia de un ataque.
Nivel 13	Intentos de desbordamiento de búfer o URL más grande de lo normal, se toma como una alerta relevante.
Nivel 14	Correlaciones de eventos que se consideran como un ataque, se considera de alta relevancia para los eventos de seguridad.
Nivel 15	Alerta sobre un ataque exitoso. Se debe tener cuidado porque puede ser un falso positivo, por esta razón se denomina de alta relevancia e importancia y requiere de la atención inmediata.

Además de los dos tipos de reglas mencionados anteriormente, OSSEC también cuenta con componentes que le facilitan el trabajo de escaneo y monitorización de un computador, estos son:

***Syslog*.** Toma los eventos ocurridos del SO como la autenticación de las cuentas de usuarios, cambios del *kernel*[15], errores de instalación de software, conexiones exitosas con el SO, entre otros.

Agent log events. El agente es el software encargado de tomar los registros de sucesos del SO que está analizando. Dependiendo de la configuración realizada puede ampliar o disminuir el alcance del agente, brindándole mayor o menor eficiencia, de esta forma, además de ser el mensajero del IDS, también se convierte en el cerebro de los eventos para el *host* [36].

***Syscheck alerts*.** Las alertas *syscheck* son aquellas que se generan al momento de realizar la conexión entre el servidor y el agente, y éste último encuentra errores o diferencias en los archivos monitorizados por el agente. Por omisión las reglas para OSSEC son almacenadas en la ruta */var/OSSEC/rules/* y se pueden reconocer por el nombre que tienen; por ejemplo: para apache se llama "*apache_rules.xml*", así mismo para otro software como MySql sería "*mysql_rules.xml*".

Adicionalmente, las reglas de OSSEC utilizan decodificadores que correlacionan los eventos ocurridos en el sistema, y agentes que envían los eventos ocurridos a un servidor el cual se encargará de informarle al administrador del IDS el evento o los eventos, de tal forma que él tome las medidas pertinentes.

[15] Kernel: o núcleo de Linux. Es el encargado de que el software y el hardware del computador trabajen juntos y correctamente, igualmente gestiona servicios y llamadas al SO.

Por otra parte, el agente realiza una suma de comprobación con MD5/SHA1 a los archivos del SO que tiene por omisión, de esta forma mira el tamaño del archivo, los cambios que se hayan generado como permisos, o cualquier alteración en la suma, tomando como referencia el valor que tenía en el momento de la configuración o instalación del agente

Para la comprobación de archivos, OSSEC por omisión comprueba los archivos de la carpeta *System32* en el SO Windows; para Linux utiliza la ruta */etc/usr/bin, /usr/sbin*, y */sbin*. OSSEC permite agregar o quitar rutas a ser analizadas [27].

A continuación se muestra parte del código de configuración de OSSEC donde se encuentran los directorios que serán analizados:

```
<syscheck>
<directories check_all="yes">/etc,/usr/bin,/usr/sbin</directories>
<directories check_all="yes">/bin,/sbin</directories>
</syscheck>
<syscheck>
<directories check_all="yes">%WINDIR%/system32</directories>
</syscheck>
```

No obstante, las reglas en OSSEC poseen un ID o identificador que va desde 010000 hasta 109999 dependiendo del tipo de regla, los ID para Windows están entre 18100–18499 llamados "*Windows system rules*", los ID que se encuentran desde la 00000 hasta 00999 son reglas reservadas y propias para OSSEC por lo tanto no se recomienda modificarlas.

En la figura 25 se muestra el funcionamiento de OSSEC cuando analiza el computador donde se encuentra el agente y lo que ocurre al enviar los mensajes al servidor y cómo éste interpreta los mensajes. Primero, el Agente OSSEC (1) toma la información que tiene sobre el computador que se está analizando y envía la información al servidor para que la decodifique (2), después la analice (3), y por

último genere una alerta (4) para que el administrador tome medidas según el tipo de alerta.

Figura 25. Flujo de registros de sucesos en OSSEC

6.1.3. Identificación de reglas en Prelude para *host.* Prelude también maneja la arquitectura cliente/servidor como la mayoría de los IDS; éste tiene funciones similares a las de Snort como el envío de mensajes al servidor por medio del formato IDMEF (ver figura 26), formato común para el envío de alertas de intrusión basada en XML; así mismo, el protocolo (TCP o UDP) utilizado para la transferencia de mensajes es totalmente independiente al formato IDMEF.

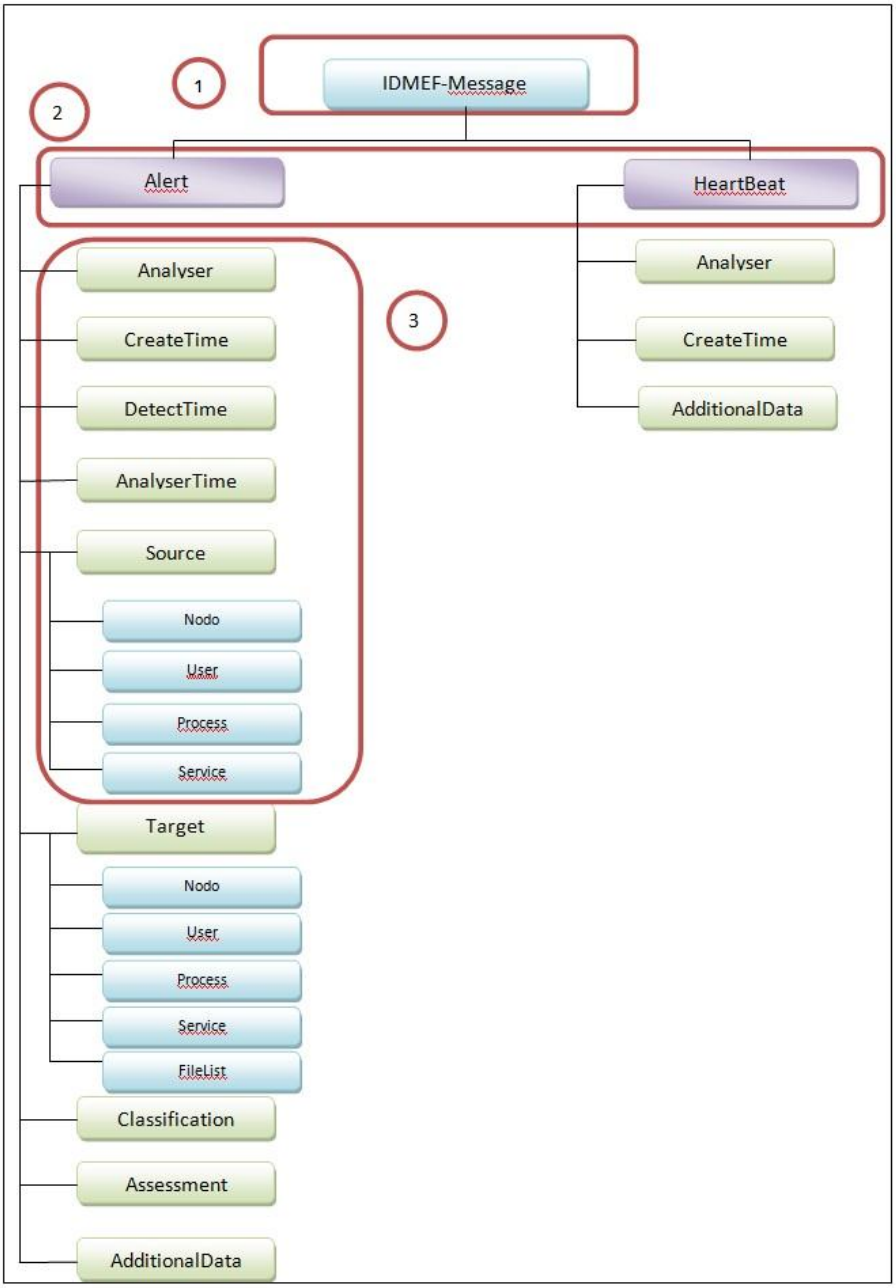
***Syslog*.** Prelude al igual que Snort maneja alertas *syslog* para analizar los cambios y/o eventos ocurridos en el SO. El *syslog* opera bajo las mismas reglas de Snort (ver tabla 5) y utiliza IDMEF para enviar los mensajes al servidor por medio de *prelude-lml* para que *Prelude Correlator* determine el nivel de amenaza.

Tabla 5. Niveles de alerta *syslog* de Prelude

Número de Alerta	Descripción
0	Se clasifica como emergencia enunciando que el SO está inutilizable.
1	Indica una alerta informando al administrador que debe tomar medidas sobre ese error o alerta.
2	Informa que algo en el SO posee o tiene algo crítico.
3	Informa sobre errores del SO o sus componentes.
4	Alerta de peligro o condiciones de peligro.
5	Noticias sobre el estado del computador monitoreado.
6	Mensajes informativos del SO.
7	Depuración de mensajes de bajo nivel.

El formato que IDMEF ofrece un tipo de mensajes que está compuesto por pequeños mensajes. Como se puede observar en la figura 26, el mensaje principal es *IDMEF-Message* (1) y este a su vez tiene dos subclases (2) que también son mensajes, y dentro de ellas otras subclases para ofrecer un mensaje con información más detallada y completa (3) [37].

Figura 26. Estructura de mensajes IDMEF en Prelude



6.1.4. Identificación de reglas de Sguil para *host* Al ser un IDS basado en Snort, Sguil comparte funciones y características con Snort, pero la más notable es el manejo de las reglas, ya que en la configuración de Sguil es necesario instalar las reglas de Snort para su funcionamiento. De ésta manera, las reglas que utiliza Sguil para emitir alertas se encuentran identificadas…en la sección 6.1.1 …

6.2. PARÁMETROS DE CARACTERIZACIÓN

Una vez identificadas las reglas y por consiguiente las funciones en común que poseen los cuatro IDS seleccionados, se hace necesario realizar una lista de parámetros que permitirán desarrollar el estudio comparativo entre ellos y, con base en los resultados obtenidos del estudio, proceder a seleccionar el más efectivo de los IDS (ver tabla 6).

Tabla 6. Parámetros de caracterización de los IDS

Parámetro	Descripción
Registros en el sistema	Reconoce cambios ocurridos en el SO.
Llaves de registro	Analiza cambios en las llaves de registro.
Conexiones realizadas hacia el agente/cliente	Conexiones que ocurrieron en el SO.
Checksum	Valor que permite verificar la integridad de los archivos del SO.
Algoritmo *hash*	Identifica el algoritmo *hash* utilizado.
Falsos positivos	Mensajes emitidos que no representan una alerta.
Ausencia de falsos negativos	Mensajes emitidos con un nivel errado de alerta o que no emitió alerta.

Carpetas por omisión	Configuración de directorios que tienen los IDS por omisión.
Detección de *Rootkit*	Posibles *Rootkit* en el agente/cliente.
Frecuencia de escaneo	Tiempo establecido por omisión para el escaneo o intercambio de información entre el cliente y el servidor.

Cada uno de los parámetros mencionados en la tabla 6, se convertirá en un punto a evaluar a cada herramienta. Con el objetivo de establecer de manera clara las acciones que se realizarán, a continuación se muestran las acciones o pruebas a realizar sobre los IDS y de esta forma evaluar su respuesta en los diferentes parámetros.

Tabla 7. Parámetros y descripción de las pruebas a realizar

Parámetro	Prueba
Registros en el sistema	Reconocer los cambios en el SO como instalación/desinstalación de software y eliminación de archivos. Alteración a los programas de inicio del SO.
Llaves de registro	Alterar llaves de registro utilizando herramientas que posee el troyano. Agregar llaves de registro al SO.
Conexiones exitosas hacia el agente/cliente	Realizar conexiones remotas al agente/cliente por medio de algún protocolo.
Checksum	Alterar o modificar un archivo cambiando su *checksum* y observar las acciones del IDS.
Algoritmo *hash*	Identificar el algoritmo utilizado por el IDS y colocar a prueba el algoritmo *hash* con herramientas que posee el troyano.

Falsos positivos	Observar alertas emitidas por actividades legítimas.
Ausencia de falsos negativos	Efectuar acciones a través del troyano al agente/cliente para descubrir eventos no detectados por el IDS.
Carpetas por omisión	Observar las carpetas que el IDS monitoriza por omisión.
Detección de *rootkit*	Emplear la función de *rootkit* del troyano para observar la respuesta del IDS.
Frecuencia de escaneo	Observar dentro de los archivos de configuración el valor del tiempo establecido para realizar escaneos al cliente.

Una vez establecidos los parámetros a evaluar a cada uno de los IDS, es necesario establecer un valor porcentual para cada parámetro, que permita establecer de forma cuantitativa qué herramienta IDS presenta mejor desempeño que las demás.

Para establecer este valor es necesario estimar qué parámetros son más relevantes y otorgar a estos mayor valor que los demás.

En la tabla 8 se muestra el peso de cada parámetro a evaluar, expresado en porcentaje, encontrado en la columna “PESO”; para cada parámetro ese será el máximo valor que pueden obtener, y ese valor se obtiene después de realizar la evaluación por medio de la rúbrica para evaluar el desempeño de las herramientas IDS (ver anexo I); al final la suma de cada peso dará como resultado los puntos que obtuvo la herramienta IDS ante el uso de un troyano.

Tabla 8. Peso de los Parámetros de Caracterización de los IDS

PARÁMETRO	Peso (%)
Registros en el sistema	25
Llaves de registro.	10
Conexiones exitosas hacia el agente/cliente	15
Checksum	15
Falsos positivos	4
Ausencia de falsos negativos	9
Detección de *rootkit*	7
Carpetas por omisión	6
Algoritmo *hash*	5
Frecuencia de escaneo	4
TOTAL	**100**

Una vez realizadas las pruebas y obtenidos los resultados y los puntajes otorgados a cada IDS, es necesario caracterizar el desempeño de cada herramienta según los puntos logrados; es por esto que se estableció en la tabla 9 la clasificación que se puede dar a cada IDS según los valores obtenidos en las

pruebas con los troyanos, conociendo de esta forma el rendimiento de cada herramienta IDS (ver anexo I).

Tabla 9. Categorías de clasificación de los IDS de acuerdo al rendimiento

Categoría	Rango de valores/puntaje	Clasificación
4	76-100	EXCELENTE
3	46-75	BUENO
2	11 - 45	REGULAR
1	0-10	MALO

7. PRUEBAS Y ANÁLISIS PARA LA ELECCIÓN DEL IDS

Para efectuar las pruebas y análisis de cada herramienta IDS se definió utilizar la herramienta antivirus Avira …ver sección 4.8 … la cual debió ser desactivada al momento de realizar estas pruebas debido a que no permitía el correcto desarrollo de las pruebas bloqueando al troyano antes de ser ejecutado. Para las pruebas en el caso específicos de los IDS serán descargadas las versiones actualizadas de las reglas tomando como referencia la fecha en que se realizarán las pruebas; en el caso que no se encuentren o no posea reglas actualizables, se crearán reglas si es posible, de lo contrario se utilizarán las reglas que el IDS tenga por omisión.

Las pruebas a cada IDS con los troyanos se realizarán de la siguiente forma:

1) Se lanza el troyano al computador que el IDS esté analizando
2) Se realizan las pruebas por medio del troyano establecidas en la tabla 7 …ver sección 6.2...
3) Se analiza la forma que el IDS mitiga las acciones del troyano, teniendo como referencia los parámetros establecidos …ver sección 6.2 …
4) Se evalúa por medio de la rúbrica (ver anexo I) el impacto causado por el troyano al SO Windows teniendo en cuenta los parámetros establecidos …ver sección 6.2 …
5) Se clasifica al IDS con los valores obtenidos al ser evaluado con la rúbrica, teniendo en cuenta los parámetros de clasificación establecidos en la tabla 9.
6) Se repiten los pasos del primero al quinto para cada IDS con los cuatro troyanos.
7) Se promedian los valores obtenidos del IDS con los cuatro troyanos para clasificarlo según los parámetros de clasificación establecidos en la tabla 9 ..ver sección 6.2 …
8) Se repiten los pasos del primero al séptimo con el siguiente IDS.

9) Se clasifica cuál IDS fue el mejor según los puntos obtenidos y la clasificación realizada en el paso séptimo.
10) Se toma al mejor IDS según el paso noveno y se repiten los pasos del primero al séptimo con los troyanos alterados.

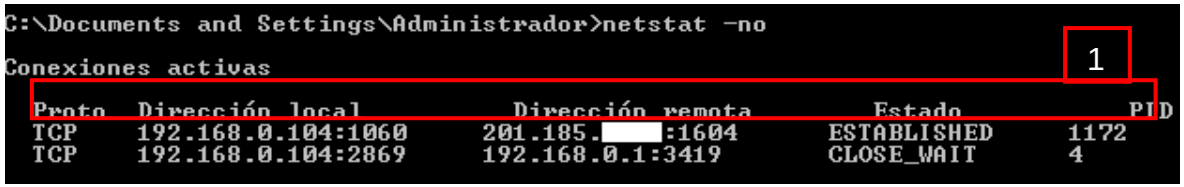
7.1. PRUEBAS Y ANÁLISIS A OSSEC

Tabla 10. Parámetros en común de OSSEC para las pruebas con los troyanos

Parámetro	Prueba	Resultados y comentarios
Algoritmo *hash*	Reconocer el algoritmo *hash* implementado por el IDS.	OSSEC implementa conjuntamente para la monitorización de sus archivos y carpetas, dos algoritmos *hash*, MD5 y SHA1, generando una alerta cuando la suma que realizan los algoritmos *hash* varia, la alerta contiene el valor inicial del *hash* y el nuevo valor, indicando al administrador que hubo un cambio.
Carpetas escaneadas por omisión	Establecer las carpetas por omisión escaneadas por el IDS.	Dentro de la configuración básica del agente OSSEC para el SO Windows las carpetas a monitorear son. *C:/WINDOWS* *C:/WINDOWS/System32* *C:/Document and Settings/all user/start menu/programs/startup*
Frecuencia de escaneo	Buscar dentro del archivo de configuración el tiempo establecido por omisión para el escaneo del agente.	Dentro de la configuración de OSSEC la frecuencia establecida por omisión para el escaneo del computador o *host* es de 72000 segundos, que es equivalente a 20 horas. Sin embargo este escaneo es aplicado en caso de no reportarse acciones por parte del usuario.

Tabla 11. Pruebas de parámetros aplicadas a OSSEC con el troyano *Darkcomet*

Parámetro	Prueba	Resultados y comentarios
Registros en el sistema	Reconocer instalaciones o desinstalaciones de programas como *utorrent*, *Winrar*, *Notepad* ++ , y/o archivos como: *System.ini, Win.ini*	El agente OSSEC de Windows reconoce la eliminación de un archivo efectuado por medio del troyano, emitiendo una alerta *Syscheck* con ID 551 de nivel 7, comprobando la integridad de éste. El agente OSSEC de Windows reconoce la instalación y desinstalación de programas realizados por el usuario, emitiendo alertas como *WinEvtLog* y *Rootcheck* con ID variable y nivel de alerta relacionado con el ID al servidor de OSSEC en Linux.
	2011 Dec 19 12:34:51 Rule Id: 18146 level: 5 Location: (ossecA2) 192.168.0.14->WinEvtLog Src IP: Administrador Application Uninstalled.	
Llaves de registro	Alterar/modificar y/o crear llaves de registro para el SO por medio del troyano.	Se logró alterar por medio del troyano algunas llaves de registro como *HKEY_LOCAL_MACHINE/SOFTWARE/Classes/Batfile* monitoreadas por el agente OSSEC de Windows y no se obtuvo ninguna respuesta por parte del IDS. Se consiguió agregar llaves de registro a rutas monitoreadas como: *HKEY_LOCAL_MACHINE/SOFTWARE/Classes/Batfile/aer.reg* por el agente OSSEC de Windows sin que éste emitiera algún tipo de repuesta.

<table>
<tr><th>Parámetro</th><th>Prueba</th><th>Resultados y comentarios</th></tr>
<tr><td rowspan="2">Conexiones exitosas hacia el agente</td><td>Realizar conexiones remotas desde el troyano al agente/cliente</td><td>Se consiguió realizar de manera exitosa una conexión (ver figura 27) usando el protocolo TCP (1) por medio del troyano al computador víctima, sin que el agente OSSEC de Windows detectara alguna anomalía o emitiera alerta.</td></tr>
<tr><td colspan="2">Figura 27. Conexión exitosa de Darkcomet no detectada por OSSEC
 
</td></tr>
<tr><td>Checksum</td><td>Alterar y/o modificar un archivo monitoreado por el IDS y/o modificar un archivo que sea importante para el SO como win.ini, autoexec.bat y observar las acciones del IDS.</td><td>Se consiguió alterar a través del troyano un archivo monitoreado por el agente OSSEC de Windows (C:/WINDOWS/win.ini), obteniendo una alerta de tipo Syscheck con ID 550 y de nivel 7, donde se indica el cambio de Checksum para el archivo mencionado anteriormente.</td></tr>
</table>

Parámetro	Prueba	Resultados y comentarios
Falsos positivos	Realizar actividades comunes de usuario tales como programas de mensajería instantánea como *skype*, mirar correo electrónico como *gmail* y utilizar páginas de entretenimiento como *youtube* o *facebook* y observar si genera alertas.	OSSEC reconoce cualquier programa instalado como alerta. En la mayoría de las veces las instalaciones son realizadas por el usuario o administrador y son software deseado, pero no reconoce como peligroso acciones comunes como acceder a páginas web tales como *google*, *Youtube*, *gmail*, *Hotmail*, etc. Además lanza una alerta al momento de iniciar una videollamada a través de *Facebook*, ésta alerta es de nivel 10.
	2011 Dec 21 09:30:21 Rule Id: 18154 level: 10. Location: (ossecA2) 192.168.0.15->WinEvtLog. Src IP: Administrador. Multiple Windows error events. WinEvtLog: Application: ERROR(4099): WmiAdapter: Administrador: A2: A2: (no message). WinEvtLog: Application: ERROR(4099): WmiAdapter: Administrador: A2: A2: (no message)	
Ausencia de falsos negativos	Realizar actividades por medio del troyano que puedan comprometer el funcionamiento del SO y observar si el IDS genera alguna alerta.	Al escanear puertos abiertos del computador, procesos activos y abrir consolas remotas, el IDS no detectó nada anómalo y tampoco generó alertas.

Parámetro	Prueba	Resultados y comentarios
Detección de *rootkit*	Camuflar el troyano en un proceso legitimo del SO o en llaves de registro.	Al implementar la opción de *rootkit* de Darckomet sobre OSSEC, éste pasa totalmente desapercibido para el agente de manera tal que no se recibe ningún tipo de alerta a esta opción.

Tabla 12. Resultados del funcionamiento de OSSEC ante la activación de Spy-net

Parámetro	Prueba	Resultados y comentarios
Registro en el sistema	Reconocer instalaciones o desinstalaciones de programas como *utorrent*, *Winrar*, *Notepad* ++, y/o archivos como: *System.ini, Win.ini*	EL IDS reconoce la instalación y desinstalación de programas. Al modificar un archivo monitoreado por el IDS es detectado y emite una alerta.

<table>
<tr><th>Parámetro</th><th>Prueba</th><th>Resultados y comentarios</th></tr>
<tr><td>Llaves de registro</td><td>Alterar/modificar y/o crear llaves de registro para el SO por medio del troyano.</td><td>Se logró crear una llave de registro en las rutas de escaneo que tiene el agente OSSEC (HKEY_LOCAL_MACHINE/ SOFTWARE/Classes/cmdfile/ EditFlags), pero no se detectó la creación y modificación de la llave por parte del IDS.</td></tr>
<tr><td rowspan="2">Conexiones exitosas al agente</td><td>Realizar conexiones remotas desde el troyano al agente/cliente</td><td>Se logró crear una conexión por medio del troyano con el computador monitoreado, sin que el agente OSSEC generara algún tipo de alerta. (ver figura 28)</td></tr>
<tr><td colspan="2">Figura 28. Conexión exitosa de Spy-net no detectada por OSSEC
 
</td></tr>
<tr><td>Checksum</td><td>Alterar y/o modificar un archivo monitoreado por el IDS y/o modificar un archivo que sea importante para el SO como win.ini, autoexec.bat y observar las acciones del IDS.</td><td>A través de Spy-net se accedió a una carpeta que contiene un archivo de extensión “.txt”, el cual se encontraba monitorizado por el IDS, al momento de realizar cualquier modificación al archivo, el IDS detecta el cambio por medio de Checksum.</td></tr>
</table>

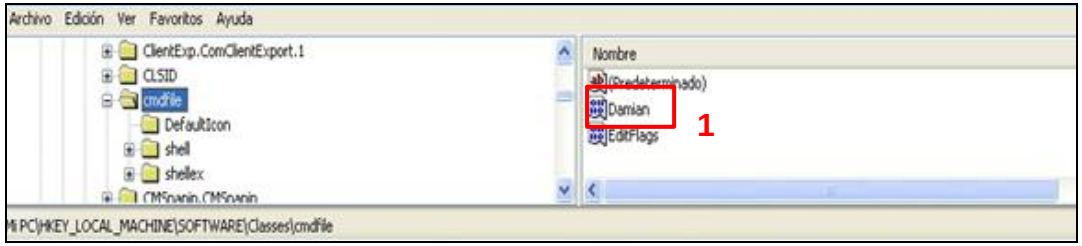
Parámetro	Prueba	Resultados y comentarios
	011 Dec 21 08:27:52 Rule Id: 503 level: 3. Location: (WinOSSEC) 192.168.0.13->ossec Src IP: gent started: 'WinOSSEC->192.168.0.102'. Ossec agent started. ** Alert 1324474135.5152: mail - ossec,syscheck, 2011 Dec 21 08:28:55 (WinOSSEC) 192.168.0.102->syscheck Rule: 550 (level 7) -> 'Integrity checksum changed.' Integrity checksum changed for: 'C:\Archivos de programa/CreadaDesdeSpynet/paraSpyNet.txt' Size changed from '18' to '25' Old md5sum was: '66248029b109e1ecfdf17df6b52d4172'. New md5sum is : 'd088692f4b27304522078d42e9e83662' Old sha1sum was: '3628b4957ce93a8bba7353c791b6022e4db1b524'. New sha1sum is : 'a03a5d2f414afb610bc264af40918bce3d176e2e'	
Falsos positivos	Realizar actividades comunes de usuario tales como programas de mensajería instantánea como *skype*, mirar correo electrónico como *gmail* y utilizar páginas de entretenimiento como *youtube* o *facebook* y observar si genera alertas.	Al navegar por internet el IDS no genera ninguna alerta; sin embargo, al utilizar un programa que requiera algún tipo de conexión como *skype,* el agente OSSEC genera una alerta.

Parámetro	Prueba	Resultados y comentarios
Ausencia de falsos Negativos	Realizar actividades por medio del troyano que puedan comprometer el funcionamiento del SO y observar si el IDS genera alguna alerta.	Por medio del troyano se efectuaron acciones como descarga de programas, copia de archivos, y ejecución de consolas remotas, pero el IDS no generó ningún tipo de alerta.
Detección de *rootkit*s	Camuflar el troyano en un proceso legítimo del SO o en llaves de registro.	Se logró ocultar el troyano dentro de los servicios del SO, sin que el agente OSSEC lograra reconocer esta acción.

Tabla 13. Pruebas de parámetros aplicadas a OSSEC con el troyano *Poison Ivy*

Parámetro	Prueba	Resultados y comentarios
Registros en el sistema	Reconocer instalaciones o desinstalaciones de programas como *utorrent*, *Winrar*, *Notepad* ++, y/o archivos como: *System.ini, Win.ini*	El agente OSSEC en Windows responde positivamente a la instalación de un programa dentro de una carpeta monitoreada, enviando una alerta *Syscheck*. Sin embargo, el agente OSSEC de Windows no presenta respuesta alguna a la eliminación del archivo monitoreado (*C:/WINDOWS/system.ini*) por medio del troyano implementado.

<table>
<tr><th>Parámetro</th><th>Prueba</th><th>Resultados y comentarios</th></tr>
<tr><td rowspan="2">Llaves de Registro</td><td>Alterar/modificar y/o crear llaves de registro para el SO por medio del troyano.</td><td>Se logró alterar por medio del troyano una llave de registro monitoreada por el sistema:(HKEY_LOCAL_MACHINE/SOFTWARE/Classes/cmdfile/Edit Flags) y como resultado el IDS no emitió alerta alguna.

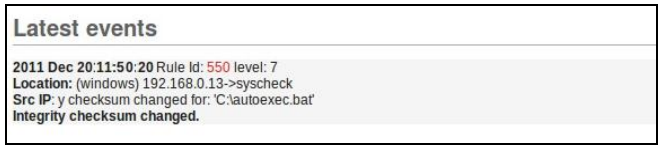
Por otra parte, al crear una llave de registro (ver figura 29) de nombre “Damian” (1) en una ruta monitoreada, el agente OSSEC no presentó respuesta a tal hecho.</td></tr>
<tr><td colspan="2">Figura 29. Creación llave de registro por medio de Poison Ivy no detectada por OSSEC
 
</td></tr>
<tr><td>Conexiones exitosas hacia el agente</td><td>Realizar conexiones remotas desde el troyano al agente/cliente.</td><td>Por medio del troyano se consiguió establecer una conexión entre el servidor y el cliente sin que el agente OSSEC lanzara algún tipo de respuesta.</td></tr>
</table>

<table>
<tr><th>Parámetro</th><th>Prueba</th><th>Resultados y comentarios</th></tr>
<tr><td rowspan="2">Checksum</td><td>Alterar y/o modificar un archivo monitoreado por el IDS y/o modificar un archivo que sea importante para el SO como win.ini, autoexec.bat y observar las acciones del IDS.</td><td>Después de efectuar el cambio dentro de un archivo monitoreado por el agente OSSEC (C:/autoexe.bat), éste genera una alerta (ver figura 30).</td></tr>
<tr><td colspan="2">Figura 30. Alerta generada por cambio en el checksum por OSSEC con Poison Ivy.
 
</td></tr>
<tr><td>Falsos Positivos</td><td>Realizar actividades comunes de usuario tales como programas de mensajería instantánea como skype, mirar correo electrónico como gmail y utilizar páginas de entretenimiento como youtube o facebook y observar si genera alertas.</td><td>Se utilizó el chat por web que tiene Hotmail y el IDS no generó ninguna alerta; sin embargo, al inicializar la aplicación de video llamada Skype, el agente OSSEC genera una alerta.</td></tr>
</table>

Parámetro	Prueba	Resultados y comentarios
	2011 Dec 22 11:53:12 Rule Id: 503 level: 3 Location: (windows) 192.168.0.13->ossec Src IP: gent started: 'windows->192.168.0.13'. Ossec agent started. ** Alert 1324572886.6432: - ossec,rootcheck, 2011 Dec 22 11:54:46 (windows) 192.168.0.13->rootcheck Rule: 514 (level 2) -> 'Windows application monitor event.' Application Found: Chat/IM/VoIP - *Skype*. Process: C:\Archivos de programa\Skype\Phone\Skype.exe.	
Ausencia de falsos Negativos	Realizar actividades por medio del troyano que puedan comprometer el funcionamiento del SO y observar si el IDS genera alguna alerta.	Al realizar acciones a través del troyano como descargar un archivo monitoreado por el agente OSSEC, no se genera ningún tipo de alerta a ésta acción. Tampoco al abrir consolas de forma remota.
Detección de *rootkit*	Camuflar el troyano en un proceso legítimo del SO o en llaves de registro.	Se consiguió esconder el troyano dentro de los servicios del sistema de forma que cada vez que se inicie sesión, el troyano realizará una conexión, a ésta acción el agente OSSEC no presentó ningún tipo de respuesta.

Tabla 14. Pruebas de parámetros aplicadas a OSSEC con el troyano *Bifrost*

Parámetro	Prueba	Resultados y comentarios
Registros en el sistema	Reconocer instalaciones o desinstalaciones de programas como *utorrent*, *Winrar*, *Notepad* ++, y/o archivos como:*System.ini, Win.ini.*	OSSEC responde de manera esperada al realizar alguna instalación/desinstalación de algún programa enviando una alerta al administrador informando lo ocurrido.
Llaves de registro	Alterar/modificar y/o crear llaves de registro para el SO por medio del troyano.	Luego de alterar una llave de registro del SO el agente OSSEC no reportó ni generó ninguna alerta ante ese cambio.
Conexiones exitosas hacia el agente	Realizar conexiones remotas desde el troyano al agente/cliente.	Una vez se logró la conexión entre el cliente y servidor del troyano *Bifrost*, ésta es registrada por el SO por medio del listado de conexiones activas; sin embargo, el agente OSSEC no registra ningún tipo de alerta a ésta conexión.
Checksum	Alterar y/o modificar un archivo que sea monitoreado por el IDS y/o modificar un archivo que sea importante para el SO como *win.ini*, *autoexec.bat* y observar las acciones del IDS.	Por medio de la opción *File manager* que ofrece *Bifrost*, se accedió al computador víctima para alterar el archivo *autoexec.bat* y de esta forma cambiar su *Checksum*. Para esta acción el agente OSSEC generó una alerta (ver figura 32) indicando que ocurrió un cambio en el archivo.

Parámetro	Prueba	Resultados y comentarios
	Figura 31. Alerta generada por cambio en el *checksum* por OSSEC con *Bifrost*.  	
Falsos positivos	Realizar actividades comunes de usuario tales como programas de mensajería instantánea como *skype*, mirar correo electrónico como *gmail* y utilizar páginas de entretenimiento como *youtube* o *facebook* y observar si genera alertas.	Al efectuar labores como navegar por internet, el IDS no registra ningún tipo de alerta, pero al intentar instalar algún tipo de aplicación el agente reporta esta acción y genera una alerta al administrador.
Ausencia de falsos negativos	Realizar actividades por medio del troyano que puedan comprometer el funcionamiento del SO y observar si el IDS genera alguna alerta.	Una vez se logró acceder al computador víctima, se realizaron acciones como copiar y mover archivos del sistema a otras carpetas monitoreadas por el agente OSSEC, sin recibir ningún tipo de respuesta por parte del IDS a estas acciones.

Parámetro	Prueba	Resultados y comentarios
Detección de *rootkit*	Camuflar el troyano en un proceso legítimo del SO o en llaves de registro.	Con *Bifrost* se logró hacer que el troyano se camuflara como un archivo de inicio del SO, sin obtener ningún tipo de alerta o acción por parte del IDS.

Las pruebas realizadas con los troyanos arrojaron resultados interesantes como se puede evidenciar en la tabla 15, el grupo de troyanos con los que se realizaron las pruebas fueron detectados por OSSEC, y los parámetros que se detectaron en los 4 troyanos fueron los mismos.

Por otra parte, las pruebas con OSSEC demostraron lo eficaz que puede llegar a ser esta herramienta para la seguridad informática en el área de HIDS, es una herramienta que no lentifica[16] al SO y sí brinda una poderosa ayuda para detectar posibles intrusos dentro del computador, aunque el cliente de OSSEC trabaja en Windows, él no funcionaría si no posee una conexión con el servidor de OSSEC el cual solo puede ser instalado en Linux, y por ende, se requiere de conocimientos en Linux para la correcta implementación del servidor.

En conclusión, Las pruebas demostraron la eficacia de OSSEC al detectar los cambios en el SO como nuevas instalaciones de software, alteración y/o modificación de archivos necesarios para el correcto funcionamiento del SO, pese a que no detectó las conexiones realizadas del troyano, ni acciones como abrir una consola remota, si reconoce cambios de configuración del SO que se pudiesen realizar desde el troyano por medio del *Checksum* que implementa para monitorear los archivos.

Luego de realizadas las pruebas y según la tabla 9 de rendimiento ante el uso de troyanos, OSSEC se clasifica como BUENO en la categoría 3 con un puntaje promedio de 52,02.

---

[16] Lentifica: Proviene del verbo lentificar, y según la RAE "*Imprimir lentitud a alguna operación o proceso, disminuir su velocidad.*" [http://buscon.rae.es/draeI/SrvltConsulta?LEMA=lentificar]

Tabla 15. Puntaje total obtenido por OSSEC

Parámetro	Puntos obtenidos por Troyano			
	Darkcomet	**Spy-Net**	**Poison Ivy**	**Bifrost**
Registros en el sistema	21,25	21,25	21,25	21,25
Llaves de registro	0	0	0	0
Conexiones exitosas hacia el agente/cliente	0	0	0	0
Checksum	15	15	15	15
Falsos positivos	1,8	1,8	1,8	1,8
Ausencia de falsos negativos	0	0	0	0
Detección de *rootkit*	0	0	0	0
Carpetas por omisión	6	6	6	6
Algoritmo *hash*	5	5	5	5
Frecuencia de escaneo	3	3	3	3
Valores totales obtenidos por OSSEC	**52,05**	**52,05**	**52,05**	**52,05**

Los parámetros alcanzados por OSSEC en las pruebas fueron:

- **Registros del sistema**: Reconoció los diferentes tipos de instalaciones/desinstalaciones efectuadas al computador, enviando una alerta al servidor de OSSEC, pero no reconoció cambios en el inicio del SO, por ende el puntaje que obtuvo fue de 21,25 de 25 puntos posibles.

- ***Checksum:*** Reconoció cambios en archivos y/o carpetas por medio del *checksum*, para posteriormente enviar la alerta al servidor de OSSEC por ende el puntaje que obtuvo fue el máximo: 15 puntos.
- **Carpetas por omisión:** OSSEC viene pre-configurado para escanear archivos y/o carpetas importantes para el SO por ende el puntaje que obtuvo fue el máximo: 6 puntos.
- **Algoritmo *hash*:** OSSEC posee dos algoritmos *hash* que son SHA1 y MD5 para monitorizar los archivos y/o carpetas por ende el puntaje que obtuvo fue el máximo: 5 puntos.
- **Frecuencia de escaneo:** El escaneo que realiza OSSEC por omisión después de ser instalado es cada 72.000 segundos, aproximadamente 20 horas, y puede ser modificado por el administrador, y el escaneo de instalaciones en el SO las realiza constantemente por ende el puntaje que obtuvo fue de 3 de 4 puntos posibles.

Los parámetros no alcanzados por OSSEC en las pruebas fueron:

- **Llaves de registro**: OSSEC no logró reconocer cambios efectuados a las llaves de registro del SO, por ende el puntaje que obtuvo fue de 0 de 10 puntos posibles.

- **Conexiones exitosas hacia el agente/cliente**: No reconoció en ningún momento la conexión del troyano al SO, por ende el puntaje que obtuvo fue de 0 de 15 puntos posibles.
- **Ausencia de falsos negativos:** No logró detectar escaneo de puertos, ni consolas remotas, ni la presencia del troyano en el SO, por ende el puntaje que obtuvo fue de 0 de 9 puntos posibles.
- **Detección de *rootkit***: No detectó la forma de ocultarse el troyano en el SO, por ende el puntaje que obtuvo fue de 0 de 7 puntos posibles.

- **Falsos positivos:** detectó el uso de programas de mensajería instantánea como alertas, por ende el puntaje que obtuvo fue de 1,8 de 4 puntos posibles.

7.2. PRUEBAS Y ANÁLISIS A SNORT

Tabla 16. Parámetros en común de Snort para las pruebas con los troyanos

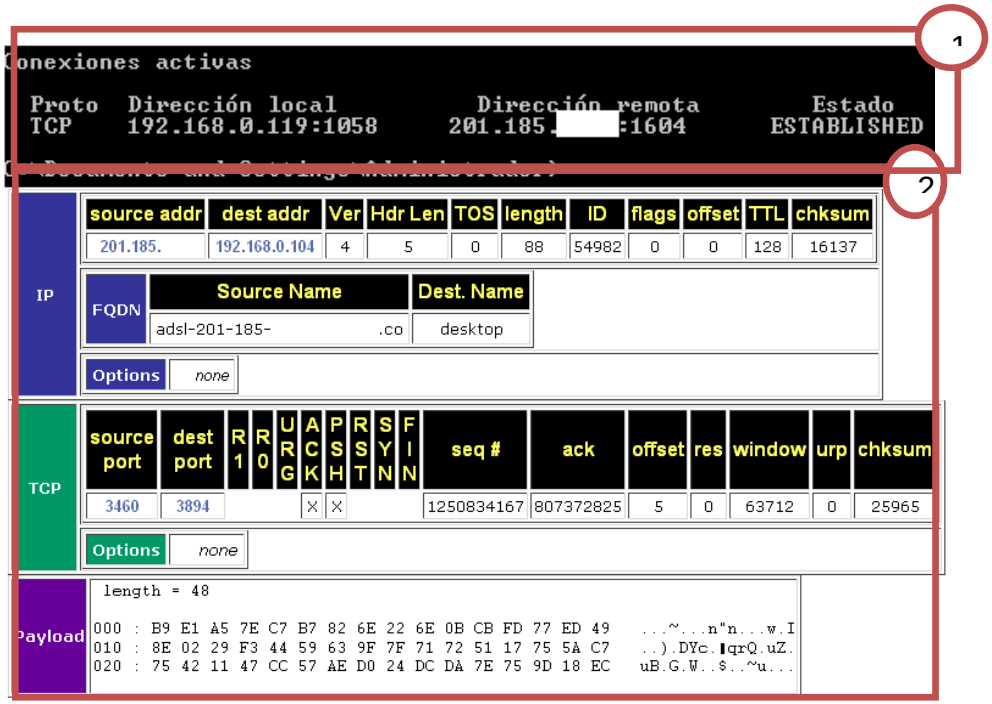
Parámetro	Prueba	Resultados y comentarios
Algoritmo *hash*	Reconocer el algoritmo *hash* implementado por el IDS.	Snort implementa funciones *hash* para la monitorización de archivos y/o carpetas del SO cuando se ha cambiado configuraciones del SO.
Carpetas escaneadas por omisión	Establecer las carpetas por omisión escaneadas por el IDS.	Snort, configurado para trabajar con HIDS, no realiza la monitorización de ningún tipo de carpeta, lo que hace es escanear constantemente el tráfico que entra y sale del computador a ser monitoreado.
Frecuencia de escaneo	Buscar dentro del archivo de configuración el tiempo establecido por omisión para el escaneo del agente.	La frecuencia de escaneo de Snort no depende de ninguna cantidad de tiempo, ya que el servidor y el cliente están en constante comunicación permitiendo que la frecuencia de escaneo sea constante mientras ambos estén funcionando.

Tabla 17. Pruebas de parámetros aplicadas a Snort con el troyano *Darkcomet*

Troyano	**Darkcomet**	
Parámetro	**Prueba**	**Resultados y comentarios**
Registros en el sistema	Reconocer instalaciones o desinstalaciones de programas como utorrent, Winrar, Notepad ++, y/o archivos como: System.ini, Win.ini.	Se realizaron instalaciones de software para uso común como: *winrar, Notepad* ++, entre otros, sin obtener respuesta alguna por parte del IDS sobre las acciones efectuadas en el computador.
Llaves de registro	Alterar/modificar y/o crear llaves de registro para el SO por medio del troyano.	Snort no reconoce cambios efectuados a las llaves de registro que se encuentran en el SO, y tampoco genera ningún tipo de alerta con respecto a la eliminación de estas.
Conexiones exitosas hacia el agente	Realizar conexiones remotas desde el troyano al agente/cliente.	El computador reconoce la conexión que tiene el troyano con él. Utilizando el comando "*netstat –v*" (ver figura 32), se ve la conexión realizada por omisión del troyano al computador (1) que a su vez también fue detectada por el IDS (2).

Parámetro	Prueba	Resultados y comentarios

Figura 32. Conexión exitosa de *Darkcomet* y detectada por Snort



Parámetro	Prueba	Resultados y comentarios
Checksum	Alterar y/o modificar un archivo que sea monitoreado por el IDS y/o modificar un archivo que sea importante para el SO como win.ini, autoexec.bat y observar las acciones del IDS.	Snort no detecta las modificaciones o alteraciones realizadas a archivos ni documentos, el *syslog* es revisado por omisión únicamente en Linux.
Falsos positivo	Realizar actividades comunes de usuario tales como programas de mensajería instantánea como skype, mirar correo electrónico como gmail y utilizar páginas de entretenimiento como youtube o facebook y observar si genera alertas.	Snort reconoce la mayoría de las descargas realizadas desde el correo electrónico y generó alertas al utilizar cuentas asociadas de *no-ip*[17] que era un programa legítimo del usuario, por ende no debía generar ningún tipo de alerta. Así mismo, al navegar por sitios web como *youtube* generó constantemente alertas del tipo *http_inspect* el cual era un error de configuración que contenía el archivo *snort.conf* al momento de descargar las reglas oficiales.

[17] *No-ip*: Es un servicio de DNS dinámico para empresas y hogares que permite la asignación de un nombre a una dirección IP.

Parámetro	Prueba	Resultados y comentarios
Ausencia de falsos negativos	Realizar actividades por medio del troyano que puedan comprometer el funcionamiento del SO y observar si el IDS genera alguna alerta.	El IDS detecta la conexión principal del troyano, pero no detecta acciones realizadas desde *Darkcomet,* tales como: abrir una consola remota, observar los puertos abiertos del computador, entre otros.
Detección de *rootkit*	Camuflar el troyano en un proceso legítimo del SO o en llaves de registro.	*Darkcomet* puede camuflarse como proceso en el SO y modificar las llaves de registro del SO sin que el IDS pueda detectarlo.

Tabla 18. Pruebas de parámetros aplicadas a Snort con el troyano *Spy-Net*

Troyano	***Spy-Net***	
Parámetro	**Prueba**	**Resultados y comentarios**
Registros en el sistema	Reconocer instalaciones o desinstalaciones de programas como *utorrent*, *Winrar*, *Notepad* ++, y/o archivos como: *System.ini, Win.ini*	Por medio de *Spy-net* se accedió al computador cliente y después de realizar algunas acciones no se detecta la instalación y desinstalación de software en el SO.

Parámetro	Prueba	Resultados y comentarios
Llaves de registro	Alterar/modificar y/o crear llaves de registro para el SO por medio del troyano.	Se realizaron cambios a las llaves de registro sin obtener alguna alerta del IDS.
Conexiones exitosas hacia el agente	Realizar conexiones remotas desde el troyano al agente/cliente.	El IDS reconoce la primera conexión efectuada para conectar el troyano con la víctima, pero las otras conexiones establecidas desde el troyano no fueron detectadas por el IDS.
Checksum	Alterar y/o modificar un archivo monitoreado por el IDS y/o modificar un archivo que sea importante para el SO como *win.ini*, *autoexec.bat* y observar las acciones del IDS.	Ninguna de las acciones fue detectada por el IDS, debido a que el IDS solo analiza el *syslog* de Linux.

Parámetro	Prueba	Resultados y comentarios
Falsos positivos	Realizar actividades comunes de usuario tales como: programas de mensajería instantánea como *skype*, mirar correo electrónico como *gmail* y utilizar páginas de entretenimiento como *youtube* o *facebook* y observar si genera alertas.	Los falsos positivos generados fueron algunas descargas hechas a través del navegador de internet y algunas conexiones con programas externos como *no-ip* y constantemente emitió alertas *http_inspect* al navegar por sitios web como *gmail*.
Ausencia de falsos negativos	Realizar actividades por medio del troyano que puedan comprometer el funcionamiento del SO y observar si el IDS genera alguna alerta.	Snort no reporta ninguna de la acciones como crear o examinar carpetas del SO, espiar escritorio y copia de archivos, pero detecta la conexión que existe del troyano en el computador.
Detección de *rootkit*	Camuflar el troyano en un proceso legítimo del SO o en llaves de registro	Se utilizó como *rootkit* el camuflaje en los procesos del SO, para lo cual el IDS no logró identificar esa forma de intrusión.

Tabla 19. Pruebas de parámetros aplicadas a Snort con el troyano *Poison Ivy*

Troyano	**Poison Ivy**	
Parámetro	**Prueba**	**Resultados y comentarios**
Registros en el sistema	Reconocer instalaciones o desinstalaciones de programas como *utorrent*, *Winrar*, *Notepad* ++, y/o archivos como: *System.ini, Win.ini*	Snort no reconoce cuando se instala/desinstala un software o se elimina un archivo del SO.
Llaves de registro	Alterar/modificar y/o crear llaves de registro para el SO por medio del troyano.	Snort no reconoce cuando por medio del troyano se cambia, se altera o se crea una nueva llave de registro en el computador monitoreado.
Conexiones exitosas hacia el agente	Realizar conexiones remotas desde el troyano al agente/cliente.	El IDS reconoce la conexión inicial del troyano al conectarse con el computador, pero no las realizadas después de esa primera conexión. .
Checksum	Alterar y/o modificar un archivo monitoreado por el IDS y/o modificar un archivo que sea importante para el SO como *win.ini*, *autoexec.bat* y observar las acciones del IDS	Snort no reconoce cuando se altera o se modifica algún archivo debido a que esa característica la cumple el *syslog* y esa función solo se encuentra disponible para el SO Linux.

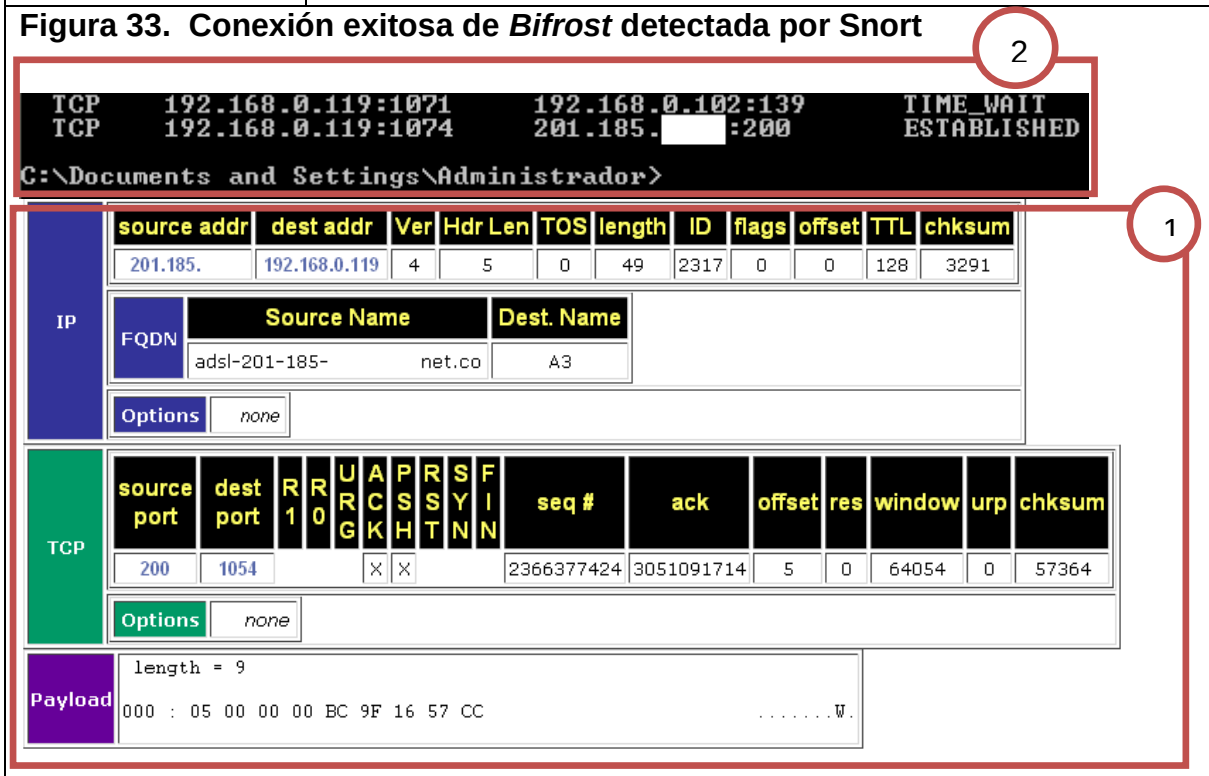
Parámetro	Prueba	Resultados y comentarios
Falsos positivos	Realizar actividades comunes de usuario tales como: programas de mensajería instantánea como *skype*, mirar correo electrónico como *gmail* y utilizar páginas de entretenimiento como *youtube* o *facebook* y observar si genera alertas.	Ante acciones comunes realizadas por un usuario, Snort generó alertas al realizar descargas desde el correo electrónico. También genero alertas *http_inspect* al navegar por sitios web como *youtube*.
Ausencia de falsos negativos	Realizar actividades por medio del troyano que puedan comprometer el funcionamiento del SO y observar si el IDS genera alguna alerta.	Se realizaron pruebas como cambios de llaves de registro, escaneo de puertos, visualización del escritorio, entre otros, sin obtener ninguna alarma por parte del IDS, solo obtuvo respuesta por parte del IDS al momento de establecer la conexión del troyano.
Detección de *rootkit*	Camuflar el troyano en un proceso legítimo del SO o en llaves de registro.	Se logró camuflar el troyano como un servicio del SO sin obtener respuesta alguna del IDS a esta acción.

Tabla 20. Pruebas de parámetros aplicadas a Snort con el troyano *Bifrost*

Troyano	**Bifrost**	
Parámetro	**Prueba**	**Resultados y comentarios**
Registros en el sistema	Reconocer instalaciones o desinstalaciones de programas como *utorrent*, *Winrar*, *Notepad* ++, y/o archivos como: *System.ini, Win.ini*	Snort no reconoce los cambios ocurridos a un software determinado para el SO Windows, no genera alertas de ninguna clase a las acciones realizadas.
Llaves de registro	Alterar/modificar y/o crear llaves de registro para el SO por medio del troyano.	Se logró alterar por medio del troyano una llave de registro monitoreada por el sistema: (*HKEY_LOCAL_MACHINE/ SOFTWARE/Classes/ cmdfile/EditFlags*) y como resultado el IDS no emitió alerta alguna. Por otra parte, al crear una llave de registro de nombre "*Damian*" el IDS no generó ni reportó ninguna alerta.

Parámetro	Prueba	Resultados y comentarios
Conexiones exitosas hacia el agente	Realizar conexiones remotas desde el troyano al agente/cliente.	Snort reconoce la intrusión al computador en el momento de realizar la conexión emitiendo una alerta indicando la presencia de un troyano en el computador (1), mostrando la dirección IP origen y destino, y el SO, también reconoce la conexión (2) (Ver figura 33).

Figura 33. Conexión exitosa de *Bifrost* detectada por Snort



Parámetro	Prueba	Resultados y comentarios
Checksum	Alterar y/o modificar un archivo monitoreado por el IDS y/o modificar un archivo que sea importante para el SO como *win.ini*, *autoexec.bat* y observar las acciones del IDS.	Snort puede analizar y revisar al SO por medio del *syslog* pero esa función solo la puede efectuar en distribuciones Linux, por consiguiente no se generó ninguna alerta en el SO Windows.
Falsos positivos	Realizar actividades comunes de usuario tales como : programas de mensajería instantánea como *skype*, mirar correo electrónico como *gmail* y utilizar páginas de entretenimiento como *youtube* o *facebook* y observar si genera alertas	Snort reconoce las descargas realizadas desde una cuenta de correo electrónico hechas por el usuario con una alerta de nivel bajo. También genera alertas del tipo *http_inspect* al navegar por sitios web como *youtube*.

Parámetro	Prueba	Resultados y comentarios
Ausencia de falsos negativos	Realizar actividades por medio del troyano que puedan comprometer el funcionamiento del SO y observar si el IDS genera alguna alerta	Snort reconoce la conexión realizada al momento de ejecutar el troyano, pero no reconoce acciones realizadas después de esa primera conexión, permitiendo abrir consolas remotas, escaneo de puertos abiertos del computador, observar y terminar procesos del SO, entre otros.
Detección de *rootkit*	Camuflar el troyano en un proceso legítimo del SO o en llaves de registro.	Snort no reconoce cambios en el modo de ejecución del troyano, tampoco reconoce acciones como ocultar el troyano en un proceso legítimo del SO y tampoco en el inicio del SO.

Las pruebas realizadas con los troyanos en Snort obtuvieron resultados sorpresivos como se puede observar en la tabla 21; aunque todos los troyanos fueron detectados, los parámetros detectados fueron pocos, en consecuencia el puntaje final fue relativamente bajo. Los parámetros detectados por Snort en los cuatro troyanos fueron los mismos, por ende los resultados de Snort frente a cada troyano fue el mismo.

La forma en que trabaja Snort es por medio de reglas que pueden ser descargadas desde la página oficial de Snort o creadas de forma específica por el usuario o administrador. Las reglas operan por medio de patrones, para éste proyecto fue necesario la creación de reglas para cada troyano (ver anexo J: Ejemplo de creación de reglas para Snort), debido a que con las reglas descargadas de la página oficial de Snort no fueron detectados los troyanos.

A partir de los patrones o firmas que genera el troyano cuando efectúa la conexión se pueden crear reglas para él, es de esta forma que el IDS puede detectarlo, pero las acciones que realiza el troyano después de establecer la conexión entre el cliente y el servidor son completamente cifradas, lo que hace prácticamente imposible poder detectar patrones y crear reglas para acciones específicas realizadas por el troyano, debido a que cada acción del troyano genera un patrón o firma totalmente diferente al momento de capturar tráfico que llega al computador y eso se evidenció en las pruebas, específicamente en la ausencia de falsos negativos.

Por otra parte, Snort generó constantemente alertas *http_inspect no content-length or transfer-encoding in http_response*, alerta que se generó por un error que

contenía las reglas VRT[18] descargadas desde el sitio web de Snort en el archivo *snort.conf*, como resultado generaba alertas al visitar la mayoría de sitios web emitiendo falsos positivos. VRT junto con Snort publicaron el nuevo archivo de *snort.conf*[19] para ser descargado por cualquier persona que tenga ese u otro problema derivado del archivo de configuración *snort.conf*.

En conclusión, Snort demostró la efectividad que posee para detectar las conexiones que se efectuaron al computador a analizar, otorgando información relevante del intruso, como la dirección IP y el nombre del equipo.

Después de realizadas las pruebas y según la tabla 9 de rendimiento ante el uso de troyanos, Snort se clasifica como REGULAR en la categoría 2 con un puntaje promedio de 31,5, valor que se logró por medio de la evaluación con la rúbrica (ver anexo I).

Los parámetros alcanzados por Snort en las pruebas fueron:

- **Conexiones exitosas hacia el agente cliente:** Snort reconoció y detectó que el troyano estaba presente en el computador. El puntaje que obtuvo fue el máximo: 15 puntos.
- **Ausencia de falsos negativos:** Al poder reconocer que un troyano estaba presente en el computador Snort logró reconocer que verdaderamente ocurrió algo en el computador. El puntaje que obtuvo fue 4,5 de 9 puntos posibles.
- **Algoritmo *hash*:** Reconoce cambios en configuraciones del SO.

---

[18] VRT (*Vulnerability Research Team)):* Grupo de expertos en seguridad informática que trabajan de forma proactiva para evaluar y responder ante las tendencias de actividades *hacking.*

[19] Snort.conf: archivo de configuración actualizado por Snort y VRT [En línea] Disponible: http://www.snort.org/vrt/snort-conf-configurations/

- **Frecuencia de escaneo:** El escaneo que realiza Snort es constante, no tiene límite de tiempo. El puntaje que obtuvo fue el máximo: 4 puntos.

Tabla 21. Puntaje total obtenido por Snort

Parámetro	Puntos obtenidos por Troyano			
	Darkcomet	**Spy-Net**	**Poison Ivy**	**Bifrost**
Registros en el sistema	0	0	0	0
Llaves de registro	0	0	0	0
Conexiones exitosas hacia el agente/cliente	15	15	15	15
Checksum	0	0	0	0
Falsos positivos	3	3	3	3
Ausencia de falsos negativos	4,5	4,5	4,5	4,5
Detección de *rootkit*	0	0	0	0
Carpetas por omisión	0	0	0	0
Algoritmo *hash*	5	5	5	5
Frecuencia de escaneo	4	4	4	4
Valores obtenidos por Snort	**31,5**	**31,5**	**31,5**	**31,5**

Los parámetros no alcanzados por Snort en las pruebas fueron:

- **Llaves de registro:** No logró reconocer cambios efectuados a las llaves de registro del SO. El puntaje que obtuvo fue 0 de 10 puntos posibles.
- **Registros en el sistema:** No reconoció instalaciones/desinstalaciones en el SO, tampoco reconoció alteraciones a carpetas o archivos. El puntaje que obtuvo fue 0 de 25 puntos posibles.

- ***Checksum*:** No reconoció cambios de archivos y/o carpetas por medio del *checksum*. El puntaje que obtuvo fue 0 de 15 puntos posibles.
- **Falsos positivos:** Reconoció descargas legítimas del usuario como alertas. El puntaje que obtuvo fue 0 de 4 puntos posibles.
- **Detección de *rootkit*:** No detectó la forma de ocultarse del SO. El puntaje que obtuvo fue 0 de 7 puntos posibles.
- **Carpetas por omisión:** No analizó ningún tipo de carpeta y/o archivo en el SO. El puntaje que obtuvo fue 0 de 6 puntos posibles.

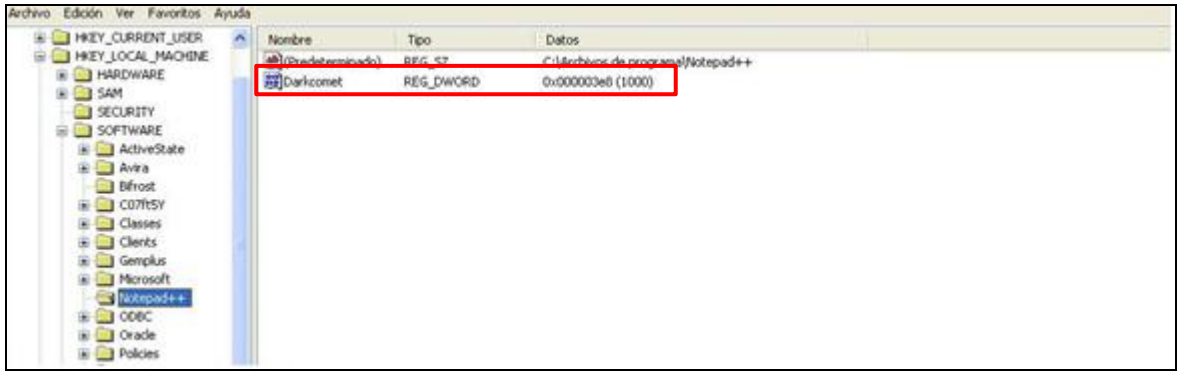
7.3. PRUEBAS Y ANÁLISIS A SGUIL

Tabla 22. Parámetros en común de Sguil para las pruebas con los troyanos

Parámetro	Prueba	Resultados y comentarios
Algoritmo *hash*	Reconocer el algoritmo *hash* implementado por el IDS.	En la configuración del sensor de Sguil no se especifica el uso de ningún tipo de algoritmo *hash*, pero durante la ejecución de las pruebas se pudo comprobar que éste realiza *Checksum* por medio de las funciones *hash* MD5 y SHA1.
Carpetas escaneadas por omisión	Establecer las carpetas por omisión escaneadas por el IDS.	Sguil no establece ningún tipo de escaneo de carpetas, debido a que al igual que Snort está constantemente analizando el tráfico que pasa a través de la tarjeta de red hacia el computador monitoreado.
Frecuencia de escaneo	Buscar dentro del archivo de configuración el tiempo establecido por omisión para el escaneo del agente.	Para la versión de Sguil cliente 0.7 implementada en este proyecto el tiempo establecido por omisión en el que el sensor y el servidor actualizan la información es de 15 segundos.

Tabla 23. Pruebas de parámetros aplicadas a Sguil con el troyano *Darkcomet*

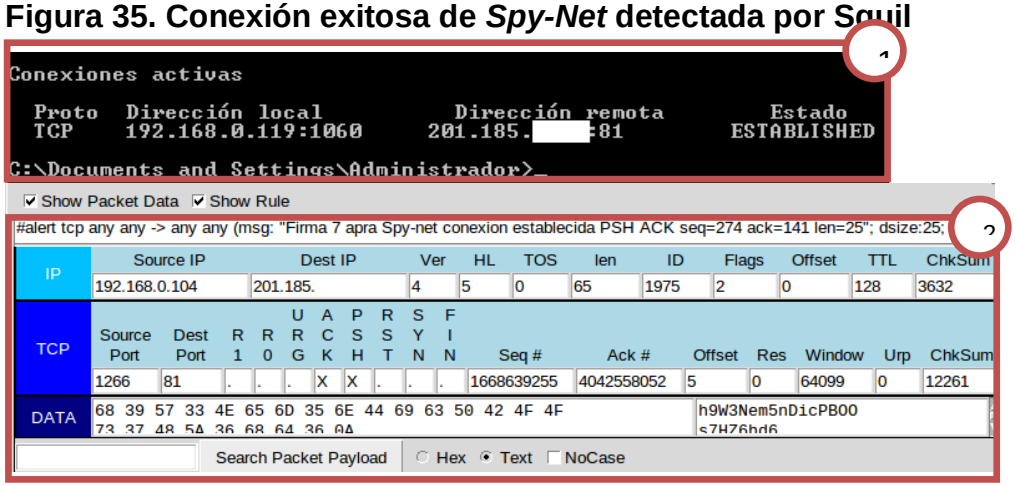
Troyano	**Darkcomet**	
Parámetro	**Prueba**	**Resultados y comentarios**
Registros en el sistema	Reconocer instalaciones o desinstalaciones de programas como *utorrent*, *Winrar*, *Notepad* ++, y/o archivos como: *System.ini, Win.ini*	Desde el troyano se logró acceder al computador donde se encuentra el sensor Sguil, y al realizar la acción de instalación/desinstalación de programas no se recibe ningún tipo de respuesta por parte del IDS.
Llaves de registro	Alterar/modificar y/o crear llaves de registro para el SO por medio del troyano.	A través de la opción *Remote Registry* que ofrece el troyano, se accedió a las llaves de registro del computador monitoreado, permitiendo crear una llave (Ver figura 34) y modificar algunas otras, sin embargo el sensor de Sguil no reportó ninguna de estas acciones.

<table>
<tr><th>Parámetro</th><th>Prueba</th><th>Resultados y comentarios</th></tr>
<tr><td></td><td colspan="2">Figura 34. Creación llave de registro por medio de Darkcomet no detectada por Sguil.
 
</td></tr>
<tr><td>Conexiones exitosas hacia el agente</td><td>Realizar conexiones remotas desde el troyano al agente/cliente.</td><td>Se logró realizar exitosamente la conexión entre el cliente y el servidor del troyano, y para este caso el sensor Sguil genera una alerta informando que el troyano Darkcomet se encuentra en ese computador.</td></tr>
<tr><td>Checksum</td><td>Alterar y/o modificar un archivo monitoreado por el IDS y/o modificar un archivo que sea importante para el SO como win.ini, autoexec.bat y observar las acciones del IDS.</td><td>A pesar de permitir la configuración de HIDS sobre Windows, Sguil realiza Checksum solamente a los archivos de configuración que se encuentran tanto en el cliente como en el sensor, pero no sobre ningún otro tipo de archivo.</td></tr>
</table>

Parámetro	Prueba	Resultados y comentarios
Falsos positivos	Realizar actividades comunes de usuario tales como: programas de mensajería instantánea como *skype*, mirar correo electrónico como *gmail* y utilizar páginas de entretenimiento como *youtube* o *facebook* y observar si genera alertas.	Al realizar acciones como consultar el correo y visitar páginas web, el IDS no reporta en ningún caso nada inusual por parte del usuario.
Ausencia de falsos negativos	Realizar actividades por medio del troyano que puedan comprometer el funcionamiento del SO y observar si el IDS genera alguna alerta.	Se implementaron las funciones de *Keylogger* y *Active Ports*, logrando hasta cerrar el sensor de Sguil en Windows sin obtener ningún tipo de alerta en el servidor de Sguil a estas acciones, pero la conexión realizada por medio del troyano si es reportada por parte del IDS.
Detección de *rootkit*	Camuflar el troyano en un proceso legítimo del SO o en llaves de registro.	Se consiguió camuflar el troyano como un proceso del SO, e instalar un ejecutable dentro de una carpeta del sistema, pero no se obtuvo respuesta o alerta del IDS a ésta acción.

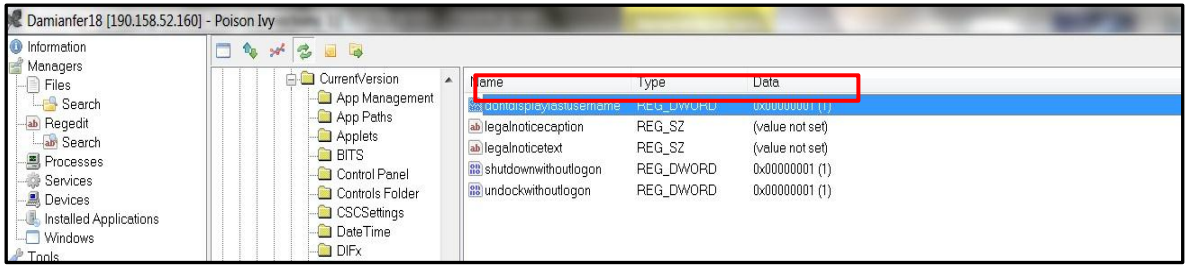
Tabla 24. Pruebas de parámetros aplicadas a Sgui con el troyano *Spy-Net*

Troyano	**Spy-Net**	
Parámetro	**Prueba**	**Resultados y comentarios**
Registros en el sistema	Reconocer instalaciones o desinstalaciones de programas como *utorrent*, *Winrar*, *Notepad* ++, y/o archivos como: *System.ini, Win.ini*	De manera oculta se consiguió instalar una aplicación (*winrar*), sin recibir respuesta del sensor de Sguil a esta acción.
Llaves de registro	Alterar/modificar y/o crear llaves de registro para el SO por medio del troyano	Por medio de la conexión establecida se pudo realizar la creación de una llave de registro y no se generó ningún tipo de respuesta del IDS.
Conexiones exitosas hacia el agente	Realizar conexiones remotas desde el troyano al agente/cliente.	El troyano creó una conexión TCP por el puerto 81 (1), obteniendo una respuesta del IDS (2), como se muestra en la figura 35.

Parametro	Prueba	Resultados y comentarios
	Figura 35. Conexión exitosa de *Spy-Net* detectada por Sguil  Conexiones activas Proto Dirección local Dirección remota Estado TCP 192.168.0.119:1060 201.185.[redacted]:81 ESTABLISHED C:\Documents and Settings\Administrador>_ Show Packet Data Show Rule #alert tcp any any -> any any (msg: "Firma 7 apra Spy-net conexion establecida PSH ACK seq=274 ack=141 len=25"; dsize:25; IP: Source IP 192.168.0.104; Dest IP 201.185.; Ver 4; HL 5; TOS 0; len 65; ID 1975; Flags 2; Offset 0; TTL 128; ChkSum 3632 TCP: Source Port 1266; Dest Port 81; R1 . ; R0 . ; URG . ; ACK X; PSH X; RST . ; SYN . ; FIN . ; Seq # 1668639255; Ack # 4042558052; Offset 5; Res 0; Window 64099; Urp 0; ChkSum 12261 DATA: 68 39 57 33 4E 65 6D 35 6E 44 69 63 50 42 4F 4F 73 37 48 5A 36 68 64 36 0A — h9W3Nem5nDicPBOO s7HZ6hd6 Search Packet Payload Hex Text NoCase	
Checksum	Alterar y/o modificar un archivo monitoreado por el IDS y/o modificar un archivo que sea importante para el SO como *win.ini*, *autoexec.bat* y observar las acciones del IDS.	Por medio del troyano se consiguió alterar el contenido del archivo de configuración de Sguil (*Sguil.conf*) y el IDS no presentó respuesta alguna a esta alteración.

Parámetro	Prueba	Resultados y comentarios
Falsos positivos	Realizar actividades comunes de usuario tales como: programas de mensajería instantánea como *skype*, mirar correo electrónico como *gmail* y utilizar páginas de entretenimiento como *youtube* o *facebook* y observar si genera alertas.	Al realizar actividades comunes por un usuario en el computador como utilizar (*skype*, *gmail*, y *facebook*), el sensor del IDS no reporta ningún tipo de anomalía a estas acciones.
Ausencia de falsos negativos	Realizar actividades por medio del troyano que puedan comprometer el funcionamiento del SO y observar si el IDS genera alguna alerta	Por medio de *Spy-Net* se consiguió acceder al computador de la víctima, copiar archivos de gran importancia para el SO, y abrir consolas remotas sin recibir ninguna notificación por parte del IDS. Aunque la conexión realizada con el troyano fue detectada.
Detección de *rootkit*	Camuflar el troyano en un proceso legítimo del SO o en llaves de registro.	Al utilizar la función *rootkit* de *Spy-Net* se logró camuflar el troyano para que se conecte una vez inicie sesión el usuario de Windows.

Tabla 25. Pruebas de parámetros aplicadas a Sguil con el troyano *Poison Ivy*

Troyano	**Poison Ivy**	
Parámetro	**Prueba**	**Resultados y comentarios**
Registros en el sistema	Reconocer instalaciones o desinstalaciones de programas como *utorrent*, *Winrar*, *Notepad* ++, y/o archivos como: *System.ini, Win.ini*	Por medio del *file manager* que posee *Poison Ivy*, se consiguió copiar el instalador de un programa en el disco duro del computador víctima, e intentar instalar la aplicación sin recibir ninguna alerta a las acciones.
Llaves de registro	Alterar/modificar y/o crear llaves de registro para el SO por medio del troyano.	Usando la función de *Regedit* contenida en *Poison Ivy* se obtuvo acceso a las llaves de registro, alterando el valor de una llave de registro (*dontdisplaylastusername*) (Ver figura 36), cambiando el último digito (0 cero) por un (1 uno) y el IDS no reportó nada.
	Figura 36. Creación llave de registro por medio de *Spy-Net* no detectada por Sguil. 	

Parámetro	Prueba	Resultados y comentarios
Conexiones exitosas hacia el agente	Realizar conexiones remotas desde el troyano al agente/cliente.	Al ejecutar el troyano *Poison* Ivy éste crea una conexión TCP a través del puerto 3460, conexión mostrada por el SO, y el IDS reporta este tipo de conexión con una alerta.
Checksum	Alterar y/o modificar un archivo monitoreado por el IDS y/o modificar un archivo que sea importante para el SO como *win.ini*, *autoexec.bat* y observar las acciones del IDS.	A través de la conexión que se creó con el troyano se logró acceder a la carpeta de Sguil y alterar algunos parámetros del archivo de configuración del IDS, sin recibir ninguna respuesta.
Falsos positivos	Realizar actividades comunes de usuario tales como: programas de mensajería instantánea como *skype*, mirar correo electrónico como *gmail* y utilizar páginas de entretenimiento como *youtube* o *facebook* y observar si genera alertas	El sensor no genera ningún tipo de reporte o alerta a las acciones realizadas, acciones como descargas por el explorador de internet, creación de carpetas y archivos, uso de programas de mensajería instantánea como *skype* o páginas de entretenimiento como *facebook*.

Parámetro	Prueba	Resultados y comentarios
Ausencia de falsos negativos	Realizar actividades por medio del troyano que puedan comprometer el funcionamiento del SO y observar si el IDS genera alguna alerta.	Al remplazar el archivo de inicio del SO *boot.ini* por *boot2.ini* y abrir consolas remotas el IDS no generó ningún tipo de alerta. Sin embargo, el agente Sguil reporta que se ha realizado una conexión y emite una alerta.
Detección de *rootkit*	Camuflar el troyano en un proceso legítimo del SO o en llaves de registro.	Por medio de *Poison Ivy* se logró instalar el troyano dentro de una carpeta del SO, de forma tal que el servidor se conectara cada vez que el usuario inicie sesión, ante esta acción el sensor de Sguil no reportó ningún tipo de alerta.

Tabla 26. Pruebas de parámetros aplicadas a Sguil con el troyano *Bifrost*

Troyano	Bifrost	
Parámetro	**Prueba**	**Resultados y comentarios**
Registros en el sistema	Reconocer instalaciones o desinstalaciones de programas como *utorrent*, *Winrar*, *Notepad* ++, y/o archivos como: *System.ini, Win.ini*.	Por medio de *bifrost* se consiguió acceder al equipo víctima y desinstalar una aplicación (*winrar*), sin que el IDS registrara ningún tipo de alerta a esta acción.

Parámetro	Prueba	Resultados y comentarios
Llaves de registro	Alterar/modificar y/o crear llaves de registro para el SO por medio del troyano.	Se logró alterar por medio del troyano una llave de registro monitoreada por el sistema: (*HKEY_LOCAL_MACHINE/ SOFTWARE/Classes/ cmdfile/EditFlags*) y como resultado el IDS no emitió alerta alguna. Por otra parte, al crear una llave de registro de nombre "*Damian*" el IDS no generó ni reportó ninguna alerta.
Conexiones exitosas hacia el agente	Realizar conexiones remotas desde el troyano al agente/cliente.	*Bifrost* establece una conexión a través del protocolo TCP usando el puerto 200, esta conexión es establecida una vez se ejecuta el troyano, siendo reconocida por el SO y por el IDS.
Checksum	Alterar y/o modificar un archivo monitoreado por el IDS y/o modificar un archivo que sea importante para el SO como *win.ini*, *autoexec.bat* y observar las acciones del IDS	A través del *file manager* que ofrece *bifrost* se consiguió acceso al archivo de configuración de Sguil, logrando alterar el nombre del archivo, a lo que el IDS no presentó ningún tipo de alerta.
Falsos positivos	Realizar actividades comunes de usuario tales como: programas de mensajería instantánea como *skype*, mirar correo electrónico como *gmail* y utilizar páginas de entretenimiento como *youtube* o *facebook* y observar si genera alertas.	Se realizaron acciones de usuario comunes como: navegar por Internet, realizar descargas, y establecer una video llamada por *Skype*, sin registrar ningún tipo de alerta del Sensor de Sguil.

Parámetro	Prueba	Resultados y comentarios
Ausencia de falsos negativos	Realizar actividades por medio del troyano que puedan comprometer el funcionamiento del SO y observar si el IDS genera alguna alerta	Por medio del troyano se consiguió copiar archivos del sistema y reubicarlos en nuevas rutas, el IDS genera una alerta advirtiendo de la conexión, pero a las acciones como crear carpetas dentro de directorios importantes para el SO, el IDS no generó ninguna alerta.
Detección de *rootkit*	Camuflar el troyano en un proceso legítimo del SO o en llaves de registro.	Al implementar la función de *rootkit* ofrecida por *bifrost* éste se logra camuflar como servicio del SO, lo cual le permitirá conectarse con el cliente una vez se inicie sesión y el IDS no lo detectó.

Los resultados de las pruebas con Sguil mostraron similitudes a las realizadas con Snort debido a que Sguil está basado en el motor de Snort, brindándole a Sguil métodos de detección similares a los que Snort posee. Sin embargo, al utilizar Sguil no se reportaron ni se registraron tantos falsos positivos, esto se debe a que las configuraciones y reglas en Sguil son analizadas intentando evitar la generación de falsos positivos. Los parámetros detectados por Sguil según la tabla 27 fueron pocos, en consecuencia el puntaje final fue relativamente bajo.

Adicionalmente, la interfaz gráfica que ofrece Sguil no es amigable y puede en algún momento confundir al administrador, debido que puede llegar a ser totalmente diferente de otras herramientas IDS, aunque como las otras herramientas IDS permite conocer en las alertas datos importantes como dirección IP origen y destino, puerto, hora de recibo de las alertas, nombre del sensor y estatus de la alerta; además, su compleja configuración e implementación no es un punto favorable para éste IDS.

Sguil posee deficiencias en la detección de cambios en el SO del computador monitoreado, ya que ningún cambio fue reportado. Sguil presenta alertas a las conexiones que existen entre el cliente y el servidor del troyano, pero no reporta ningún tipo de alerta a las acciones que se pueden efectuar después de tener la conexión establecida entre el cliente y el servidor del troyano.

Las funciones *hash* implementadas por el IDS son solo para la verificación de la integridad de los mensajes enviados entre el servidor y el cliente/sensor del IDS, pero no son usadas en la verificación de la integridad de los archivos del cliente/sensor.

El IDS Sguil después de realizar las pruebas se posicionó en la categoría 2 obteniendo una calificación de REGULAR según la tabla rendimiento del IDS ante el uso de troyanos (ver tabla 9) con un puntaje total de 32,5.

Los parámetros alcanzados por Sguil fueron:

- **Conexiones exitosas hacia el agente/cliente**: Reconoció y detecto que el troyano estaba presente en el computador. El puntaje obtenido es el máximo: 15 puntos.
- **Falsos positivos:** No surgieron falsos positivos El puntaje obtenido es el máximo: 4 puntos.
- **Ausencia de falsos negativos:** Al poder reconocer que un troyano estaba presente en el computador Sguil logró reconocer que verdaderamente ocurrió algo en el computador. El puntaje obtenido fue 4,5 de 9 puntos posibles.
- **Algoritmo *hash*:** Posee dos algoritmos *hash* que son SHA1 y MD5 para monitorizar al SO. El puntaje que obtuvo fue el máximo: 5 puntos.
- **Frecuencia de escaneo:** El escaneo que realiza Snort es constante, no tiene límite de tiempo. El puntaje que obtuvo fue el máximo: 4 puntos.

Los parámetros no alcanzados por Sguil fueron:

- **Llaves de registro:** No logró reconocer cambios efectuados a las llaves de registro del SO. El puntaje que obtuvo fue 0 de 10 puntos posibles.
- **Registros en el sistema:** No reconoció instalaciones/desinstalaciones en el SO, tampoco reconoció alteraciones a carpetas o archivos. El puntaje que obtuvo fue 0 de 25 puntos posibles.
- ***Checksum*:** no reconoció cambios de archivos y/o carpetas por medio del *checksum*. El puntaje que obtuvo fue 0 de 15 puntos posibles.
- **Detección de *rootkit*:** No detectó la forma de ocultarse del SO. El puntaje que obtuvo fue 0 de 7 puntos posibles.
- **Carpetas por omisión:** No analizó ningún tipo de carpeta y/o archivo en el SO. El puntaje que obtuvo fue 0 de 6 puntos posibles.

Tabla 27. Puntaje total obtenido por Sguil

Parámetro	Puntos obtenidos por Troyano			
	Darkcomet	**Spy-Net**	**Poison Ivy**	**Bifrost**
Registros en el sistema	0	0	0	0
Llaves de registro	0	0	0	0
Conexiones exitosas hacia el agente/cliente	15	15	15	15
Checksum	0	0	0	0
Falsos positivos	4	4	4	4
Ausencia de falsos negativos	4,5	4,5	4,5	4,5
Detección de rootkit	0	0	0	0
Carpetas por omisión	0	0	0	0
Algoritmo hash	5	5	5	5
Frecuencia de escaneo	4	4	4	4
Valores obtenidos por Sguil	**32,5**	**32,5**	**32,5**	**32,5**

7.4. PRUEBAS Y ANÁLISIS A PRELUDE

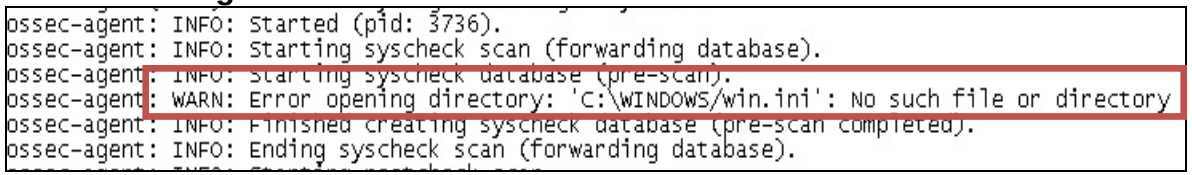
Tabla 28. Parámetros en común de Prelude para las pruebas con los troyanos

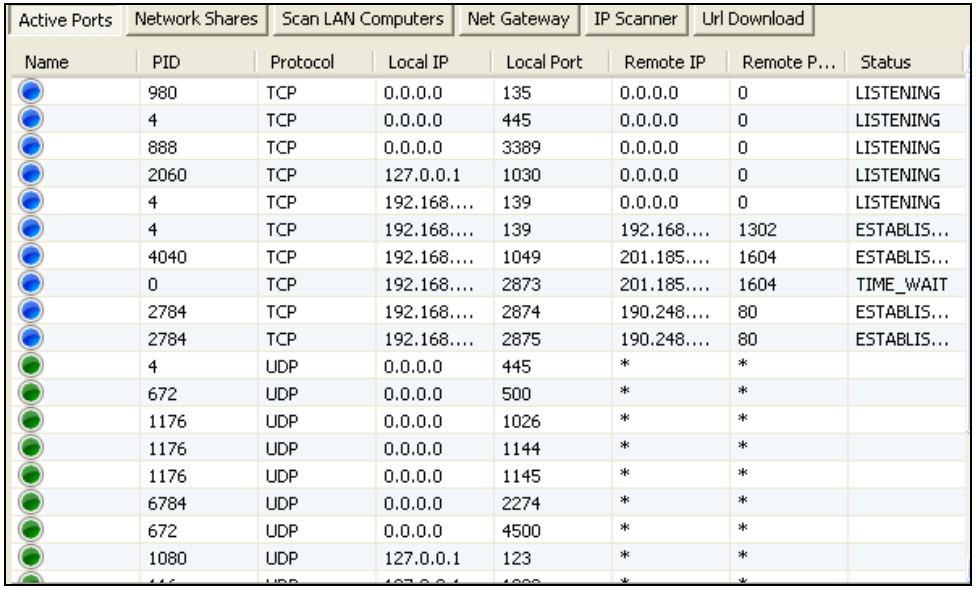
Parámetro	Prueba	Resultados y comentarios
Algoritmo *hash*	Reconocer el algoritmo *hash* implementado por el IDS.	Prelude en su configuración como HIDS necesita como agente la implementación de OSSEC, esto implica que los algoritmos *hash* que utiliza los toma de OSSEC, es decir MD5 y SHA1.
Carpetas escaneadas por omisión	Establecer las carpetas por omisión escaneadas por el IDS.	Prelude, en conjunto con OSSEC, establece las siguientes rutas a monitorear: *C:/WINDOWS* *C:/WINDOWS/System32* *C:/Document and Settings/all user/start menu/programs/startup*
Frecuencia de escaneo	Buscar dentro del archivo de configuración el tiempo establecido por omisión para el escaneo del agente.	Como el agente que usa Prelude pertenece a OSSEC, la frecuencia establecida por omisión para el escaneo del computador o *host* es de 72000 segundos.

Tabla 29. Pruebas de parámetros aplicadas a Prelude con el troyano *Darkcomet*

Troyano	**Darkcomet**	
Parámetro	**Prueba**	**Resultados y comentarios**
Registros en el sistema	Reconocer instalaciones o desinstalaciones de programas como *utorrent*, *Winrar*, *Notepad* ++, y/o archivos como: *System.ini, Win.ini*	Reconoce la instalación de una aplicación en Windows mostrando la alerta "*Windows application monitor event*" detectado por el *rootchek*, informando qué tipo de aplicación o aplicaciones fueron instaladas y mostrando la ruta donde fue instalada. Si se realiza una instalación que no se encuentre en la base de datos del IDS, no se mostrará información relevante, como consecuencia solo informará que un software fue instalado.
Llaves de registro	Alterar/modificar y/o crear llaves de registro para el SO por medio del troyano.	Se creó por medio del troyano la llave de registro (*HKEY_LOCAL_MACHINE/SOFTWARE/Classes/Batfile/aer*) sin ser emitida una alerta por el IDS. Tampoco fue alertado un cambio realizado a una llave existente en la misma ruta (*HKEY_LOCAL_MACHINE/SOFTWARE/Classes/Batfile*)
Conexiones exitosas hacia el agente	Realizar conexiones remotas desde el troyano al agente/cliente.	La conexión inicial entre el troyano y la víctima fue exitosa (1), sin embargo fue detectada la conexión por el IDS (2) (ver figura 37).

Parámetro	Prueba	Resultados y comentarios
	Figura 37. Conexión exitosa de *Darkcomet* y detectada por Prelude de *Spy-Net* 	
Checksum	Alterar y/o modificar un archivo monitoreado por el IDS y/o modificar un archivo que sea importante para el SO como *win.ini*, *autoexec.bat* y observar las acciones del IDS.	Desde el troyano se accedió a la carpeta Windows para posteriormente eliminar el archivo llamado "*win.ini*" el cual forma parte importante de la configuración inicial para el SO. Como resultado no se obtuvo ninguna respuesta del IDS, siendo este archivo monitoreado por el mismo. Al revisar los registros de suceso se encuentra que efectivamente generó una advertencia (ver figura 38) indicando problemas con ese archivo, pero no fue manifestado como alerta en el servidor.

<table>
<tr><th>Parámetro</th><th>Prueba</th><th>Resultados y comentarios</th></tr>
<tr><td></td><td colspan="2">Figura 38. Advertencia obtenida al eliminar el archivo win.ini en los registros de sucesos del agente OSSEC en Windows con Darkcomet
 
</td></tr>
<tr><td rowspan="2">Falsos positivos</td><td>Realizar actividades comunes de usuario tales como: programas de mensajería instantánea como skype, mirar correo electrónico como gmail y utilizar páginas de entretenimiento como youtube o facebook y observar si genera alertas.</td><td>Se realizaron actividades cotidianas en el computador a analizar, pero se recibieron constantemente alertas http_inspect al momento de navegar por sitios web como gmail, youtube, entre otros. Los falsos positivos se generaron por una configuración en el archivo snort.conf que contenía por omisión Snort en el momento de la descarga y actualización de las reglas.</td></tr>
<tr><td colspan="2">Figura 39. Falso positivo por actividades legítimas de usuario emitido por Prelude en las pruebas de Darkcomet
 
</td></tr>
<tr><td>Ausencia de falsos negativos</td><td>Realizar actividades por medio del troyano que puedan comprometer el funcionamiento del SO y observar si el IDS genera alertas.</td><td>Detecta la conexión del troyano pero no reconoce los cambios que este realiza para camuflarse en el computador víctima.
No reconoce un escaneo de puertos (ver figura 40).</td></tr>
</table>

<table>
<tr><th>Parámetro</th><th>Prueba</th><th>Resultados y comentarios</th></tr>
<tr><td></td><td colspan="2">Figura 40. Escaneo remoto de puertos con Darkcomet al computador analizado por Prelude
 

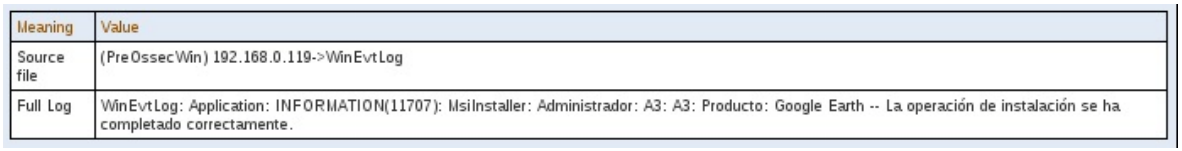
<table>
<tr><th>Name</th><th>PID</th><th>Protocol</th><th>Local IP</th><th>Local Port</th><th>Remote IP</th><th>Remote P...</th><th>Status</th></tr>
<tr><td></td><td>980</td><td>TCP</td><td>0.0.0.0</td><td>135</td><td>0.0.0.0</td><td>0</td><td>LISTENING</td></tr>
<tr><td></td><td>4</td><td>TCP</td><td>0.0.0.0</td><td>445</td><td>0.0.0.0</td><td>0</td><td>LISTENING</td></tr>
<tr><td></td><td>888</td><td>TCP</td><td>0.0.0.0</td><td>3389</td><td>0.0.0.0</td><td>0</td><td>LISTENING</td></tr>
<tr><td></td><td>2060</td><td>TCP</td><td>127.0.0.1</td><td>1030</td><td>0.0.0.0</td><td>0</td><td>LISTENING</td></tr>
<tr><td></td><td>4</td><td>TCP</td><td>192.168....</td><td>139</td><td>0.0.0.0</td><td>0</td><td>LISTENING</td></tr>
<tr><td></td><td>4</td><td>TCP</td><td>192.168....</td><td>139</td><td>192.168....</td><td>1302</td><td>ESTABLIS...</td></tr>
<tr><td></td><td>4040</td><td>TCP</td><td>192.168....</td><td>1049</td><td>201.185....</td><td>1604</td><td>ESTABLIS...</td></tr>
<tr><td></td><td>0</td><td>TCP</td><td>192.168....</td><td>2873</td><td>201.185....</td><td>1604</td><td>TIME_WAIT</td></tr>
<tr><td></td><td>2784</td><td>TCP</td><td>192.168....</td><td>2874</td><td>190.248....</td><td>80</td><td>ESTABLIS...</td></tr>
<tr><td></td><td>2784</td><td>TCP</td><td>192.168....</td><td>2875</td><td>190.248....</td><td>80</td><td>ESTABLIS...</td></tr>
<tr><td></td><td>4</td><td>UDP</td><td>0.0.0.0</td><td>445</td><td>*</td><td>*</td><td></td></tr>
<tr><td></td><td>672</td><td>UDP</td><td>0.0.0.0</td><td>500</td><td>*</td><td>*</td><td></td></tr>
<tr><td></td><td>1176</td><td>UDP</td><td>0.0.0.0</td><td>1026</td><td>*</td><td>*</td><td></td></tr>
<tr><td></td><td>1176</td><td>UDP</td><td>0.0.0.0</td><td>1144</td><td>*</td><td>*</td><td></td></tr>
<tr><td></td><td>1176</td><td>UDP</td><td>0.0.0.0</td><td>1145</td><td>*</td><td>*</td><td></td></tr>
<tr><td></td><td>6784</td><td>UDP</td><td>0.0.0.0</td><td>2274</td><td>*</td><td>*</td><td></td></tr>
<tr><td></td><td>672</td><td>UDP</td><td>0.0.0.0</td><td>4500</td><td>*</td><td>*</td><td></td></tr>
<tr><td></td><td>1080</td><td>UDP</td><td>127.0.0.1</td><td>123</td><td>*</td><td>*</td><td></td></tr>
</table></td></tr>
<tr><td>Detección de rootkit</td><td>Camuflar el troyano en un proceso legítimo del SO o en llaves de registro.</td><td>Se logró camuflar el troyano en un servicio del SO sin que el IDS identificara esa acción.</td></tr>
</table>

Tabla 30. Pruebas de parámetros aplicadas a Prelude con el troyano *Bifrost*

<table>
<tr><th>Troyano</th><th colspan="2">Bifrost</th></tr>
<tr><th>Parámetro</th><th>Prueba</th><th>Resultados y comentarios</th></tr>
<tr><td rowspan="2">Registros en el sistema</td><td>Reconocer instalaciones o desinstalaciones de programas como utorrent, google earth, Winrar, Notepad ++, y/o archivos como:

System.ini, Win.ini</td><td>Reconoce la instalación de alguna aplicación en Windows (ver figura 41) mostrando la alerta “Windows application monitor event” detectado por el rootchek informando que fue instalado un nuevo programa en el SO y muestra la ruta donde fue instalada.

El IDS genera información de la aplicación instalada consultando una base de datos sobre aplicaciones que se pueden instalar en el SO, si no encuentra la aplicación instalada en la base de datos, solamente informará que una aplicación fue instalada.</td></tr>
<tr><td colspan="2">Figura 41. Alerta emitida ante la instalación de un software en Prelude con Bifrost

 
<table>
<tr><th>Meaning</th><th>Value</th></tr>
<tr><td>Source file</td><td>(PreOssecWin) 192.168.0.119->WinEvtLog</td></tr>
<tr><td>Full Log</td><td>WinEvtLog: Application: INFORMATION(11707): MsiInstaller: Administrador: A3: A3: Producto: Google Earth -- La operación de instalación se ha completado correctamente.</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>Parámetro</th><th>Prueba</th><th>Resultados y comentarios</th></tr>
<tr><td>Llaves de registro</td><td>Alterar/modificar y/o crear llaves de registro para el SO por medio del troyano.</td><td>Se creó por medio del troyano la llave de registro (HKEY_LOCAL_MACHINE/SOFTWARE/Classes/Batfile/aer) sin que el IDS emitiera alguna alerta.

Tampoco fue alertado un cambio realizado a una llave existente en la misma ruta (HKEY_LOCAL_MACHINE/SOFTWARE/Classes/Batfile).</td></tr>
<tr><td rowspan="2">Conexiones exitosas hacia el agente</td><td>Realizar conexiones remotas desde el troyano al agente/cliente.</td><td>Se obtuvo una conexión exitosa del troyano por medio del protocolo TCP, desde el computador cliente al servidor (1) y el IDS la reconoció como una alerta (2) (ver figura 42).</td></tr>
<tr><td colspan="2">Figura 42. Conexión exitosa de Bifrost y detectada por Prelude
 
</td></tr>
</table>

<table>
<tr><th>Parámetro</th><th>Prueba</th><th>Resultados y comentarios</th></tr>
<tr><td>Checksum</td><td>Alterar y/o modificar un archivo monitoreado por el IDS y/o modificar un archivo que sea importante para el SO como win.ini, autoexec.bat y observar las acciones del IDS.</td><td>Desde el troyano se accedió a la carpeta Windows para posteriormente eliminar el archivo llamado “win.ini” el cual forma parte importante de la configuración inicial para el SO. Como resultado no se obtuvo ninguna respuesta del IDS, siendo este archivo monitoreado por él, al revisar los registros de sucesos se encuentra que efectivamente generó una advertencia pero no fue manifestado como alerta en el servidor.</td></tr>
<tr><td rowspan="2">Falsos positivos</td><td>Realizar actividades comunes de usuario tales como: programas de mensajería instantánea como skype, mirar correo electrónico como gmail y utilizar páginas de entretenimiento como youtube o facebook y observar si genera alertas.</td><td>Se realizaron actividades cotidianas en el computador a analizar y el IDS emitió continuamente alertas http_inspect al navegar por la mayoría de sitios web tales como gmail, google, youtube, entre otros. Las alertas fueron generadas a raíz de un error de configuración en el archivo snort.conf suministrado por Snort.</td></tr>
<tr><td colspan="2">Figura 43. Falso positivo por actividades legítimas de usuario emitido por Prelude en las pruebas de Bifrost.
 
</td></tr>
</table>

Parámetro	Prueba	Resultados y comentarios
Ausencia de falsos negativos	Efectuar acciones a través del troyano al agente para descubrir errores en la generación o no de las alertas.	Detecta la conexión entre el cliente y el servidor del troyano. Se ocultó el troyano dentro de otro proceso sin obtener alguna alerta del IDS; además, se realizó un escaneo de puertos sin ser detectado como anómalo.
Detección de *rootkit*	Camuflar el troyano en un proceso legítimo del SO o en llaves de registro.	Se camufló el troyano como un servicio del SO sin que el IDS identificara esa acción.

Tabla 31. Pruebas de parámetros aplicadas a Prelude con el troyano *Spy-Net*

Troyano	*Spy-Net*	
Parámetro	**Prueba**	**Resultados y comentarios**
Registros en el sistema	Reconocer instalaciones o desinstalaciones de programas como utorrent, Winrar, Notepad ++, y/o archivos como: System.ini, Win.ini	Detecta la instalación de aplicaciones nuevas en el SO emitiendo una alerta al servidor Prelude desde el sensor OSSEC.

<table>
<tr><th>Parámetro</th><th>Prueba</th><th>Resultados y comentarios</th></tr>
<tr><td>Llaves de registro</td><td>Alterar/modificar y/o crear llaves de registro para el SO por medio del troyano.</td><td>Se modificaron y se crearon llaves existentes, y el IDS en ningún momento reconoció los cambios efectuados.</td></tr>
<tr><td rowspan="2">Conexiones exitosas hacia el agente</td><td>Realizar conexiones remotas desde el troyano al agente/cliente.</td><td>Se realizó exitosamente la conexión entre el cliente y el servidor del troyano (1) pero fue detectada por el IDS (2) (ver figura 44). Al realizar escaneos de puertos abiertos al SO, y al cambiar la dirección IP de conexión del cliente del troyano, no se obtuvo ninguna alerta por parte del IDS.</td></tr>
<tr><td colspan="2">Figura 44. Conexión exitosa de Spy-Net y detectada por Prelude
 
</td></tr>
<tr><td>Checksum</td><td>Alterar y/o modificar un archivo monitoreado por el IDS y/o modificar un archivo que sea importante para el SO como win.ini, autoexec.bat y observar las acciones del IDS.</td><td>Desde el troyano se accedió a una carpeta monitoreada por el IDS y se alteró el contenido en ella, como resultado emitió una alerta en los primeros minutos después de realizar la modificación (ver figura 45).</td></tr>
</table>

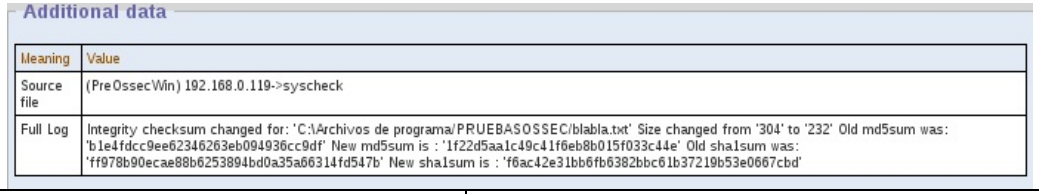
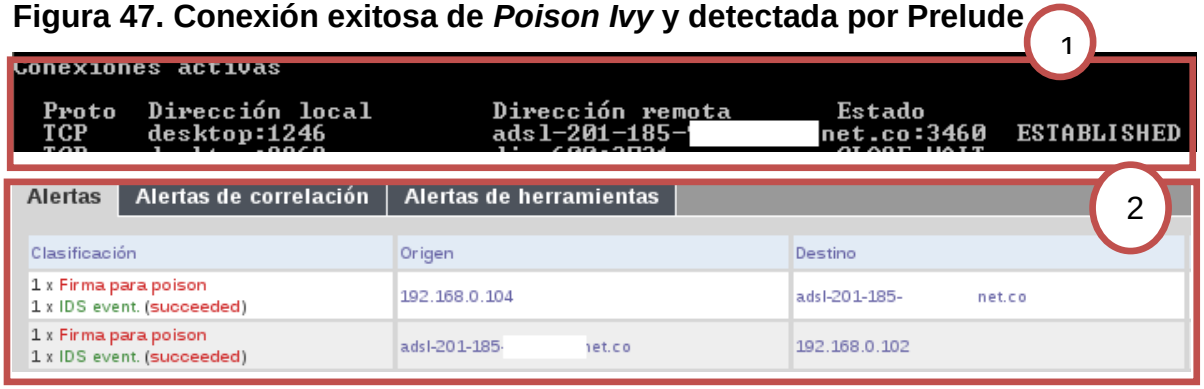
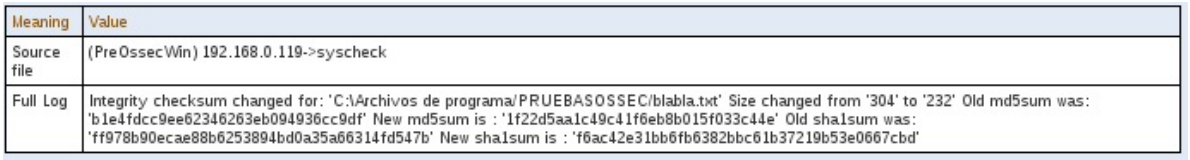
<table>
<tr><th>Parámetro</th><th>Prueba</th><th>Resultados y comentarios</th></tr>
<tr><td></td><td colspan="2">Figura 45. Alerta generada por cambio en el checksum por Prelude con Spy-Net
 
</td></tr>
<tr><td rowspan="2">Falsos positivos</td><td>Realizar actividades comunes de usuario tales como: programas de mensajería instantánea como skype, mirar correo electrónico como gmail y utilizar páginas de entretenimiento como youtube o facebook y observar si genera alertas.</td><td>Se realizaron actividades cotidianas en el computador a analizar y en el servidor. El IDS generó constantemente alertas http_inspect al momento de navegar por sitios web tales como gmail, youtube, google, entre otros. El falso positivo fue generado por el archivo de configuración snort.conf suministrado por Snort al descargar las reglas.</td></tr>
<tr><td colspan="2">Figura 46. Falso positivo por actividades legítimas de usuario emitido por Prelude en las pruebas de Spy-net
 
</td></tr>
<tr><td>Ausencia de falsos negativos</td><td>Realizar actividades por medio del troyano que puedan comprometer el funcionamiento del SO y observar si el IDS genera alguna alerta.</td><td>Detecta la conexión del troyano entre el cliente y el servidor. Se ocultó el troyano dentro de otro proceso sin obtener alguna alerta del IDS. Se realizó un escaneo de puertos abiertos al SO sin ser detectado como anómalo.</td></tr>
<tr><td>Detección de rootkit</td><td>Camuflar el troyano en un proceso legítimo del SO o en llaves de registro.</td><td>Se camufló el troyano como un servicio del SO sin que el IDS identificara esa acción, y se instaló como servicio de arranque del SO sin presentar alguna alerta.</td></tr>
</table>

Tabla 32. Pruebas de parámetros aplicadas a Prelude con el troyano *Poison Ivy*

Troyano	**Poison Ivy**	
Parámetro	**Prueba**	**Resultados y comentarios**
Registros en el sistema	Reconocer instalaciones o desinstalaciones de programas como utorrent, Winrar, Notepad ++, y/o archivos como: System.ini, Win.ini	Detecta la instalación de nuevo software al computador que se está monitoreando emitiendo una alerta al servidor de Prelude.
Llaves de registro	Alterar/modificar y/o crear llaves de registro para el SO por medio del troyano.	Se crearon y modificaron algunas llaves de registro que formaban parte del escaneo para el IDS sin obtener alguna alerta por parte del agente que se encontraba en el computador.
Conexiones exitosas hacia el agente	Realizar conexiones remotas desde el troyano al agente/cliente.	El troyano realizó la conexión satisfactoriamente entre el cliente y su servidor (1), posteriormente el IDS emitió una alerta indicando la conexión del troyano (2) (ver figura 47).

Parámetro	Prueba	Resultados y comentarios
	Figura 47. Conexión exitosa de *Poison Ivy* y detectada por Prelude 	
Checksum	Alterar y/o modificar un archivo monitoreado por el IDS y/o modificar un archivo que sea importante para el SO como *win.ini*, *autoexec.bat* y observar las acciones del IDS.	Se realizaron cambios a archivos que previamente se habían agregado a la configuración del IDS para determinar el funcionamiento de los *Checksum* empleados por el IDS. El IDS detecta el cambio y emite una alerta al servidor informando que ocurrieron cambios en la suma de *Checksum* del archivo (ver figura 48).
	Figura 48. Alerta generada por cambio en el *checksum* por Prelude con *Poison Ivy* 	

<table>
<tr><th>Parámetro</th><th>Prueba</th><th>Resultados y comentarios</th></tr>
<tr><td rowspan="2">Falsos positivos</td><td>Realizar actividades comunes de usuario tales programas de mensajería instantánea como skype, mirar correo electrónico como gmail y utilizar páginas de entretenimiento como youtube o facebook y observar si genera alertas</td><td>Se realizaron actividades cotidianas en el computador a analizar y el IDS detectó continuamente alertas del tipo http_inspect al navegar por páginas web tales como gmail, youtube, google, entre otras. (ver figura 49).</td></tr>
<tr><td colspan="2">Figura 49. Falso positivo por actividades legítimas de usuario emitido por Prelude en las pruebas de Poison Ivy
 
</td></tr>
<tr><td>Ausencia de falsos negativos</td><td>Realizar actividades por medio del troyano que puedan comprometer el funcionamiento del SO y observar si el IDS genera alguna alerta.</td><td>Detecta la conexión realizada desde el troyano, pero no detecta escaneo de puertos abiertos en el computador, y tampoco reconoce los cambios efectuados a llaves de registro.</td></tr>
<tr><td>Detección de rootkit</td><td>Camuflar el troyano en un proceso legítimo del SO o en llaves de registro.</td><td>Se hizo pasar el troyano como un servicio del SO colocándolo en el arranque del mismo sin que se emitiera una alerta.</td></tr>
</table>

Las pruebas realizadas con los troyanos en Prelude obtuvieron buenos resultados como se puede ver en la tabla 33, al trabajar Prelude con los IDS OSSEC y Snort como sensores, Prelude logró detectar la mayoría de los parámetros establecidos (ver tabla 7).

Prelude demostró la capacidad que posee para interactuar con los IDS que manejen la arquitectura IDMEF, unificando mensajes y alertas en una sola interfaz, de esta forma se pueden utilizar diferentes IDS como agentes o sensores de Prelude, logrando así ser un IDS Hibrido.

Para que Prelude trabajara como HIDS se precisó utilizar otro IDS que funcionaran de forma nativa como HIDS, en este caso se escogió OSSEC, el cual posee un agente que trabaja tanto en Linux como en Windows. Las alertas emitidas por Prelude con OSSEC fueron en tiempo real y sin mayores diferencias comparadas con las alertas emitidas por el servidor nativo de OSSEC, aunque la interfaz de Prelude es estéticamente más agradable, no representa un punto a favor como tal; la real diferencia o ventaja es la forma de unificar la información de todos los sensores que se pueden agregar a Prelude sin importar si el sensor agregado sea un IDS diferente a Prelude, los dos IDS van a trabajar sin contratiempos.

Adicionalmente, se instaló Snort que también trabaja con la arquitectura de mensajes IDMEF, logrando de esta forma incorporarse y configurarse como sensor en tiempo real para Prelude. Por otra parte, si se compara la interfaz y la información suministrada por el sensor de Snort a Prelude no es tan completa como lo haría Snort en la interfaz propia, pero no hace gran diferencia tomando como base los resultados o las detecciones que realiza.

Como en las pruebas de Snort…ver sección 7.2. … las reglas creadas funcionaron sin inconvenientes y se obtuvieron los mismos resultados en las detecciones de los troyanos por parte de Snort como sensor de Prelude.

El IDS Prelude después de realizar las pruebas se posicionó en la categoría 3 obteniendo una calificación de BUENO según la tabla rendimiento del IDS ante el uso de troyanos (ver tabla 9) con un puntaje total de 69,75.

Tabla 33. Puntaje total obtenido por Prelude

Parámetro	**Puntos obtenidos por Troyano**			
	Darkcomet	**Spy-Net**	**Poison Ivy**	**Bifrost**
Registros en el sistema	21,25	21,25	21,25	21,25
Llaves de registro	0	0	0	0
Conexiones exitosas hacia el agente/cliente	15	15	15	15
Checksum	15	15	15	15
Falsos positivos	0	0	0	0
Ausencia de falsos negativos	4,5	4,5	4,5	4,5
Detección de *rootkit*	0	0	0	0
Carpetas por omisión	6	6	6	6
Algoritmo *hash*	5	5	5	5
Frecuencia de escaneo	3	3	3	3
Valores obtenidos por Prelude	**69,75**	**69,75**	**69,75**	**69,75**

Los parámetros alcanzados por Prelude en las pruebas fueron:

- **Registros en el sistema:** Reconoció los diferentes tipos de instalaciones/desinstalaciones efectuadas al computador, enviando una alerta al servidor, pero no reconoció cambios en el inicio del SO. El puntaje que obtuvo fue 21,25 de 25 puntos posibles.
- **Conexiones exitosas hacia el agente/cliente:** Reconoció y detecto que el troyano estaba presente en el computador. El puntaje que obtuvo fue el máximo: 15 puntos.
- ***Checksum*:** Reconoció cambios en archivos y/o carpetas por medio del checksum, para posteriormente enviar la alerta al servidor de OSSEC. El puntaje que obtuvo fue el máximo: 15 puntos.
- **Ausencia de falsos negativos:** Al poder detectar que un troyano realizó una conexión al computador analizado se logró reconocer que verdaderamente ocurrió un cambio significativo en el computador. El puntaje que obtuvo fue 4,5 de 9 puntos posibles.
- **Carpetas por omisión:** Está pre-configurado para escanear archivos y/o carpetas importantes para el SO. El puntaje que obtuvo fue el máximo: 6 puntos.
- **Algoritmo *hash*:** Posee dos algoritmos *hash* que son SHA1 y MD5 para monitorizar los archivos y/o carpetas. El puntaje que obtuvo fue el máximo: 5 puntos.
- **Frecuencia de escaneo:** El escaneo lo puede realizar constantemente y en tiempo real para registrar posibles intrusiones al computador, también permite configurar tiempo para escanear archivos y/o carpetas. El puntaje que obtuvo fue el máximo: 4 puntos.

Los parámetros no alcanzados por Prelude en las pruebas:

- **Llaves de registro:** No logró reconocer cambios efectuados a las llaves de registro del SO. El puntaje que obtuvo fue 0 de 10 puntos posibles.

- **Falsos positivos:** Detectó el uso de programas de mensajería instantánea como alertas y descargas del correo electrónico como alertas. El puntaje que obtuvo fue 0 de 4 puntos posibles.
- **Detección de *rootkit*:** No detectó la forma de ocultarse el troyano en el SO. El puntaje que obtuvo fue 0 de 7 puntos posibles.

7.5. CONCLUSIONES DE LOS RESULTADOS OBTENIDOS EN LAS PRUEBAS

A partir de las pruebas realizadas a los IDS con los troyanos se logró determinar qué IDS consiguió detectar la mayor cantidad de parámetros de caracterización establecidos (ver tabla 7) al ser evaluados por medio de la rúbrica (ver anexo I).

Cada IDS obtuvo puntajes diferentes en los cuales se pudo evidenciar las ventajas y desventajas que se pueden tener al momento de utilizar alguna de las herramientas IDS analizadas. Estos resultados se pueden evidenciar en la tabla 34 donde se detallan los puntajes totales obtenidos por las herramientas IDS.

Tabla 34. Puntajes y clasificación final de los IDS

Categoría	Puntos	Herramienta IDS	Clasificación
3	69,75	Prelude	BUENO
3	52,05	OSSEC	BUENO
2	32,5	Sguil	REGULAR
2	31,5	Snort	REGULAR

Prelude obtuvo la categoría 3 que lo clasifica como BUENO con el puntaje de 69,75. OSSEC se ubicó en la categoría 3 clasificado como BUENO con un puntaje de 52,05. Sguil se ubicó en la categoría 2 clasificado como REGULAR con un puntaje de 32,5. Snort se ubicó en la última posición con la categoría de REGULAR con un puntaje de 31,5.

Tomando como base el estudio realizado y los puntajes obtenidos por las herramientas, el IDS seleccionado para efectuar las pruebas con los troyanos modificados será Prelude, debido a que obtuvo mayor puntaje (ver tabla 34) al momento de detectar las acciones efectuadas a través de los troyanos.

8. PRUEBA Y ANÁLISIS DE LA HERRAMIENTA IDS PRELUDE CON TROYANOS ALTERADOS

Para efectuar las pruebas y análisis de esta sección se utilizará el IDS que obtuvo mayor puntaje en el capítulo anterior, que, para este caso de estudio fue Prelude. Junto con el IDS seleccionado se utilizará la herramienta antivirus Avira, la cual fue seleccionada acorde al estudio realizado por *PassMark* …ver sección 4.8 …. Adicionalmente los troyanos utilizados serán alterados intentando que no sean detectados por las herramientas antivirus, además se observará si esas modificaciones generan algún cambio en la detección por parte del IDS.

Para modificar un *malware* o un troyano no existe una única guía o un único método y se basa especialmente en experiencias realizadas debido a que todo funciona en modo de prueba y error. Cada troyano o *malware* es totalmente diferente a otro y las modificaciones que se implementen a un troyano específico posiblemente no servirá para otro troyano, por este motivo algunas personas utilizan *crpyter*s. La función de un *crpyter* es cifrar el troyano o *malware* dentro de otro programa, y de esta forma intentar hacerlo pasar como inofensivo ante las herramientas antivirus. La ventaja al utilizar un *crpyter* radica en que uno puede funcionar para varios troyanos ahorrando tiempo y trabajo debido que al realizar una modificación a un troyano, *malware* o *crpyter* puede durar horas, días y hasta meses [38].

Al utilizar motores de búsqueda en internet se pueden encontrar y descargar *crpyter*s, aunque todos son detectados por las herramientas antivirus como software malicioso, por ende requieren ser modificados al igual que se haría con un troyano.

Las razones de utilizar un *crpyter* se deben a que los troyanos son conocidos y analizados profundamente por los desarrolladores de las herramientas antivirus, por ende las firmas encontradas por las herramientas antivirus en los troyanos son sensibles a modificaciones, esto quiere decir que en el momento de modificar la firma o firmas que se encuentran en el troyano es muy probable que se dañe el ejecutable del troyano volviéndolo inservible. Al utilizar un *crpyter* se obtiene un troyano con un código modificado, por ende diferente al original y sin firmas conocidas por las herramientas antivirus. En el anexo K se detalla el proceso para modificar un *crpyter* para después ser utilizado con el troyano.

Adicionalmente, a los troyanos se les asignarán puertos de conexión diferentes a los de las pruebas anteriores; así mismo, se colocarán contraseñas para observar si este tipo de configuración afecta o interfiere en el momento de la detección por parte de la herramienta IDS.

Las pruebas se llevaran a cabo de la siguiente forma:

1) Se tomarán los cuatro troyanos de las pruebas anteriores y un IDS que será Prelude debido al puntaje que obtuvo en las pruebas …ver sección 7.5. ...
2) Los troyanos serán modificados, alterados y cifrados, de esta forma se podrá cambiar el código del troyano logrando ser un troyano con un código diferente al original pero sin interrumpir las funcionalidades que el troyano posee. Esto se puede lograr por medio de un *crypter,* los cuales deben también ser alterados y cifrados, para ello se utilizan editores hexadecimales que permiten la modificación hexadecimal y/o binaria del código de los troyanos para poder dejarlos indetectables. Además, al momento de la configuración del troyano, cuando se asignan los puertos de conexión entre el cliente y el servidor, se asignarán puertos diferentes a los establecidos por omisión en las pruebas anteriores con los troyanos … ver

sección 7…, también se les asignarán contraseñas que serán empleadas en el momento de conectarse el cliente con el servidor del troyano.

3) Los troyanos modificados serán ejecutados evaluando de esta forma el desempeño del IDS Prelude en los parámetros establecidos …ver sección 6.2 … en las pruebas también estará presente la herramienta antivirus Avira …ver sección 4.8 … instalada y activada en los computadores que se realizarán las pruebas.

Tabla 35. Parámetros en común de Prelude para las pruebas con los troyanos modificados

Parámetro	Prueba	Resultados y comentarios
Algoritmo *hash*	Reconocer el algoritmo *hash* implementado por el IDS.	Implementa conjuntamente para la monitorización de sus archivos y carpetas, dos algoritmos *hash*: MD5 y SHA1, enviando siempre el cálculo nuevo y antiguo en cada alerta emitida.
Carpetas escaneadas por omisión	Establecer las carpetas por omisión escaneadas por el IDS.	Dentro de la configuración básica del agente para el SO Windows las carpetas a monitorear son: *C:/WINDOWS* *C:/WINDOWS/System32* *C:/Document and Settings/all user/start menu/programs/startup*
Frecuencia de escaneo	Buscar dentro del archivo de configuración el tiempo establecido por omisión para el escaneo del agente.	Dentro de la configuración, la frecuencia establecida por omisión para el escaneo del computador o *host* es de 72.000 segundos, que es equivalente a 20 horas. Sin embargo, este escaneo es aplicado en caso de no reportarse acciones por parte del usuario.

Tabla 36. Pruebas de parámetros aplicadas a Prelude con el troyano *Spy-Net* modificado

Troyano	***Spy-Net***	
Parámetro	**Prueba**	**Resultados y comentarios**
Registros en el sistema	Reconocer instalaciones o desinstalaciones de programas como *utorrent*, *Winrar*, *Notepad* ++, y/o archivos como: *System.ini, Win.ini*	Reconoce la instalación/desinstalación realizadas al SO que se está escaneando por el IDS; al emitir la alerta presenta información sobre el programa que fue instalado/desinstalado consultando la base de datos del agente.
	2012 Ene 11 10:34:51 Rule Id: 18146 level: 5 *Location: (WinOssec) 192.168.0.119->WinEvtLog* *Src IP: administrador* *Application Uninstalled.*	
Llaves de registro	Alterar/modificar y/o crear llaves de registro para el SO por medio del troyano.	Se crearon llaves de registro en rutas donde el agente realiza escaneos, sin obtener una alerta por parte del IDS en su servidor.
Conexiones exitosas hacia el agente	Realizar conexiones remotas desde el troyano al agente/cliente.	Se ejecutó el troyano sin problemas y se conectó exitosamente con su servidor en el computador víctima (ver figura 50) pero el IDS reconoció la conexión y emitió una alerta.

<table>
<tr><th>Parámetro</th><th>Prueba</th><th>Resultados y comentarios</th></tr>
<tr><td></td><td colspan="2">Figura 50. Conexión exitosa de Spy-Net modificado y detectada por Prelude
 
</td></tr>
<tr><td rowspan="2">Checksum</td><td>Alterar y/o modificar un archivo monitoreado por el IDS y/o modificar un archivo que sea importante para el SO como win.ini, autoexec.bat y observar las acciones del IDS</td><td>Se reconocen los cambios efectuados a archivos que se encuentren en su ruta de escaneo ofreciendo información de cuándo y qué archivo fue modificado.</td></tr>
<tr><td colspan="2">Location: (WinOSSEC) 192.168.0.119->ossec
Src IP: gent started: 'WinOSSEC->192.168.0.119'.
Ossec agent started.
** Alert 1329382985.24996: mail - ossec,syscheck,
2012 Feb 16 04:03:05 (WinOSSEC) 192.168.0.119->syscheck
Rule: 551 (level 7) -> 'Integrity checksum changed again (2nd time).'
Integrity checksum changed for: 'C:\Archivos de programa/PRUEBASOSSEC/blabla.txt'
Size changed from '207' to '218'
Old md5sum was: 'c8d4badcc2ffe56347d090e4009cee06'
New md5sum is : '28199eaf323f9f3939ccd6b245712cf3'
Old sha1sum was: 'd6cff45cc0866144eb8718c99dd726cd4a028c86'
New sha1sum is : 'cf562775fe439b6983bd4f9c54c6eaaa6eadf6a5'</td></tr>
</table>

Parámetro	Prueba	Resultados y comentarios
Falsos positivos	Realizar actividades comunes de usuario tales como: programas de mensajería instantánea como *skype*, mirar correo electrónico como *gmail* y utilizar páginas de entretenimiento como *youtube* o *facebook* y observar si genera alertas	Al realizar ciertas actividades como instalación/desinstalación de algún programa de mensajería instantánea como *skype* genera una alerta informando los cambios en el SO. Al navegar por sitios web como *youtube*, y *gmail*, entre otros, no generó ningún tipo de alerta.
	2012 Ene 10:40:21 Rule Id: 18154 level: 10 *Location: (ossecA2) 192.168.0.15->WinEvtLog* *Src IP: Administrador* *Multiple Windows error events.* *WinEvtLog: Application: ERROR(4099): WmiAdapter: Administrador: A2:* *A2: (no message)* *WinEvtLog: Application: ERROR(4099): WmiAdapter: Administrador: A2:* *A2: (no message)*	
Ausencia de falsos negativos	Realizar actividades por medio del troyano que puedan comprometer el funcionamiento del SO y observar si el IDS genera alguna alerta.	No detectó el cambio de llaves de registro cuando se realizó el escaneo al SO, siendo esas llaves parte de la ruta de escaneo; además, no detectó el uso de una consola remota desde *Spy-Net* (Ver figura 51). Sin embargo, el IDS detectó la conexión del troyano con su cliente y servidor.

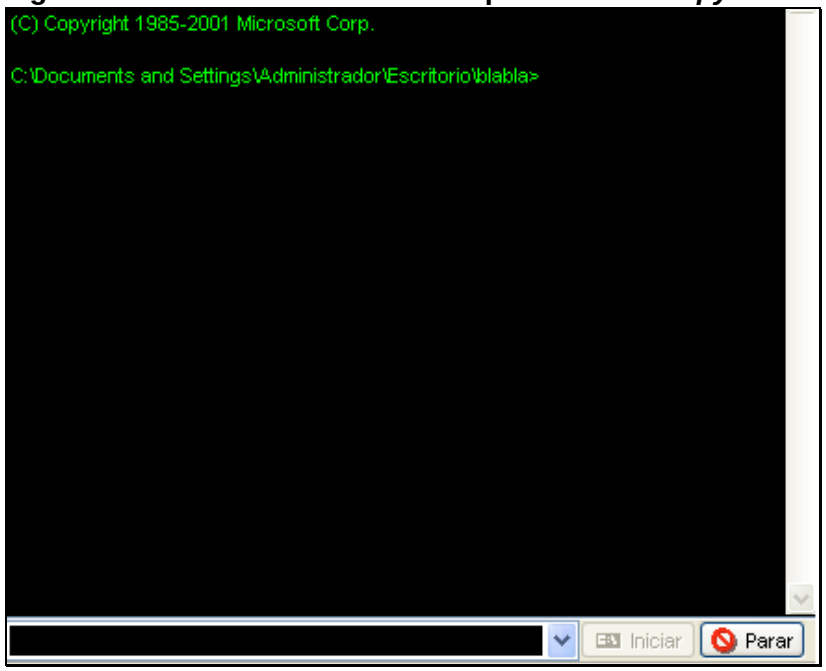
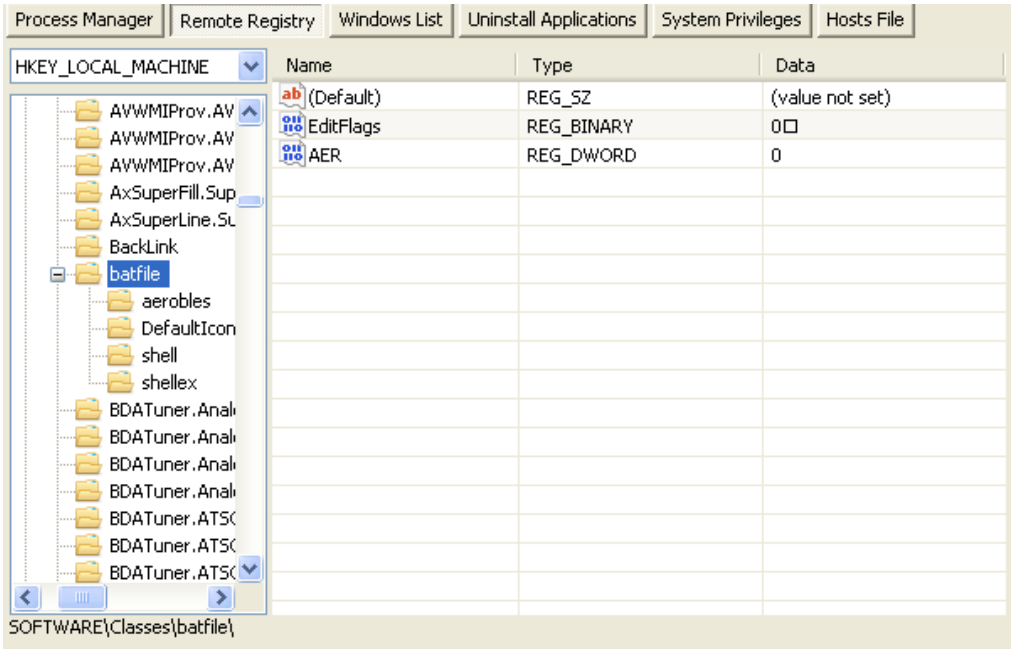
Parámetro	Prueba	Resultados y comentarios
	Figura 51. Consola remota abierta por medio de *Spy-Net* modificado no detectada por Prelude 	
Detección de *rootkit*	Camuflar el troyano en un proceso legítimo del SO o en llaves de registro.	Se hizo pasar el troyano como un servicio del SO colocándolo en el arranque del mismo sin que se emitiera una alerta. Desde la creación del troyano se configuró para que se hiciera pasar por otro proceso, de esta forma fue invisible para el IDS y también para las personas que miren los procesos, pasando de forma oculta para la herramienta antivirus.

Tabla 37. Pruebas de parámetros aplicadas a Prelude con el troyano *Darkcomet* modificado

<table>
<tr><th>Troyano</th><th colspan="2">Darkcomet</th></tr>
<tr><th>Parámetro</th><th>Prueba</th><th>Resultados y comentarios</th></tr>
<tr><td rowspan="2">Registros en el sistema</td><td>Reconocer instalaciones o desinstalaciones de programas como utorrent, Winrar, Notepad ++, y/o archivos como:

System.ini, Win.ini</td><td>Al momento de instalar/desinstalar alguna aplicación o software el agente del IDS emite una alerta al servidor indicando de que ocurrió un cambio en el SO y muestra qué software fue instalado/desinstalado si se encuentra en la base de datos del agente ese software.</td></tr>
<tr><td colspan="2">Location: (WinOSSEC) 192.168.0.119->ossec
Src IP: gent started: 'WinOSSEC->192.168.0.119'.
Ossec agent started.
** Alert 1329383798.26410: mail - windows,
2012 Feb 16 04:16:38 (WinOSSEC) 192.168.0.119->WinEvtLog
Rule: 18146 (level 5) -> 'Application Uninstalled.'
User: Administrador
WinEvtLog: Application: INFORMATION(11724): MsiInstaller: Administrador: A3: A3: Producto: Google Earth -- La eliminación se ha completado correctamente.</td></tr>
</table>

<table>
<tr><th>Parámetro</th><th>Prueba</th><th>Resultados y comentarios</th></tr>
<tr><td rowspan="2">Llaves de registro</td><td>Alterar/modificar y/o crear llaves de registro para el SO por medio del troyano.</td><td>Se modificó una llave de registro que el agente escanea por omisión, sin obtener alguna alerta por parte del IDS (ver figura 52).</td></tr>
<tr><td colspan="2">Figura 52. Creación llave de registro por medio de Darkcomet modificado no detectada por Prelude
 
</td></tr>
</table>

<table>
<tr><th>Parámetro</th><th>Prueba</th><th>Resultados y comentarios</th></tr>
<tr><td rowspan="2">Conexiones exitosas hacia el agente</td><td>Realizar conexiones remotas desde el troyano al agente/cliente</td><td>Desde el cliente del troyano al servidor en el computador víctima se realizó la conexión de forma exitosa (1), pero fue detectada por el IDS (2) (ver figura 53).</td></tr>
<tr><td colspan="2">Figura 53. Conexión exitosa de Darkcomet modificado y detectada por Prelude
 
</td></tr>
<tr><td>Checksum</td><td>Alterar y/o modificar un archivo monitoreado por el IDS y/o modificar un archivo que sea importante para el SO como win.ini, autoexec.bat y observar las acciones del IDS</td><td>Se alteró el contenido de algunos archivos que eran escaneados por el IDS, como consecuencia el agente respondió satisfactoriamente ante ese evento emitiendo una alerta al servidor mostrando el Checksum anterior y la ruta del archivo que fue alterado.</td></tr>
</table>

Parámetro	Prueba	Resultados y comentarios
	Location: (WinOSSEC) 192.168.0.119->ossec Src IP: gent started: 'WinOSSEC->192.168.0.119'. Ossec agent started. ** Alert 1329382985.24996: mail - ossec,syscheck, 2012 Feb 16 04:03:05 (WinOSSEC) 192.168.0.119->syscheck Rule: 551 (level 7) -> 'Integrity checksum changed again (2nd time).' Integrity checksum changed for: 'C:\Archivos de programa/PRUEBASOSSEC/blabla.txt' Size changed from '207' to '218' Old md5sum was: 'c8d4badcc2ffe56347d090e4009cee06' New md5sum is : '28199eaf323f9f3939ccd6b245712cf3' Old sha1sum was: 'd6cff45cc0866144eb8718c99dd726cd4a028c86' New sha1sum is : 'cf562775fe439b6983bd4f9c54c6eaaa6eadf6a5'	

Parámetro	Prueba	Resultados y comentarios
Falsos positivos	Realizar actividades comunes de usuario tales como: programas de mensajería instantánea como *skype*, mirar correo electrónico como *gmail* y utilizar páginas de entretenimiento como *youtube* o *facebook* y observar si genera alertas	Genera alertas al utilizar aplicaciones de mensajería instantánea como *skype* y también con algunas descargas realizadas desde el correo. Al navegar por sitios como *gmail*, *youtube*, entre otros, el IDS no generó ninguna alerta.
	2012 Ene 10:40:21 Rule Id: 18154 level: 10 Location: (ossecA2) 192.168.0.15->WinEvtLog Src IP: Administrador Multiple Windows error events. WinEvtLog: Application: ERROR(4099): WmiAdapter: Administrador: A2: A2: (no message) WinEvtLog: Application: ERROR(4099): WmiAdapter: Administrador: A2: A2: (no message)	

Parámetro	Prueba	Resultados y comentarios
Ausencia de falsos negativos	Realizar actividades por medio del troyano que puedan comprometer el funcionamiento del SO y observar si el IDS genera alguna alerta	No reconoce cuando se realizan escaneos para revisar qué puertos se encuentran abiertos y los procesos que tenga el SO, aunque el IDS sí reconoce la conexión efectuada entre el servidor y el cliente del troyano.
Detección de *rootkit*	Camuflar el troyano en un proceso legítimo del SO o en llaves de registro.	Se inyectó el troyano como un servicio del SO sin que el agente emitiera alguna alerta.

Tabla 38. Pruebas de parámetros aplicadas a Prelude con el troyano *Bifrost* modificado

Troyano	Bifrost	
Parámetro	**Prueba**	**Resultados y comentarios**
Registros en el sistema	Reconocer instalaciones o desinstalaciones de programas como *utorrent*, *Winrar*, *Notepad* ++, y/o archivos como: *System.ini, Win.ini*	Registra los cambios al instalar/desinstalar alguna aplicación o software en el SO emitiendo una alerta al servidor.

Parámetro	Prueba	Resultados y comentarios
	2012 Ene 11 10:34:51 Rule Id: 18146 level: 5 *Location: (WinOssec) 192.168.0.119->WinEvtLog* *Src IP: Administrador* *Application Uninstalled.*	
Llaves de registro	Alterar/modificar y/o crear llaves de registro para el SO por medio del troyano.	Al modificar una llave de registro del SO no se obtiene ninguna alerta referente a ese cambio por el agente.
Conexiones exitosas hacia el agente	Realizar conexiones remotas desde el troyano al agente/cliente.	El troyano realizó la conexión entre el cliente y el servidor exitosamente (1), sin embargo, el IDS detectó esta conexión y generó una alerta (2) (ver figura 54).

Figura 54. Conexión exitosa de *Bifrost* modificado y detectada por Prelude

Parámetro	Prueba	Resultados y comentarios
Checksum	Alterar y/o modificar un archivo monitoreado por el IDS y/o modificar un archivo que sea importante para el SO como *win.ini*, *autoexec.bat* y observar las acciones del IDS	Al momento de cambiar/modificar algún archivo que el IDS monitorea se envía una alerta para el servidor del IDS indicando qué archivo fue alterado, mostrando la ruta y los datos de la suma.
Falsos positivos	Realizar actividades comunes de usuario tales como: programas de mensajería instantánea como *skype*, mirar correo electrónico como *gmail* y utilizar páginas de entretenimiento como *youtube* o *facebook* y observar si genera alertas	Los falsos positivos que se pudieron encontrar fueron alertas de software de mensajería instantánea. Al navegar por páginas web tales como *youtube*, *gmail*, entre otras, el IDS no generó ninguna alerta.
Ausencia de falsos negativos	Realizar actividades por medio del troyano que puedan comprometer el funcionamiento del SO y observar si el IDS genera alguna alerta.	Reconoce y detecta la conexión que se efectúa entre el cliente y el servidor del troyano, pero no reconoce ciertas actividades como escaneo de puertos abiertos o de procesos en el computador.
Detección de *rootkit*	Camuflar el troyano en un proceso legítimo del SO o en llaves de registro.	Se colocó en la carpeta de Windows como un programa legítimo (ver figura 55) sin que OSSEC emitiera algún tipo de alerta y también se inyectó dentro de un proceso del SO sin que el agente lo reconociera como una alerta.

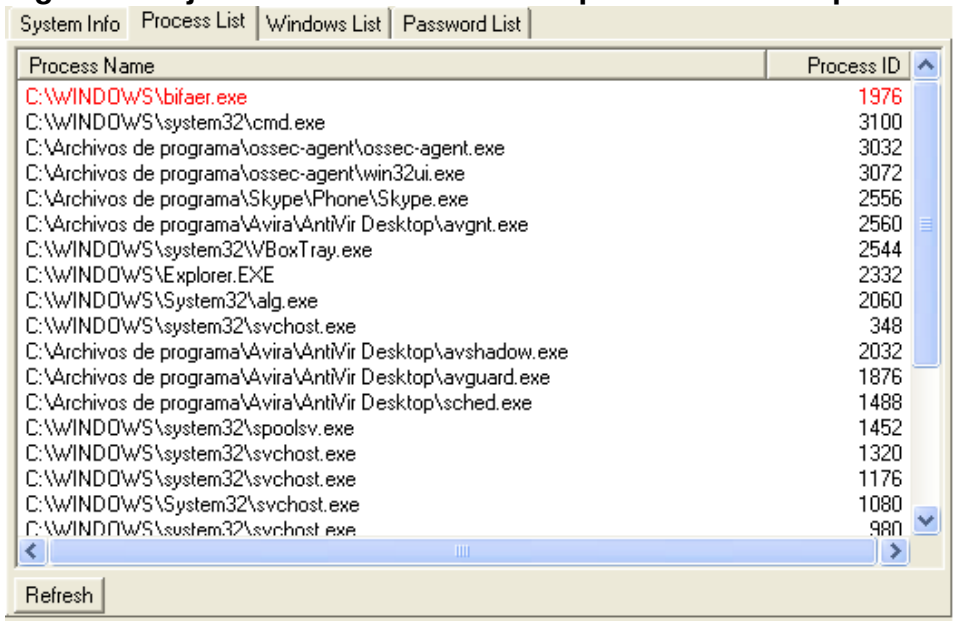
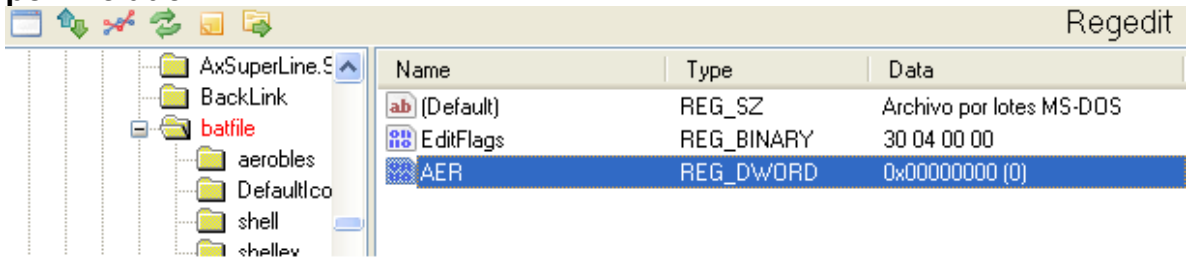
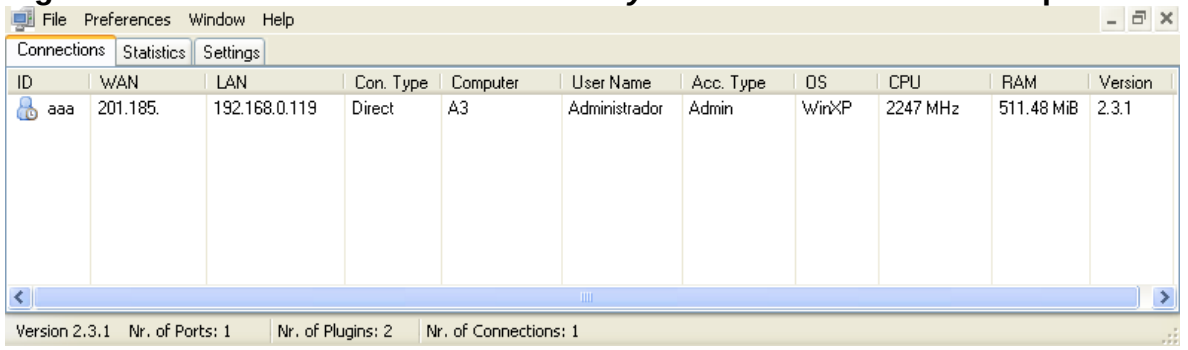
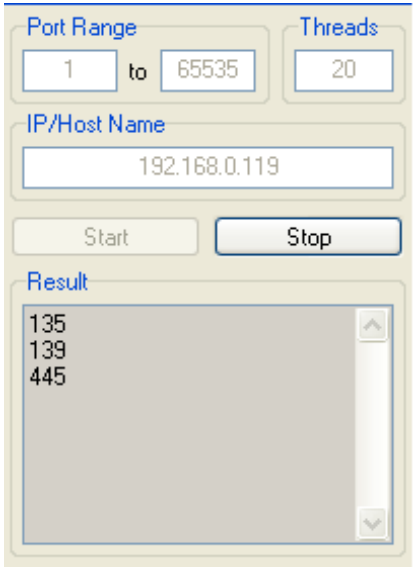
Parámetro	Prueba	Resultados y comentarios
	Figura 55. Ejecución de *Bifrost* como proceso del computador analizado por Prelude 	

Tabla 39. Pruebas de parámetros aplicadas a Prelude con el troyano *Poison Ivy* modificado

Troyano	**Poison Ivy**	
Parámetro	**Prueba**	**Resultados y comentarios**
Registros en el sistema	Reconocer instalaciones o desinstalaciones de programas como *utorrent*, *Winrar*, *Notepad* ++, y/o archivos como: *System.ini, Win.ini*	Registra cualquier instalación que se efectúa en el computador donde se encuentra el agente asociado a Prelude.
	Rule: 18147 (level 5) -> 'Application Installed.' *User: Administrador* *WinEvtLog: Application: INFORMATION(11707): MsiInstaller: Administrador: A3: A3: Producto: Google Earth -- La operación de instalación se ha completado correctamente.*	
Llaves de registro	Alterar/modificar y/o crear llaves de registro para el SO por medio del troyano.	Se creó por medio del troyano una llave de registro (ver figura 56) para el SO en donde el agente asociado realiza escaneos, pero el agente no reconoció la nueva llave de registro como una alerta.

<table>
<tr><th>Parámetro</th><th>Prueba</th><th>Resultados y comentarios</th></tr>
<tr><td></td><td colspan="2">Figura 56. Creación llave de registro por medio de Poison Ivy modificado no detectada por Prelude
 
</td></tr>
<tr><td rowspan="2">Conexiones exitosas hacia el agente</td><td>Realizar conexiones remotas desde el troyano al agente/cliente.</td><td>Se ejecutó el troyano en el computador víctima, la conexión fue exitosa (ver figura 57) pero el agente de Prelude no reconoció la conexión, por ende no emitió ninguna alerta.</td></tr>
<tr><td colspan="2">Figura 57. Conexión exitosa de Poison Ivy modificado no detectado por Prelude
 
</td></tr>
</table>

Parámetro	Prueba	Resultados y comentarios
Checksum	Alterar y/o modificar un archivo monitoreado por el IDS y/o modificar un archivo que sea importante para el SO como *win.ini*, y observar las acciones del IDS.	Se modificó un archivo que era escaneado por omisión en el IDS, también se modificó un archivo que no se encontraba por omisión, creándole la ruta en el archivo de configuración; para ambos archivos el IDS generó una alerta reconociendo los cambios en ellos.
Falsos positivos	Realizar actividades comunes de usuario tales como: programas de mensajería instantánea como *skype*, mirar correo electrónico como *gmail* y utilizar páginas de entretenimiento como *youtube* o *facebook* y observar si genera alertas	Al utilizar programas de mensajería instantánea como *skype,* Prelude emite una alerta. Al navegar por sitios web tales como *gmail*, y *youtube*, entre otros, el IDS no generó ningún tipo de alerta.
Ausencia de falsos negativos	Realizar actividades por medio del troyano que puedan comprometer el funcionamiento del SO y observar si el IDS genera alguna alerta.	La conexión establecida por el troyano con el cliente no fue detectada por el IDS, tampoco detectó escaneo de procesos y de puertos abiertos del computador.

Parámetro	Prueba	Resultados y comentarios
	Figura 58. Escaneo remoto de puertos con *Poison Ivy* modificado no detectado por Prelude 	
Detección de *rootkit*	Camuflar el troyano en un proceso legítimo del SO o en llaves de registro.	Se camufló el troyano como un proceso legítimo del SO y el agente asociado no reconoció dicho proceso como peligroso.

El IDS Prelude demostró la efectividad que posee al trabajar junto con otras herramientas IDS. Las detecciones frente a los troyanos modificados fue similar a las encontradas antes de ser modificados ... ver sección 7.4 ...

El IDS Prelude después de realizar las pruebas con los troyanos modificados se posicionó en la categoría 3 obteniendo una calificación de BUENO según la tabla rendimiento del IDS ante el uso de troyanos (ver tabla 9) con un puntaje promedio total de 65,32 (Ver tabla 40).

Tabla 40. Puntaje total obtenido por Prelude con los troyanos modificados

Parámetro	**Puntos obtenidos por Troyano**			
	Darkcomet	**Spy-Net**	**Poison Ivy**	**Bifrost**
Registros en el sistema	21,25	21,25	21,25	21,25
Llaves de registro	0	0	0	0
Conexiones exitosas hacia el agente/cliente	15	15	0	15
Checksum	15	15	15	15
Falsos positivos	0	0	0	0
Ausencia de falsos negativos	4,5	4,5	0	4,5
Detección de *rootkit*	0	0	0	0
Carpetas por omisión	6	6	6	6
Algoritmo *hash*	5	5	5	5
Frecuencia de escaneo	3	3	3	3
Valores obtenidos por Prelude	**69,75**	**69,75**	**52,05**	**69,75**

Los parámetros alcanzados por Prelude en las pruebas con los troyanos modificados:

- **Registros en el sistema:** Reconoció los diferentes tipos de instalaciones/desinstalaciones efectuadas al computador, enviando una alerta al servidor, pero no reconoció cambios en el inicio del SO. El puntaje que obtuvo fue 21,25 de 25 puntos posibles.
- **Conexiones exitosas hacia el agente cliente:** Reconoció y detecto que el troyano estaba presente en el computador. El puntaje que obtuvo fue el máximo: 15 puntos.
- ***Checksum*:** Reconoció cambios en archivos y/o carpetas por medio del checksum, para posteriormente enviar la alerta al servidor de OSSEC. El puntaje que obtuvo fue el máximo: 15 puntos.
- **Ausencia de falsos negativos:** Al poder detectar que un troyano realizó una conexión al computador analizado se logró reconocer que verdaderamente ocurrió un cambio significativo en el computador. El puntaje que obtuvo fue 4,5 de 9 puntos posibles.
- **Carpetas por omisión:** Está pre-configurado para escanear archivos y/o carpetas importantes para el SO. El puntaje que obtuvo fue el máximo: 6 puntos.
- **Algoritmo *hash*:** Posee dos algoritmos *hash* que son SHA1 y MD5 para monitorizar los archivos y/o carpetas. El puntaje que obtuvo fue el máximo: 5 puntos.
- **Frecuencia de escaneo:** El escaneo lo puede realizar constantemente y en tiempo real para registrar posibles intrusiones al computador, también permite configurar tiempo para escanear archivos y/o carpetas. El puntaje que obtuvo fue el máximo: 4 puntos.

Los parámetros no alcanzados por Prelude en las pruebas con los troyanos modificados:

- **Llaves de registro:** No logró reconocer cambios efectuados a las llaves de registro del SO. El puntaje que obtuvo fue 0 de 10 puntos posibles.
- **Falsos positivos:** Detectó el uso de programas de mensajería instantánea como alertas y descargas del correo electrónico como alertas. El puntaje que obtuvo fue 0 de 4 puntos posibles.
- **Detección de *rootkit*:** No detectó la forma de ocultarse el troyano en el SO. El puntaje que obtuvo fue 0 de 7 puntos posibles.
- **Conexiones exitosas hacia el agente/cliente:** Un solo troyano después de realizar las modificaciones no fue detectado por el IDS, el troyano fue Poison Ivy, por ello obtuvo 0 puntos de 15 puntos posibles solo para ese troyano.

Las diferencias de estas pruebas con las anteriores radicaron en el cambio del puerto de conexión del troyano, en el uso de contraseña para el cliente al momento de iniciar conexión con el servidor del troyano, aunque algunos troyanos poseían contraseñas por omisión y otros no poseían ninguna, para ambos casos las contraseñas fueron cambiadas o agregadas según el caso. Posteriormente el troyano fue cifrado con el *crpyter* modificado y no detectado por la herramienta antivirus logrando dejar indetectable también al troyano ante la herramienta antivirus.

Para el IDS fueron utilizadas las mismas reglas creadas en las pruebas anteriores logrando detectar la mayoría de los troyanos, demostrando el grado de confiabilidad de la herramienta IDS contra la herramienta antivirus, solamente uno de los troyanos llamado *Poison Ivy* logró evadir la herramienta IDS y antivirus, los demás troyanos solo evadieron la herramienta antivirus.

En la figura 59 se muestra la comparación por medio de un editor hexadecimal el troyano *Poison Ivy* detectado (1) y el troyano *Poison Ivy* no detectado (2) tanto por la herramienta antivirus cuanto la herramienta IDS. Se pueden evidenciar los cambios que se realizan al código, las partes de color verde representan el código

que se encuentra igual para ambos troyanos y el color amarillo indica las partes que son diferentes.

Adicionalmente, por medio del programa *Wireshak,* el mismo utilizado para crear las reglas para el IDS en las primeras pruebas, se puede observar el cambio generado al momento de conectarse el cliente y el servidor del troyano no detectado (ver figuras 60 y 61). El troyano genera una firma totalmente nueva no registrada en las reglas creadas para el IDS lo que le permite no ser detectado y de esta forma se logra efectuar la conexión del troyano al computador víctima sin generar sospecha ni alerta de la herramienta IDS.

Figura 59. Comparación hexadecimal del troyano *Poison Ivy* detectado y no detectado

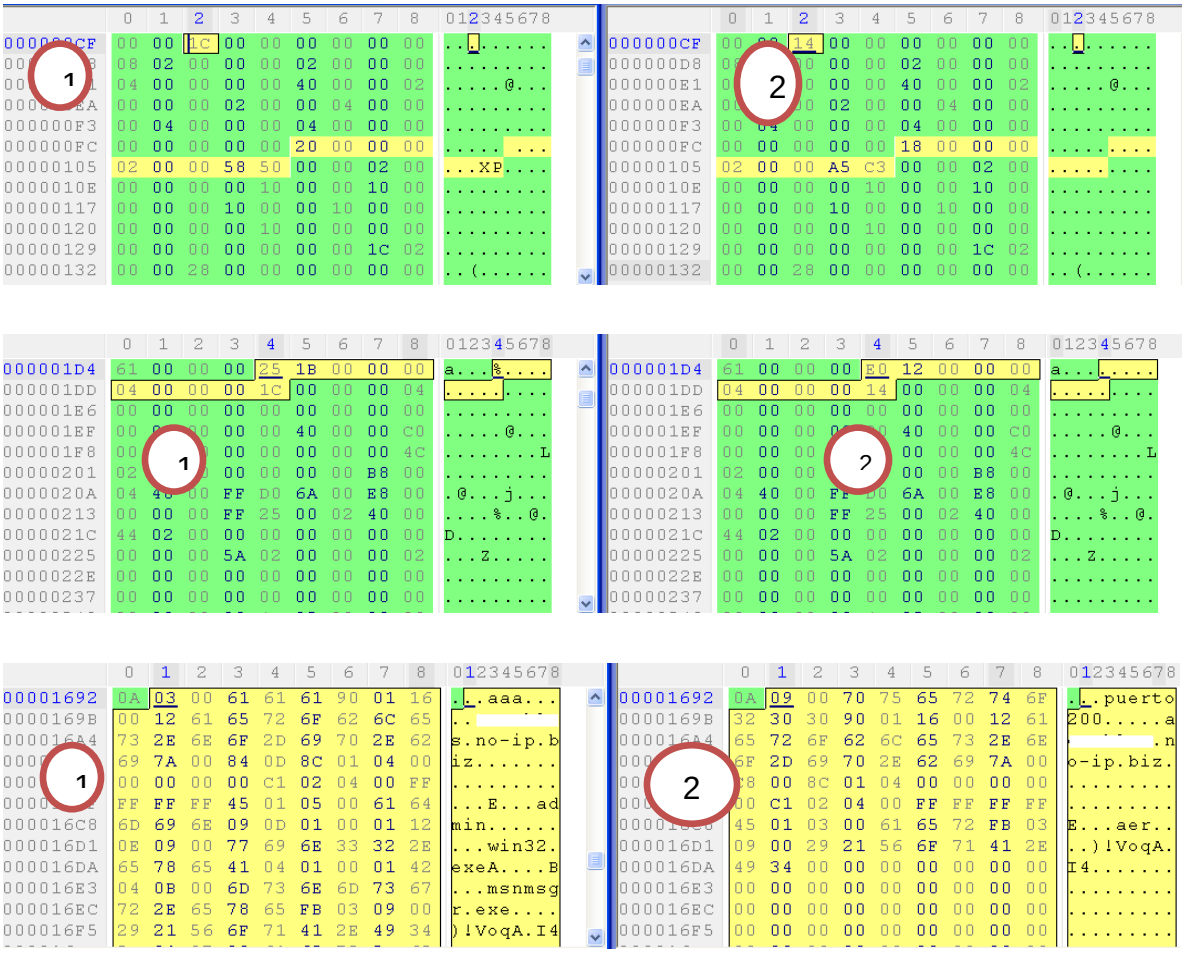


Figura 60. Firma del troyano *Poison Ivy* detectado por el IDS

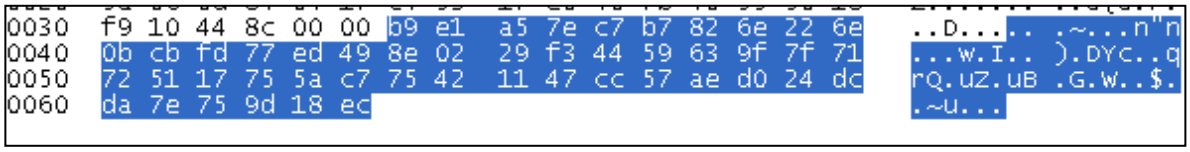
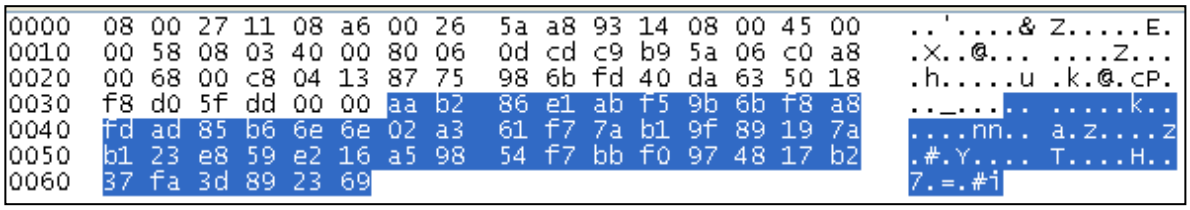


Figura 61. Firma del troyano *Poison Ivy* no detectado por el IDS



En conclusión, con la información obtenida del troyano se evidencian las razones por las cuales el IDS no logró detectarlo. La razón fundamental por la cual el troyano no fue detectado por el IDS no radica exclusivamente en el uso de un *crpyter,* puesto que los otros troyanos fueron detectados por el IDS aún siendo modificados por él, radica en el cambio de puerto y el uso de contraseña para la conexión entre el cliente y el servidor del troyano, debido a que al cambiar esas opciones que *Poison Ivy* posee por omisión se logra generar un flujo diferente en el código, generando una firma nueva y totalmente diferente, caso contrario con los otros troyanos analizados, que aún siendo modificados, cifrados y con puertos y contraseñas diferentes fueron detectados por el IDS.

CONCLUSIONES

En la actualidad el uso de herramientas informáticas para detección de ataques y/o intrusiones a sistemas de información son vitales en el área de seguridad informática, brindando un papel importante al momento de referirse de protección. El uso de herramientas informáticas tales como IDS y antivirus permiten obtener un nivel de seguridad parcial, razón por la cual se debe estar en constante investigación para el uso de nuevos métodos y herramientas de protección.

En el caso particular de intrusiones por medio de troyanos, se presentaron algunas vulnerabilidades que aún poseen tanto las herramientas IDS como los antivirus frente al empleo de este tipo de *malware*. A pesar que las herramientas IDS brindan mayor protección comparadas con las herramientas antivirus, las herramientas IDS son más complejas de administrar, y requieren de personal calificado para su uso y mantenimiento.

Así mismo, la efectividad de las herramientas IDS depende en gran parte del tipo de intrusiones que se deseen mitigar, por ello actualmente se están utilizando sistemas que permitan integrar varios IDS logrando unificar mensajes y alertas, obteniendo como resultado la efectividad que cada IDS posee. Al analizar los diferentes tipos de herramientas IDS empleadas en este proyecto se logró conocer la capacidad que tiene cada IDS para mitigar las intrusiones efectuadas por medio de troyanos, también se demostró la capacidad que tienen las herramientas para trabajar conjuntamente.

Los resultados de este proyecto, además de demostrar la importancia que ofrecen las herramientas IDS para la seguridad informática, permiten a conocer las debilidades y fortalezas que estas herramientas poseen para el caso específico de troyanos, dejando un precedente y una semilla para estudios posteriores que se

deseen desarrollar para mejorar la detección de troyanos o para mitigar otro tipo de ataques o intrusiones.

RECOMENDACIONES

Se recomienda profundizar en los métodos de conexión que realizan los troyanos al momento de comunicarse con su servidor, con el fin de poder mejorar las técnicas de detección con troyanos.

Por otro lado, se recomienda para futuros estudios de las herramientas de detección de intrusos realizar un análisis sobre intrusiones realizados por otro tipo *malware,* con el fin conocer y desarrollar métodos que permitan mitigar futuras intrusiones. Así mismo, explorar más a fondo las ventajas y desventajas que se obtienen al poder trabajar conjuntamente con diferentes herramientas IDS de forma hibrida.

De igual manera, se recomienda a la facultad de ingeniería informática de la Universidad Pontificia Bolivariana incluir dentro de la cátedra de seguridad informática el estudio y aplicación de las herramientas de detección de intrusos, debido al gran potencial de estas herramientas para detectar eventos ocurridos dentro de un computador.

REFERENCIAS

[1] J. Mieres.(2009). Ataques informáticos, debilidades de seguridad comúnmente explotadas. Evil Fingers. [Online]. Available: https://evilfingers.com/publications/white_AR/01_Ataques_informaticos.pdf

[2] RSA Security. (2008). Moderación de ataques con intermediarios y troyanos. [Online]. Available: http://www.rsa.com/products/consumer/whitepapers/9910_MITM_WP_0708_LE.pdf

[3] Netwitness. (2010, Enero). "Kneber Botnet" Targets Corporate Networks and Credentials. [Online]. Available: http://netwitness.com/about/press-releases/2010-netwitness-discovers-massive-zeus-compromise

[4] Panda Security.(2011, Junio). Informe trimestral PandaLabs. [Online]. Available: http://prensa.pandasecurity.com/wp-content/uploads/2011/07/Informe-PandaLabs-Q2-2011.pdf

[5] Microsoft. (2010, Enero). Documento informativo de seguridad. Microsoft TechNet. [Online]. Available: http://www.microsoft.com/latam/technet/seguridad/alerta/979352.mspx

[6] Panda Security. (2012).Informe Trimestral PandaLabs. [Online]. Available: http://press.pandasecurity.com/press-room/reports/

[7] Y. Namestnikov. (2011). Desarrollo de las amenazas informáticas en el segundo trimestre de 2011. [Online]. Available: http://www.viruslist.com/sp/analysis?pubid=207271138

[8] P. Aguilera. López. Seguridad informática. 1 Ed. EDITEX. 2010.

[9] R. Braginski. (2010, Febrero). Casi 2.000 computadoras Argentinas afectadas por un ciberataque global. [Online]. Available: http://edant.clarin.com/diario/2010/02/18/um/m-02142991.htm

[10] Microsoft. (2011, Junio). What Is the Security Intelligence Report. 11° vol. [Online]. Available: http://www.microsoft.com/security/sir/default.aspx

[11] Statcounter. (2011). Top 5 Operating Systems from January to December 2011. [Online]. Available: http://gs.statcounter.com/#os-ww-monthly-201101-201112

[12] A. Kostin (2011, Diciembre). Pagos online: Comididad y seguridad. [Online] http://www.viruslist.com/sp/hackers/analysis?pubid=207271154

[13] A. Gostev (2011, Febrero) Kaspersky Security Bulletin 2010: Boletín de seguridad Desarrollo de las amenazas informáticas en 2010 [Online] http://www.viruslist.com/sp/hackers/analysis?pubid=207271113

[14] The Information Assurance Technology Analysis Center (IATAC). (2009, Septiembre). Intrusion Detection Systems. N° 6. [Online]. Available: http://iac.dtic.mil/iatac/download/intrusion_detection.pdf

[15] Trend Micro. (2010). Getting started with OSSEC. [Online]. Available: http://www.ossec.net/main/getting-started-with-ossec

[16] B. Visscher. (2007). About Sguil. [Online]. Available: http://http://sguil.sourceforge.net/docs.html

[17] PRELUDE. (2005). [Online]. Available: http://www.prelude-technologies.com/en/welcome/index.html

[18] Sourcefire. (2010). About Snort. [Online].Available: http://www.snort.org/snort

[19] Darkcodersc Software. (2011) DarkComet Remote Administration Tool Project. [Online]. Available: http://www.darkcomet-rat.com/about

[20] Poison Ivy.(2006, Sptiembre). About Poison Ivy. [Online]. Available: http://www.poisonivy-rat.com/

[21] W. Stallings. (2005). Cryptography and Network Security Principles and Practices. 4ta ed. Prentice Hall. 744 p.

[22] K. Lai y D. Wren. (2010, Enero). Internet Security Products Performance Benchmarking. [Online]. Available: http://www.passmark.com/ftp/antivirus_10-performance-testing-ed4.pdf

[23] University of Michigan. (2005, Octubre). TOR (The Onion Router). DEPARTMENT OF LSAIT. [Online]. Available: http://lw.lsa.umich.edu/lsait/admin/TOR%20Routing%20Infomation%20.pdf

[24] J. A. Sáez. Muñoz. (2007). Malware. Universidad de Granada. [Online]. Available: http://lsi.ugr.es/~ig1/docis/aluwork/Malware.pdf

[25] A. Barrera. García-orea. (2010). Presente y futuro de los IDS. [Online]. Available: http://www.neurosecurity.com/whitepapers/futureIDS.pdf

[26] The Snort Preoject. (2011, Diciembre). Snort User Manual 2.9.2. [Online]. Available: http://www.snort.org/assets/166/snort_manual.pdf

[27] A. Hay, D. Cid y R. Bray. 1 Ed. OSSEC HIDS. *Host*-Based Intrusion Detection guide. Burlington: Syngress Publishing. 2008. Pp. 416.

[28] TCL Developers Xchange.(2009). Welcome to Tcl developers xchange. [Online]. Available: http://www.tcl.tk/about/

[29] R. Danyliw. (2010). The Analysis Console for Intrusion Databases. [Online]. Available: http://www.andrew.cmu.edu/user/rdanyliw/snort/snortacid.html

[30] Phplot Project. (2005). Getting Started with PHPlot. [Online]. Available: http://phplot.sourceforge.net/phplotdocs/starting.html

[31] J. Lim. (2011). ADOdb Database Abstraction Library for PHP. [Online]. Available: http://adodb.sourceforge.net/

[32] The PHP Group. (2012). What is PHP. [Online] . Avaliable: http://www.php.net/

[33] D. Burks. (2011, Julio). Security Onion. [Online]. Available: http://securityonion.blogspot.com/

[34] R. Gerhards. (2009, Agosto). The BSD syslog Protocol. [Online]. Available: http://tools.ietf.org/html/rfc5424

[35] A. Gómez Vieites. Enciclopedia de la Seguridad Informática. 2 Ed. Ra-Ma. 2006, 828 p.

[36] Trend Micro. (2010). OSSEC Architecture. [Online]. Available: http://www.ossec.net/main/ossec-architecture

[37] H. Debear, D. Curry, B. Feinstein. (2007, Marzo) The intrusion Detection Message Exchange Format (IDMEF) [Online]. Avaliable: http://www.ietf.org/rfc/rfc4765.txt

[38] J.M. Aquilina and E Casey *Malware Forensics: Investigating and Analyzing Malicious Code.* United States of America: Syngress, 2008, pp. 340-342