UNIVERSIDAD NACIONAL DE COLOMBIA

Diseño e implementación del mecanismo para la producción de registros electrónicos y evidencia digital de los incidentes en redes inalámbricas 802.11x

Ing. Oscar Javier Moreno Delgado

Universidad Nacional de Colombia

Departamento de Ingeniería de Sistemas e Industrial

Bogotá, Colombia

2013

Diseño e implementación del mecanismo para la producción de registros electrónicos y evidencia digital de los incidentes en redes inalámbricas 802.11x

Ing.Oscar Javier Moreno Delgado

Tesis o trabajo de investigación presentada como requisito parcial para optar al título de:

Magister en Ingeniería de Telecomunicaciones

Director (a):

Ing. Ingrid Patricia Páez Parra, PhD

Línea de Investigación:

Redes y Sistemas de Telecomunicaciones

Grupo de Investigación:

Grupo de Investigación en Teleinformática de la Universidad Nacional de Colombia - GITUN

Universidad Nacional de Colombia

Departamento de Ingeniería de Sistemas e Industrial

Bogotá, Colombia

2013

Nota de Aceptación

_____

_____

_____

_____

Directora

_____

Jurado

Bogotá D.C., 2013

(Dedicatoria o lema)

A mis padres que lo dieron todo por Mi, especialmente - a mi madre que me acompaña siempre.

A Yury, mi esposa, compañera y sobretodo amiga, que siempre ha creído en mí y me ha apoyado en todos los malos momentos, sin pedir nada a cambio.

A mis hijos, Camila y Anthony, que dan Sentido a mi vida y que siempre están y estarán en mi corazón con su amor.

Agradecimientos

Deseo expresar mi más sincero agradecimiento a la doctora Ph.D., Ingeniería de Telecomunicaciones, Ingrid Patricia Páez Parra, quien además de transmitirme su vocación investigadora, me oriento, ayudo y estimulo constantemente y directamente en todos los aspectos de la tesis durante la ejecución de la misma. Agradecerle la plena confianza que siempre me ha demostrado, así como la dedicación y la atención que en todo momento me ha ofrecido.

A mis compañeros y Profesores de la Universidad Nacional de Colombia de la Maestría de Telecomunicaciones, por su colaboración durante mi estadía en la universidad.

A todos los participantes que contribuyeron de una u otra forma para que yo pudiese terminar mi tesis a cabalidad.

Declaración

Me permito afirmar que he realizado la presente tesis de manera autónoma y con la única ayuda de los medios permitidos y no diferentes a los mencionados en la propia tesis. Todos los pasajes que se han tomado de manera textual o figurativa de textos publicados y no publicados, los he reconocido en el presente trabajo.

Bogotá, D.C., 17.12.2013

_____

Oscar Javier Moreno Delgado

Resumen

El siguiente proyecto presenta una solución a la inexistencia de un procedimiento eficiente y eficaz que ayude a manejar los incidentes en la red inalámbrica 802.11, la cual consiste en construir una arquitectura para la producción de evidencia digital, desarrollando un diseño e implementación para registrar eventos en forma de registros electrónicos, fortaleciendo la admisibilidad y relevancia de los mismos, es decir, que la información obtenida (evidencia digital), sea confiable e integra, de tal forma que pueda ser útil para esclarecer un futuro incidente informático.

Palabras Claves: Redes inalámbricas, evidencia digital, logs, estampado cronológico, hash.

Abstract

The following project presents a solution to the lack of an efficient and effective method to help manage the incidents on a 802.11 wireless network, which consists in constructing an architecture for the production of digital evidence, developing a design and implementation to record events in the form of electronic records, strengthening the relevance and admissibility thereof, this means that the information obtained (digital evidence), is reliable and integrated, so that it can be useful to clarify the future data incidents.

Keywords: Wireless networks, digital evidence, logs, chronological pattern, hash.

Contenido

Pág.

Agradecimientos ..... **VII**

Resumen ..... **IX**

Lista de figuras ..... **XIII**

Lista de Símbolos y abreviaturas **2**

Introducción **6**

1. Estado del Arte **9**
1.1 Definiciones 9
1.2 Historia 20
1.3 Metodología 26
1.4 Requerimientos Legales 27

2. Parámetros y limitaciones **28**
2.1 Parámetros mínimos de hardware y software 28
- Creación de usuario para conexión SSH clientes 32
2.2 Limitaciones 33

3. Diseño **37**
3.1 Arquitectura 37
3.1.1 Diagrama de arquitectura. 37
3.1.2 Diagrama de flujo. 38
3.2 Construcción de Snort para Windows. 39
3.3 Administración y creación de Scrip para Snort 43

4. Implementación **46**
- Despliegue de HashStamp 46
- Configuración de clientes Windows 47
4.1 Crear evidencia de un directorio: 52
4.2 Verificar evidencia de un directorio: 54

5. Validación y análisis de resultados **57**
5.1 Caso con Logs predeterminados 58
5.2 Caso con la solución propuesta 60

6. Conclusiones y recomendaciones **73**
6.1 Conclusiones 73

6.2 Recomendaciones 74
6.3 Trabajos futuros 74

7. Bibliografía 75

Lista de figuras

Pág.

Figura 1-1.: Actividad de tratamiento de riesgos [28]. 22
Figura 2-1.: Instalación de Snort en Linux Debian. 28
Figura 2-2.: Configuración de Snort. 28
Figura 2-3.: Creación de directorios para almacenar evidencia. 29
Figura 2-4.: Instalación de SSH. 29
Figura 2-5.: Permisos para directorios contenedores de evidencia. 29
Figura 2-6.: Descarga de JRE en Debian [41]. 29
Figura 2-7.: Creación de directorio java y descompresión del mismo. 30
Figura 2-8.: Edición de archivo bashrc. 30
Figura 2-9.: Instalación certificado DigiStamp 30
Figura 2-10.: Descarga de Snort 31
Figura 2-11.: Reglas Snort 31
Figura 2-12.: Creación de usuario para conexión ssh clientes. 32
Figura 2-13.: Instalación de WinPcap 32
Figura 2-14.: Instalación de SSHFS. 32
Figura 2-15.: Configuración y montaje de SSHFS. 33
Figura 2-16.: Disco de archivos de logs. 33
Figura 2-17.: Carna Logs – Nmap. 34
Figura 2-18.: Metasploit [47]. 35
Figura 3-1.: Diagrama de Arquitectura. 37
Figura 3-2.: Diagrama de Flujo. 38
Figura 3-3.: Extracción de reglas de Snort 39
Figura 3-4.: Listado de reglas de Snort 39
Figura 3-5.: Ejecución de Snort usando archivo de reglas 39
Figura 3-6.: Edición de archivo snort.conf 40
Figura 3-7.: Edición de archivo snort.conf y cambio de ruta de reglas para Windows 40
Figura 3-8.: Reporte de errores de reglas de Snort 41
Figura 3-9.: Edición de archivo snort.conf y se construye regla que contiene error. .. 41
Figura 3-10.: Reporte de errores de reglas de Snort. 41
Figura 3-11.: Re-direccionar los archivos logs a un directorio predeterminado. 42
Figura 3-12.: Validación exitosa de reglas de Snort. 42
Figura 3-13.: Listado de interfaces. 42
Figura 3-14.: Script que permiten administrar Snort. 43

Figura 3-15.: Instalación del Servicio Snort. 43
Figura 3-16.: Iniciar servicio Snort. 44
Figura 3-17.: Detener Servicio Snort. 44
Figura 3-18.: Desinstalar Servicio Snort. 45
Figura 3-19.: Verificación de errores de construcción reglas de Snort. 45
Figura 4-1.: Certificado DigiStamp. 46
Figura 4-2.: Solicitud de password. 46
Figura 4-3.: Borrado de certificado de base de datos. 47
Figura 4-4.: Selección de directorio y extracción de Snort. 47
Figura 4-5.: Detener servicio de Snort. 47
Figura 4-6.: Desinstalar servicio Snort. 48
Figura 4-7.: Modificación de ruta de grabación de archivos de evidencia. 48
Figura 4-8.: Instalar servicio Snort. 49
Figura 4-9.: Selección de la interfaz de red donde escuchará Snort. 49
Figura 4-10.: Iniciar servicio Snort. 49
Figura 4-11.: Directorio de incidentes registrados. 50
Figura 4-12.: Verificación de errores de construcción de reglas de Snort. 51
Figura 4-13.: Reglas compiladas de Snort. 51
Figura 4-14.: Ejecución del programa HashStamp [49]. 52
Figura 4-15.: Ventana del programa HashStamp. 52
Figura 4-16.: Usuario y password TimeStamping 53
Figura 4-17.: Selección de directorio a estampar y directorio de evidencia destino. 53
Figura 4-18.: Procesar evidencia 53
Figura 4-19.: Mensaje emergente de estampado exitoso. 54
Figura 4-20.: Directorio con hash y estampas de evidencia. 54
Figura 4-21.: Ventana de selección de archivos hash y estampado (HashStamp). 55
Figura 4-22.: Archivo hash a verificar. 55
Figura 4-23.: Archivo estampado cronológico (TimeStamping) a comparar. 55
Figura 4-24.: Proceso de validación exitosa. 56
Figura 4-25.: Proceso de verificación no exitosa. 56
Figura 5-1.: Escaneo de puertos Zenmap. 58
Figura 5-2.: Búsqueda de incidente en Log de Windows. 59
Figura 5-3.: Ejecución de Metasploit. 59
Figura 5-4.: Vulnerabilidad MS08-067 [52]. 59
Figura 5-5.: Búsqueda de incidente metasploit en Log de Windows. 60
Figura 5-6.: Escaneo de puertos mediante Zenmap [53]. 61
Figura 5-7.: Archivos tcpdump.log y alert.ids. 61
Figura 5-8.: Evidencia obtenida en log alert.ids 62
Figura 5-9.: Vulnerabilidad MS08-067 [52]. 62
Figura 5-10.: Metasploit ejecutado con éxito. 63
Figura 5-11.: Evidencia en log de ataque metasploit. 63
Figura 5-12.: Evidencia de log de ataque metasploit mediante Wireshark 63
Figura 5-13.: Visor de eventos en Windows. 67
Figura 5-14.: Registro de windows 67

Figura 5-15.: Hash en windows 68
Figura 5-16.: Logs en windows 68
Figura 5-17.: Logs en Windows y su alteración 69
Figura 5-18.: Logs de eventos en Linux 69
Figura 5-19.: Comparación de los principales algoritmos [59]. 70
Figura 5-20.: Estadística de utilización de estampado cronológico [60]. 71
Figura 5-21.: Entidades que prestan el servicio de estampado cronológico [61]. 72

Lista de tablas

Pág.

Tabla 1: Escala de Probabilidad y detección [29] 21
Tabla 2: Comparación de casos propuestos..... 65

Lista de Símbolos y abreviaturas

Abreviatura	**Término**
Snort.conf	Archivo de configuración del programa para reporte de incidentes
7s	Archivador de ficheros libre desarrollado por Igor Pavlov
AFS Kerbeos	AFS incluye su propia implementación de Kerberos, el kaserver, versión 4.
Alert.ids	Archivo generado por Snort de incidentes
arp	El Address Resolution Protocol (protocolo de resolución de direcciones).
BackOrifice	Programa de control remoto de ordenadores que funciona bajo un servidor y un cliente.
Cer	Extensión de archivo de DigiStamp
CGI	*Common Gateway Interface:* Tecnología que se usa en los servidores web.
Cifrado	Es un método que permite aumentar la seguridad de un mensaje o de un archivo mediante la codificación del contenido, de manera que sólo pueda leerlo la persona que cuente con la clave de cifrado adecuada para descodificarlo.
DigiStamp	Instituto Nacional de estándares y tecnología
DNS	Domain Name System o DNS: Es un sistema de nomenclatura jerárquica para computadoras, servicios o cualquier recurso conectado a Internet o a una red privada.

ethernet	Estándar de redes de área localpara computadores con acceso al medio por detección de la onda portadora y con detección de colisiones (CSMA/CD).
fddi	Interfaz de Datos Distribuida por Fibra (FDDI: Fiber Distributed Data Interface).
FileSlake	Espacio de almacenamiento de datos que existe desde el final del archivo hasta el final de la última clúster asignado al archivo.
ftp	File Transfer Protocol. Protocolo de Transferencia de Archivos.
GUI	Es un programa informático que actúa de interfaz de usuario, utilizando un conjunto de imágenes y objetos gráficos para representar la información y acciones disponibles en la interfaz.
Hash	Un hash es un algoritmo criptográfico para generar clave en orden unidireccional
Http	Hypertext Transfer Protocol o HTTP. Protocolo de transferencia de hipertexto.
Https	Hypertext Transfer Protocol Secure, más conocido por sus siglas HTTPS, es un protocolo de aplicación basado en el protocolo HTTP, destinado a la transferencia segura de datos de Hipertexto.
Icmp	Protocolo de Mensajería de control de Internet.
Ids	Sistema de detección de intrusos.
imap	Permite acceder a varios clientes al mismo buzón, facilitando el acceso posterior a los mensajes de correo disponibles en el servidor mediante correo web.
Jar	Extensión de archivo de lenguaje Java
Lat, , moprc, mopdl	Abreviatura para ether proto p, donde p es uno de los protocolos anteriores a ellos.
Máquina de *turing*	Es un dispositivo que manipula símbolos sobre una tira de cinta de acuerdo a una tabla de reglas.

Metasploit	Tests de intrusión en redes o en servidores
Mismidad	Datos obtenidos iguales a los presentados
Nmap	Programa de rastreo de puertos
OpenBSD	Es un sistema operativo libre tipo Unix multiplataforma, basado en 4.4BSD. Es un descendiente de NetBSD, con un foco especial en la seguridad y la criptografía.
OpenWrt	Es una distribución de Linux basada en firmware usada para dispositivos empotrados tales como routers personales.
P7s	Extensión de archivos de compresión
Parches	En informática, un parche consta de cambios que se aplican a un programa, para corregir errores, agregarle funcionalidad, actualizarlo, etc.
Payload	Genera ejecutable a partir de una carga útil de Metasploit.
Pcap	Interfaz de programación de aplicaciones
Pgp	Proteger información distribuida a través de internet.
Pkt	Cisco Packer Tracer
pop3	Descarga los mensajes eliminándolos del servidor. Los mensajes de correo electrónico ya no se encuentran disponibles por correo web o un programa de correo.
rarp	Son las siglas en inglés de Reverse Address Resolution Protocol (Protocolo de resolución de direcciones inverso).
Rules	Reglas utilizadas en Snort
Service Pack	Consisten en un grupo de actualizaciones que corrigen y mejoran aplicaciones y sistemas operativos.
Sha	Extensión de archivo del Hash sha512

SMB	Server Message Block. Es Un protocolo que pertenece que permite compartir Archivos e Impresoras (entre Otras Cosas). Es utilizado principalmente en computadores con sistema operativo Microsoft Windows y DOS.
smtp	Simple Mail Transfer Protocol. (SMTP) Protocolo para la transferencia simple de correo electrónico.
Sniffer	Herramienta que detecta las conexiones de otras pc en tu red LAN y se utiliza frecuentemente para test de seguridad y penetración.
Snort	Software de sistema de detección de intrusos.
Spammers	Sujeto o persona que hace Spam.
Tcp	Protocolo de Control de Transmisión.
Tcpdump.log	Archivo generado por Snort de incidentes para ser graficados por una analizador
telnet	Telecommunication Network. Es el nombre de un protocolo de red que nos permite viajar a otra máquina para manejarla remotamente.
TimeStamping	Estampado cronológico
Traceroute	Es una herramienta de diagnóstico para mostrar el camino de los paquetes en una red IP y su retardo en tránsito.
Tsa	Autoridad de sellado de tiempo (extensión de archivos al ser estampados)
Tsr	Extensión de sello de tiempo de respuesta
Tunel	Se conoce como túnel o tunneling a la técnica que consiste en encapsular un protocolo de red sobre otro (protocolo de red encapsulador) creando un túnel de información dentro de una red de computadoras.
Udp	Protocolo de datagramas.
Wireshark	Analizador de Protocolos
X.509	Infraestructuras de claves públicas
Zip	Compresión de archivos

Introducción

Actualmente no existe ningún procedimiento eficiente y eficaz que ayude a manejar los incidentes en la red inalámbrica por falta de un mecanismo que brinde confianza, ya que los métodos existentes no logran obtener datos confiables; los cuales, al ser llevados a un tribunal, cumplan con un papel importante dentro de la investigación y procesamiento de los delincuentes informáticos ya que la judicialización, depende únicamente de la autenticidad en los mismos.

Es por esta razón que en la presente investigación, se diseñó e implementó un mecanismo que cumple los requisitos de integridad, figura [4-20], de los datos, fortaleciendo la admisibilidad y relevancia de los mismos con su respectivo estampado cronológico otorgado por una entidad certificadora figura [4-24], para dar cumplimiento a la ley de datos informáticos, capitulo [1], parágrafo 1.4, [1], [2], permitiendo que un futuro incidente sea parte de un juicio.

En la primera parte, se hace referencia al estado del arte, la cual contiene definiciones, capitulo [1], historia, capitulo [1], parágrafo 2, y la metodología, capitulo [1], parágrafo 3, de cadena de custodia, como también los lineamientos de la legislación colombiana sobre delitos informáticos capitulo [1], parágrafo 4.

En el capítulo 2, se presentan los parámetros mínimos de hardware y software del servidor parágrafo 2.1, a, como de los clientes, parágrafo 2.1, b, explicando su respectiva instalación, los requisitos de construcción del instalador Snort, las limitaciones, parágrafo 2.2, que permiten verificar el mecanismo planteado mediante un escaneo de puertos (Nmap), figura [2-17], y un metasploit, figura [2-18], que permite control total del equipo identificado como víctima.

El escaneo de puertos se realiza mediante el aplicativo Nmap, la cual es considerada la herramienta más eficiente y popular a la hora de mostrar vulnerabilidades en una máquina determinada [3], en la figura [5-2], se presentan las vulnerabilidades por las cuales posteriormente son aprovechadas mediante un metaexploit, figura [5-4], que nos dará el control total de la máquina, en este caso denominada víctima.

En el capítulo 3, se presenta el diseño con el diagrama de arquitectura, parágrafo 3.1.1, y el diagrama de flujo, parágrafo 3.1.2, a su vez, se explica la creación de reglas de Snort, parágrafo 3.2, y la creación de cinco scrip, parágrafo 3.3, que consisten en instalar, figura [3-15], iniciar, figura [3-16], detener, figura [3-17], desinstalar, figura [3-18], y verificación de errores de construcción de las reglas de snort, figura [3-17]. Se construye el instalador del snort para los sistemas operativos Windows y linux.

Es importante resaltar que estos scripts fueron creados ya que con ellos se evita de forma manual subir los servicios de snort y la verificación de reglas que permitirán la detección de incidentes, figura [3-14].

En el capítulo 4, se realiza la implementación que consiste en la recolección de evidencia de incidentes mediante dos ataques: Zenmap, figura [5-6], y MS08-067, figura [5-9], para los casos con logs predeterminados, capítulo 5, parágrafo 5,1 y con la solución propuesta, capítulo 5, parágrafo 5,2; para este último, se crea la evidencia en un directorio con el hash y la estampa, figura [4-20], y su proceso de verificación exitoso, figura [4-24], garantizando la autenticidad de los mismos.

En el capítulo 5, se valida el diseño e implementación de la solución planteada donde se muestra un atacante haciendo escaneo de puertos con Zenmap, figura [5-6], y un ataque informático de control total a la maquina víctima, figura [5-9], siendo este registrado únicamente con los logs predeterminados del sistema operativo, capítulo 5, parágrafo 5.1, desde la figura [5-13] a la figura [5-18].

A su vez, se presenta el mismo ataque registrado anteriormente, pero en esta ocasión demostrando que el atacante fue registrado mediante la solución planteada, la cual se logra evidenciar en los archivos tcpdump.log y alert.ids, figura[5-7], la captura del ataque, figura [5-8] y figura [5-12], con la solución planteada, capítulo 5, parágrafo 5.2.

Por otra parte, se hace un cuadro comparativo de los principales algoritmos que están siendo utilizados actualmente frente al algoritmo propuesto, SHA-512, En la anterior gráfica, se puede determinar que el SHA-512, que genera el aplicativo, permite evitar ataques de fuerza bruta para descifrar el algoritmo, además de evitar ataques de colisiones. Figura [5-19].

Encontramos igualmente los resultados y análisis de resultados del diseño e implementación del mecanismo para la producción de registros electrónicos y evidencia digital de incidentes mediante una tabla de comparación de casos propuestos, Capítulo 5, Tabla 2, mostrando los aportes del proyecto.

Finalmente, se muestran dos gráficas referentes a la estadística de utilización de estampado cronológico, figura [5-20], y las entidades que prestan el servicio de estampado cronológico, figura [5-21].

En el capítulo 6, son presentadas las conclusiones del proyecto realizado, las cuales sirven como punto de partida y referencia para futuras investigaciones realizadas con los temas tratados en la tesis.

1. Estado del Arte

1.1 Definiciones

Se definen dos conceptos que ayudan a contextualizar el trabajo de investigación tales como: registros electrónicos y evidencia digital.

1.1.1. Registros electrónicos: son un tipo de evidencia física que está construida de campos magnéticos y pulsos electrónicos que pueden ser recolectados y analizados con herramientas y técnicas especiales [1].

1.1.2. **Evidencia digital:** Se define evidencia digital a los datos que constan en formato electrónico y que constituyen elementos de prueba, comprendiendo las etapas de extracción, procesamiento e interpretación [2].

1.1.3. **IDS:** Sistema de detección de intrusos. Implementa un motor de detección de ataques y barrido de puertos que permite registrar, alertar y responder ante cualquier anomalía previamente definida como patrones que corresponden a ataques, barridos, intentos aprovechar alguna vulnerabilidad, análisis de protocolos, conocidos. Todo en tiempo real, más detalles en [4, 5].

1.1.4. **Snort:** Es un sistema de código de red de prevención y detección de intrusiones abierta (IPS - IDS) desarrollado por Sourcefire. La combinación de los beneficios de la firma, el protocolo y la inspección basada en anomalías [6].

1.1.5. **Estampado cronológico:** Estampado cronológico. Es sellado de tiempo confiable del proceso que se lleva de manera segura en el tiempo, tanto de la creación como de la modificación de un documento electrónico. La seguridad aquí

significa que nadie, ni siquiera el dueño del documento, debe ser capaz de cambiarlo una vez que ha sido guardado, debido a que la integridad de aquel que realizó el sellado de tiempo nunca debe ser comprometida.

La parte administrativa involucra poner en marcha una infraestructura de sellado de tiempo confiable disponible públicamente para obtener, procesar y renovar sellados de tiempo.

Las estampas expiran después de un cierto período de tiempo, como un año, y un documento firmado con una clave caducada no debe ser aceptado. Sin embargo, hay muchos casos en que sea necesario para los documentos firmados que se consideran legalmente válido por mucho más tiempo que la validez del certificado; arrendamientos y contratos a largo plazo son algunos ejemplos. Al registrar el contrato con una autoridad de sellado de tiempo digital (TSA) en el momento de su firma, la firma se puede validar, incluso después de que expire la clave.

Si todas las partes en el contrato guardan una copia del sello de tiempo, cada uno puede probar que el contrato fue firmado con claves válidas. Muy importante es el hecho de que el sello de tiempo puede probar la validez de un contrato, incluso si la clave de uno de los firmantes es comprometida en algún momento después de la firma del contrato [7].

1.1.6. **Hash o función resumen:** Los algoritmos Hash, o de resumen, se constituyen un tipo especial de criptosistemas. Estos, a diferencia de los algoritmos simétricos o asimétricos, no utilizan el concepto de clave. Para estos algoritmos existe un nuevo término llamado: fingerprint o huella digital o resumen o hash [8].

Una función Hash toma un mensaje de entrada de longitud arbitraria y genera un código de longitud fija. La salida de longitud fija se denomina hash del mensaje original.

Los criptosistemas Hash presentan las siguientes características:

- **Unidireccionalidad:** este concepto significa que deberá ser computacionalmente muy difícil, por no decir imposible, obtener el mensaje M (original).
- **Compresión:** a partir de un mensaje de cualquier longitud, el hash H(M) debe tener una longitud fija. Normalmente mucho menor.
- **Coherente:** la misma entrada (mensaje original) siempre deberá producir la misma salida (mensaje original).
- **Facilidad de Cálculo:** debe ser fácil calcular la función Hash H(M) a partir de un mensaje M.
- **Único:** Imposible encontrar dos mensajes que generen el mismo hash.
- **Difusión:** el resumen H(M) debe ser una función compleja de todos los bits del mensaje M.

Principales algoritmos criptográficos de tipo hash:

a) **MD5** (Message Digest Algorithm 5, Algoritmo de Ordenación de Mensajes 5): es un algoritmo desarrollado por RSA Data Security, Inc. MD5 es una función hash de 128 bits, que toma como entrada un mensaje de cualquier tamaño y produce como salida un resumen del mensaje de 128 bits.

b) **SHA** (Secure Hash Algorithm): la familia SHA (Secure Hash Algorithm, Algoritmo de Hash Seguro) es un sistema de funciones hash criptográficas relacionadas de la Agencia de Seguridad Nacional de los Estados Unidos y publicadas por el National Institute of Standards and Technology (NIST).

c) **SHA-1** ha sido examinado muy de cerca por la comunidad criptográfica, y no se ha encontrado ningún ataque efectivo. No obstante, en el año 2004, un número de ataques significativos fueron divulgados sobre funciones criptográficas de hash con una estructura similar a SHA-1; esto ha planteado dudas sobre la seguridad a largo plazo de SHA-1.

d) **SHA-0 y SHA-1** producen una salida resumen de 160 bits de un mensaje, que puede tener un tamaño máximo de 264 bits, y se basa en principios similares a los usados MD5.

e) **SHA-2** produce una salida resumen de 256 (para SHA-256) o 512 (para SHA-512) y difiere a SHA-1 en que el algoritmo contempla algunas constante adicionales; así mismo, el tamaño del resumen es diferente al igual que el número de rondas.

1.1.7. **ISO/IEC 27005:** La norma ISO/IEC 27005:2008 es una guía para la gestión de riesgos de seguridad de la información, de acuerdo con los principios ya definidos en otras normas de la serie 27000.

Sustituye (y actualiza) las partes 3 y 4 de la norma ISO TR 13335 (Técnicas para la gestión de la seguridad IT y Selección de salvaguardas, respectivamente) y se convierte en la guía principal para el desarrollo de las actividades de análisis y tratamiento de riesgos en el contexto de un SGSI. Constituye, por tanto, una ampliación del apartado 4.2.1 de la norma ISO 27001, en el que se presenta la gestión de riesgos como la piedra angular de un SGSI, pero sin prever una metodología específica para ello [9].

1.1.8. **UTF-8:** *8-bit Unicode Transformation Format*, es un formato de codificación de caracteres Unicode e ISO 10646, la cual utiliza símbolos de longitud variable. Está definido como estándar por la RFC 3629, de la Internet Engineering Task Force (IETF) [10]. Actualmente es una de las tres posibilidades de codificación reconocidas por Unicode y lenguajes web [11].

Sus características principales son:

- Es capaz de representar cualquier carácter Unicode.
- Usa símbolos de longitud variable (de 1 a 4 bytes por carácter Unicode).
- Incluye la especificación US-ASCII de 7 bits, por lo que cualquier mensaje ASCII se representa sin cambios.
- Incluye sincronía. Es posible determinar el inicio de cada símbolo sin reiniciar la lectura desde el principio de la comunicación.

- No superposición. Los conjuntos de valores que puede tomar cada byte de un carácter multibyte, son disjuntos, por lo que no es posible confundirlos entre sí.

1.1.9. **Salt:** Una salt, se utiliza normalmente para almacenar hashes de contraseñas de forma segura (tal que no puede ser leído por otros). Cuando se agrega una cadena aleatoria a la contraseña, esto la vuelve mucho más difícil. El cálculo de este nuevo hash se hace de la siguiente forma:

Código en JAVA:

```
import java.security.MessageDigest;

public byte[ ] getHash(String password) throws NoSuchAlgorithmException {
    MessageDigest digest = MessageDigest.getInstance("SHA-2");
    digest.reset();
    byte[ ] input = digest.digest(password.getBytes("UTF-8"));
}
```

1.1.10. **JRE:** Es el encargado del consumo de memoria de los objetos y no el compilador, ya que en Java no hay punteros sino referencias a objetos, el código compilado contiene identificadores sobre los objetos que luego el JRE traduce en direcciones de memoria.

En el momento de ejecutar una aplicación JAVA, el JRE utiliza un proceso llamado class loader que realiza la carga del lenguaje contenido en las clases JAVA [12].

1.1.11. **Digistamp:** De acuerdo al estándar RFC 3161, un sellado de tiempo confiable es un Sellado de tiempo emitido por un Tercero Confiable (Digistamp), que actúa como una autoridad de sellado de tiempo (TSA). Es usado para demostrar la existencia de alguna información antes de cierta fecha (ej. contratos, información de investigación, registros médicos, etc.) sin la posibilidad de que el dueño pueda cambiar el sellado de tiempo. Múltiples TSA, pueden ser usadas para aumentar la confiabilidad y reducir la vulnerabilidad del sistema.

1.1.12. **WinPcap:** Es la herramienta que permite acceder a la conexión entre capas de red en entornos Windows. Permite a las aplicaciones capturar y trasmitir los paquetes de red puenteando la pila de protocolos; y tiene útiles características adicionales que incluyen el filtrado de paquetes a nivel del núcleo, un motor de generación de estadísticas de red y soporte para captura de paquetes.

Por otra parte, consiste en un controlador, que extiende el sistema operativo para proveer acceso de red a bajo nivel, y una biblioteca que se usa para acceder fácilmente a las capas de red de bajo nivel. Esta biblioteca también contiene la versión de Windows de la bien conocida API de Unix, libpcap [13].

1.1.13. **SSHFS:** *Secure SHell FileSystem (SSHFS),* (interprete de ordenes seguras), es un protocolo que nos sirve para acceder a una maquina remota a través de la red y estando en otra máquina, parecer que estamos en ella. Un servidor SSH escucha por defecto en el puerto 22; trabaja de forma parecida a telnet, pero lo hace de forma cifrada, por lo que si interceptan el mensaje y no tienen la contraseña no podrán leer el contenido.

Permite copiar datos de forma segura (tanto ficheros sueltos como simular sesiones FTP cifradas), gestionar claves RSA para no escribir claves al conectar a los dispositivos y pasar los datos de cualquier otra aplicación por un canal seguro tunelizado mediante SSH. Se utiliza en sistema de archivos linux (y otros sistemas operativos con una implementación FUSE, tal como en Mac OS X), que opera sobre archivos en una computadora remota usando un entorno seguro de acceso como si los tuviéramos en nuestro pc, gracias al sistemas de archivos en espacio de usuario fuse(*Filesystem in User Space*) y usando un entorno seguro gracias al protocolo SSH [14].

1.1.14. RSA: Es un sistema de cifrado que usa un algoritmo imposible de solucionar a día de hoy. Se creé que con la llegada de sistemas cuánticos y su rapidez se puedan llegar a solucionar.

Es una pareja de claves ya que crean a pares y de forma que una no sirve sin la otra, una clave es privada y la otra pública [15].

- **Clave privada:**

Se usa para descifrar y es la que debe estar solo presente en la máquina cliente y protegerla.

- **Clave pública:**

Se usa para cifrar y se puede repartir sin temor a que con ella podamos descifrar lo que va por el medio.

Esta clave solo cifra y no puede usarse para descifrar ni lo que se haya cifrado con ella.

Solo la clave privada asociada a ella puede descifrar lo que se haya cifrado con la clave pública.

1.1.15. **Computación cuántica:** Es un paradigma de computación distinto al de la computación clásica. Se basa en el uso de qubits en lugar de bits, y da lugar a nuevas puertas lógicas que hacen posibles nuevos algoritmos.

Una misma tarea puede tener diferente complejidad en computación clásica y en computación cuántica, lo que ha dado lugar a una gran expectación, ya que algunos problemas intratables pasan a ser tratables. Mientras que un computador clásico equivale a una máquina de Turing, un computador cuántico equivale a una máquina de Turing cuántica [16].

1.1.16. Qubit o cubit (del inglés *quantum bit*, bit cuántico): Es un sistema cuántico con dos estados propios y que puede ser manipulado arbitrariamente. Esto es, se trata de un sistema que solo puede ser descrito correctamente mediante la mecánica cuántica, y que solamente tiene dos estados bien distinguibles mediante medidas físicas. También se entiende por qubit la información que contiene ese sistema cuántico de dos estados posibles. El qubit es la unidad mínima y por lo tanto constitutiva de la teoría de la información cuántica. Es un concepto fundamental para la computación cuántica y para la criptografía cuántica, el análogo cuántico del bit en informática [17].

1.1.17. **Tcpdump:** Es un herramienta en línea de comandos cuya utilidad principal es analizar el tráfico que circula por la red.

a. Permite al usuario capturar y mostrar a tiempo real los paquetes transmitidos y recibidos en la red a la cual el ordenador está conectado.

b. tcpdump funciona en la mayoría de los sistemas operativos UNIX: Linux, Solaris, BSD, Mac OS X, HP-UX y AIX entre otros. En esos sistemas, tcpdump hace uso de la biblioteca libpcap para capturar los paquetes que circulan por la red.

c. Existe una adaptación de tcpdump para los sistemas Windows que se llama WinDump y que hace uso de la biblioteca Winpcap.

d. En UNIX y otros sistemas operativos, es necesario tener los privilegios del root para utilizar tcpdump.

1.1.18. **Darknet** - la internet oculta: En las últimas dos décadas, ciber-criminales y quienes buscan evitar el acecho de las autoridades han hallado refugio en lo que se conoce como la 'red oscura' o 'darknet' (su nombre inglés). Poco se sabe de ella, pero el público comenzó a tomar conciencia de su envergadura en los últimos meses, cuando el grupo de hackers Anonymous anunció el hackeo de decenas de sitios ocultos en ella.

Se dice que Darknet es más grande que la Internet que conocemos. Según la descripción de diversos grupos de hackers, si se compara Internet con un iceberg, la punta del iceberg representa la Internet que conocemos: accesible a través de motores de búsqueda como Google, con todos los sitios web más famosos, y a la cual se llega por medio de un dominio URL de nivel superior (incluyendo .com, .net, .org, etc.) [18].

1.1.19. Nmap: El motor de secuencias de comandos de Nmap es una herramienta para los scripts creados por el usuario. Este poder se demuestra en la suite de scripts diseñados para inspeccionar de Windows a través del protocolo SMB. Muchas tareas de huellas se pueden realizar, incluyendo la búsqueda de cuentas de usuarios, recursos compartidos abiertos, y contraseñas débiles. Todas estas

tareas están bajo una común biblioteca de autenticación, y compartir, lo que da a los usuarios una interfaz común y familiar [19]**.**

Algunas de las herramientas que se asemejan a Nmap [3]:

- **Netcat:** La navaja multiuso para redes.Una utilidad simple para Unix que lee y escribe datos a través de conexiones de red usando los protocolos TCP o UDP. Está diseñada para ser una utilidad del tipo "back-end" confiable que pueda ser usada directamente o fácilmente manejada por otros programas y scripts. Al mismo tiempo, es una herramienta rica en características, útil para depurar {debug} y explorar, ya que puede crear casi cualquier tipo de conexión que podamos necesitar y tiene muchas habilidades incluidas.
- **TCPDump / WinDump:** El sniffer clásico para monitoreo de redes y adquisición de información.

  Tcpdump es un conocido y querido analizador de paquetes de red basado en texto. Puede ser utilizado para mostrar los encabezados de los paquetes en una interfaz de red {"network interface"} que concuerden con cierta expresión de búsqueda. Podemos utilizar esta herramienta para rastrear problemas en la red o para monitorear actividades de la misma. Hay una versión {port} para Windows llamada WinDump. TCPDump es también la fuente de las bibliotecas de captura de paquetes Libpcap y WinPcap que son utilizadas por Nmap y muchas otras utilidades.
- **Hping2:** Una utilidad de observación para redes similar a ping pero con esteroides.

  hping2 ensambla y envía paquetes de ICMP/UDP/TCP hechos a medida y muestra las respuestas. Fue inspirado por el comando ping, pero ofrece mucho más control sobre lo enviado. También tiene un modo traceroute bastante útil y soporta fragmentación de IP. Esta herramienta es particularmente útil al tratar de utilizar funciones como las de traceroute/ping o analizar de otra manera, hosts detrás de un firewall que bloquea los intentos que utilizan las herramientas estándar.
- **GFI LANguard:** Un escáner de red no-libre para Windows. LANguard escanea redes y reporta información como el nivel de "service pack"

de cada máquina, faltas de parches de seguridad, recursos compartidos, puertos abiertos, servicios/aplicaciones activas en la computadora, datos del registro, passwords débiles, usuarios y grupos; y más. Los resultados del escaneo se muestran en un reporte en formato HTML, que puede ser modificado a gusto propio o consultado. Aparentemente, una versión gratuita está disponible para prueba y usos no comerciales.

- **Ettercap:** Es un interceptor/sniffer/registrador para LANs con ethernet basado en terminales. Soporta disecciones activas y pasivas de varios protocolos (incluso aquellos cifrados, como HTTPS). También es posible la inyección de datos en una conexión establecida y filtrado al vuelo {"on the fly"} y aun manteniendo la conexión sincronizada. Muchos modos de sniffing fueron implementados para darnos un set poderoso y completo de sniffing. También soporta plugins. Tiene la habilidad para comprobar si estamos en una LAN con switches o no, y de identificar huellas de sistemas operativos para dejarnos conocer la geometría de la LAN.
- **John the Ripper:** Es un cracker de passwords rápido, actualmente disponible para muchos sabores de Unix (11 son oficialmente soportados, sin contar arquitecturas diferentes). Su propósito principal es detectar passwords de Unix débiles. Soporta varios tipos de hashes de password, que son comúnmente encontrados en varios sabores de Unix, así como también AFS de Kerberos y las "hashes" de Windows NT/2000/2003/2008/XP. Otros varios tipos de hashes se pueden agregar con algunos parches que contribuyen algunos desarrolladores.
- **OpenSSH / SSH:** Una manera segura de acceder a computadoras remotas. Deriva de la versión de ssh de OpenBSD, que a su vez deriva del código de ssh pero de tiempos anteriores a que la licencia de ssh se cambiara por una no libre. ssh (secure shell) es un programa para loggearse en una máquina remota y para ejecutar comandos en una máquina remota. Provee de comunicaciones cifradas y seguras entre dos hosts no confiables, sobre una red insegura. También se pueden redirigir conexiones arbitrarias de TCP/IP sobre este canal seguro.
- **Sam Spade:** Herramienta de consulta de redes de distribución gratuita. Provee de una interfaz de usuario gráfica (GUI) consistente y de una implementación de varias tareas de investigación de red útiles. Fue diseñada con

la idea de rastrear spammers en mente, pero puede ser útil para muchas otras tareas de exploración, administración y seguridad. Incluye herramientas como ping, traceroute, explorador de web crudo, transferencia de zona de DNS, búsqueda en sitios web, entre otras. Los que no son usuarios de Windows pueden disfrutar de las versiones online de muchas de sus herramientas.

- **ISS Internet Scanner:** Evaluación de vulnerabilidades a nivel de Aplicación. Internet Scanner comenzó en el '92 como un pequeño escáner. ISS creció hasta ser una enorme empresa con una amplia gama de productos de seguridad. El escáner de Internet de ISS es bastante bueno, pero cuenta.
- **Tripwire:** Es un comprobador de integridad de archivos y directorios. Es una herramienta que ayuda a administradores y usuarios de sistemas monitoreando alguna posible modificación en algún set de archivos. Si se usa regularmente en los archivos de sistema, Tripwire puede notificar a los administradores del sistema, si algún archivo fue modificado o reemplazado, para que se puedan tomar medidas de control de daños a tiempo.
- **L0phtCrack 4:** Aplicación de recuperación y auditoría de passwords para Windows. Crackea los passwords de Windows a partir de las hashes que puede obtener (por medio de acceso apropiado) de máquinas con Windows NT/2000/2003/2008/XP, independientes, servidores en red, controladores primarios de red. En algunos casos, puede snnifear hashes directamente desde el cable. También tiene numerosos métodos de generar suposiciones de passwords (diccionario).

1.1.20. **RPC:** Asignación de puerto dinámico de Remote Procedure Call (RPC) es utilizada por las aplicaciones de administración remota como administrador de protocolo de configuración dinámica de Host (DHCP). Asignación dinámica de puertos RPC le indicará que el programa RPC para usar un puerto aleatorio determinado por encima de 1024.

Los clientes que utilizan los servidores de seguridad que desee controlar qué puertos usan RPC para que su enrutador del firewall puede estar configurado para reenviar sólo estos puertos de protocolo de Control de transmisión (TCP) [20].

1.2 Historia

A nivel mundial, y en especial en Estados Unidos a mediados de los años 60, tuvieron que hacer frente a la manipulación de la información y el espionaje de datos, para los que no había legislación penal. En esos años, el debate se centró en la elaboración de una respuesta jurídica [21].

La introducción de la interfaz gráfica en los años noventa, a la que siguió un rápido aumento del número de usuarios de internet, engendró nuevos desafíos. La información colocada legalmente en internet en un país pasó a estar disponible en todo el mundo, incluso en aquellos países en que su publicación no era legal como Estados Unidos, España [22]. Otro aspecto preocupante de los servicios en línea es el lugar en el que se encuentra el delincuente informático ya que puede ser por completo distinto del lugar en el que éste comete su ciberdelito y convirtiéndose ahora en ciberdelitos trasnacionales [23].

El siglo XXI han predominado métodos nuevos y sofisticados para delinquir tales como la “pesca de datos” o “phishing” [24]. y los ataques con redes zombi o “botnets” [25]; el uso de tecnologías que resultan aún más difíciles de controlar para los que investigan comunicaciones con transmisión de voz sobre protocolo de Internet (VoIP) y la informática en nube “cloud computing” [26].

Las estadísticas relativas a la ciberdelincuencia no suelen mencionar los delitos por separado, y las pocas estadísticas que existen sobre el impacto del delito cibernético no son por lo general, lo suficientemente detalladas como para proporcionar información fidedigna sobre la escala o el alcance de los delitos. Sin esos datos, es difícil cuantificar el impacto del delito cibernético en la sociedad y elaborar estrategias para combatirlo [27], sin embargo, la ISO/IEC 27005 [28], sobre gestión de riesgos y seguridad de la información, los cuales permiten valorar el riesgo, tanto personal como de hardware y software, identificar activos, amenazas, controles existentes, vulnerabilidades, impactos y probabilidades con base a una escala que permite estimar entre nula y extrema la probabilidad y su evaluación que permite de una escala de extremadamente alto a una extremadamente bajo la posibilidad de no detección.

Esta evaluación y tratamiento del riego, consiste en la identificación de la gama de opciones para tratarlo, la evaluación de dichas opciones, la preparación de planes para el tratamiento del riesgo y su implementación.

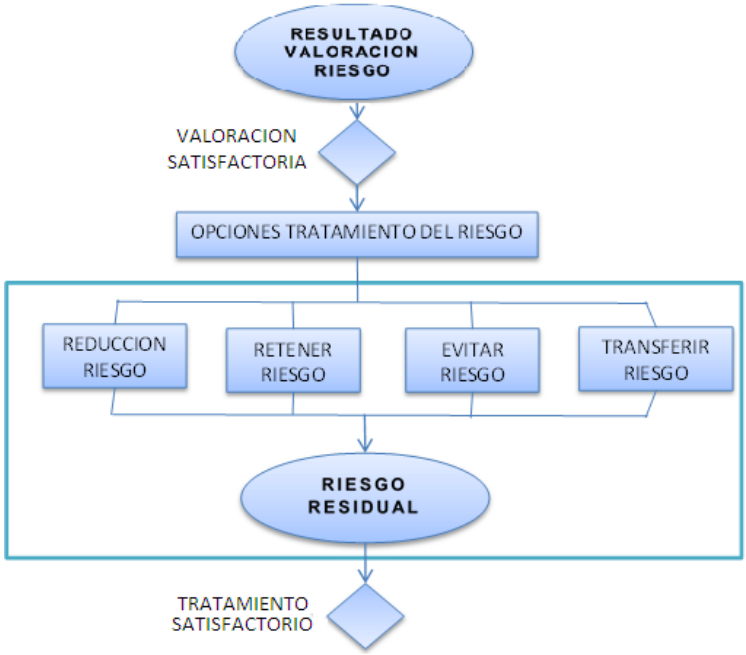
Hay cuatro opciones disponibles para su tratamiento, las cuales consisten en: reducción del riesgo, retención (aceptación) del riesgo, evitar el riesgo y transferir el riesgo.

Tabla 1: Escala de Probabilidad y detección [29]

Probabilidad		**Score**
Nula	Rara vez ocurre	0
Muy baja	2 - 3 veces cada 5 años	1
Baja	Una vez por año	2
Media	Hasta dos veces por año	3
Alta	Hasta una vez por mes	4
Muy alta	Más de una vez por mes	5
Extrema	Varias veces a la semana o al día	6

Detección		**Posibilidad no detección**
Extremadamente alto	Rara vez ocurre	1
Alto	2 - 3 veces cada 5 años	2
Medio	Una vez por año	3
Bajo	Hasta dos veces por año	4
Extremadamente bajo	Varias veces a la semana o al día	5

Figura 1-1.: Actividad de tratamiento de riesgos **[28]**.



En Colombia se ha venido trabajando en grupos de investigación tales como GECTI, entre otros, grupo de Investigación en Comercio Electrónico, Telecomunicaciones e Informática en seguridad de redes inalámbricas de la Universidad de los Andes. Cuentan con líneas de investigación como evidencia digital, delitos informáticos, propiedad intelectual en el contexto digital dirigidos por ingenieros en la rama tanto nacional como internacional como lo son: Ing. Jeimy José Cano Martínez y el Ing. Rafael Hernando Gamboa Bernate [30].

A Nivel Mundial, a partir del año 2001 el Standards Australia (Estandares de Australia) se encarga de trabajar en proyectos de investigación relacionados con evidencia digital y publicaron para el año 2003 la “Guía para el manejo de evidencia en IT” [31], aunque esta guía no está disponible.

Por otra parte, existen herramientas las cuales nos permiten análisis de vulnerabilidades, de las cuales las más destacadas hasta el momento son [32], [33]:

Alternativas a Snort: Comparativa entre snort, nessus, openvas, tcpdump y retina.

a. OPENVAS vs NESSUS

OPENVAS:

- La primera diferencia entre NESSUS y OPENVAS es la cantidad de plugins con los que cuentan, aproximadamente **48268** plugins para el análisis de vulnerabilidades mientras que OPENVAS por el momento tiene 25505. Hay una diferencia de casi el doble de plugins, esto le da una mayúscula ventaja a NESSUS.
- La segunda diferencia es que OPENVAS es más complejo en cuanto a instalación y configuración inicial, esto debido a que tiene una arquitectura de funcionamiento distinta la cual puede complicar la instalación y configuración inicial.
- El tiempo es otra variable importante en esta comparación, para efectos de esta prueba, un servidor Windows 2003 SP1, los tiempos son:

- NESSUS: 10 minutos aproximadamente
- OPENVAS: 18 minutos aproximadamente

- OPENVAS sólo tiene una versión, la cual es libre y gratis para usar; este es el principal motor que debe impulsar el apoyo hacia esta herramienta; además por ser gratuito esta herramienta no tiene un limitante de direcciones IP que pueden ser escaneados.
- La principal diferencia y por la cual aún OPENVAS no es altamente usado está relacionado a la cantidad de plugins, OPENVAS aún no detecta vulnerabilidades CRITICAS y que ya deberían estar implementadas sobre la misma.

Lo que llama la atención de sobremanera es que OPENVAS encuentra menos vulnerabilidades que NESSUS pero lo que preocupa es que no haya encontrado la vulnerabilidad altamente conocido MS08-067.

b. RETINA vs NESSUS

RETINA:

- NESSUS y OPENVAS realizan escaneo de vulnerabilidades a aplicaciones web, mientras que RETINA en su versión COMMUNITY (gratuita) no ofrece esta característica.
- Su instalación es extremadamente sencilla, basta dar unos clicks y la tendremos instalada, por el momento sólo hay una versión para Windows de esta herramienta y es una aplicación de ESCRITORIO, es decir, no se puede instalar en un servidor y usar a través de red.

c. SNORT vs NESSUS y TCPDUMP

SNORT:

Aunque Nessus frente a los anteriormente expuestos, es uno de los que mejor realiza el análisis de vulnerabilidades, pero teniendo en cuenta SNORT, éste es capaz de cumplir funciones con las cuales no cuentan las demás como:

a. Escuchar en modo promiscuo en un interfaz de red, es decir, atendiendo tanto a los paquetes dirigidos a dicho interfaz como a los que no le correspondería escuchar, y mostrar o registrar localmente en el disco duro dicho tráfico. En este modo de operación es igual al programa "tcpdump", hasta el punto de que comparte el sistema de registro en el disco de tráfico.

b. Escuchar en modo promiscuo en un interfaz de red, pero atender a una serie de reglas más avanzadas que las de tcpdump, lo cual puede servir para analizar y depurar tráfico y protocolos de red.

c. Frente a tcpdump, podemos decir que Snort no solo realiza las mismas tareas, sino que las realiza de forma más eficientes mediante reglas que permiten el registro de incidentes.

Por otra parte, se pueden generar los archivos de evidencia en extensión igual que la generada por tcpdump, la cual permitirá el mismo ser observada en otras herramientas como Wireshark.

d. Como herramienta para el análisis avanzado del tráfico de red, obviamente también en modo promiscuo, y utilizando una base de reglas sofisticada para detectar y avisar sobre tráfico no deseado que esté circulando por nuestra red.

e. SNORT no se limita exclusivamente al protocolo IP+TCP/UDP/ICMP, aunque sobresalga su especialización en ellos. También es capaz de escuchar otros protocolos como ethernet, fddi, arp, rarp, lat y mopdl.

f. Para ser capaz de filtrar y analizar el tráfico lo mejor posible, snort lleva una serie de módulos que le permiten:

- recomponer el tráfico fragmentado
- recomponer las cadenas de caracteres enviadas a través de servicios como telnet, ftp, smtp, http, pop3 e imap, lo que le permite explorar el tráfico desde el nivel de aplicación y no sólo del nivel de red.
- entender el tráfico http, como p.ej. la expansión de caracteres " " = "%20"
- entender el tráfico rpc
- reconocer el tráfico generado por el BackOrifice, independientemente de los puertos que use
- detectar escaneos remotos en busca de puertos abiertos

Además, está diseñado de modo que sea fácil incorporarle nuevos módulos para expandir su funcionalidad.

1.3 Metodología

1.2.1. Cadena de Custodia y embalaje de la evidencia

La informática forense es la ciencia de identificar, preservar, analizar y presentar datos almacenados electrónicamente en un medio computacional, datos que son evidencias de un delito informático [34].

Basándose en esta definición, esta metodología se basa en 5 partes perfectamente definidas, las cuales se deben tener en cuenta al realizar la respectiva cadena de custodia, su embale hasta ser entregada como evidencia:

a. ASEGURAR LA ESCENA.
b. IDENTIFICAR EVIDENCIAS.
c. PRESERVAR EVIDENCIAS.
d. ANALIZAR EVIDENCIAS.
e. PRESENTACION E INFORMES DEL ANÁLISIS.

Las TRES primeras partes, deberían en lo posible realizarse en la escena del delito.

La CUARTA en el laboratorio forense.

La ÚLTIMA parte se hará en un tribunal de justicia o delante de un cliente (puesto que será la presentación de unos informes).

Esta metodología es empleada por toda organización que después de obtener la evidencia, tenga su respectivo procedimiento para dar inicio a la cadena de custodia.

La empresa Adalid abogados cuenta con un grupo de expertos especializado en informática forense y herramientas que obtienen imágenes forenses de evidencia digital; sin embargo, el aplicativo que utilizan llamado certievidencia [7], herramienta propietaria de Adalid abogados, obtiene el respectivo hash de todo el contenido de los medios o dispositivos de almacenamiento, garantizando la integridad de la imagen obtenida a través de un cifrado md5 de 128 [35], la cual contiene una seria de colisiones de hash

problemas de seguridad detectados desde 1996; la cual fue remplazado por un algoritmo llamado sha1, la cual se ve comprometido igualmente desde el año 2005 por colisiones de hash [36].

Es por esta razón, que al mecanismo planteado se le implemento un algoritmo SHA-512, la cual, a la actualidad no ha sido vulnerado. Al igual que el anteriormente descrito, al algoritmo se le agrega el salt para hacer aún más difícil su recuperación. La longitud del salt puede variar [37].

1.4 Requerimientos Legales

Los delitos informáticos intentan ser disminuidos con la creación de la ley 1273 de 2009 expedida por el senado de la república, y buscan concienciar a las empresas a tomar medidas de seguridad, de lo contrario recibirán una sanción económica si no son capaces de recuperarse de un incidente, o facilitar datos importantes sobre el incidente [38]. Por otra parte, la ley 527 de 1999, define y reglamenta el acceso y uso de los mensajes de datos , la cual en el capítulo 6,7 y 8 de la presente ley, permite dar parámetros exactos para que una evidencia sea válida mediante la integridad de los datos obtenidos [39].

2. Parámetros y limitaciones

2.1 Parámetros mínimos de hardware y software

a. Servidor:

- **Hardware**
  - Procesador de 1.5 Ghz con un núcleo.
  - 512 MB de RAM
  - 250 GB de disco duro.
- **Software**
  - **Linux:** Debian Wheezy o versiones superiores [40].
  - **Configuración:** Servidor de evidencias digitales

Figura 2-1.: Instalación de Snort en Linux Debian**.**

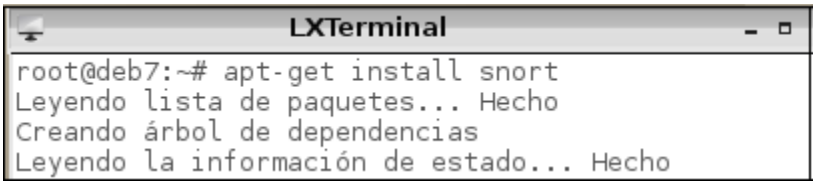


Figura 2-2.: Configuración de Snort**.**

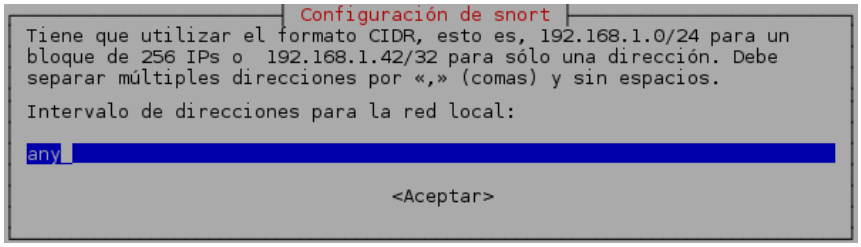


Figura 2-3.: Creación de directorios para almacenar evidencia**.**

Asignación de permisos de acceso total al folder de la red (/EvidenciaRed/Red192.168.1.0), y ninguno al grupo del propietario ni a otros usuarios por seguridad:

- SSH para Linux en servidor Debian Wheezy

Figura 2-4.: Instalación de SSH**.**

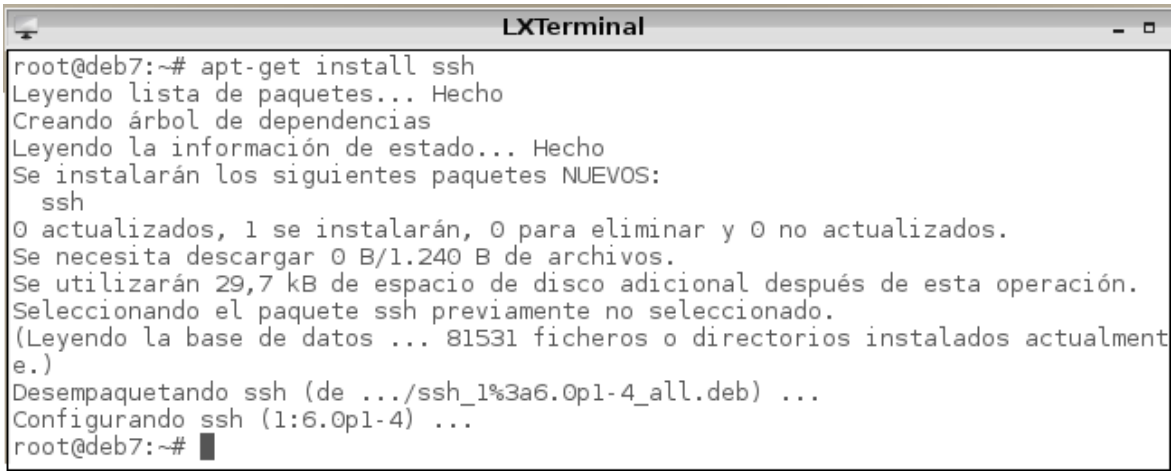


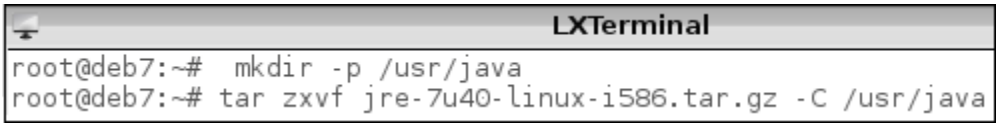
Figura 2-5.: Permisos para directorios contenedores de evidencia**.**

- Instalación y despliegue de JRE en Debian

Figura 2-6.: Descarga de JRE en Debian **[41]**.

Se crea el directorio de java como root y se descomprime el tar.gz.

Figura 2-7.: Creación de directorio java y descompresión del mismo**.**

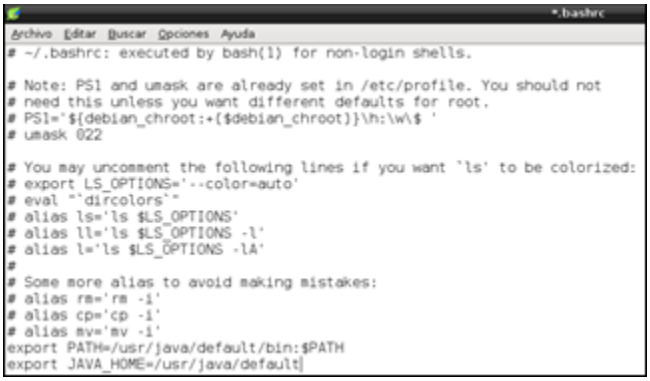


Se copian los archivos a /usr/share creando posteriormente el enlace simbólico default, para establecer la variable de entorno PATH de Java:

Se edita el archivo ~/.bashrc con un editor de texto plano, si no existe lo creamos y añadimos lo siguiente al final del mismo.

export PATH=/usr/java/default/bin:$PATH
export JAVA_HOME=/usr/java/default

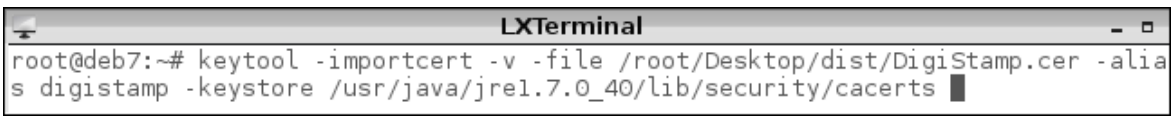
Figura 2-8.: Edición de archivo bashrc**.**



Certificado de DigiStamp en los repositorios de confianza raíz de Java.

Ahora se procederá a instalar el certificado de DigiStamp en los repositorios de confianza raíz de Java:

Figura 2-9.: Instalación certificado DigiStamp



b. **Cliente**:

- **Hardware**
  - Procesador de 800 Mhz con un núcleo.
  - 256 MB de RAM
  - 120 de disco duro.

- **Software**

- **Sistema Operativo:** Windows XP SP3 o superior [42].
- **Snort**

Debido que Snort, no cuenta con un instalador para Windows [5], es necesario descargar de la página el existente y crear uno a partir del mismo, de tal forma que pueda ser instalado en servidores y clientes, sin que requiera configuraciones adicionales.

Figura 2-10.: Descarga de Snort

Figura 2-11.: Reglas Snort

- **Creación de usuario para conexión SSH clientes**

Figura 2-12.: Creación de usuario para conexión ssh clientes**.**

- SSHFS Manager Vs.0.0.1.5: Permite hacer conexión entre máquinas para que la información viaje cifrada, permitiendo guardar de forma segura la información en el servidor [43].

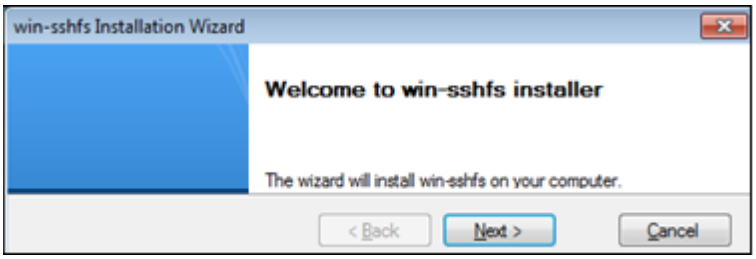
Posteriormente, se procede a instalar WinPcap, la cual permite la captura de paquetes con el Snort [6].

Figura 2-13.: Instalación de WinPcap**.**

- **SSHFS:**

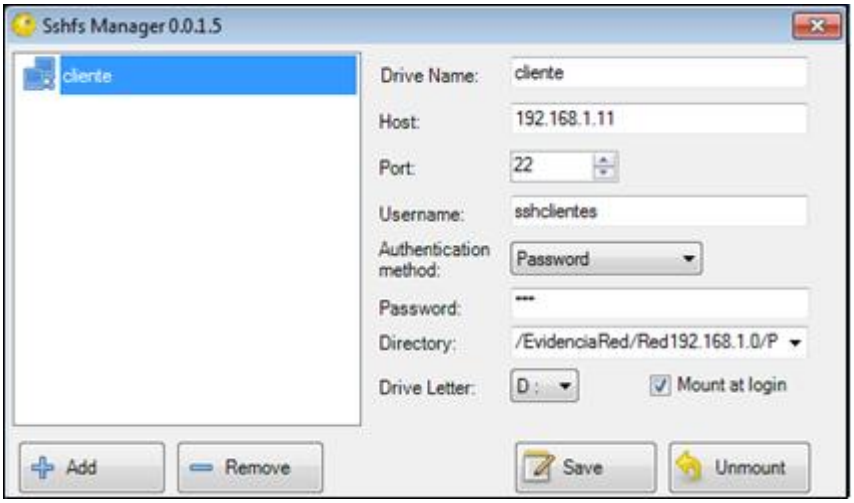
Finalizada la instalación de WinPcap, se procede a instalar SSHFS:

Figura 2-14.: Instalación de SSHFS.



A continuación se procede a configurar el SSHFS para que se conecte al PC servidor, con el usuario y clave creado en la instalación de herramientas del servidor, además debe indicar el directorio que contendrá los registros electrónicos:

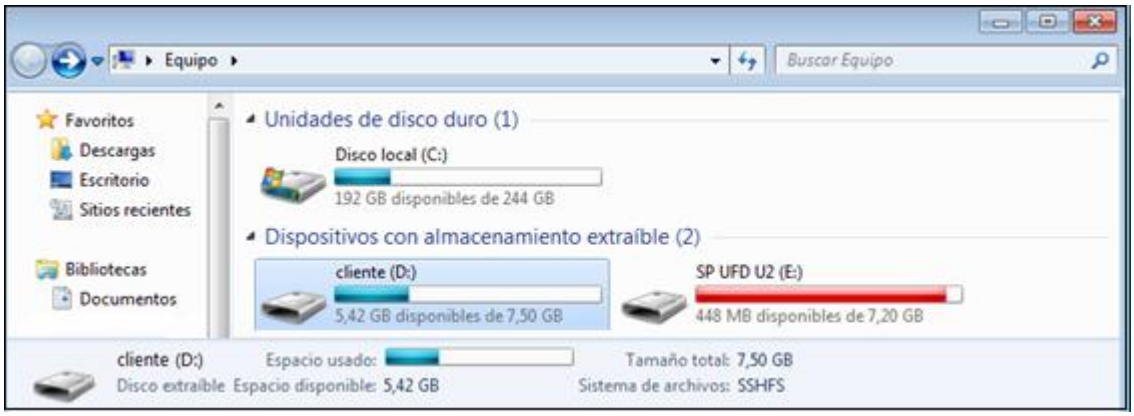
Figura 2-15.: Configuración y montaje de SSHFS**.**



Es ideal que utilice un directorio por cada PC que vaya a guardar registros electrónicos en el servidor. Haga clic en el botón "Mount", para montar la unidad en el PC.

Se observa que se creó el disco donde se guardará los archivos de log.

Figura 2-16.: Disco de archivos de logs**.**



2.2 Limitaciones

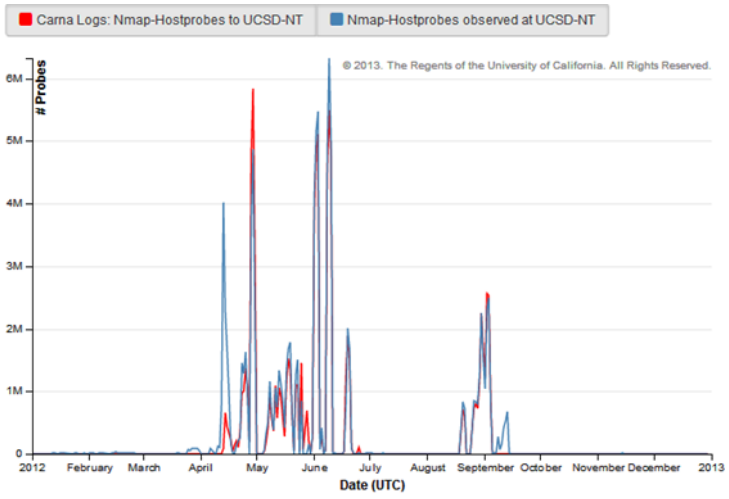
En la validación del mecanismo, se utilizaron dos tipos de ataques, las cuales permiten demostrar que estos tipos de eventos son registrados en logs mediante el sistema de

detección de intrusos, la integridad del mismo con el Hash (SHA-512), y su estampado cronológico la cual permite garantizar la información obtenida del mismo.

El primer ataque efectuado, se realizó mediante Nmap, que consiste en un escáner de red, la cual es una herramienta de seguridad considerada como una de las mejores [44] herramientas gratuitas en existencia [19].

Se utilizó esta herramienta ya que en la parte académica, se están trabajando actualmente proyectos las cuales consisten en detectar evidencia del tráfico capturado que fue sondeado en el año 2012 en toda la red IPv4, utilizando una red de bots (botnet llamado "carna", la cual consiste en escaneo de puertos utilizando dispositivos embebidos inseguros). Con la Red de Telescopios UCSD (una gran red oscura) [18], se trabaja detectando tráfico que posteriormente es seleccionado en paquetes que consisten en una sonda nmap (compuesto por cuatro tipos diferentes de paquetes), que la red de bots Carna utiliza y que permite visualizar muestras recolectadas, el número total de sondas que se observó en el telescopio se detallan en la línea azul. Si bien estas sondas pueden haber sido generados por cualquier host de Internet, el gran aumento visible entre abril y septiembre de 2012, coincide con los registros distribuidos por los autores de la botnet (línea roja), que muestra evidencia de esta actividad de exploración ilegal [45].

Figura 2-17.: Carna Logs – Nmap.



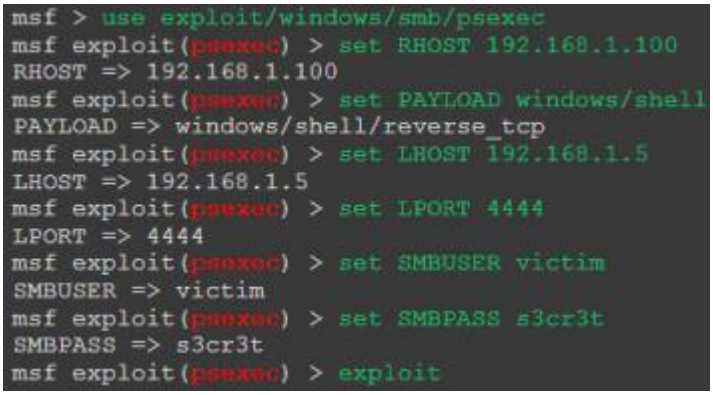
Por otra parte, Nmap ("Network Mapper") [19], es una utilidad de código abierto para la exploración de la red o la auditoría de seguridad, scripts que permite identificar versiones

de software en remoto, vulnerabilidades, enumeración de usuarios, directorios etc., y que su finalidad es escanear rápidamente grandes redes, se utilizó para demostrar que el mecanismo planteado logra detectar este tipo de escaneo y registrarlo correctamente en el servidor de logs.

Por otra parte, se aprovecha la vulnerabilidad MS08-067, que es un metasploit [46], que permite el control total del equipo identificado como víctima, aplicado sobre el servicio RPC de Windows [20].

Se utiliza este procedimiento ya que es una de las vulnerabilidades más utilizadas para lograr obtener control total de la maquina objetivo.

Figura 2-18.: Metasploit **[47]**.



Por otra parte, este ataque puede realizarse con otros procedimientos para permanecer ocultos sin que sea detectado por ningún dispositivo sin importar que este sea un firewall que controla el tráfico de la red [47].

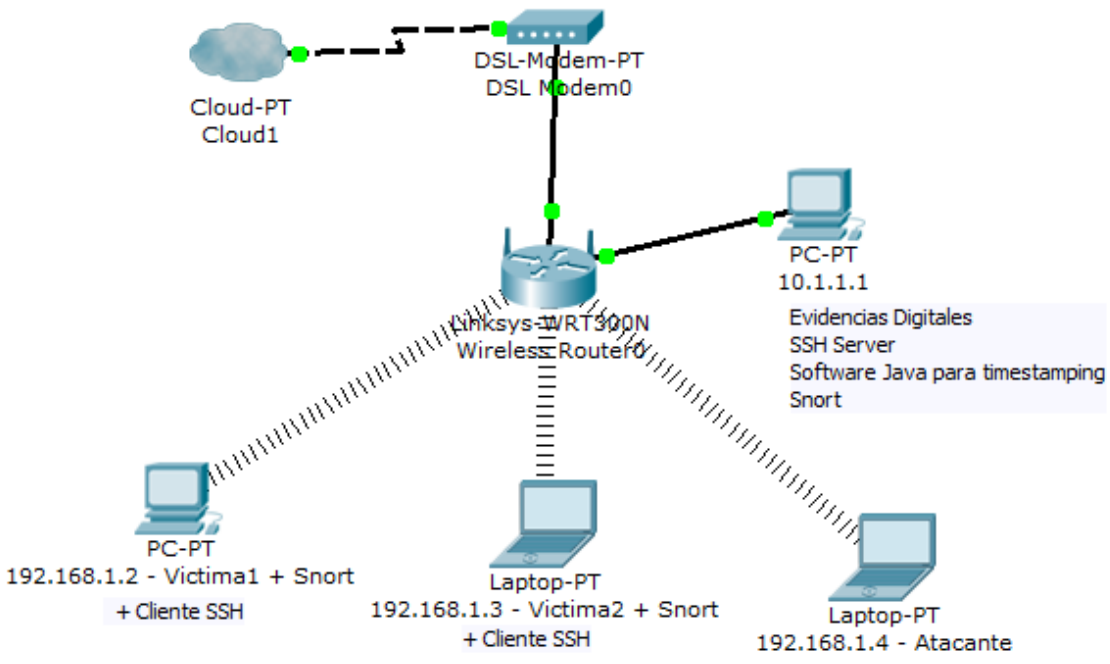
3. Diseño

3.1 Arquitectura

3.1.1 Diagrama de arquitectura.

A continuación se presenta el diagrama de arquitectura la cual permite en forma general explicar la arquitectura que permitirá registrar los eventos en el servidor de logs.

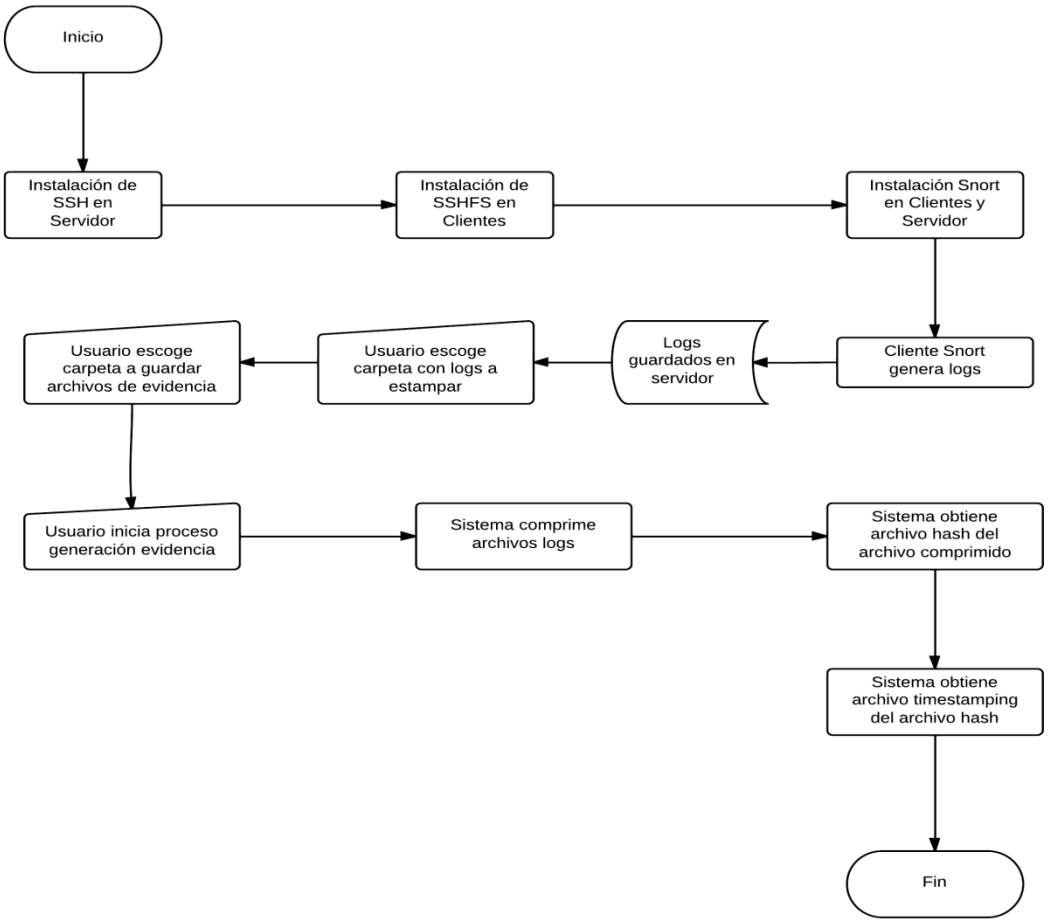
Figura 3-1.: Diagrama de Arquitectura.



3.1.2 Diagrama de flujo.

Figura 3-2.: Diagrama de Flujo.

Se detalla paso a paso lo que hace el mecanismo desde la instalación del túnel por medio del protocolo SSH, hasta el momento que el sistema obtiene el archivo de estampado cronológico del hash, a partir de los registros obtenidos por Snort.



3.2 Construcción de Snort para Windows.

Teniendo descomprimido el Snort (Capitulo 2.1 sección b), se procede abrir el directorio de reglas las cuales algunas son modificadas y otras serán creadas en su totalidad para la detección de los incidentes.

Figura 3-3.: Extracción de reglas de Snort

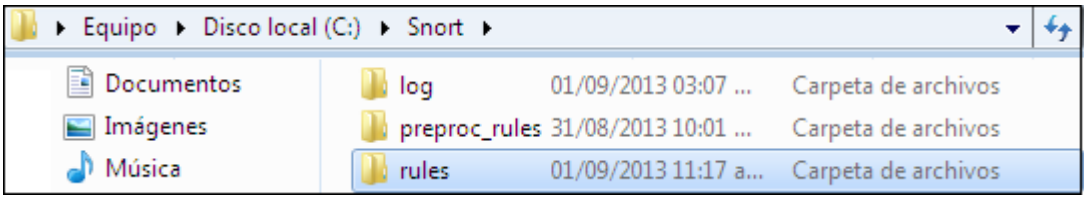
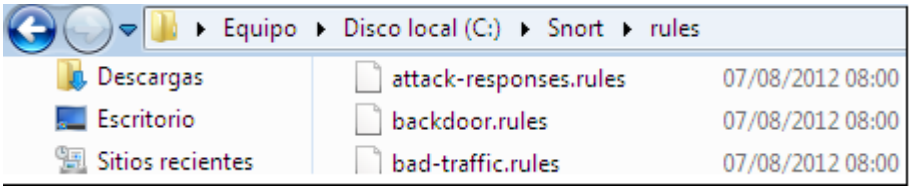


Figura 3-4.: Listado de reglas de Snort



a) Se ingresa en modo consola dentro del directorio C:\Snort\bin y se ejecute el Snort usando archivo de configuración específico, el cual indica cuales archivos de reglas se van a usar.

Figura 3-5.: Ejecución de Snort usando archivo de reglas

```
C:\Snort>snort -c c:\snort\etc\snort.conf
Detection:
   Search-Method = AC-Full-Q
    Split Any/Any group = enabled
    Search-Method-Optimizations = enabled
    Maximum pattern length = 20
ERROR: c:\snort\etc\snort.conf(247) Could not stat dynamic module path "/usr/loc
al/lib/snort_dynamicpreprocessor/": No such file or directory.

Fatal Error, Quitting..

C:\Snort\bin>_
```

b) Al hacer el proceso anterior, aparecerán los errores de reglas de Snort las cuales deben ser reparadas.

c) Con base en el error generado anteriormente, se procede a editar el archivo snort.conf con algún editor de texto; para este caso, se utilizó Notepad++ [48], la cual permitirá encontrar más fácilmente la línea que contiene el error para ser corregido.

En la línea 247 de la figura 3,6, del archivo snort.conf, nos informa que la ruta de las librerías snort_dynamicpreprocessor esta incorrecta ya que por defecto trae las de linux. Por tal motivo se corrige con base a la ruta para Windows.

Antes de reparar las líneas que aparecerán como error, se deben comentarear todas las líneas que comiencen con la palabra preprocessor, la cual en Linux lo que hace es desprender los paquetes en sus componentes (desarman los paquetes), para verificar si el paquete es malicioso o no (Se envía paquete End to End la cual al enviar un paquete se descompone en paquetes más pequeños y así el Snort no lo detecta y en la maquina destino arma el paquete para retransmitir a destino). Windows no puede desensamblar estos paquetes para verificar los mismos, es por esta razón que debemos comentar las líneas en Windows que comiencen con la palabra preprocessor y así evitamos estos errores en el archivo de configuración de reglas.

Figura 3-6.: Edición de archivo snort.conf

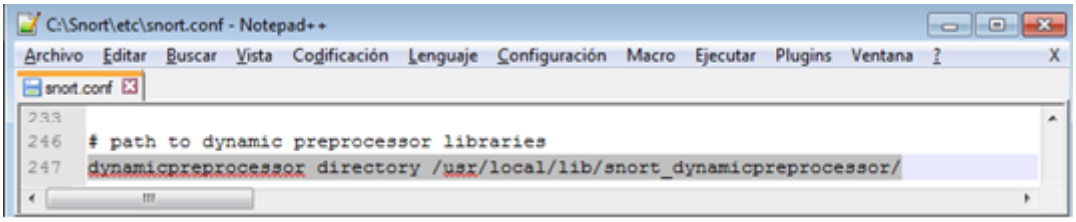
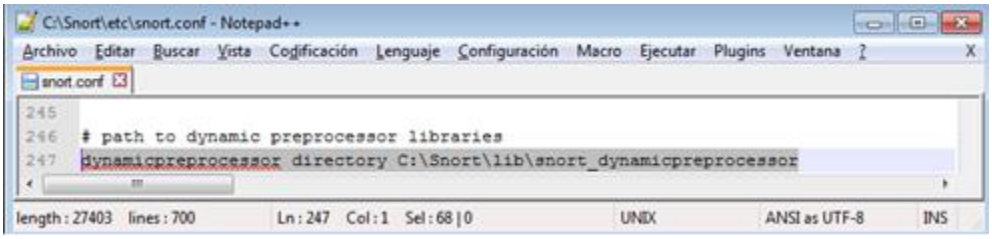


Figura 3-7.: Edición de archivo snort.conf y cambio de ruta de reglas para Windows



d) Se ejecuta nuevamente Snort usando archivo de configuración específico, el cual indicará si contienen otras reglas con errores para ser nuevamente corregidas, la cual se observa que en la línea 253 no tiene la dirección correcta de las librerías de reglas dinámicas snort_dynamicrule la cual esta línea se comenta ya que para Windows no se utilizan estas reglas.

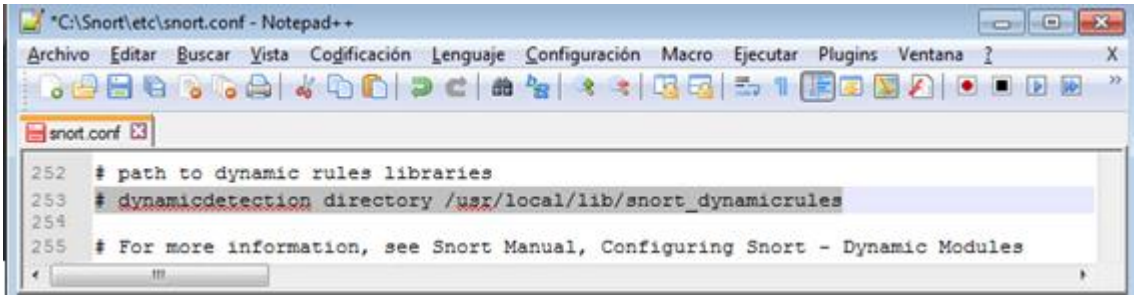
Figura 3-8.: Reporte de errores de reglas de Snort

```
Detection:
   Search-Method = AC-Full-Q
    Split Any/Any group = enabled
    Search-Method-Optimizations = enabled
    Maximum pattern length = 20
ERROR: c:\snort\etc\snort.conf(253) Could not stat dynamic module path "/usr/loc
al/lib/snort_dynamicrules": No such file or directory.

Fatal Error, Quitting..

C:\Snort\bin>
```

Figura 3-9.: Edición de archivo snort.conf y se construye regla que contiene error.



e) Se ejecuta las veces que sea necesario el Snort hasta que ya no genere el mismo errores; en este caso se ejecuta el comando C:\Snort\bin>snort -c c:\snort\etc\snort.conf la cual nuevamente nos muestra otro error que indica que se debe definir la ruta del LOG donde serán grabados los registros obtenidos del snort. Para el caso propuesto, se selecciona el directorio C:\Snort\log. Allí quedaran los archivos pcap y alert.ids.

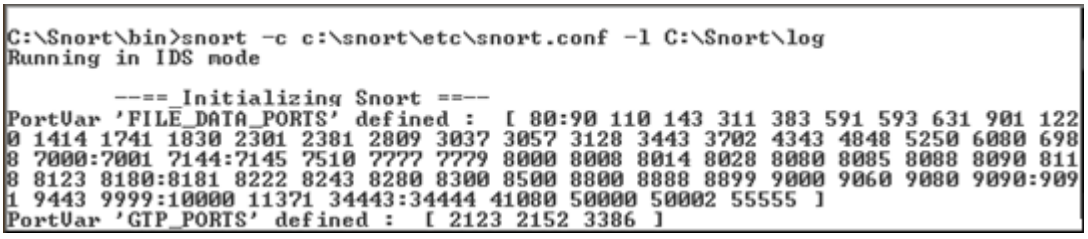
Figura 3-10.: Reporte de errores de reglas de Snort.

```
Detection:
   Search-Method = AC-Full-Q
    Split Any/Any group = enabled
    Search-Method-Optimizations = enabled
    Maximum pattern length = 20
Tagged Packet Limit: 256
Log directory = log
ERROR: OpenAlertFile() => fopen() alert file log/alert.ids: No such file or dire
ctory
Fatal Error, Quitting..

C:\Snort\bin>
```

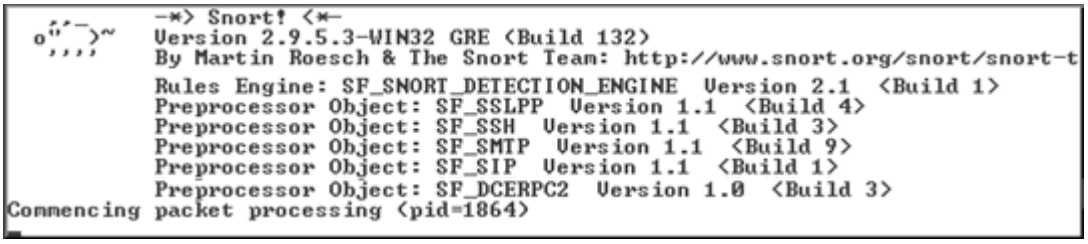
f) Para el error anterior, ejecutamos el comando: snort -c c:\snort\etc\snort.conf -l C:\Snort\log que permitirá re-direccionar los archivos logs a un directorio predeterminado.

Figura 3-11.: Re-direccionar los archivos logs a un directorio predeterminado**.**



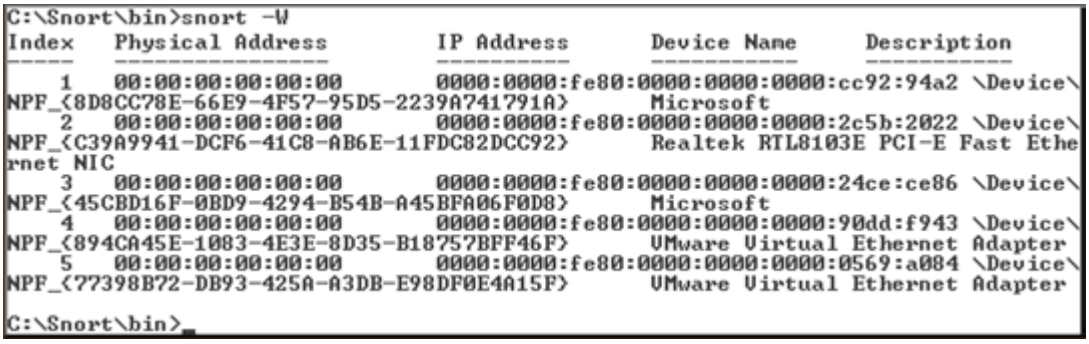
g) Después de la respectiva depuración de las reglas de Snort nos debe aparecer una ventana la cual nos indicara que el proceso es exitoso y que escuchara el tráfico en la interfaz de red por defecto.

Figura 3-12.: Validación exitosa de reglas de Snort**.**



h) Para listar las interfaces se usa el comando: snort –W, la cual permitirá manualmente seleccionar la interfaz a utilizar para el escaneo de la red.

Figura 3-13.: Listado de interfaces**.**

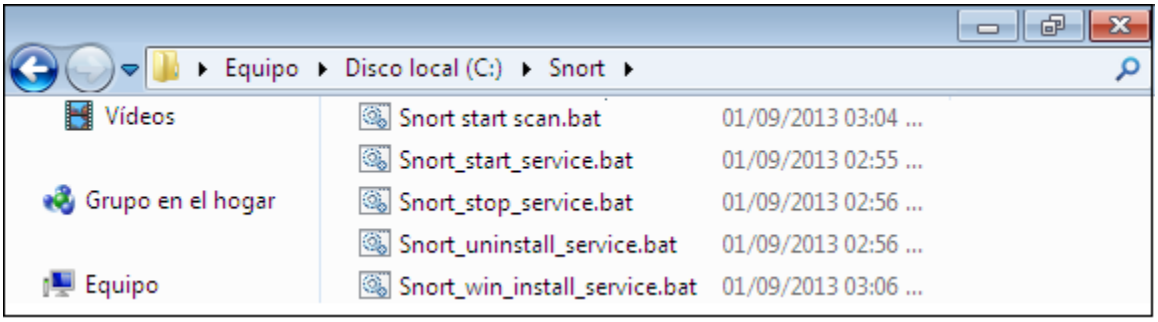


i) Con base en lo anterior, se construyen 5 Script que permitirá administrar el Snort.

3.3 Administración y creación de Scrip para Snort

En esta parte, se crean 5 Script que permiten instalar, detener, y desinstalar el servicio, al igual que la verificación de errores de construcción de reglas de Snort, permitiendo administrarlo de forma automática.

Figura 3-14.: Script que permiten administrar Snort**.**

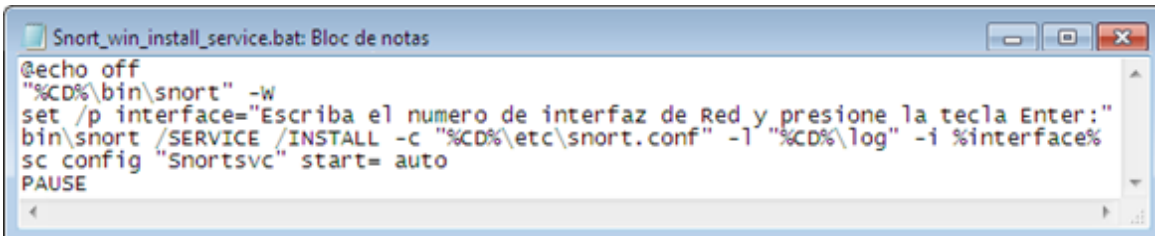


Detalle de los Script:

a. Script Snort_win_install_service.bat: permitirá instalar el servicio de Snort.

Código del Script:

Figura 3-15.: Instalación del Servicio Snort**.**



Explicación:

"%CD%\bin\snort" –W

Permite listar las interfaces que están en el computador.

set /p interface=<texto>

Permite recibir el número de la interfaz seleccionada por el usuario para instalar el servicio sobre esa interfaz.

bin\snort /SERVICE /INSTALL -c "%CD%\etc\snort.conf" -l "%CD%\log" -i %interface%

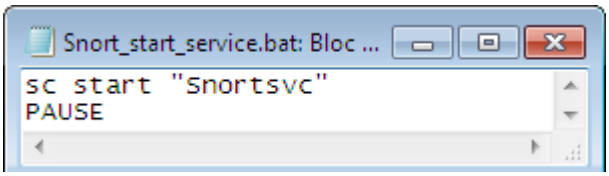
Instala el servicio de snort para que se ejecute usando el archivo de configuración que se encuentra en el directorio actual, subdirectorio \etc\snort.conf; directorio de logs y el número de la interfaz dada.

sc config "Snortsvc" start= auto

Activa el servicio ya instalado como inicio automático.

b. Script Snort_start_service.bat: permite iniciar el servicio de Snort.

Figura 3-16.: Iniciar servicio Snort**.**



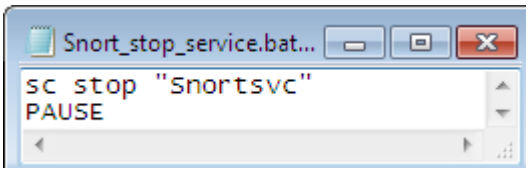
Explicación:

sc start "Snortsvc"

Inicia ejecución del servicio "Snortsvc".

c. Script Snort_stop_service.bat: Permite detener el servicio snort Snortsvc.

Figura 3-17.: Detener Servicio Snort**.**



Explicación:

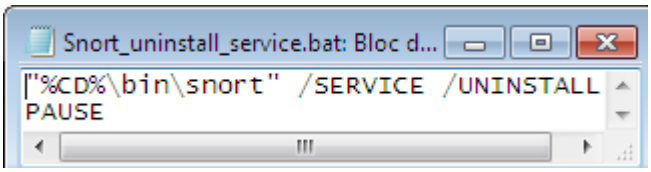
sc stop "Snortsvc"

Detiene ejecución del servicio "Snortsvc".

d. Script Snort_uninstall_service.bat: Permite desinstalar el servicio snort.

Código del Script:

Figura 3-18.: Desinstalar Servicio Snort**.**



Explicación:

"%CD%\bin\snort" /SERVICE /UNINSTALL

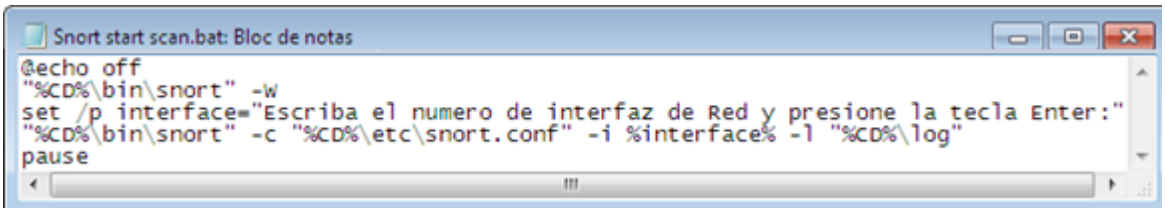
Desinstala el servicio de snort ubicado en el directorio actual.

e. El Script Snort start scan.bat tiene dos funcionalidades:

- Revisión de errores: Permite ejecutar el Snort con el fin de mirar errores que puedan estarse generando por una mala construcción de reglas.
- Permite ejecutar el Snort sin necesidad de crear y administrar servicios en Windows.

Código del Script:

Figura 3-19.: Verificación de errores de construcción reglas de Snort**.**



Explicación:

"%CD%\bin\snort" –W

Permite listar las interfaces que están en el computador.

set /p interface=<texto>

Permite recibir el número de la interfaz seleccionada por el usuario para instalar el servicio sobre esa interfaz.

*"%CD%\bin\snort" -c "%CD%\etc\snort.conf" -i %interface% -l "%CD%\log"*Creación de directorios de log por red/computador.

4. Implementación

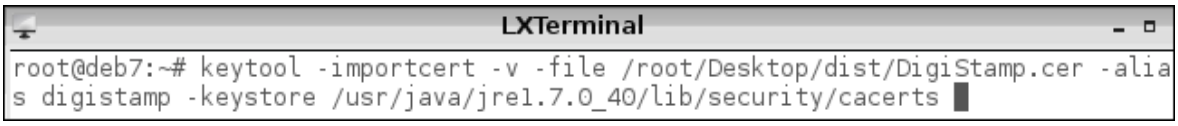
Después de tener los parámetros necesarios en el diseño del mecanismo que permite registrar por medio de logs los incidentes en redes inalámbricas y con base en las limitaciones, capítulo 2, parágrafo 2-2, se implementa el mecanismo que permite garantizar la integridad de la evidencia.

- Despliegue de HashStamp

a. Certificado Digital

Como primera medida, se procede a instalar el certificado de Digistamp en los repositorios de confianza raíz de Java:

Figura 4-1.: **C**ertificado DigiStamp**.**



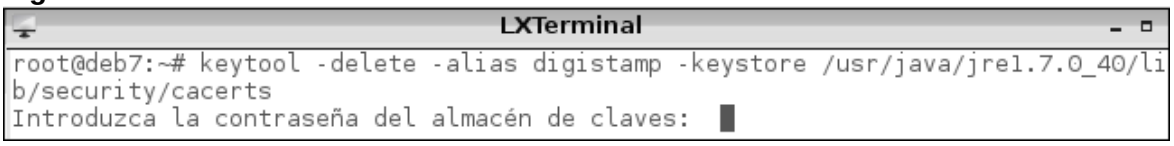
El cual solicitará la clave del almacén de claves, la cual es "changeme" por defecto en la instalación de Java:

Figura 4-2.: Solicitud de password**.**

Pregunta si desea confiar en este certificado, escriba "si":

Para borrar el certificado de la BD de certificados, escriba el siguiente comando:

Figura 4-3.: Borrado de certificado de base de datos**.**

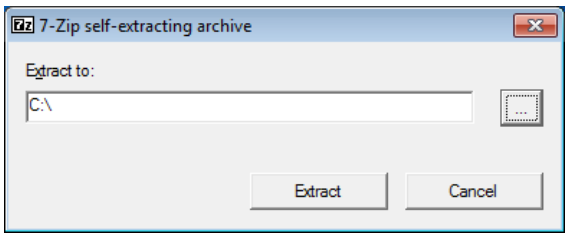


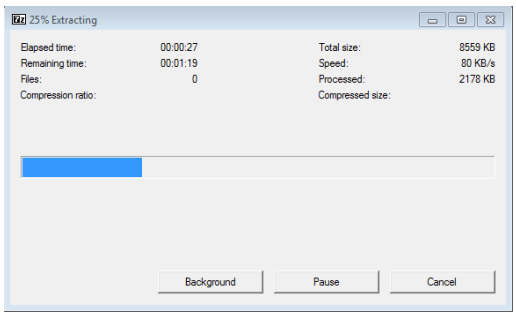
- Configuración de clientes Windows

Posteriormente, se procede a instalar el Snort para Windows, usando el aplicativo anteriormente elaborado. Se selecciona el directorio donde se extraerá el directorio Snort y sus archivos.

En la unidad C, directorio Snort, contiene los sub_directorios de reglas (rules), de registros de eventos (log), y 5 script creados para la administración del Snort.

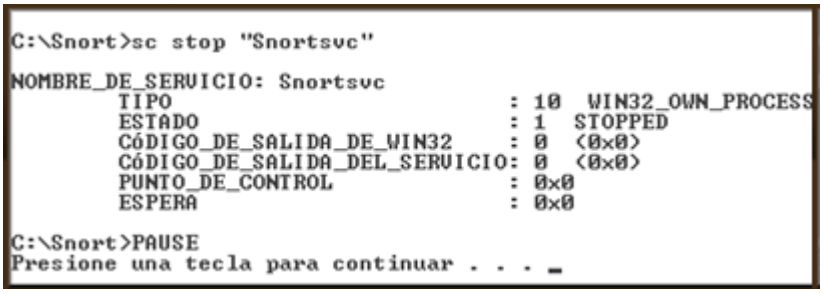
Figura 4-4.: Selección de directorio y extracción de Snort**.**





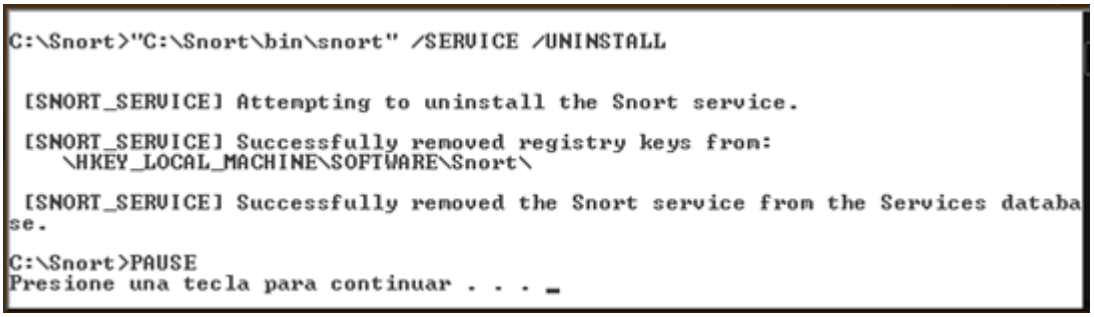
a. Como primera medida, se detiene el servicio Snort ejecutando el scrip Snort_stop_service.bat, en el cliente para evitar conflictos con la implementación del nuevo mecanismo.

Figura 4-5.: Detener servicio de Snort**.**



b. Posteriormente, se desinstala algún servicio que puede estar instalado en el momento, se ejecuta el Script "Snort_uninstall_service.bat".

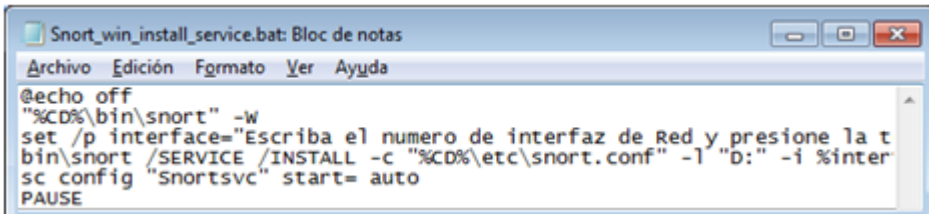
Figura 4-6.: Desinstalar servicio Snort**.**



c. Ahora se procede a ingreso del directorio Snort en la unidad C y se edita el script, Snort_win_install_service.bat, para indicarle la ruta donde se almacenarán los logs, la cual debe ser en la unidad montada con el programa SSHFS.

   En este caso se coloca la ruta de la unidad D.

Figura 4-7.: Modificación de ruta de grabación de archivos de evidencia**.**



- Ejecución de Scrip:

Se ejecuta el Script Snort_win_install_service.bat que permitirá instalar el servicio de Snort al igual que la selección de interfaz por la cual el equipo obtendrá la evidencia digital.

Figura 4-8.: Instalar servicio Snort**.**

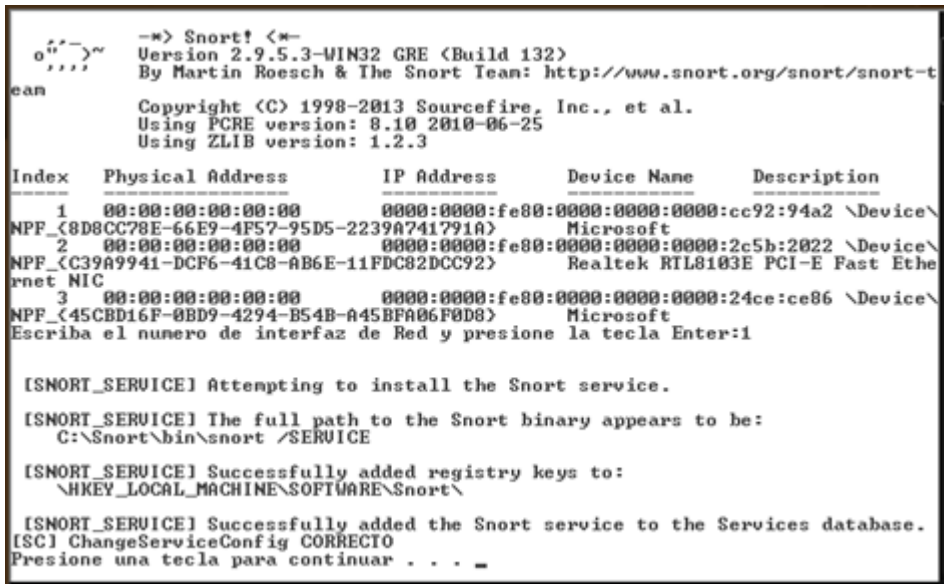
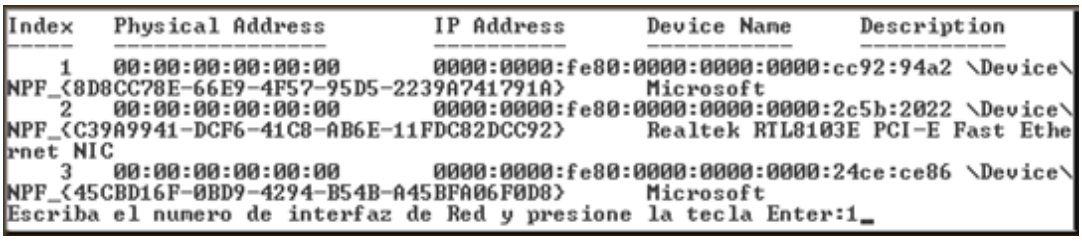
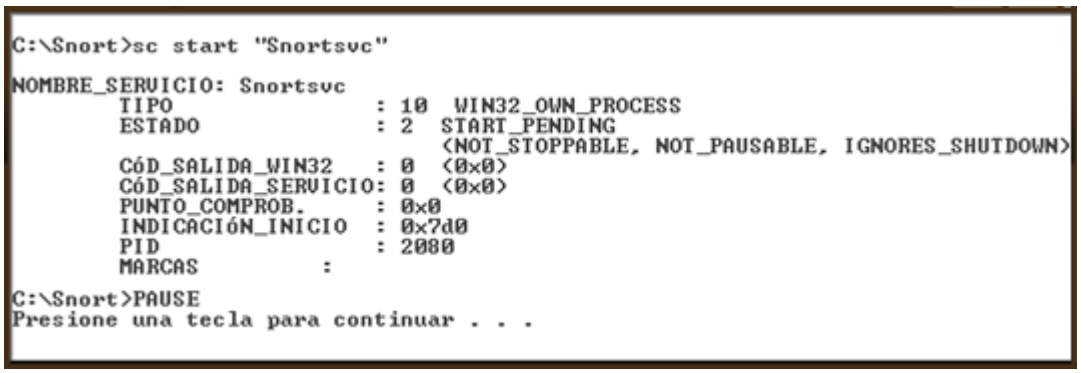


Figura 4-9.: Selección de la interfaz de red donde escuchará Snort**.**



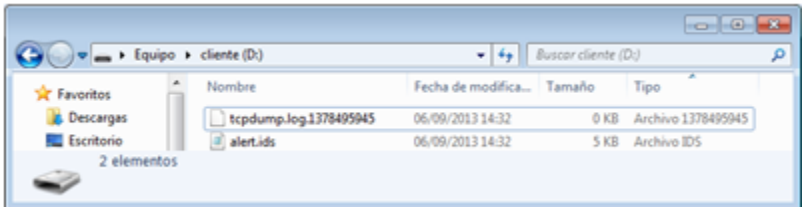
d. Se ejecuta el Script Snort_start_service.bat, que permite iniciar el servicio de Snort.

Figura 4-10.: Iniciar servicio Snort**.**



A continuación se observa el directorio donde se crearon los archivos de evidencia que se actualizan de forma automática con los registros de incidentes.

Figura 4-11.: Directorio de incidentes registrados**.**



Estos archivos son:

Alert.ids: Registra en texto plano los incidentes efectuados por el atacante en la red a un equipo determinado.

tcpdump.log.xxxxxxx: En este archivo registra los incidentes en formato pcap con la diferencia frente al archivo alert.ids, la cual éste solo se genera una vez, pero el tcpdump se construye uno por cada ejecución del Snort.

- Detener Servicio Snort.
  Para detener el servicio de Snort en Windows, se ejecuta el Script Snort_stop_service.bat.

e. Revisión de errores de Snort.

El Script Snort "start scan.bat" tiene dos funcionalidades:

- Revisión de errores: Permite ejecutar el Snort con el fin de mirar errores que puedan estarse generando por una mala construcción de reglas.
- Permite ejecutar el Snort sin necesidad de crear y administrar servicios en Windows.

Figura 4-12.: Verificación de errores de construcción de reglas de Snort**.**

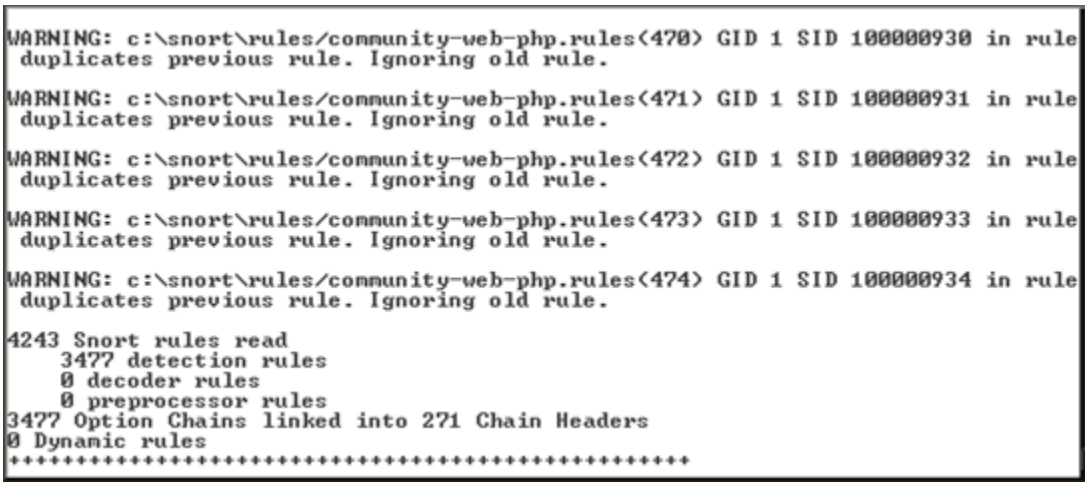
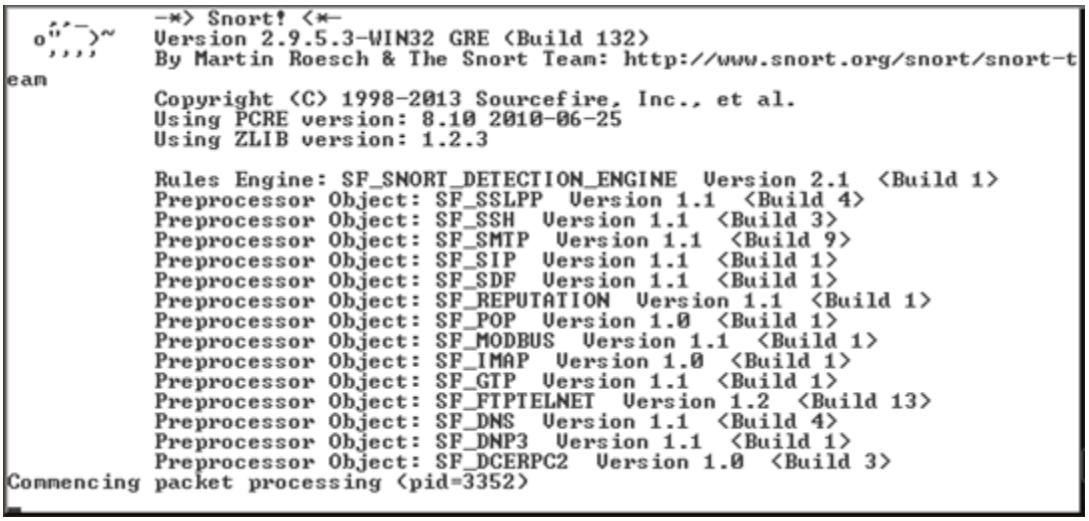


Figura 4-13.: Reglas compiladas de Snort**.**



Finalmente, al verificar que las reglas, los servicios y el directorio donde quedan almacenados los incidentes está correcto, se procede al despliegue del aplicativo HashStamp, que permitirá crear evidencia de una carpeta y verificar la evidencia de la misma, a través del capítulo de validación (Capitulo 5).

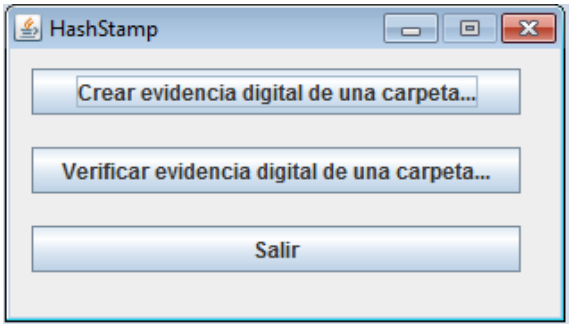
- Despliegue de HashStamp:

Se copia el directorio "dist", con el ejecutable java, en cualquier directorio y se ejecuta usando el siguiente comando: java -jar /<rutadeljar>

Figura 4-14.: Ejecución del programa HashStamp **[49]**.

Aparecerá la ventana:

Figura 4-15.: Ventana del programa HashStamp.



A continuación se relata el funcionamiento de cada opción:

Crear evidencia de un directorio: Efectúa el proceso de obtención de evidencia de un directorio generando una copia comprimida, un archivo hash [50] del zip y un archivo de estampa de estampado cronológico del hash.

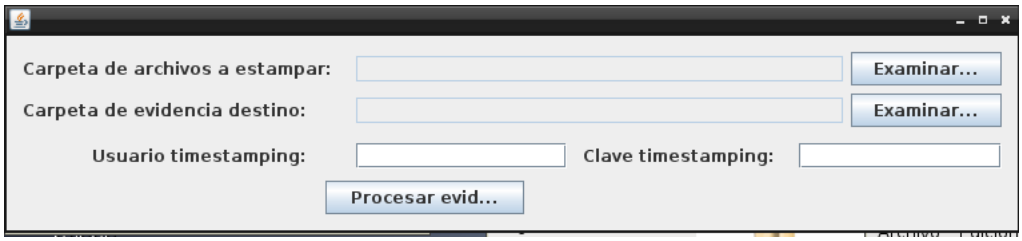
a. ***Validar evidencia de un directorio:*** Valida la estampa del archivo, recibiendo el archivo hash y el archivo de estampado cronológico, usando el certificado de DigiStamp.
b. ***Salir:*** Cierra el programa, cerrando todas las ventanas abiertas.

A continuación, el paso a paso de las opciones del software:

4.1 Crear evidencia de un directorio:

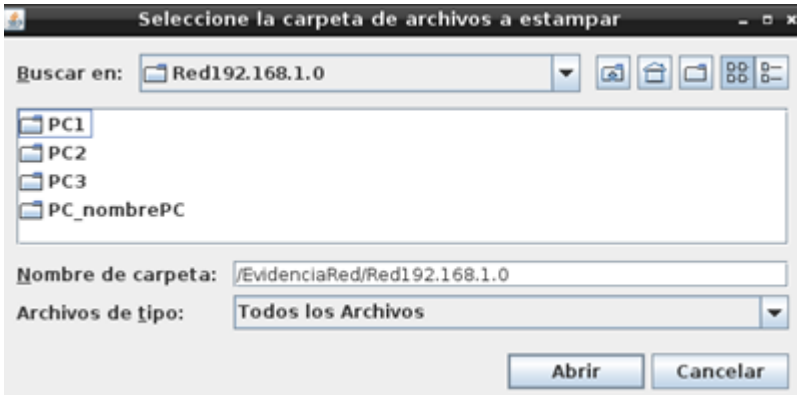
A continuación se procede a c*rear evidencia digital de un directorio por medio del aplicativo HashStamp.*

Figura 4-16.: Usuario y password TimeStamping



El usuario selecciona el directorio de archivos a estampar, para ello hace clic en el botón "Examinar" que se encuentra al lado del cuadro de texto correspondiente al directorio de archivos a estampar:

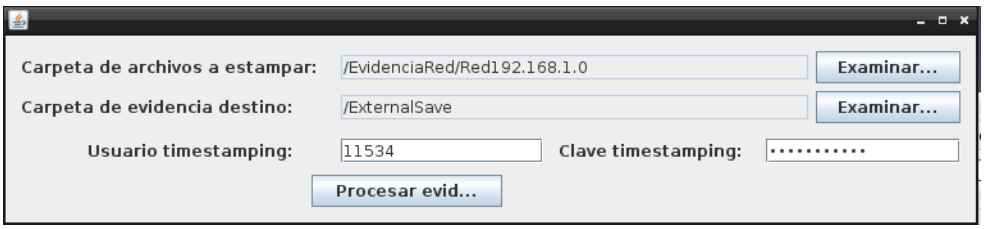
Figura 4-17.: Selección de directorio a estampar y directorio de evidencia destino.



El usuario selecciona el directorio y hace clic en "Abrir".

El usuario repite el proceso anterior con el directorio destino donde quedará el archivo zip con los logs de snort, el archivo hash del zip y el archivo de firma PCKS #7 el cual es el archivo de estampado cronológico. También ingresa el usuario y clave del sistema de estampado cronológico, en este caso el usuario y clave de Digistamp requerido para generar estampas.

Figura 4-18.: Procesar evidencia



El usuario presiona el botón "Procesar Evidencia". Para la generación de la evidencia digital, el software obedece los siguientes pasos:

Se guarda todos los archivos del directorio a obtener la evidencia, comprimiendo el contenido del directorio con la ruta completa. El archivo tomará el nombre: "ev_<*YYYYMMDDHHMISS*>.*zip*"

Se obtiene el hash del archivo generado en el paso anterior, con el algoritmo SHA-512 y se guarda el archivo con el nombre: "hash_<*YYYYMMDDHHMISS*>.*sha*"

Se obtiene el estampado de tiempo del archivo generado en el paso anterior, generando un archivo de firma PCKS 7 con el nombre: "tsr_<*YYYYMMDDHHMISS*>.*p7s*"

Figura 4-19.: Mensaje emergente de estampado exitoso.

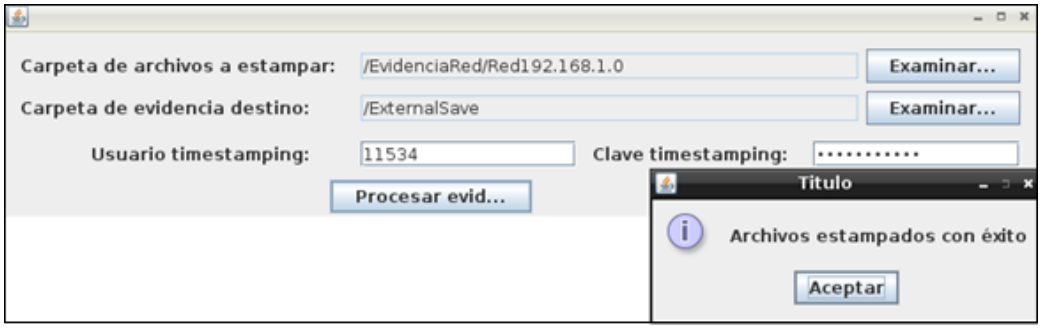
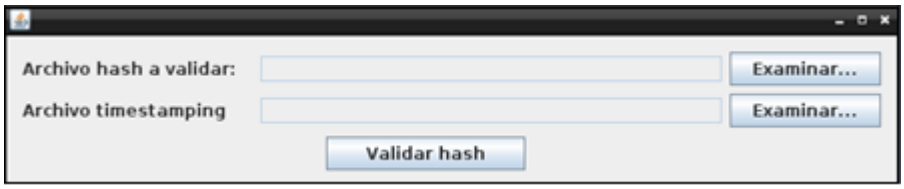


Figura 4-20.: Directorio con hash y estampas de evidencia.

4.2 Verificar evidencia de un directorio:

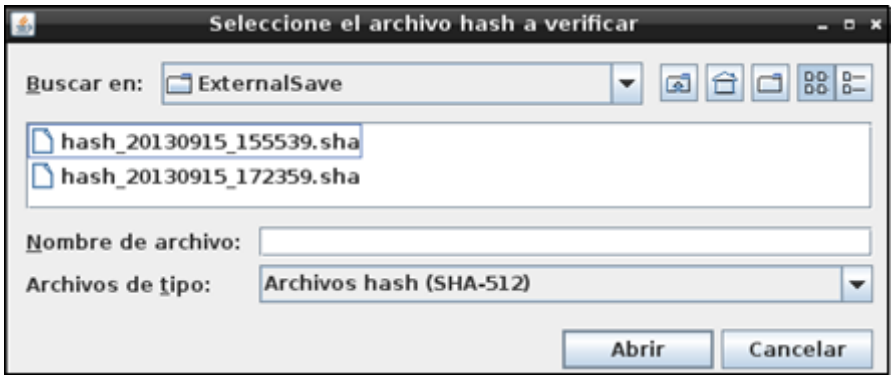
Al presionar la opción *Crear evidencia de un directorio*, aparecerá la siguiente ventana:

Figura 4-21.: Ventana de selección de archivos hash y estampado (HashStamp).



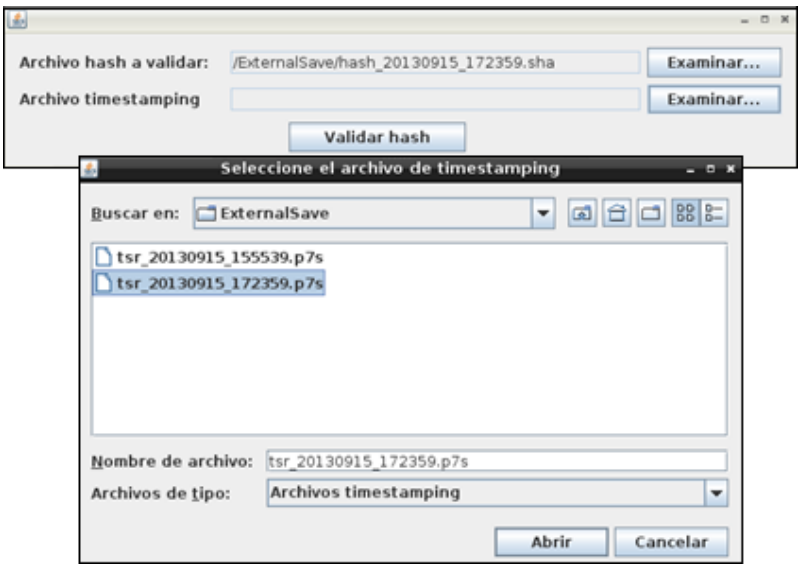
Para verificar la evidencia generada es necesario seleccionar el archivo de hash. Para ello, haga clic en el botón "Examinar..." que se encuentra al lado del cuadro de texto correspondiente a la ruta del archivo hash:

Figura 4-22.: Archivo hash a verificar.



El sistema solo permitirá seleccionar archivos con extensión .sha. Repita el proceso para seleccionar el archivo de estampado cronológico (TimeStamping):

Figura 4-23.: Archivo estampado cronológico (TimeStamping) a comparar.

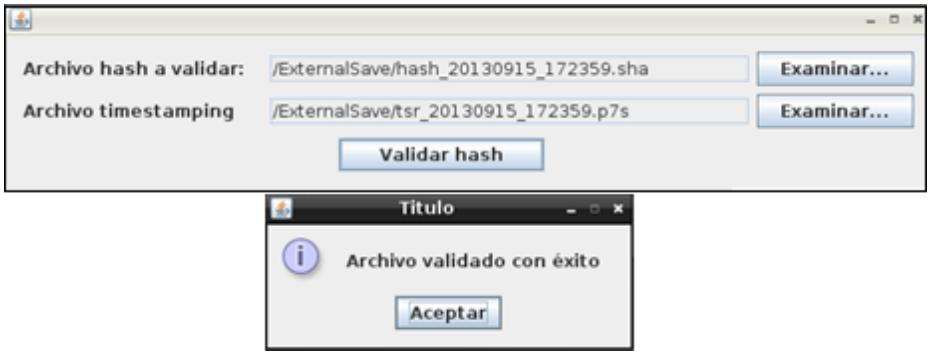


Haga clic en el botón "Procesar Hash".

El proceso de validación de estampado cronológico (TimeStamping) valida el archivo de hash contra el archivo de TimeStamping calculando el hash (SHA1) del archivo de hash, luego valida el TimeStamping contra el certificado.

Si la validación es exitosa, muestra mensaje:

Figura 4-24.: Proceso de validación exitosa.



Caso contrario, en el que no coincide el archivo de hash contra el archivo de estampado cronológico (TimeStamping), muestra mensaje de error indicando que la estampa de tiempo no corresponde al archivo enviado (archivo de hash):

Figura 4-25.: Proceso de verificación no exitosa.

5. Validación y análisis de resultados

En el desarrollo de este capítulo, se valida la implementación mediante pruebas a dos casos específicos, uno, por defecto, con logs predeterminados y el segundo con sus funciones de registro de logs y generación de la evidencia digital.

En primer lugar, en el capítulo 3, se diseñó una arquitectura para la producción de registros electrónicos y evidencia digital de incidentes, para lo cual fue necesario utilizar una solución apoyada por el software Snort e SSHFS, para diseñar y producir registros electrónicos en un servidor remoto.

Posteriormente, se diseñó un aplicativo que permite la transformación de los registros electrónicos, producidos por Snort, en evidencia digital admisible para un juicio, generando copia de los registros electrónicos en un archivo comprimido, obteniendo un archivo hash con el algoritmo SHA-512 [51], garantizando la integridad de la evidencia.

Se evidencia en el capítulo 4, figura 4-11, que los incidentes quedan registrados en dos tipos de archivos generados por Snort, como lo son: los archivos pcap e ids, las cuales, el primero permitió con un analizador de tráfico o interprete de este tipo de archivos, mostrar el incidente ocasionado dentro de la red, y logrando mostrar en el capítulo 5, figura 5-12, en un grado más de detalle aspectos como ip origen, ip destino, fecha, hora. El otro tipo de archivo (ids), registra el incidente en texto plano y fácil de interpretar, permitiendo en poco tiempo identificar lo sucedido, demostrando igualmente el incidente ocasionado sin el nivel de detalle que el anterior; pero con la ventaja que traerá aspectos importantes del ataque como

protocolos utilizados, fecha y hora, ip origen y destino, tipo de ataque utilizado entre otros.

Se validó la implementación mediante pruebas a dos casos específicos, uno con los registro de logs por defecto que lleva la máquina y el segundo con sus funciones de registro de logs y generación de la evidencia digital, la cual el segundo permitió identificar incidentes que ocurrieron en la red mediante ataques a equipos identificados como victimas sin importar el sistema operativo que contengan los mismos; a su vez se demostró que no solamente el ataque puede provenir de otra máquina Windows sino que el atacante en este caso utilizo tres tipos de sistemas operativos como Windows 7, Linux BackTrack y debian wheezy para ingresar a la red y apoderarse de una máquina en su control total.

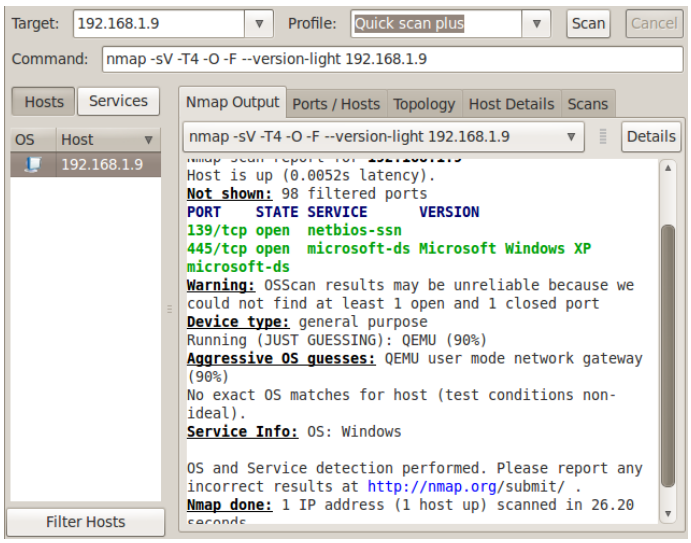
A continuación se detallan los dos tipos de casos que permitieron validar el mecanismo de reporte de incidentes mediante logs generados por snort.

5.1 Caso con Logs predeterminados

A continuación se valida el mecanismo mediante un ataque efectuado por un atacante para vulnerar nuestro sistema sin que se logre evidenciar el mismo:

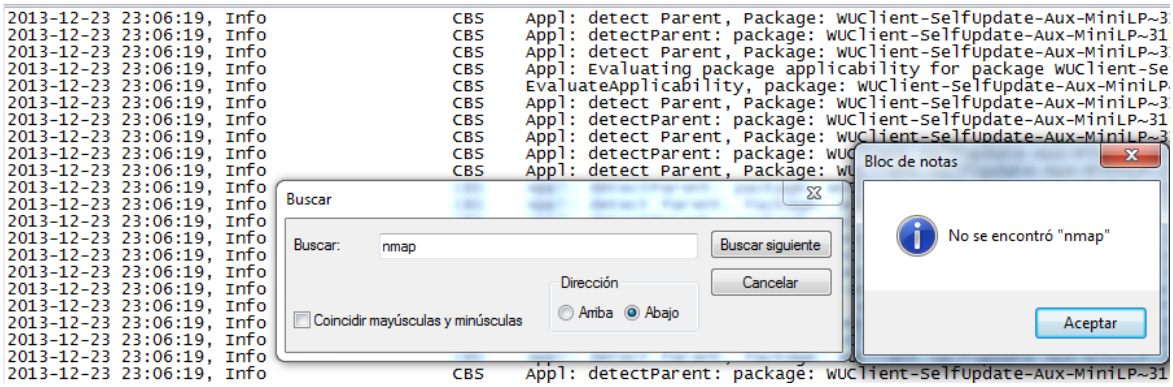
1) Escaneo de puertos a un XP SP3:

Figura 5-1.: Escaneo de puertos Zenmap.



Al ejecutar el anterior escaneo por medio de Zenmap, se evidencia como primera medida los puertos abiertos que cuenta la máquina víctima, en nuestro caso, lo que se requiere, es que el sistema operativo, genere una alerta o cree una base de conocimiento de incidentes para ser utilizados posteriormente en el esclarecimiento de un hecho, pero al ser buscados en el log que contiene Windows, en este caso, la víctima, inmediatamente después del escaneo efectuado, se observa que no contiene el incidente ocurrido por falta de un mecanismo que evidencie estos registros en el momento del ataque.

Figura 5-2.: Búsqueda de incidente en Log de Windows.

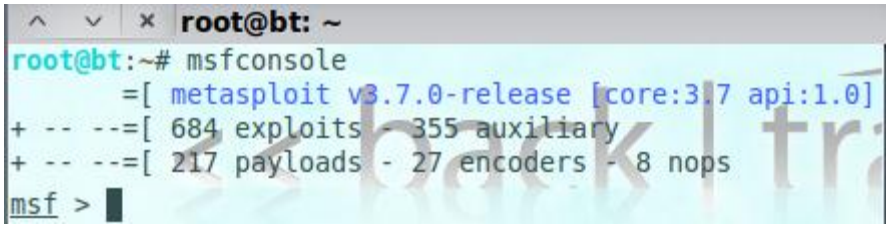


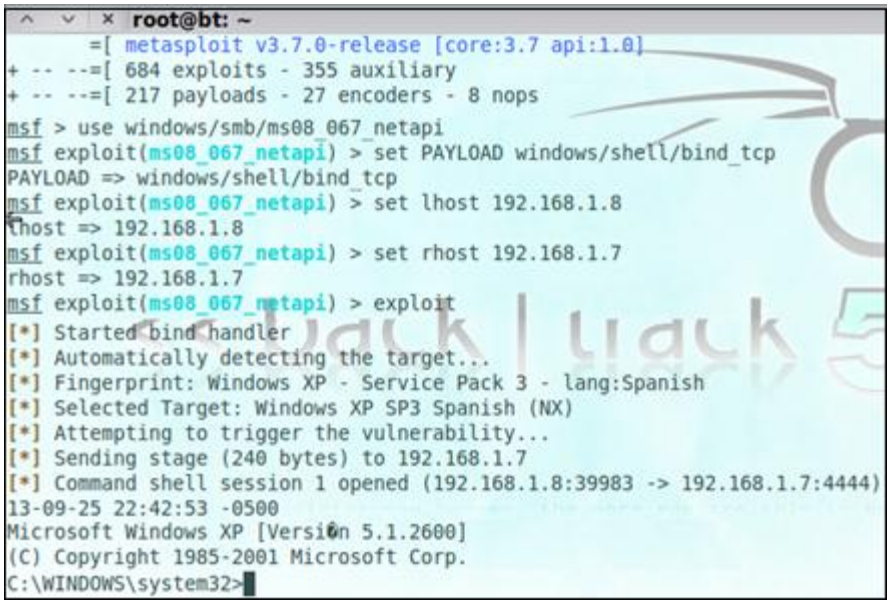
2) Se aprovecha la vulnerabilidad MS08-067 aplicada sobre el servicio RPC de Windows [20].

Figura 5-3.: Ejecución de Metasploit.

Se escriben los parámetros del host origen (LHOST), host destino (RHOST) y PAYLOAD (windows/shell/bind_tcp) para que pueda ser ejecutado con éxito en Windows.

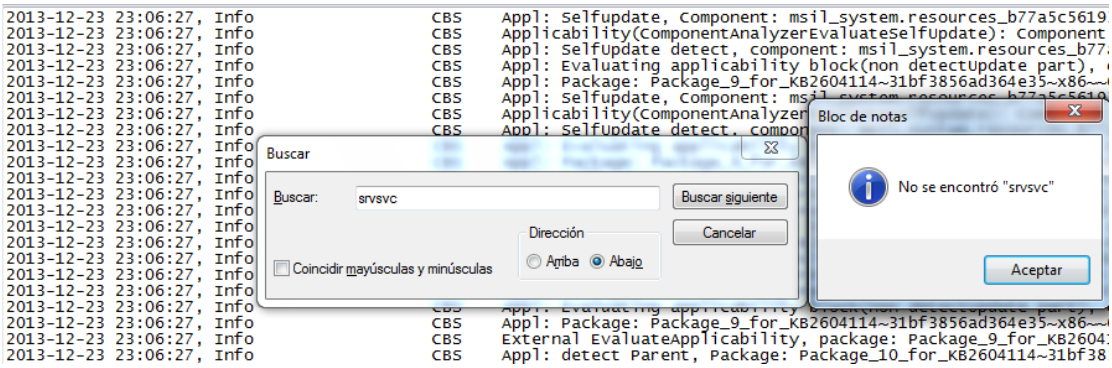
Figura 5-4.: Vulnerabilidad MS08-067 **[52]**.





En este caso, el metasploit logra obtener el control total de la maquina víctima, la cual es comprobado en la anterior figura ya que nos muestra la ruta donde se encuentra el atacante, en este caso en el directorio Windows\system32\, e indicando con el cursor que está a la espera de alguna ejecución por parte de quien efectuó el hecho.

Figura 5-5.: Búsqueda de incidente metasploit en Log de Windows.

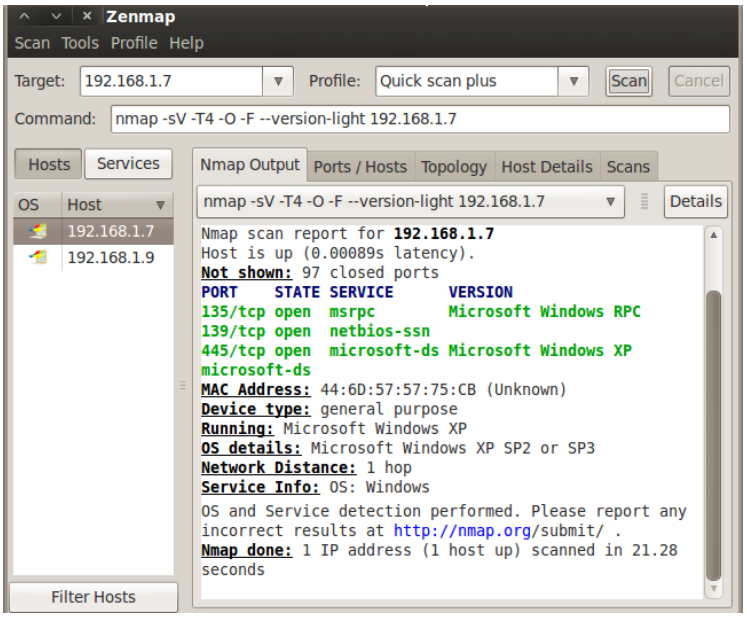


5.2 Caso con la solución propuesta

A continuación se presentan los pasos para la validación de la implementación propuesta, donde se muestra un atacante haciendo escaneo de puertos y un ataque informático, registrándolo usando la solución propuesta:

1) Escaneo de puertos a un XP SP3:

Figura 5-6.: Escaneo de puertos mediante Zenmap **[53]**.

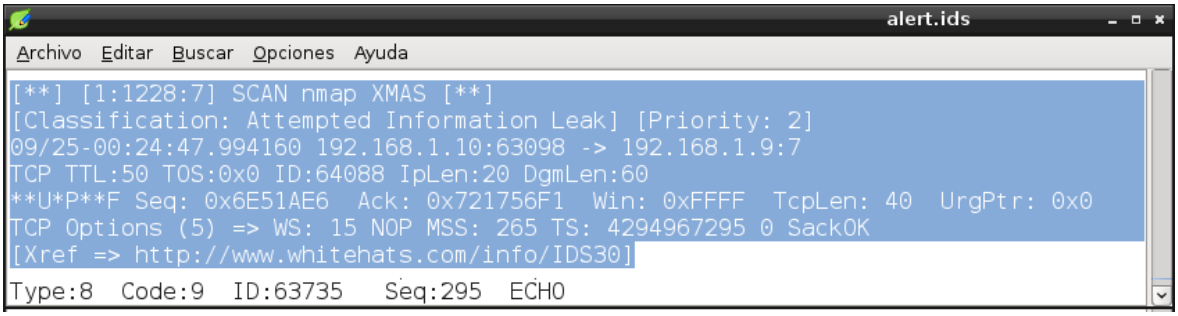


Para verificar que el mecanismo ha generado los respectivos logs de incidentes en formato alert.ids y tcpdump.log A continuación se observa el directorio donde se crearon los archivos de evidencia que se actualizan de forma automática con los datos obtenidos en los registros.

Figura 5-7.: Archivos tcpdump.log y alert.ids**.**

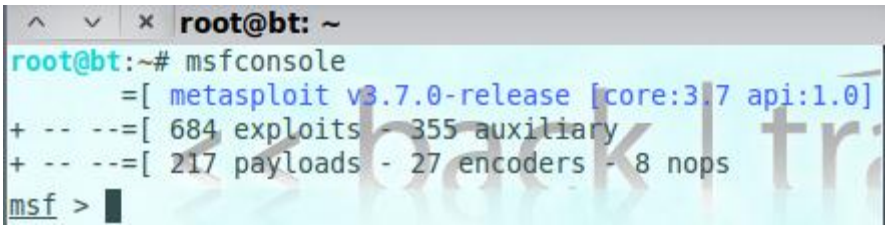
Abrimos el archivo generado como evidencia alert.ids que se encuentra en el servidor
Debian:

Figura 5-8.: Evidencia obtenida en log alert.ids



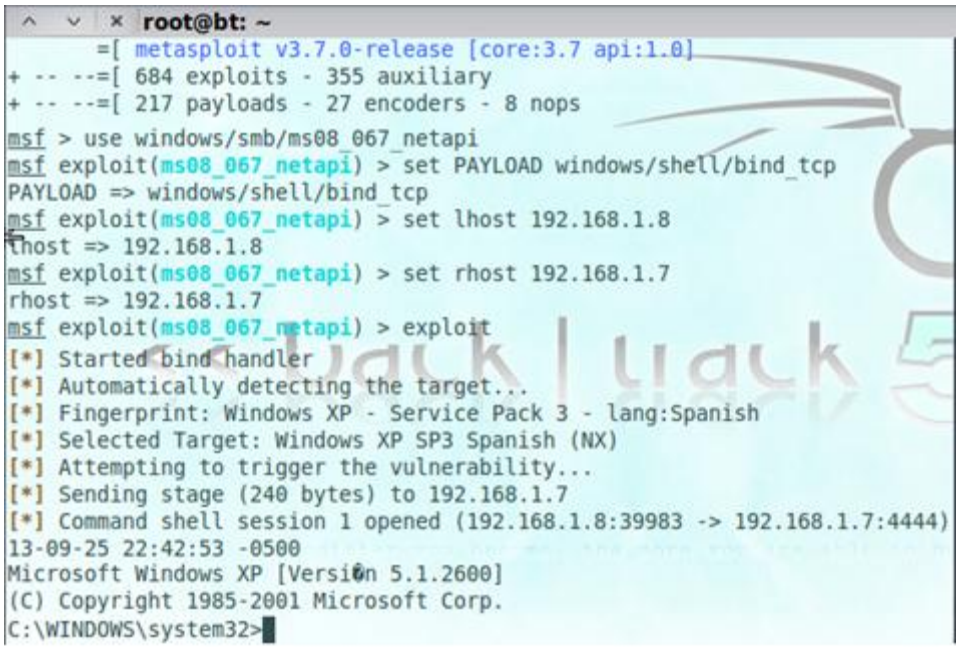
2) Se aprovecha la vulnerabilidad MS08-067 aplicada sobre el servicio RPC de Windows [20].

Figura 5-9.: Vulnerabilidad MS08-067 **[52]**.



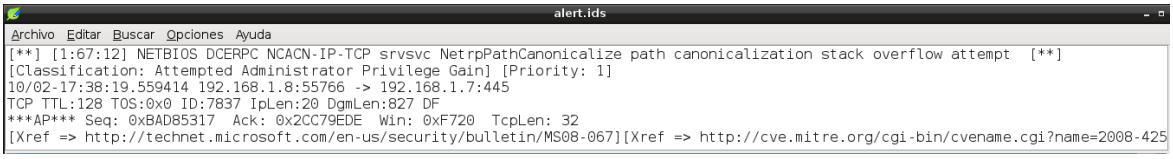
Se escriben los parámetros del host origen (LHOST), host destino (RHOST) y PAYLOAD (windows/shell/bind_tcp) para que pueda ser ejecutado con éxito en Windows.

Figura 5-10.: Metasploit ejecutado con éxito.



Se efectuó el ataque con éxito, registrándose en el log del Snort en el servidor de evidencia digitales:

Figura 5-11.: Evidencia en log de ataque metasploit.



Se ejecuta el programa javaTimesStamping.jar, usando el comando *java -jar <ruta delTimesStamping.jar>/TimesStamping.jar.*

Figura 5-12.: Evidencia de log de ataque metasploit mediante Wireshark

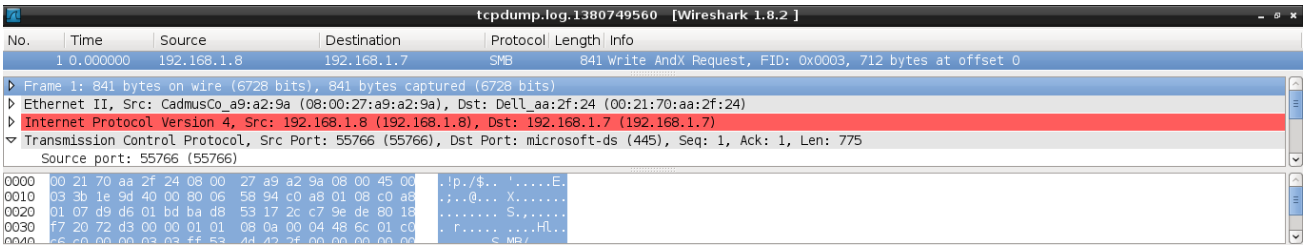


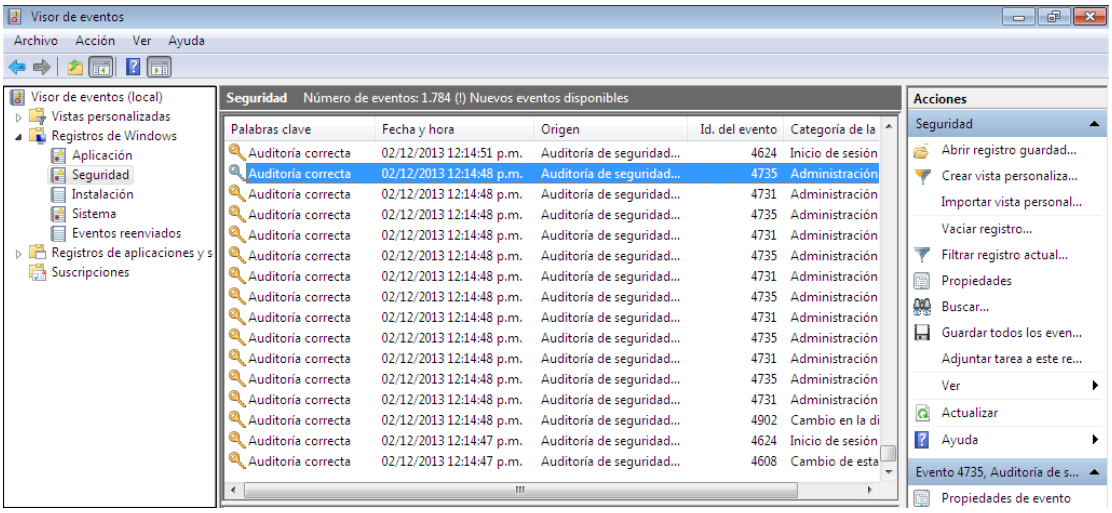
Tabla 2: Comparación de casos propuestos.

	IDS	HASH	ESTAMPA	REGLAS	LOGS
SIN CASO PROPUESTO	* En el visor de Windows, se encuentran los registros, la cual contienen las aplicaciones, seguridad, instalación, sistema y eventos reenviados que permiten registrar los eventos ocurridos. Cabe anotar que estos eventos son volátiles y que son borrados con solo apagar la maquina o dándole clic en vaciar registro. Figura [5-13, 5-14].	* En Windows el fichero SAM se almacena en la ruta C:\Windows\ system32\config\sam, Figura [5-15], aunque si se intenta acceder desde el mismo Windows no se podrá ya que no se tiene permisos para ver este fichero y además está en binario. Gracias a un investigador francés apodado "Gentil Kiwi", ya es accesible en texto claro por medio de un aplicativo [54]. * En Linux Kali, existe una herramienta que permite sacar los has de md5 y sha1 pero no el propuesto SHA-512 [55].	* Múltiples empresas prestan el servicio de estampado cronológico como DigiStamp, certicamaras, entre otras, en los diferentes sistemas operativos, pero aun así se requiere de que la entidad envíe un perito forense experto para realizar dicha tarea [56].	* En Windows no se pueden modificar las reglas, es por esta razón que Microsoft creó una herramienta denominada System Center 2012 R2 Configuration Manager que permite establecer directivas de seguridad y supervisar el estado al mismo tiempo que le das acceso a los usuarios a las aplicaciones preferidas de los dispositivos que elijan [57]. * En Linux al igual que en Windows, no se pueden crear reglas de incidentes, Se puede establecer reglas iptables para enrutar el tráfico a ciertas máquinas, tales como a un servidor HTTP o FTP dedicado [58].	* En Windows, los logs se encuentran en la ruta C:\Windows\Logs\CBS, la cual contiene un archivo llamado CBS.log, la cual contiene los eventos realizados hasta el momento; sin embargo, este tipo de archivo se puede alterar como un archivo de texto simple. Figura [5-16, 5-17]. * En linux los encontramos en la carpeta /var/logs, pero al igual que en Windows podemos modificarlos o borrarlos según sea el caso. Figura [5-18].
CON CASO PROPUESTO	* Decodificador del paquete. * Motor de detección (Comparación contra firmas).	* El SHA-512 que genera el aplicativo, permite evitar ataques de fuerza bruta para descifrar el algoritmo. * Evita ataques frente a colisiones. Figura [5-19].	* Aunque muchas entidades ofrecen el servicio de estampado cronológico, ninguna lo hace	* Zenmap * MS08-067 * Sin importar el tipo de incidente, las reglas en el aplicativo se pueden actualizar o crear dado el	* Los logs no se guardan en la máquina (Victima), para evitar modificar la evidencia, sino en un servidor remoto.

	* Loggin y sistema de alerta. * Plugins de salida.	* Se crea directorio con hash y estampas de evidencia. Figura [4-20].	automáticamente mediante un aplicativo o sin la presencia de un perito informático. * Se crea directorio con hash y estampas de evidencia. Figura [4-20].	caso, cada vez que se requiera.	* Los logs se envían a través de un túnel SSH, instantáneamente en el momento de su elaboración para evitar ataques en el medio (man in the middle) [6], de la evidencia digital por el medio la cual están organizados. * Permite guardar registros electrónicos de acuerdo a cada computador en la red. * Se muestra el registro del ataque también en formato pcap que permite ser abierto en un analizador de tráfico como Wireshark para un mayor detalle del incidente. Figura [5-12].

Detalle de tabla 2. Casos propuestos.

Figura 5-13.: Visor de eventos en Windows



En la anterior figura, podemos detallar el visor de Windows la cual contiene los registros de Windows incluido la parte de seguridad.

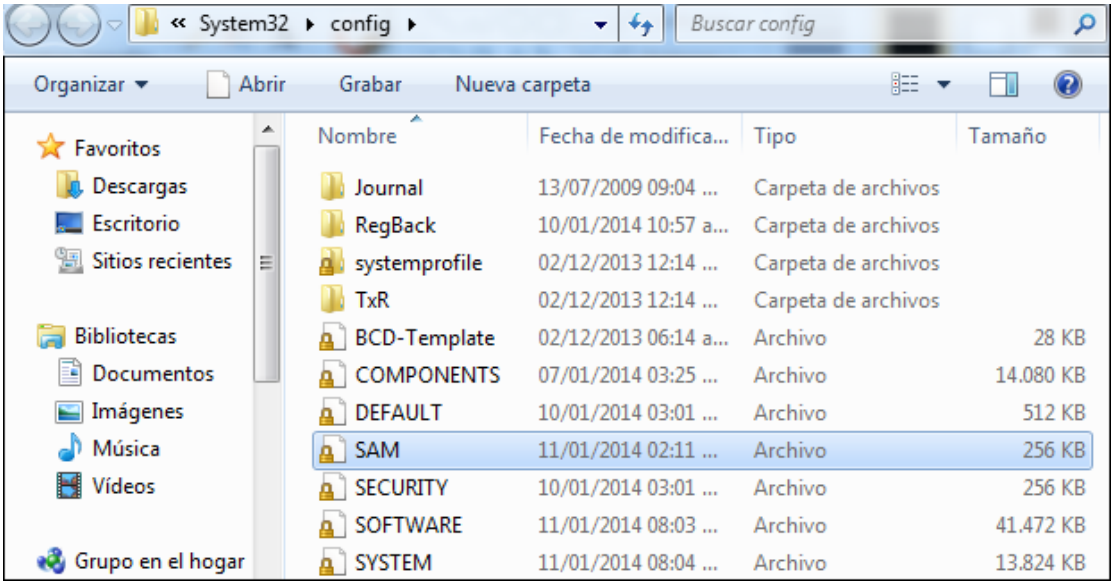
Figura 5-14.: Registro de windows

Registro de Windows

- SYSTEM: configuración del servicios y hardware. (HKEY_LOCAL_MACHINE\SYSTEM).
  - Nombre del Equipo.
  - Zona Horaria.
  - Interfaces de Red y su configuración.
  - Redes Wireless a las que se ha conectado el equipo.
  - Programas de inicio automático.
  - Recursos Compartidos del sistema.
  - Fecha de ultimo apagado del equipo y número de veces que se apagó el equipo.
- NTUSER.dat: (HKEY_CURRENT_USER).
  - Historial de Búsqueda: NTUSER.dat\Software\Microsoft\Search Assistant\ACMru
  - URLs Tipeadas: NTUSER.dat\Software\Microsoft\Internet Explorer\TypedURLs
  - Documentos Recientes: NTUSER.dat\Software\Microsoft\Windows\CurrentVersion\Explorer\RecentDocs
  - Ventanas de Diálogos Open/Save:
    - Save: NTUSER.dat\Software\Microsoft\Windows\CurrentVersion\ComDlg32\OpenSavePIDIMRU
    - Open: NTUSER.dat\Software\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32\OpenSaveMRU
  - Último comando ejecutado: NTUSER.dat\Software\Microsoft\Windows\CurrentVersion\Explorer\RunRMU
  - Historial de Ejecución de Aplicaciones: NTUSER.dat\Software\Microsoft\Windows\CurrentVersion\Explorer\UserAssis\{GUID}\count

En la figura -14, se observa el registro de Windows, la cual contiene un reporte exacto de la máquina, tanto de hardware, software instalado.

Figura 5-15.: Hash en windows



En la anterior grafica se ubica el directorio la cual contiene el hash de Windows llamado SAM.

Figura 5-16.: Logs en windows

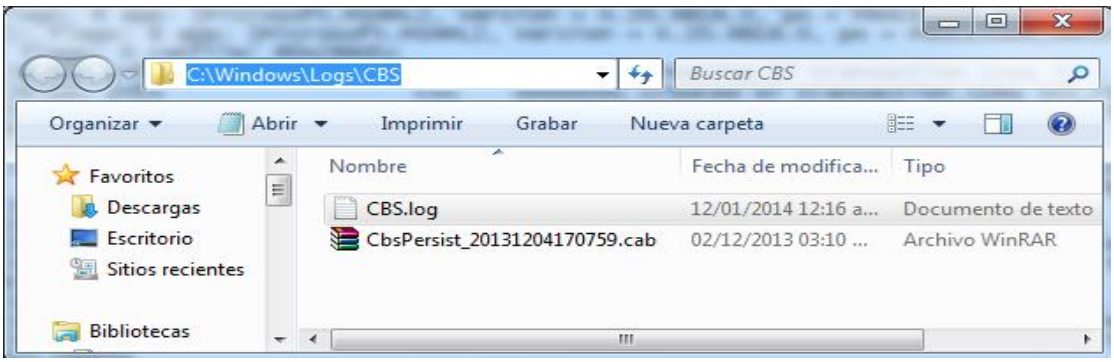
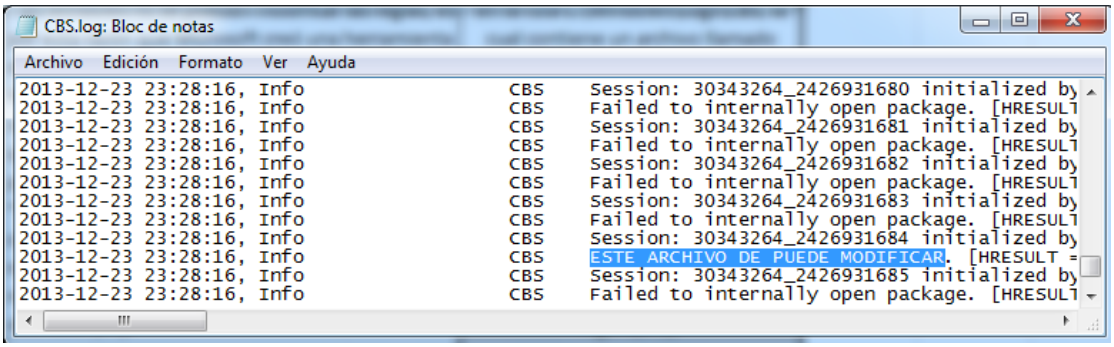


Figura 5-17.: Logs en Windows y su alteración



El log CBS.log, o log de Windows, puede ser alterado tan solo con un editor de texto y permitiendo guardar a su vez los cambios efectuados.

Figura 5-18.: Logs de eventos en Linux

Archivos de registro comunes (pueden variar según la distro):

- /var/log/message: registro de mensajes generales del sistema
- /var/log/auth.log: log de autenticación
- /var/log/kern.log: registro del kernel
- /var/log/cron.log: registro de crond
- /var/log/maillog: registro del servidor de mails
- /var/log/qmail/ : registro de Qmail
- /var/log/httpd/: registro de errores y accesos a Apache
- /var/log/lighttpd: registro de errores y accesos a Lighttpd
- /var/log/boot.log : registro de inicio del sistema
- /var/log/mysqld.log: registro de la base de datos MySQL
- /var/log/secure: log de autenticación
- /var/log/utmp or /var/log/wtmp : registro de logins

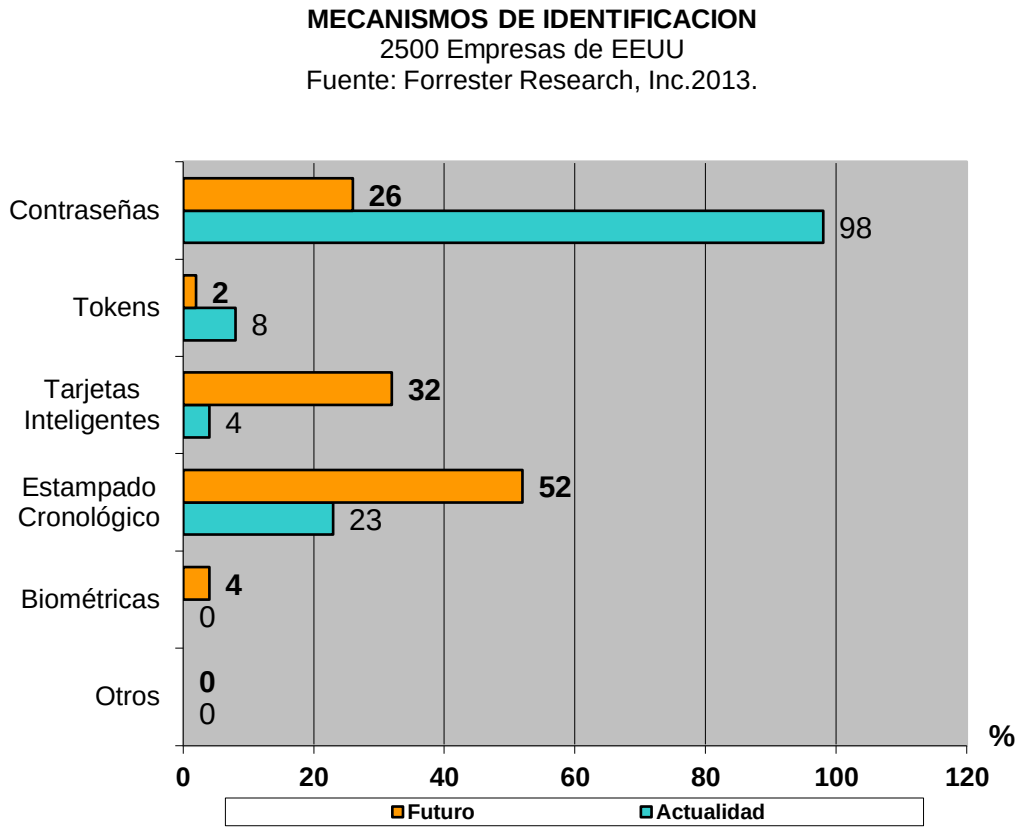
Igualmente que en Windows, Linux cuenta con su propio registro de logs la cuales se encuentran ubicados en el directorio que se presenta en la grafica 5-18.

Figura 5-19.: Comparación de los principales algoritmos **[59]**.

Algoritmo	Tamaño De Salida	Tamaño De La Logitud	Tamaño De Palabra	Rondas	ATAQUE MAS CONOCIDO / Colisiones
MD4	128	64	32	48	SI (3)
MD5	128	64	32	64	SI ($2^{20.96}$)
SHA-0	160	64	32	80	SI ($2^{33.6}$)
SHA-1	160	64	32	80	SI (2^{51})
SHA-256/224	256/224	64	32	64	Teoricamente [4] ($2^{28.5}$-24)
SHA-512/384	512/384	128	64	80	Teoricamente [4] ($2^{32.5}$-24)
SHA-512	224/256/384/512[5]	-	64	24	No

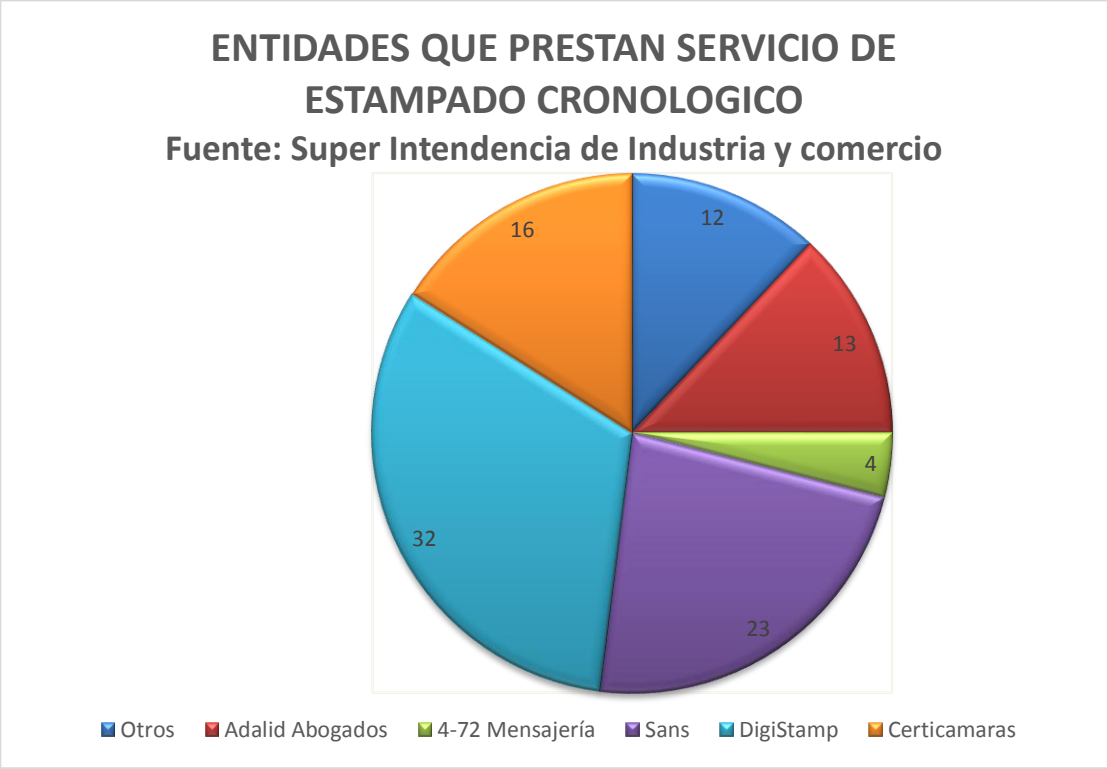
En la anterior gráfica, se puede determinar que el SHA-512, que genera el aplicativo, permite evitar ataques de fuerza bruta para descifrar el algoritmo, además de evitar ataques de colisiones. Figura [5-19].

Figura 5-20.: Estadística de utilización de estampado cronológico **[60]**.



La anterior estadística, muestra la tendencia actual y futura del estampado cronológico frente a otros medios de identificación, la cual se puede observar que aunque no muchas empresas o personas en general conocen o utilizan el servicio, es el segundo más utilizado hasta el momento y con tendencia a subir.

Figura 5-21.: Entidades que prestan el servicio de estampado cronológico **[61]**.



La estadística de utilización del estampado cronológico de algunas entidades está dadas frente al costo que tiene cada entidad y el servicio que este ofrece, aunque todas cuenten con lo reglamentario para que la evidencia estampada sea válida ante un tribunal.

Otras entidades que utilizan el servicio de estampado cronológico son mencionadas igualmente por la Super Intendencia de Industria y Comercio [62].

- Personas Jurídicas
- Públicas o privadas
- Nacionales o extranjeras
- Cámaras de comercio
- Notarías ó consulados

6. Conclusiones y recomendaciones

6.1 Conclusiones

Los archivos obtenidos hash, mediante el algoritmo SHA-512 que genera el aplicativo, garantizan la integridad de la evidencia, con un tamaño de la salida de 512^5, la cual frente a los utilizados actualmente, cuentan con el mismo sistema de cifrado pero ocasionan colisiones dentro del mismo, como es descrito en la figura[5-7], del capítulo de validación y análisis de resultados.

Los registros electrónicos de incidentes, obtenidos en evidencia digital, teniendo en cuenta la integridad mediante el algoritmo (SHA-512), sufren transformación con la construcción del Snort para Windows (y sus respectivas reglas); como se observa en el capítulo 3, sección 3.2.

En la construcción de Snort para Windows, se muestra la creación de Scrips que permiten, instalar, detener, y desinstalar el servicio, al igual que la verificación de errores de construcción de reglas y su administración de forma automática administrando Snort, frente al propuesto por Sourcefire, como se puede verificar en el parágrafo 3-2, figura[3-14].

Los archivos obtenidos mediante el algoritmo SHA-512, no se guardan en la máquina (Victima), para evitar modificar la evidencia, viajan por un túnel, por medio del protocolo SSH, instantáneamente en el momento de su creación, para evitar ataques en el medio, y permitir guardar los registros electrónicos de acuerdo a cada computador en la red, figura [4-17], como se puede comprobar en el capítulo 2, figura [2-11].

El aplicativo HashStamp, permite sacar estampado cronológico de los archivos mostrándolos después de su ejecución, en un directorio con hash y estampas de evidencia, figura [4-20], la cual demuestra que se hace de forma automática frente a

las diferentes entidades que prestan este servicio (figura 5-21), como se describe en el capítulo 5, parágrafo 5-2, referente al caso con solución propuesta, figura [4-15].

6.2 Recomendaciones

Además de implementar esta solución en redes inalámbricas, se recomienda implementarla en cualquier dispositivo que soporte la instalación de Snort, tales como PC de escritorio, servidores, garantizando la seguridad en toda la red para que así los incidentes sean identificados y grabados en logs que permitirán ser utilizados posteriormente como evidencia frente a un tribunal para esclarecer o afirmar un posible delito informático.

6.3 Trabajos futuros

Esta solución puede ser mejorada instalando Snort en un router que soporte OpenWRT, las cuales permiten la producción de registros electrónicos y redireccionar los mismos hacia un servidor remoto, para evitar un ataque en el medio y fortaleciendo lo planteado en el proyecto.

Por otra parte, se podría incorporar alarmas mediante plugins (SNMP), indicando al administrador de la red el incidente ocurrido y así tomar una acción frente a este hecho.

7. Bibliografía

[1] E. C., Digital Evicence and Computer Crime, 1St edition ed., A. Press, Ed., MG-Hill, 2000, p. 4.

[2] J. J. C. Martínez, El Peritaje Informático y la Evidencia Digital en Colombia: Conceptos, Retos y Propuestas, Primera Edicion ed., U. D. L. Andes, Ed., Bogota: Universidad de los Andes, Facultad de Derecho, 2010, p. 23.

[3] Insegure.org, «Las 75 Herramientas de Seguridad Más Usadas,» 05 2003. [En línea]. Available: http://insecure.org/tools/tools-es.html. [Último acceso: 01 07 2013].

[4] K. J. C. &. C. Gerg, Managing Intrusion Detection with Open Source Tools - Snort and IDS Tools, Primera edicion ed., M. Loukides, Ed., O´reilly Media Inc, 2004, p. 03.

[5] I. SOURCEFIRE, «Snort,» Snort, 16 09 2013. [En línea]. Available: http://www.snort.org/snort-downloads?. [Último acceso: 16 09 2013].

[6] S. Snort, «Snort,» SourceFire, 01 11 2013. [En línea]. Available: http://www.snort.org/. [Último acceso: 17 12 2013].

[7] A. Abogados, «Adquisición de Imagenes Forenses,» Bogota, 2013.

[8] F. I. P. STANDARDS, «Secure Hash Standard (SHS),» United States Of America, 2012.

[9] I. 27005:2008, «27005:2008,» 15 06 2008. [En línea]. Available: http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=42107. [Último acceso: 02 11 2013].

[10] F. Yergeau, «INTERNET STANDARD,» 11 2003. [En línea]. Available: http://tools.ietf.org/html/rfc3629. [Último acceso: 10 08 2013].

[11] E. R. Harold, «Java I/O,» 2nd Edition ed., O'Reilly Media, Inc., 2006, p. 474.

[12] T. Groussard, «JAVA 7,» de *Los fundamentos del lenguaje Java*, Cataluña, Estelle Dechenaud, 2012, p. 16.

[13] T. I. S. w. p. c. library, «WinPcap,» Riverbed Technolog, 2013. [En línea].

Available: http://www.winpcap.org/. [Último acceso: 01 07 2013].

[14] SteveMcIntyre, «SSH,» Debian.org, 19 06 2013. [En línea]. Available: https://wiki.debian.org/SSH. [Último acceso: 10 12 2013].

[15] J. R. A. Alfonso Muñoz Muñoz, Cifrado de las comunicaciones digitales: de la cifra clásica al algoritmo RSA, OxWORD Computing, 2013, 2013.

[16] U. d. Extremadura, «Computación cuántica,» *Hiperenciclopédica de divulgación del saber,* vol. Vol.8, nº No.2, 204.

[17] M. M. R. M. Á. A. Isidro Valentín Rodríguez, «Estructura de Qubits como un Factor de Realce de Coherencia en una Computadora Cuántica de un Solo Camino,» *Centro de Investigación en computación,* vol. v.11 n.4, nº ISSN 1405-5546, 2008.

[18] J. Phillips, «Darknet: la Internet oculta,» 04 11 2011. [En línea]. Available: http://www.lagranepoca.com/darknet-la-internet-oculta. [Último acceso: 30 09 2013].

[19] Ron Bowes, «Scanning Windows Deeper With the Nmap Scanning Engine,» SANS Institute InfoSec Reading Room, 19 06 2009. [En línea]. Available: http://www.sans.org/reading-room/whitepapers/testing/scanning-windows-deeper-nmap-scanning-engine-33138. [Último acceso: 15 12 2013].

[20] M. Corporation, «Cómo configurar la asignación dinámica de puertos RPC para trabajar con servidores de seguridad,» 02 12 2013. [En línea]. Available: http://support.microsoft.com/kb/154596/es. [Último acceso: 28 12 2013].

[21] N. S. F. (. S. R. I. Susan H. Nycum, The criminal law aspects of computer abuse: applicability of the state penal laws to computer abuse, Menlo Park, California: Stanford Research Institute, 1976.

[22] N. Unidas, «Congreso de las Naciones Unidas sobre Prevención del Delito y Justicia Penal,» de *Novedades recientes en el uso de la ciencia y la tecnología por los delincuentes y por las autoridades competentes en la lucha contra la delincuencia, incluido el delito cibernético*, Salvador, Brasil, 2010.

[23] U. I. d. T. R. j. c. e. ciberdelito, EL CIBERDELITO: GUÍA PARA LOS PAÍSES EN DESARROLLO, D. d. A. T. y. C. (CYB), Ed., Suiza, 2009, p. 79.

[24] S. M. Markus Jakobsson, Phishing and Countermeasures: Understanding the Increasing Problem of Electronic Identity Theft, Wiley, 2007, p. 01.

[25] C. Elisan, «Security smarts for the self-guided IT professional,» 2012.

[26] K. Jamsa, Cloud Computing, Primera Edicion ed., A. L. Company, Ed., Jones & Bartlett Publishers, 2013, p. 01.

[27] I. Walden, Computer Crimes and Digital Investigations, Oxford University Press, 2007, p. 01.

[28] I. STANDARD, ISO/lEC 27005, vol. First edition, Switzerland, 2008, p. 18.

[29] I. STANDARD, ISO/lEC 27005, First edition ed., 2008, p. 3.5.

[30] G. B. R. Cano Martínez J. José, Universidad de los Andes, Bogota, 2012.

[31] A. Ghosh, «Guidelines for the Management of IT Evidence,» UNPAN, 21 03 2004. [En línea]. Available: http://unpan1.un.org/intradoc/groups/public/documents/apcity/unpan016411.pdf. [Último acceso: 01 05 2012].

[32] S. V. N. review, «Compare Snort VS Nessus review: Snort-70%, Nessus-30%.,» 08 2013. [En línea]. Available: http://azatvs.com/snort-VS-nessus-1950376. [Último acceso: 24 09 2013].

[33] E. HACKING, «ALTERNATIVAS A NESSUS: COMPARATIVA ENTRE NESSUS, OPENVAS Y RETINA,» 13 05 2012. [En línea]. Available: http://www.el-palomo.com/2012/05/alternativas-a-nessus-comparativa-entre-nessus-openvas-y-retina/. [Último acceso: 25 09 2013].

[34] J. Bertolin, Seguridad de la Información, Madrid: Paraninfo, 2008, pp. 310,311.

[35] H. Y. Xiaoyun Wang, «How to Break MD5 and Other Hash Functions,» 2005.

[36] L. Á. Q. V. E. M. Y. M. Siler Amador Donado, «Colisiones en el algoritmo de ciframiento SHA-1,» *Revista Generación Digital,* vol. Vol. 8, nº No. 1., pp. 1-2, 05 06 2009.

[37] V. MEXICANH, « TIPOS DE HASHES,» *HASHCAT,* p. 20, 11 2013.

[38] S. d. l. R. d. Colombia, «Ley 1273 del 2009,» 05 01 2009. [En línea]. Available: http://www.secretariasenado.gov.co/senado/basedoc/ley/2009/ley_1273_2009.html. [Último acceso: 23 03 2011].

[39] E. c. d. Colombia, «Ley 527 de 1999,» 18 08 1999. [En línea]. Available: http://www.secretariasenado.gov.co/senado/basedoc/ley/1999/ley_0527_1999.html. [Último acceso: 01 06 2011].

[40] D. GNU/Linux, «wheezy,» 04 05 2013. [En línea]. Available: http://www.debian.org/releases/stable/. [Último acceso: 28 05 2013].

[41] M. Sharma, Openfire Administration", A Practical Step-by-step guide to rolling out a secure instant messaging service over your network, Primera Edicion ed., Packt Publishing, 2008, p. 26.

[42] M. Corporation, «Windows® XP Service Pack 3,» 06 05 2013. [En línea]. Available: http://www.microsoft.com/es-co/download/details.aspx?id=24. [Último acceso: 05 02 2013].

[43] A. K. Harnal, Linux Aplications and Administration, Cuarta Edicion ed., New Delhi: McGraw-Hill, 2009, p. 477.

[44] E. HACKING, «ALTERNATIVAS A NESSUS: COMPARATIVA ENTRE NESSUS, OPENVAS Y RETINA,» 13 05 2012. [En línea]. Available: http://www.el-palomo.com/2012/05/alternativas-a-nessus-comparativa-entre-nessus-openvas-y-retina/. [Último acceso: 02 08 2013].

[45] Anonymous, «Carna botnet scans confirmed,» 17 03 2013. [En línea]. Available: http://blog.caida.org/best_available_data/2013/05/13/carna-botnet-scans/. [Último acceso: 15 12 2013].

[46] P. Phongthiproek, «The Operation CouldBurst Attack,» 18 02 2010. [En línea]. Available: http://www.exploit-db.com/papers/13621/. [Último acceso: 15 12 2013].

[47] A. Technology, «Metasploit, Framework, Demystyfied,» *The exploit Magazine,* pp. 12-20, 08 2012.

[48] F. Software, «Notepad++,» 29 09 2013. [En línea]. Available: http://notepad-plus-plus.org/. [Último acceso: 29 09 2013].

[49] CERES, «SELLADO DE TIEMPO O TIMESTAMPING,» 23 11 2005. [En línea]. Available: http://www.cert.fnmt.es/index.php?cha=com&sec=13&page=109&lang= es. [Último acceso: 09 02 2013].

[50] J. Bertolin, Seguridad de la Información, Madrid: Paraninfo, 2008, p. 517.

[51] J. B. Juan Pablo Sarubbi, Seguridad Informatica - Tecnicas de defensa comunes bajo variantes del sistema operativo Unix, U. N. d. Luján, Ed., Luján, 2008, p. 3.4.2.

[52] M. Corporation, «Microsoft Security Bulletin MS08-067 - Critical,» Microsoft, 23 10 2008. [En línea]. Available: http://technet.microsoft.com/en-us/security/bulletin/ms08-067. [Último acceso: 01 10 2013].

[53] B. P. Angela Orebaugh, Nmap in the Enterprise: Your Guide to Network Scanning, Primera Edicion ed., S. Publishing, Ed., United States Of America: Elsevier, Inc, 2008, p. 138.

[54] A. Marki, «recuperar hash de sam,» 03 10 2012. [En línea]. Available: http://www.youtube.com/watch?v=J_F9CtcSxm8. [Último acceso: 05 12 2013].

[55] K. Linux, «Kali Linux Hash Cracking,» 11 09 2013. [En línea]. Available: http://www.youtube.com/watch?v=p7lwpAIbvv0. [Último acceso: 11 12 2013].

[56] Certicamaras, «Camara de comercio de Bogotá,» 08 08 2012. [En línea]. Available: http://www.contratos.gov.co/archivospuc1/2012/OSMC/241000052/12-13-1084438/OSMC_PROCESO_12-13-1084438_241000052_5071074.pdf. [Último acceso: 02 11 2013].

[57] M. C.-S. a. C. Platform, «System Center 2012 R2 Configuration Manager,» Microsoft, 25 10 2013. [En línea]. Available: http://www.microsoft.com/es-es/server-cloud/products/system-center-2012-r2-configuration-manager/default.aspx#fbid=ctN4NsGRYSM. [Último acceso: 26 11 2013].

[58] Fedoraproject.org, «Fedora 18 - Guía de seguridad,» Fedora, 10 2012. [En línea]. Available: http://docs.fedoraproject.org/es-ES/Fedora/18/pdf/Security_Guide/Fedora-18-Security_Guide-es-ES.pdf. [Último acceso: 12 12 2013].

[59] V. Aurora, «CICLO DE VIDA DE LOS ALGORITMOS DE HASHING,» 03 06 2011. [En línea]. Available: http://www.securitybydefault.com/2011/06/ciclo-de-vida-de-los-algoritmos-de.html. [Último acceso: 01 11 2013].

[60] FORRESTER, «CONFERENCIAS ACIS,» 01 06 2013. [En línea]. Available: http://www.acis.org.co/index.php?id=42. [Último acceso: 26 11 2013].

[61] S. d. I. y. Comercio, «COMERCIO ELECTRÓNICO-Regulación,» 23 03 2012. [En línea]. Available: http://www.sic.gov.co/. [Último acceso: 03 10 2013].

[62] S. I. d. I. y. Comercio, «COMERCIO ELECTRÓNICO-Regulación,» 18 02 2006. [En línea]. Available: http://webcache.googleusercontent.com/search?q=cache:VZfTyGBGBhEJ:www.acis.org.co/memorias/JornadasSeguridad/IJNSI/entidadcertifica.ppt+&cd=1&hl=es&ct=clnk&gl=co. [Último acceso: 20 12 2013].