

Nivel de Inmadurez de los Sistemas de Detección de Intrusiones de Red (NIDS).

Por: Alejandro Corletti

Acorletti@cybsec.com, acorletti@altransdb.com

1. Presentación:

El presente trabajo, surge de una evaluación de los distintos NIDS existentes en el mercado.

La idea es la de incluir estas nuevas tecnologías en la red de una importante empresa, para lo cual, no sólo se analizaron las opciones disponibles, sino que para que el tema no fuera abordado ligeramente, se encargó un análisis (Benchmark) de los diferentes productos existentes en el mercado, para determinar cuáles de ellos se ajustaban mejor a las características de esta red.

La evaluación de los productos (aún no finalizada), incluyó las siguientes tareas:

- a. Investigación de mercado.
- b. Reunión de información de los productos.
- c. Determinación de las características que se consideran más importantes en un IDS para esta red.
- d. Selección preliminar de un número de ellos para investigar en detalle (en la etapa final, quedaron sólo tres productos que se creyó podían ser los más adecuados a esta empresa).
- e. **Comparativa:** Sobre esta actividad es donde se hizo mayor hincapié y se dedicó más tiempo, subdividiéndola en tres partes:
 - 1) Respuesta ante ataques conocidos.
 - 2) Respuesta ante anomalías a lo determinado en las RFC correspondientes a los protocolos de la familia TCP/IP.
 - 3) Aspectos generales.
- f. Análisis de vulnerabilidades en NIDS.

Al ir avanzando en la evaluación de estos productos (en particular las tareas del punto e.) comienzan a aparecer una serie de detalles que dan origen a este trabajo que creo demuestran que **aún no se ha alcanzado un nivel de madurez adecuado** para confiar plenamente en la información que los IDS entregan y que presentan a su vez un gran número de vulnerabilidades.

Las conclusiones finales tratan de marcar los detalles más importantes y lo que creo que puede ser un curso de acción orientado a continuar mejorando estos nuevos dispositivos de seguridad.

Por último, a lo largo de este texto no se hará mención al nombre de los productos, empresa propietaria o de distribución (comercial o gratuita), pues no se trata aquí de promover un NIDS en particular, sino de plantear lo que se aprecia como estado actual de esta tecnología, independientemente de sus fabricantes. En los casos en que se presente alguna comparativa, será como **producto A, B o C**, y pido disculpas por anticipado, si en algún momento se deja translucir las características de alguno de ellos.

2. Marco de trabajo:

Se armó una pequeña red de laboratorio y a su vez se comenzó también con una etapa de prueba de los distintos productos en varias zonas de la red en producción de la empresa, tratando de trabajar con plataformas de hardware similares.

Se estudió los distintos SPLITTER del mercado, encontrando en ellos varias diferencias en sus prestaciones de trabajo en 100 Mbps, en particular cuando se trataba de tráfico full dúplex.

Los productos que lo permitían fueron comprobados sobre sistemas operativos Linux, Solaris y Windows 2000.

En todos los productos se emplearon los conjuntos de reglas (o firmas) que proporcionaban la máxima cobertura, sin personalizar ninguna de ellas, para obtener el mismo línea base o punto de partida con todos.

Si bien es un factor importante, no se pudo evaluar la pérdida de paquetes al trabajar a 100 Mbps, pero sí se tuvo en cuenta el empleo de CPU y memoria de los distintos productos.

Se está trabajando en la actualidad sobre una investigación de los distintos productos que permitan correlar eventos, para poder determinar su eficiencia en la centralización de todos los dispositivos de seguridad.

3. Breve descripción de la comparativa:

El presente trabajo se basó en tres aspectos para realizar la comparación:

- a. Respuesta ante ataques conocidos.
- b. Respuesta ante anomalías a lo determinado en las RFCs correspondientes a los protocolos de la familia TCP/IP.
- c. Comparativa de aspectos generales.

Los cuales se desarrollan a continuación.

a. Respuesta ante ataques conocidos:

Esta tarea se divide en dos partes:

- Aprovechamiento de la información recolectada a través de la actividad generada por dos empresas que desarrollaron haking ético.
- Generación de tráfico a través de distintas herramientas conocidas (Internet Security Scanner, Retina y Nessus), programas de generación de ataques realizados en PERL, y herramientas de scan de puertos y otras vulnerabilidades.

b. Respuesta ante anomalías a lo determinado en las RFC correspondientes a los protocolos de la familia TCP/IP:

Se subdividió este análisis por protocolos, empleando desarrollos propios que generaban tráfico los cuales, pudiendo o no ser ataques conocidos, no cumplían lo determinado por las RFCs correspondientes a esos protocolos. Los protocolos investigados fueron:

1) ETHERNET (encabezado MAC)(IEEE: 802.3).

Se generaron 2 patrones de tráfico: arp1.cap de tramas y Ethernet1.cap de 170 tramas con las siguientes características:

Incoherencias de solicitudes y réplicas ARP.
 Tamaños de trama incorrectos.
 Campo Length o Ethertype modificados.
 Excesivos Broadcast.
 Falsos Multicast.
 Misma dirección fuente y destino.
 Direcciones globalmente administradas erróneas.
 Errores de CRC.
 Ataques ARP.
 Modificación de tablas ARP.

- 2) **BOOTP** (RFC 1541, 1531, 1533 y 1534): Se trabajó directamente con DHCP, estas RFCs son muy claras en las combinaciones permitidas acorde a qué tipo de mensaje DHCP se trate, cualquier otra combinación no contemplada no debería generarse.

Se generó 1 patrón de tráfico: dhcp1.cap de 230 tramas con las siguientes características:

Valores erróneos en los campos:

- OP (sólo permite 1 y 2).
- HTYPE (sólo permite del 1 al 7).
- HLEN: Especifica el tipo y longitud de la dirección de Hardware (Debería estar de acuerdo con Ethernet tiene tipo 1 para 10 Mbps y longitud 6 octetos).
- HOPS: El cliente debería colocar (0), si es necesario pasar a través de distintos router , el servidor BOOTP o DHCP lo incrementará.
- TRANSACTION ID: Dependerá de las solicitudes y respuestas, debe contener un número entero que permite llevar el control entre las solicitudes y respuestas.
- SECONDS: Determina el tiempo transcurrido desde que se inició la operación.
- FLAGS: Identifica por medio del primer bit si es un Broadcast, los restantes quince deben estar puestos a cero.
- CLIENT IP ADDRESS:
- YOUR IP ADDRESS:
- SERVER IP ADDRESS:
- ROUTER IP ADDRESS:
- CLIENT HARDWARE ADDRESS:
- SERVER HOST NAME:
- BOOT FILE NAME: Debería contener el tipo de arranque(Ej: UNIX)
- OPTIONS: Define máscara de subred, hora,etc.

Obtención de información, con R_ARP, BOOT_P o DHCP.

Saturación de direcciones en servidores.

- 3) **IP** (RFC 791):

Se generó 1 patrón de tráfico: ip1.cap de 261 tramas con las siguientes características:

Errores de campo versión.
 Falsas longitudes de cabecera.
 Falsos valores de campo Protocol
 Incoherencias de combinaciones de TOS y D, T , R.
 Datagramas con ID number de igual valor.
 Datagramas con igual IP fuente y destino (ataque LAND).
 Direcciones IP reservadas.
 Errores de fragmentación.

Falsos valores de TTL
 Errores de checksum.
 Incoherencias y uso dudoso de campo opciones.
 Rellenos no múltiplos de 4 Byte.
 Análisis de comportamiento con ECN (bit 6 y 7 de TOS).
 Detección de ACL empleando IP con encabezados erróneos.

4) **ICMP (RFC 792):**

Se generó 1 patrón de tráfico: icmp1.cap de 578 tramas con las siguientes características:

Valores no permitidos en campos:

- ICMP Type.
- ICMP Code (combinaciones de códigos no existentes para determinados tipos)
- ICMP Time Stamp.
- ICMP Information request.
- ICMP Address mask request.

Mensaje de fragmentación requerida y no permitida de ICMP erróneos.

Mensajes de puerto, red o destino inalcanzable erróneos.

Empleos de traceroute.

Solicitudes de información ICMP.

Empleo de ICMP fragmentado.

ICMP con campos IP erróneos.

Mensajes TTL excedido erróneos.

Redirigido (Ataque Winfreeze).

Ping con datos.

Ping de longitud excesiva.

Combinaciones no permitidas de IP con ICMP.

Broadcast ICMP.

5) **IGMP (RFC 1112, Apéndice 1):**

Se generó 1 patrón de tráfico: igmp1.cap de 232 tramas con las siguientes características:

Errores en campos:

- Versión: Sólo es válido 1 y 2.
- Tipo: Sólo dos tipos 1 (consulta) y 2 (Reporte). (El analizador de protocolos reconoce hasta el valor 4, aún no sé qué RFC los amplía).
- No usado: sólo 0 en envío y debería ignorarse en reporte.
- Checksum: se refiere sólo a los 8 octetos del mensaje.
- Dirección de grupo: en envío debería ser 0, (abusos de grupo).

Captura y alteración de mensajes entre routers y switches.

Direcciones multicast origen.

Combinaciones de direcciones MAC 01-00-5E-XX-XX-XX con falsas IP multicast.

Anuncios de host para incorporación a grupos multicast.

Mensajes de propagación de grupos.

Falsos sondeos multicast por parte de "Routers (falsos)".

Mal empleo de protocolo DVMRP (Distance Vector Multicast Routing Protocol).

6) **UDP (RFC 768):**

Se generó 1 patrón de tráfico: udp1.cap de 119 tramas con las siguientes características:

Errores de fragmentación combinado con IP.
 Errores del campo checksum, alteración de su existencia (opcional).
 Empleo de UDP con Broadcast IP.
 Puertos fuente y destino en 0.
 Puertos fuente y destino iguales (en casos especiales).
 Errores de longitud (No puede ser menor a 8 y debería coincidir con la suma de datos y cabecera).

7) **TCP** (RFC 793, 812, 813, 879, 896 y 1122):

Se generó 1 patrón de tráfico: tcp1.cap de 757 tramas con las siguientes características:

Empleo de TCP con Broadcast IP.
 Puertos fuente y destino en 0.
 Errores de campo desplazamiento de datos.
 Empleo de los bit reservados.
 Envío de datos durante el establecimiento de sesiones.
 Alteraciones de FLAG (SYN, ACK, PSH, RST, URG y FIN).
 Empleos de combinaciones con ACK = 0.
 Segmentos NULL (todos los FLAG = 0).
 Sesiones sin completar al abrir o sin cerrar.
 Falsas combinaciones de bit URG con campo puntero de urgente.
 Errores de secuencias de envío y recepción.
 Errores de ventana.
 Errores de timestamp.
 Errores temporales.
 Errores de fragmentación.
 Ataque Tiny Fragment.
 Incoherencias y uso dudoso de campo opciones.
 Errores de longitudes de cabecera.
 Errores en MTU durante el triple handshake.
 Modificaciones de MSS.
 Análisis de comportamiento con ECN (bit 8 y 9).

8) **SNMP** (RFC 1155, 1156 y 1157):

Se generó 1 patrón de tráfico: snmp1.cap de 258 tramas (SIN TERMINAR AUN) con las siguientes características:

Pruebas con objetos de secuencias diferentes a: 1.3.6.1.

Errores en los campos:

- Versión: Solo permite 1, 2 y 3.
- Comunidad: No debería estar vacío.
- PDU: Solo puede ser GetRequest, GetNextRequest, GetResponse, SetRequest y Trap.

La PDU tiene a su vez los siguientes campos:

- PDU type: solo permite: 0 GetRequest, 1 GetNextRequest, 2 GetResponse y 3 SetRequest.
- Request ID: Valor entero que controla la correspondencia entre agente y administrador.
- Error status: solo cinco tipos de error: 0 noError, 1 tooBig, 2 noSuchName, 3 badValue, 4 readOnly y 5 genErr. (El analizador de protocolos reconoce hasta el valor 18 dec, aún no encontrará la RFC que lo amplía).

- Error index: Identifica la entrada en la lista que ocasionó el error.
 - Object/value: Define el objeto con su valor correspondiente.
- Existe también otro formato de PDU, que es el de Trap PDU, el cual tiene los siguientes campos:
- PDU Type: Solo admite el valor 4.
 - Enterprise: Identifica al administrador de la “empresa” que definió la trap.
 - Agent Address: Debe coincidir con la dirección IP del agente.
 - Generic Trap Type: solo siete valores están definidos: 0 coldStart, 1 warmStart, 2 linkDown, 3 linkUp, 4 authenticationFailure, 5 egpNeighborLoss y 6 enterpriseSpecific
 - Specific Trap Type: Empleado para identificar un Trap no genérico.
 - Timestamp: Representa el tiempo transcurrido entre la última reinicialización y la generación del presente trap.

Combinaciones falsas de la variable con su valor.

Falsificación de tráfico sobre capturas reales.

9) **Telnet** (RFC 854, 855 y 857):

Se generó 1 patrón de tráfico: telnet1.cap de 59 tramas con las siguientes características:

Empleo de comandos no válidos.

Errores en los campos:

- IAC: Debería ser FF
- Command Code: Solo están definidos los valores desde F0 a FF (este último es IAC).
- Option Negotiated: Solo están definidos los valores desde 0 a 22 hex y el FF.

Pruebas de Telnet session reconstruction.

10) **FTP** (RFC 265, 354, 412, 542, 765, 959):

Se generó 1 patrón de tráfico: ftp1.cap de 420 tramas (SIN TERMINAR AUN) con las siguientes características:

Errores en los campos:

- Descriptor: Solo puede ser 0 a 4.
- Inconsistencias con el Byte count y la cantidad de Marker.
- OPCODE: Valores permitidos son: de 00 a 0E, de 4F a 077, de 5A a 100, de FF a 377.
- SET DATA TYPE: Valores permitidos son: de 00 a 08, de 4F a 077, de 5A a 100, de FF a 377.
- ERROR CODE: Valores permitidos son: de 00 a 0B.
- MODE: Stream, Block, Compressed,
- STRUCTURE: File, Record, Page (dentro de page, inconsistencias entre: Header length, Page Index, Data length, page type y Optional field).
- COMMANDS: Combinaciones que no son: ABOR, ACCT, ALLO, APPE, etc.
- MENSAJES: Sólo admite desde el 110, 120, 125, 150, etc.

Establecimiento de sesiones activas y pasivas.

Irregularidades en puerto 20 y 21.

Ordenes y datos intercambiando puertos.

Saturación en transferencia de archivos.

11) **HTTP** (RFC 1945, 2109, 2145, 2616):

Se generó 1 patrón de tráfico: http1.cap de 142 tramas (SIN TERMINAR AUN) con las siguientes características:

Errores en los campos:

- Method: OPTIONS, GET, HEAD, POST, PUT, DELETE, TRACE, CONNECT, extension.
- Request URI: Cualqueir variante contemplada en el punto 3.2. (Uniform Resource Identifiers) de la RFC 1945.
- HTTP Version: Debería ser de la forma HTTP/1.x, (podría aparecer HTTP/0.x)
- Status Code: 1xx (informacional), 2xx(Successful), 3xx(Redirection), 4xx (Client error), 5xx (Server error),
- Reason phrase: Texto.
- Content Codings: Solo dos valores son definidos: x-gzip y x-compress.
- General-Header: cache-control, connection, date, transfer-encoding, upgrade, via, name, value, pragma y content
- Request-Header: Simple y full, referer, user-agent, accept (charset, encoding, language), autorization, from, host, if (Modified-since, match, none-match, range, unmodified-since), max-forwards, proxy-authorization, range, referer.
- Response-Header: Simple y full, status-line (Status code:1xx a 5xx, y Reason phrase), server, www-authenticate, age, location, proxy-authenticate, public, retry-after, server, vary, warning, set-cookie.
- Entity-Header: Allow, content (base, encoding, lenguaje, length, type, location, MD5, range), expires, Etag, last-modified, extension-header.

Reensamble de paquetes a otros puertos.

Estudio de HTTP session conversion.

Análisis de distintas opciones de Unicode (xC0, xC1, xE=, xF0, xF8, xFC, secuencias 0xC0AE, etc).

12) SMTP y POP (RFC 821, 1082):

Se generó 1 patrón de tráfico: smtp1.cap de 322 tramas (SIN TERMINAR AUN) con las siguientes características:

Empleo de mensajes con comandos incorrectos: Los únicos comandos válidos son: HELLO, MAIL, RECIPIENT, DATA, SEND, SOML, SAML, RESET, VERIFY, EXPAND, HELP, NOOP, QUIT y TURN (No dependientes de mayúsculas y debiendo respetar la sintaxis expresada en el punto 4.1.2 de la RFC).

Errores en código de mensaje de réplica: Los valores definidos son: 211, 214, 220, 221, 421, 250, 251,354, 450, 451, 452, del 500 al 504, 550, 552,553, 554.

Empleo de longitudes excesivas en los campos: user (64 caracteres), domain (64 caracteres), path (64 caracteres), command line (512 caracteres), reply line(512 caracteres), text line (1000 caracteres), recipients buffer (100 recipients).

Bombardeos de mail.

13) SSH (FALTA DESARROLLAR).

Pruebas sobre puerto 22

Triple Handshake sin finalizar.

Alteración de las siete tramas de establecimiento de sesión SSH.

14) DNS (RFC 1591, 1034 y 1035): (FALTA DESARROLLAR).

Valores Erróneos en:

OPCODE: QUERY, SQUERY

TYPE: A, MD, MF, MB, MG, MR, NULL, WKS, CNAME, HINFO, MX, NS, PTR, SOA, AXFR, MAILB, TXT (Se corresponden con los valores decimales desde 1 a 16).

CLASS: IN, CS, CH, HS (Se corresponden con los valores decimales desde 1 a 4).

RR: CNAME, HINFO (CPU, OS), MBRDATA, MFRDATA, MGRDATA, MINFORDATA, MRRDATA, MXRDATA, NULLRDATA, NSRDATA, PTRRDATA, SOARDATA, TXTRDATA, ARDATA, WKSRRDATA

Answer, Authority y Additional: RRs (NAME, TYPE [2octetos], CLASS [2 octetos], TTL, RDLENGTH[entero de 16 bit], RDATA).

Header Section:

- ID: Entero de 16 bit (debe guardar consistencia entre solicitud y réplica).
- QR: 1 bit, 0 (QUERY), 1 (RESPONSE).
- OPCODE: 4 bit, 0(QUERY), 1 (IQUERY), 2 (STATUS), 3 a 15 (Reservados para usos futuros) no deberían emplearse.
- AA (Authoritative Answer): 1 bit, sólo válido en respuestas
- TC (Truncation): 1 bit, se emplea para especificar que el mensaje fue truncado por ser más grande de lo permitido.
- RD (Recursion Desired): 1 bit, configurado en consulta y copiado en la respuesta.
- RA (Recursion Available): 1 bit, informa que el server soporta recursividad.
- Z: Reservado para usos futuros.
- RCODE (Response Code): 4 bit, solo como parte de una respuesta, sólo admite valores de 0 a 5 y de 6 a 15 está reservado para usos futuros, no deberían emplearse.
- QDCOUNT: 16 bit, número de entradas en la question section.
- ANCOUNT: 16 bit, número de RRs en ña answer section.
- NSCOUNT: 16 bit, número de servidores de nombres en la RR.
- ARCOUNT: 16 bit, número de RRs en la additional records section.

Question Section:

- QNAME: Secuencia de etiquetas (cada una consta de 1 octeto, seguido por este número de octetos) debe finalizar con el octeto 0.
- QTYPE: 2 octetos, son válidos todos los de TYPE y se suman del 252 al 255.
- QCLASS: 2 octetos, son válidos todos los de CLASS y se suma el valor 255, en el caso de internet es IN.

Formato de RR:

- Name: Nombre de dominio.
- Type: 2 octetos, descripto anteriormente.
- Class: 2 octetos, descripto anteriormente, especifican la clase de datos que deberán ir en RDATA.
- TTL: 32 bit, descripto anteriormente.
- RDLENGTH: 2 octetos que especifican la longitud que deberá tener RDATA.
- RDATA: cadena de longitud variable.

Empleo de tamaños superiores a los permitidos en los siguientes campos:

- Labels (63 octetos)
- names (255 octetos)
- TTL (valor positivo de 32 bit)
- mensajes UDP (512 octetos).

Falsificación de los bit parámetros (solicitud, respuesta, inversa, autoritativa, recursiva, iterativa, fallas, nombres).

Modificación en campos número de...

Modificación de campos secciones (solicitud, respuesta, autoridad e información adicional).

c. Comparativa de aspectos generales:

Los aspectos generales de esta comparativa se basaron en los siguientes puntos:

1) INSTALACIÓN:

- a. Rapidez de instalación
- b. Facilidad de instalación
- c. Soporte para MS, UNIX y LINUX
- d. Documentación de instalación.
- e. Soporte para 10 y 100 base T.
- f. Situación para 1000 Base T.
- g. Existencia de software adicional (terceras partes)
- h. Requerimientos de hardware.
- i. Requerimientos de modificación en la red para su instalación.

2) SEGURIDAD:

- a. Empleo de canales separados para escucha y transmisión de eventos.
- b. Seguridad empleada en los canales de transmisión (TCP, puertos, autenticación, criptografía, etc.).
- c. Nivel de estandarización de los mecanismos de seguridad.
- d. Claridad y facilidad en las instrucciones para el empleo de los mecanismos de seguridad.
- e. Nivel de ocultamiento de los productos (dificultad para determinar su presencia).
- f. Vulnerabilidades conocidas o detectadas.
- g. Consistencia en el sistema de control de caídas de los distintos elementos.
- h. Capacidad de funcionamiento fuera de banda en casos extremos.
- i. Alarmas de falla de cualquiera de sus componentes.
- j. Información sobre bastionamiento de la plataforma que lo soporta.

3) DETECCIÓN DE INCIDENTES:

- a. Nivel de monitoreo de eventos.
- b. Detección de incidentes externos e internos.
- c. Capacidad de detección y configuración de patrones de tráfico
- d. Generación de falsos positivos.
- e. Detección de eventos por sucesivas ocurrencias de un hecho.
- f. Detección y posibilidad de respuesta ante DoS.
- g. Detección y posibilidad de respuesta ante accesos no autorizados.
- h. Detección y posibilidad de respuesta ante ataques conocidos.
- i. Detección y posibilidad de respuesta ante actividad sospechosa
- j. Capacidad del usuario para crear reglas.
- k. Capacidad para análisis de contenidos.
- l. Capacidad de análisis de fragmentación
- m. Capacidad de correlación de eventos.
- n. Capacidad de detección de UNICODE.
- ñ. Capacidad de personalización del software y minimización de los falsos positivos.
- o. Grado de detalle y claridad en la visualización de eventos.
- p. Porcentaje de eventos detectados (es decir no pérdida de eventos).

- q. Estrategias de preprocesamiento de reglas.
- r. Estrategias de seguimiento de sesiones.
- s. Posibilidad de relevamiento de componentes de red.
- t. Posibilidad de migrar las reglas hacia otras plataformas.

4) RESPUESTA A INCIDENTES:

- a. Nivel y calidad en el envío de las alarmas
- b. Envíos de SNMP traps (y versión de SNMP)
- c. Integración de el SNMP nativo del producto con los distintos software de administración SNMP.
- d. Capacidad de logs de eventos.
- e. Capacidad de registro de eventos (nivel de detalle).
- f. Capacidad para la adopción de medidas en firewall o router.
- g. Capacidad para finalizar sesiones dudosas.
- h. Capacidad de respuestas a través de programas, scripts, ejecutables, etc.
- i. Estrategias nativas para la adopción de contramedidas.
- j. Estrategias nativas de engaño.
- k. Capacidad de interacción con otros sniffer (para lanzar seguimientos de actividad).

5) CONFIGURACIÓN:

- a. Capacidad de configuración remota.
- b. Ayudas para la configuración remota.
- c. Nivel o facilidad de preconfiguración de los elementos remotos
- d. Capacidad de ajuste de la propagación de eventos
- e. Flexibilidad en la configuración de ataques y análisis de tráfico referidos a host, servicios o protocolos.
- h. Interfaz gráfica para configuración.
- i. Posibilidad de configuración de reglas de red y de protocolos por separado.
- j. Acceso al código.

6) MONITOREO DE EVENTOS:

- a. Nivel de entorno gráfico en la visualización de eventos.
- b. Necesidad de capacitación del personal que opera cotidianamente la consola.
- c. Amigabilidad de visualización de eventos.
- d. Capacidad de resumen o consolidación de eventos múltiples en vistas breves.
- e. Capacidad de reunión de eventos (de múltiples sensores) en una misma consola.
- f. Capacidad de envío de eventos a consolas de administración SNMP.
- g. Posibilidad de detalle en la visualización de eventos.
- h. Claridad en la especificación (explicación) técnica del detalle de los eventos detectados.
- i. Ajuste a estándares en el monitoreo de eventos (CVE, Bugtraqs, etc.).
- j. Clara visualización de la prioridad de los eventos.

7) ADMINISTRACIÓN DE DATOS:

- a. Capacidad de recepción de datos de distintos productos.
- b. Calidad de la DB de almacenamiento propia.
- c. Posibilidad de empleo de ODBC sobre cualquier DB.
- d. Capacidad de exportación de su DB a otros formatos.
- e. Flexibilidad de acceso a la DB desde otros productos de consulta (Nivel de apertura de su estructura).
- f. Capacidad para la generación de reportes.

- g. Facilidad para el diseño de plantillas de reportes.
- h. Flexibilidad para la personalización de los reportes.
- i. Flexibilidad para la exportación de reportes a otros formatos (Word, CSV, etc.).
- j. Mantenimiento de la base de datos.
- k. Velocidad de procesamiento de consultas y respuestas de la DB.
- l. Acceso a la estructura de la DB.
- ll. Estrategia de almacenamiento de datos.

8) RENDIMIENTO:

- a. Capacidad de procesar el tráfico y reaccionar ante un alto volumen de tráfico.
- b. Rendimiento del producto ante el incremento de reglas personalizadas.
- c. Capacidad de rendimiento en función directa del hardware (es decir es extremo en su necesidad de hardware para su eficiente funcionamiento?).
- d. Capacidad de funcionamiento remoto ante alto tráfico de red.
- e. Rendimiento de la DB.
- f. Rendimiento de la consola.
- g. Rendimiento de los sensores.
- h. Rendimiento con varios sensores.
- i. Variaciones de rendimiento al ir incrementando la DB.

9) ARQUITECTURA:

- a. Adaptamiento a altas velocidades de red y nuevas tecnologías.
- b. Integración con otras arquitecturas, hardware y software.
- c. Nivel de estandarización de toda su arquitectura (es decir Compatibilidad con otras arquitecturas).
- d. Confiabilidad de toda la arquitectura.
- e. Costo de la arquitectura completa.
- f. Explicación de su arquitectura.
- g. Acceso a modificaciones en su arquitectura.

10) ACTUALIZACIONES:

- a. Vigencia de las actualizaciones.
- b. Integración de las actualizaciones con las reglas personalizadas.
- c. Automatización de las actualizaciones.
- d. Flexibilidad de las actualizaciones en todos sus módulos.
- e. Costos de las actualizaciones.
- g. Explicación de las actualizaciones.
- h. Posibilidad de actualizaciones en grupos de sensores o agrupamientos personalizados de eventos.

11) SOPORTE TÉCNICO:

- a. Métodos empleados para el soporte técnico.
- b. Disponibilidad del soporte técnico.
- c. Eficiencia y capacidad para solucionar los problemas.
- d. Costo del soporte técnico.
- e. Capacidad técnica y de detalle del soporte técnico.
- f. Tiempo de respuesta del soporte técnico.
- g. Apoyo en actualización y capacitación de personal que brinda el soporte técnico.

4. Resumen de algunas mediciones y datos obtenidos:

a. Medición de respuesta ante ataques conocidos (acorde lo tratado en el punto 3. a.)):

La presente tabla representa uno de los períodos de captura, (el que se creyó más representativo), y la respuesta de los productos A, B y C durante un lapso comprendido entre las 10hs y las 1430hs en el cual se generaron ataques. Para el análisis de la misma se deben tener en cuenta los siguientes conceptos:

- 1) Las reglas de los tres productos se encontraban acorde a la última actualización y cubriendo todos los ataques posibles, es decir, con la máxima cobertura.
- 2) No se personalizó ninguna de ellas para evaluar a los mismos bajo las condiciones iniciales de detección.
- 3) Se eliminaron todos los eventos repetidos, es decir, que en la tabla figura una sola instancia de cada alarma, si bien existen muchas repeticiones de los mismos.
- 4) El tráfico generado, corresponde a la actividad de hacking desarrollado por un solo ordenador en ese lapso de tiempo.
- 5) Como detalle de particular interés descubierto en esta comparativa, llama poderosamente la atención LA DISPARIDAD DE LOS EVENTOS CAPTURADOS. De la totalidad de los eventos (los mostrados en la tabla, más los repetidos), solo coinciden los IDS en su captura en el orden de un 5 a 10 %, EL RESTO QUE SON CAPTURADOS POR UNO Y/U OTRO, NO LO SON POR EL O LOS RESTANTES. Este hecho es de suma importancia para esta comparativa pues denota la importancia de estudiar los distintos IDS y personalizar paso a paso las reglas de cada uno.
- 6) La conclusión final puede ser enfocada desde dos puntos de vista, dejando al criterio del lector la elección del que crea más adecuado:
 - a. Una postura podría ser que **es preferible obtener la mayor cantidad de datos posibles** para que luego el usuario tenga mayor cantidad de elementos de juicio para personalizar las reglas. Siempre queda la libertad del usuario para poder minimizar el nivel de detalle de las capturas. El problema aquí radica en la obligación del usuario de conocer en detalle los distintos ataques y el impacto que pueden causar en su red en particular.
 - b. La teoría anterior puede ser refutada bajo la idea de **minimizar los falsos positivos**, mostrando solo lo esencial y descartando lo menos importante. Este enfoque también es discutible pues lo ideal sería que posea un conjunto de reglas con máximo nivel de detalle (Que sería de suponer que justamente son estas), y a su vez otras en las cuales cumpla la función de minimizar falsos positivos, dejando al usuario la libertad de elección.

La medición fue la siguiente:

| N° | Fecha/hora | Definición del evento | Producto A | Producto B | Producto C | Observaciones |
|----|-------------|-------------------------------|------------|------------|------------|---|
| 1. | 23/11/10:06 | Snmp-suspicious-get | SI | NO | NO | Se omiten para tratar de mantener el anonimato de los productos |
| 2. | 23/11/10:20 | Decod-http-tilde | SI | NO | NO | |
| 3. | 23/11/10:28 | Scan Proxy attempt | NO | SI | SI | |
| 4. | 23/11/10:28 | INFO - Possible Squid Scan | NO | SI | SI | |
| 5. | 23/11/10:28 | SCAN nmap fingerprint attempt | NO | SI | SI | |
| 6. | 23/11/10:28 | SCAN nmap TCP | NO | SI | NO | |
| 7. | 23/11/10:29 | RPC portmap listing | NO | SI | NO | |
| 8. | 23/11/10:29 | RPC portmap request mountd | NO | SI | NO | |
| 9. | 23/11/10:37 | Cobalt-raq-history-exposure | SI | NO | NO | |

| N ° | Fecha/hora | Definición del evento | Producto A | Producto B | Producto C | Observaciones |
|-----|-------------|---|------------|------------|------------|---------------|
| 10. | 23/11/10:37 | Webstore-misconfig | SI | NO | NO | |
| 11. | 23/11/10:37 | Pdgsoftcart-misconfig | SI | NO | NO | |
| 12. | 23/11/10:38 | Coldfusion-file-existence | SI | NO | SI | |
| 13. | 23/11/10:38 | Coldfusion-source-display | SI | NO | SI | |
| 14. | 23/11/10:38 | http-cgi-cachemgr | SI | NO | NO | |
| 15. | 23/11/10:39 | http-cgi-phf | SI | NO | NO | |
| 16. | 23/11/10:40 | http-iis-aspdot | SI | NO | NO | |
| 17. | 23/11/10:40 | iis-exair-dos | SI | NO | SI | |
| 18. | 23/11/10:40 | Ezmall2000-misconfig | SI | NO | NO | |
| 19. | 23/11/10:41 | Quikstore-misconfig | SI | NO | NO | |
| 20. | 23/11/10:50 | SCAN Proxy attempt | NO | SI | SI | |
| 21. | 23/11/10:58 | Decod-http-cookie | SI | NO | NO | |
| 22. | 23/11/11:01 | Coldfusion-admin-dos | SI | NO | NO | |
| 23. | 23/11/11:01 | Cisco-catalyst-remote-commands | SI | NO | NO | |
| 24. | 23/11/11:02 | Http-cgi-vuln | SI | NO | SI | |
| 25. | 23/11/11:14 | WEB – FRONTPAGE fourdots request | NO | SI | SI | |
| 26. | 23/11/11:14 | WEB – MISC http directory transversal | NO | SI | SI | |
| 27. | 23/11/11:15 | WEB – ISS SAM Attempt | NO | SI | SI | |
| 28. | 23/11/11:15 | WEB – MISC .htaccess access | NO | SI | NO | |
| 29. | 23/11/11:15 | WEB – MISC .htpasswd access | NO | SI | NO | |
| 30. | 23/11/11:15 | WEB-IIS jet vba access | NO | SI | SI | |
| 31. | 23/11/11:15 | WEB – MISC order.log access | NO | SI | SI | |
| 32. | 23/11/11:15 | WEB-FRONTPAGE dvwssr.dll access | NO | SI | SI | |
| 33. | 23/11/11:15 | WEB-IIS fpcount access | NO | SI | SI | |
| 34. | 23/11/11:15 | WEB-IIS _vti_inf access | NO | SI | SI | |
| 35. | 23/11/11:15 | WEB-FRONTPAGE administrators.pwd | NO | SI | SI | |
| 36. | 23/11/11:15 | WEB-FRONTPAGE authors.pwd access | NO | SI | SI | |
| 37. | 23/11/11:15 | WEB-FRONTPAGE service.pwd | NO | SI | SI | |
| 38. | 23/11/11:15 | WEB-FRONTPAGE users.pwd access | NO | SI | SI | |
| 39. | 23/11/11:15 | WEB-IIS site server config access | NO | SI | SI | |
| 40. | 23/11/11:15 | WEB-COLDFUSION cfmlsyntaxcheck.cfm access | NO | SI | SI | |
| 41. | 23/11/11:15 | WEB-COLDFUSION exampleapp access | NO | SI | SI | |
| 42. | 23/11/11:15 | WEB-COLDFUSION getfile.cfm access | NO | SI | SI | |
| 43. | 23/11/11:15 | WEB-COLDFUSION fileexists.cfm access | NO | SI | SI | |
| 44. | 23/11/11:15 | WEB-COLDFUSION snippets attempt attempt | NO | SI | SI | |
| 45. | 23/11/11:15 | WEB-CGI AT-admin.cgi access | NO | SI | NO | |
| 46. | 23/11/11:15 | WEB-MISC order.log access | NO | SI | NO | |
| 47. | 23/11/11:15 | WEB-CGI AnyForm2 access | NO | SI | SI | |
| 48. | 23/11/11:15 | WEB-MISC count.cgi access | NO | SI | NO | |
| 49. | 23/11/11:15 | WEB-MISC ultraboard access | NO | SI | NO | |
| 50. | 23/11/11:16 | WEB-CGI aglimpse access | NO | SI | SI | |
| 51. | 23/11/11:16 | WEB-MISC ax-admin.cgi access | NO | SI | NO | |
| 52. | 23/11/11:16 | WEB-MISC bigconf.cgi access | NO | SI | NO | |
| 53. | 23/11/11:16 | WEB-CGI bnbform.cgi access | NO | SI | NO | |
| 54. | 23/11/11:16 | cachemgr.cgi access | NO | SI | SI | |
| 55. | 23/11/11:16 | WEB-CGI campas access | NO | SI | SI | |

| N ° | Fecha/hora | Definición del evento | Producto A | Producto B | Producto C | Observaciones |
|------|-------------|--------------------------------------|------------|------------|------------|---------------|
| 56. | 23/11/11:16 | WEB-CGI classifieds.cgi access | NO | SI | SI | |
| 57. | 23/11/11:16 | WEB-IIS cmd.exe access | NO | SI | SI | |
| 58. | 23/11/11:16 | WEB-CGI edit.pl access | NO | SI | SI | |
| 59. | 23/11/11:16 | WEB-CGI environ.cgi access | NO | SI | SI | |
| 60. | 23/11/11:16 | WEB-CGI faxsurvey access | NO | SI | NO | |
| 61. | 23/11/11:16 | WEB-CGI filemail access | NO | SI | SI | |
| 62. | 23/11/11:16 | WEB-CGI file.pl access | NO | SI | SI | |
| 63. | 23/11/11:16 | WEB-CGI finger access | NO | SI | SI | |
| 64. | 23/11/11:16 | WEB-CGI formmail access | NO | SI | SI | |
| 65. | 23/11/11:16 | WEB-MISC get32.exe access | NO | SI | NO | |
| 66. | 23/11/11:16 | WEB-CGI glimpse access | NO | SI | SI | |
| 67. | 23/11/11:16 | WEB-MISC guestbook access | NO | SI | NO | |
| 68. | 23/11/11:16 | WEB-MISC handler access | NO | SI | NO | |
| 69. | 23/11/11:16 | WEB-CGI htmlscript access | NO | SI | SI | |
| 70. | 23/11/11:16 | WEB-IIS achg.htr access | NO | SI | SI | |
| 71. | 23/11/11:16 | WEB-IIS iisadmpwd attempt | NO | SI | SI | |
| 72. | 23/11/11:16 | WEB-IIS anot.htr access | NO | SI | SI | |
| 73. | 23/11/11:16 | WEB-CGI info2www access | NO | SI | SI | |
| 74. | 23/11/11:17 | WEB-MISC /cgi-bin/jj attempt | NO | SI | NO | |
| 75. | 23/11/11:17 | WEB-CGI maillist.pl access | NO | SI | SI | |
| 76. | 23/11/11:17 | WEB-CGI man.sh access | NO | SI | SI | |
| 77. | 23/11/11:17 | WEB-CGI NPH-publish access | NO | SI | SI | |
| 78. | 23/11/11:17 | WEB-CGI nph-test.cgi access | NO | SI | SI | |
| 79. | 23/11/11:17 | WEB-CGI perl.exe access | NO | SI | SI | |
| 80. | 23/11/11:17 | WEB-CGI perlshop.cgi access | NO | SI | SI | |
| 81. | 23/11/11:17 | WEB-CGI pfdisplay.cgi access | NO | SI | SI | |
| 82. | 23/11/11:17 | WEB-CGI phf access | NO | SI | SI | |
| 83. | 23/11/11:17 | WEB-CGI php access | NO | SI | SI | |
| 84. | 23/11/11:17 | WEB-MISC plusmail access | NO | SI | NO | |
| 85. | 23/11/11:17 | WEB-CGI rquest.exe access | NO | SI | SI | |
| 86. | 23/11/11:17 | WEB-CGI rwwwshell.pl access | NO | SI | SI | |
| 87. | 23/11/11:17 | WEB-CGI survey.cgi access | NO | SI | SI | |
| 88. | 23/11/11:17 | WEB-CGI test.cgi access | NO | SI | NO | |
| 89. | 23/11/11:17 | WEB-CGI testcounter.pl access | NO | SI | NO | |
| 90. | 23/11/11:17 | WEB-CGI view-source access | NO | SI | SI | |
| 91. | 23/11/11:17 | WEB-CGI visadmin.exe access | NO | SI | SI | |
| 92. | 23/11/11:17 | WEB-CGI w3-msql access | NO | SI | SI | |
| 93. | 23/11/11:17 | WEB-MISC webdist.cgi access | NO | SI | NO | |
| 94. | 23/11/11:17 | WEB-CGI websendmail access | NO | SI | SI | |
| 95. | 23/11/11:17 | WEB-CGI wguest.exe access | NO | SI | SI | |
| 96. | 23/11/11:17 | WEB-CGI whoisraw access | NO | SI | SI | |
| 97. | 23/11/11:17 | WEB-CGI wrap access | NO | SI | SI | |
| 98. | 23/11/11:17 | WEB-CGI www-sql access | NO | SI | SI | |
| 99. | 23/11/11:18 | WEB-CGI wwwadmin.pl access | NO | SI | SI | |
| 100. | 23/11/11:18 | WEB-MISC wwwboard.pl access | NO | SI | NO | |
| 101. | 23/11/11:18 | WEB-CGI args.bat access | NO | SI | SI | |
| 102. | 23/11/11:18 | WEB-CGI win-c-sample.exe access | NO | SI | SI | |
| 103. | 23/11/11:18 | WEB-CGI phf access | NO | SI | SI | |
| 104. | 23/11/11:18 | WEB-CGI uploader.exe access | NO | SI | NO | |
| 105. | 23/11/11:18 | WEB-MISC Ecommerce import.txt access | NO | SI | NO | |
| 106. | 23/11/11:18 | WEB-IIS asp-dot attempt | NO | SI | SI | |
| 107. | 23/11/11:18 | WEB-IIS .asp access | NO | SI | SI | |
| 108. | 23/11/11:18 | WEB-IIS iisadmpwd attempt | NO | SI | SI | |
| 109. | 23/11/11:18 | WEB-IIS codebrowser Exair access | NO | SI | SI | |
| 110. | 23/11/11:18 | WEB-IIS codebrowser SDK access | NO | SI | SI | |
| 111. | 23/11/11:18 | WEB-MISC mall log order access | NO | SI | SI | |

| N ° | Fecha/hora | Definición del evento | Producto A | Producto B | Producto C | Observaciones |
|------|-------------|--------------------------------------|------------|------------|------------|---------------|
| 112. | 23/11/11:18 | WEB-IIS codebrowser access | NO | SI | SI | |
| 113. | 23/11/11:18 | WEB-IIS msadc/msadcs.dll access | NO | SI | SI | |
| 114. | 23/11/11:18 | WEB-MISC Ecommerce checks.txt access | NO | SI | NO | |
| 115. | 23/11/11:18 | WEB-MISC Ecommerce import.txt access | NO | SI | NO | |
| 116. | 23/11/11:18 | WEB-MISC piranha passwd.php3 access | NO | SI | NO | |
| 117. | 23/11/11:18 | WEB-MISC shopping cart access access | NO | SI | NO | |
| 118. | 23/11/11:18 | WEB-MISC queryhit.htm access | NO | SI | NO | |
| 119. | 23/11/11:18 | WEB-IIS CGImailto.exe access | NO | SI | NO | |
| 120. | 23/11/11:18 | WEB-MISC cart 32 AdminPwd access | NO | SI | NO | |
| 121. | 23/11/11:19 | WEB-IIS cmd.exe access | NO | SI | SI | |
| 122. | 23/11/11:19 | WEB-MISC convert.bas access | NO | SI | NO | |
| 123. | 23/11/11:19 | WEB-MISC counter.exe access | NO | SI | NO | |
| 124. | 23/11/11:19 | WEB-IIS fpcount access | NO | SI | SI | |
| 125. | 23/11/11:19 | WEB-IIS admin access | NO | SI | SI | |
| 126. | 23/11/11:19 | WEB-IIS bdir.ht access | NO | SI | SI | |
| 127. | 23/11/11:19 | WEB-IIS MSProxy access | NO | SI | SI | |
| 128. | 23/11/11:19 | WEB-IIS newdsn.exe access | NO | SI | SI | |
| 129. | 23/11/11:19 | WEB-IIS uploadn.asp access | NO | SI | SI | |
| 130. | 23/11/11:19 | WEB-IIS search97.vts | NO | SI | SI | |
| 131. | 23/11/11:19 | WEB-MISC ws_ftp.ini access | NO | SI | NO | |
| 132. | 23/11/11:19 | decod-nmap | SI | NO | SI | |
| 133. | 23/11/11:34 | SCAN Proxy attempt | NO | SI | SI | |
| 134. | 23/11/11:34 | INFO - Possible Squid Scan | NO | SI | SI | |
| 135. | 23/11/11:36 | WEB-FRONTPAGE fourdots request | NO | SI | SI | |
| 136. | 23/11/11:36 | WEB-MISC http directory traversal | NO | SI | NO | |
| 137. | 23/11/11:36 | WEB-IIS SAM Attempt | NO | SI | SI | |
| 138. | 23/11/11:36 | WEB-MISC .htaccess access | NO | SI | NO | |
| 139. | 23/11/11:36 | WEB-MISC .htpasswd access | NO | SI | NO | |
| 140. | 23/11/11:37 | WEB-CGI AT-admin.cgi access | NO | SI | SI | |
| 141. | 23/11/11:38 | WEB-IIS fpcount access | NO | SI | SI | |
| 142. | 23/11/11:38 | WEB - IIS iisadmpwd attempt | NO | SI | SI | |
| 143. | 23/11/11:55 | iss-scan | SI | SI | SI | |
| 144. | 23/11/11:55 | ident-error | SI | SI | SI | |
| 145. | 23/11/11:55 | SCAN NMAP XMAS | NO | SI | SI | |
| 146. | 23/11/12:04 | Tcp-ooop-sent | SI | NO | SI | |
| 147. | 23/11/12:37 | win95-back-orifice | SI | NO | NO | |
| 148. | 23/11/12:46 | RPC NFS Showmount | NO | SI | SI | |
| 149. | 23/11/13:18 | Decod-ssh | SI | NO | SI | |
| 150. | 23/11/13:21 | satan-scan | SI | NO | SI | |
| 151. | 23/11/13:22 | ip-halfscan | SI | NO | SI | |
| 152. | 23/11/13:22 | trin00-daemon | SI | NO | NO | |
| 153. | 23/11/13:22 | ddos-mstream-zombie | SI | NO | SI | |
| 154. | 23/11/13:21 | decod-nmap | SI | NO | SI | |
| 155. | 23/11/13:21 | portmap-pdump | SI | NO | NO | |
| 156. | 23/11/13:21 | ip-portscan | SI | NO | NO | |
| 157. | 23/11/13:30 | http-dotdot | SI | NO | NO | |
| 158. | 23/11/13:30 | http-cgi-viewsrc | SI | NO | NO | |
| 159. | 23/11/13:30 | irix-infosrch-fname | SI | NO | NO | |
| 160. | 23/11/13:30 | http-htmlexport-file-access | SI | NO | NO | |
| 161. | 23/11/13:30 | http-cgi-phpfileread | SI | NO | SI | |
| 162. | 23/11/13:30 | sgi-pfdispaly | SI | NO | SI | |

| N ° | Fecha/hora | Definición del evento | Producto A | Producto B | Producto C | Observaciones |
|---------------------------|-------------|---|------------|------------|------------|---------------|
| 163. | 23/11/13:30 | http-dotdot | SI | NO | NO | |
| 164. | 23/11/13:30 | http-cgi-campas | SI | NO | SI | |
| 165. | 23/11/13:30 | decod-ftp-syst | SI | NO | SI | |
| 166. | 23/11/13:30 | http-cgi-faxsurvey | SI | NO | SI | |
| 167. | 23/11/13:31 | http-iis-cmd | SI | NO | SI | |
| 168. | 23/11/13:31 | cart32-admin-password | SI | NO | SI | |
| 169. | 23/11/13:31 | cart32-clientlist | SI | NO | NO | |
| 170. | 23/11/13:31 | http-cgi-vuln | SI | NO | SI | |
| 171. | 23/11/13:31 | http-website-uploader | SI | NO | SI | |
| 172. | 23/11/13:31 | http-cgi-nph | SI | NO | SI | |
| 173. | 23/11/13:31 | decod-webfinger-attempt | SI | NO | SI | |
| 174. | 23/11/13:31 | http-unix-passwords | SI | NO | SI | |
| 175. | 23/11/13:31 | http-cgi-test | SI | NO | NO | |
| 176. | 23/11/13:31 | smtp-debug | SI | NO | NO | |
| 177. | 23/11/13:31 | http-cgi-phf | SI | NO | SI | |
| 178. | 23/11/13:31 | coldfusion-sourcewindow | SI | NO | NO | |
| 179. | 23/11/13:43 | X11 xopen | NO | SI | SI | |
| 180. | 23/11/13:44 | decod-http-cookie | SI | NO | NO | |
| 181. | 23/11/13:44 | coldfusion-cfcache | SI | NO | SI | |
| 182. | 23/11/13:44 | siteserver-site-csc | SI | NO | SI | |
| 183. | 23/11/13:44 | http-head | SI | NO | SI | |
| 184. | 23/11/14:21 | DDOS Stacheldraht client-check-gag | NO | SI | NO | |
| 185. | 23/11/14:21 | DDOS Trin00 | NO | SI | NO | |
| 186. | 23/11/14:21 | DDOS shaft handler to agent | NO | SI | SI | |
| 187. | 23/11/14:21 | DDOS mstream handler ping to agent | NO | SI | SI | |
| 188. | 23/11/14:29 | RPC portmap request rstatd | NO | SI | SI | |
| 189. | 23/11/14:29 | RSERVICES rlogin root | NO | SI | SI | |
| 190. | 23/11/14:29 | INFO FTP anonymous FTP | NO | SI | SI | |
| 191. | 23/11/14:29 | FTP saint scan | NO | SI | SI | |
| 192. | 23/11/14:29 | WEB-CGI view-source directory traversal | NO | SI | SI | |
| 193. | 23/11/14:29 | WEB-MISC SGI InfoSearch fname access | NO | SI | NO | |
| 194. | 23/11/14:29 | WEB-CGI calendar access | NO | SI | SI | |
| 195. | 23/11/14:29 | WEB-MISC Poll-it access | NO | SI | NO | |
| 196. | 23/11/14:29 | WEB-MISC BigBrother access | NO | SI | NO | |
| 197. | 23/11/14:29 | WEB-MISC htgrep access | NO | SI | NO | |
| 198. | 23/11/14:29 | WEB-CGI yabb access | NO | SI | SI | |
| TOTALES DETECTADOS | | | 56 | 142 | 128 | |

b. Medición de respuesta ante anomalías a lo determinado en las RFCs correspondientes a los protocolos de la familia TCP/IP (acorde al tráfico tratado en el punto 3. b.):

NOTA: Los casilleros que se encuentran vacíos aún no fueron evaluados en su totalidad.

| | | Producto A | | | Producto B | | | Producto C | | |
|---|----------------------|-----------------|-----------------------|---------|-----------------|-----------------------|---------------------------------------|-----------------|-----------------------|---------------------------------|
| | | De tec tó | Cant. de tramas | Detalle | De tec tó | Cant. de tramas | Detalle | De tec tó | Cant. de tramas | Detalle |
| 1 | arp1.cap | NO | | | SI | 12 | BAD TRAFFIC | SI | 2 | ARP Suspicious |
| 2 | ethernet1.cap | NO | | | SI | 11 | BAD TRAFFIC | NO | | |
| 3 | icmp1.cap | NO | | | SI | 6 | ICMP: Destination Unreachable, Source | SI | 16 | Trace Route, IRDP Gateway Spoof |

| | | | | | | | | | | |
|----|------------------------------------|----|---|-----------------|----|--------------------------------------|--|----|----|----------------------|
| | | | | | | Quench, Redirect host, Redirect net. | | | | |
| 4 | dhcp1.cap | NO | | | NO | | NO | | | |
| 5 | ip1.cap | SI | 4 | IP Len Mismatch | SI | 2 | Teardrop attack | SI | 1 | UDP Bomb |
| 6 | udp1.cap | NO | | | SI | 30 | BAD TRAFFIC | SI | 1 | UDP Bomb |
| 7 | http1.cap | | | | | | | | | |
| 8 | tcp1.cap | SI | 9 | TCP FLAGS NO | SI | 18 | BAD TRAFFIC: TCP port 0, SCAN – INFO Possible Squid Scan, SCAN FIN, SCAN SYN FIN,SCAN NMPA XMAS, SCAN nmap, X11 Outgoing | SI | 2 | Nmap Scan, Pmap Dump |
| 9 | igmp1.cap | NO | | | NO | | | NO | | |
| 10 | snmp1.cap | | | | | | | | | |
| 11 | telnet1.cap | NO | | | NO | | | SI | 4 | TCP Overlap Data |
| 12 | ftp1.cap | | | | | | | | | |
| 13 | smtp1.cap | | | | | | | | | |
| 14 | ssh1.cap | | | | | | | | | |
| 15 | dns.cap | | | | | | | | | |
| 16 | ataque http con vulnerabilidad CGI | SI | 8 | WEB:CGI | SI | 17 | WEB:CGI | SI | 16 | WEB:CGI |

5. Vulnerabilidades analizadas:

Los tipos de ataques que se pueden lanzar hacia un IDS, acorde a varios artículos publicados ya, se generalizó en clasificarlos en:

- a. **Inserción:** Este concepto define un ataque en el cual el IDS acepta información que el o los destinos de esa trama descartan, es decir, que se produce una inconsistencia entre los eventos que procesa el IDS y el o los host destino. La idea de esta metodología, en principio es conseguir desincronismo entre ambos, y a través de este, saturar de información la base de datos del IDS o lograr que por medio de este descarte que realiza el host destino, al reensamblar varias tramas, interprete algo que en el caso del IDS se interpretará distinto pues con la totalidad de las tramas el resultado es otro.

Analizando alguna de las técnicas del apartado anterior, podría suceder que al enviar tráfico, por ejemplo HTTP caracter a caracter, si se logra que a través de cualquier estrategia el host descarte tramas y el IDS no, se pueden plantear casos como el siguiente:

PSEADSFHXSRWWUIOFFRKLDE

Si se lograra el descarte de: SEDFXHXRWUIFFKLE

La palabra que se podría reensamblar sería: PASSWORD en el host final, lo cual si el IDS lo hubiera detectado, casi seguramente enviaría una alarma. Si la técnica de inserción cumple su cometido el IDS recibiría: PSEADSFHXSRWWUIOFFRKLDE, sin encontrar nada anómalo en lo subrayado.

Se insiste entonces en este ejemplo para comprender la importancia de esta técnica, pues es de las más empleadas en ataques a IDS.

En general estos ataques se producen cuando un IDS es menos estricto en el procesamiento de tramas que el sistema final, y por esta razón acepta más información y descarta menos que el o

los host destino. La solución a esto es ajustar más aún la selección de tramas, pero como suele suceder en muchas cosas de la vida, la relación costo/beneficio siempre presente hace que esta decisión provoque un aumento en las técnicas de **evasión** que se tratan a continuación.

- b. **Evasión:** La idea es la inversa del caso anterior, es decir, un IDS descarta información que el o los host destinos procesan, en estos casos, estas tramas evaden el procesamiento del IDS. Si se plantea el caso inverso del ejemplo anterior, podría suceder que si se enviara el siguiente mensaje caracter a caracter:

PASSWORD

Si se lograra que el IDS descartara cualquier letra, por ejemplo la W, este sensor interpretará PASSWORD, lo cual no será ninguna anomalía, por el contrario en el host quizás se esté buscando alguna contraseña y se pueda lograr el objetivo buscado sin que el IDS se entere.

- c. **Negación de Servicio:**

Como se mencionó anteriormente el peor caso de este tipo de ataques es cuando se llega a dejar fuera de servicio al IDS, pues es un sistema "Fail open", es decir que dejaría indefensa la red. El gran inconveniente que posee un IDS ante estos ataques es que con sus únicos recursos está observando la totalidad de las conexiones de la red, prácticamente se podría comparar a un alto porcentaje de la suma de los recursos de cada host consume.

Tipos de ataques reales hacia IDS:

Algunos protocolos son relativamente simples de analizar, en estos casos, se genera la información desde un sistema a otro y luego se espera una respuesta, ejemplos de estos son ARP, UDP, DNS, etc. Otros protocolos son más complejos y es necesario realizar un seguimiento del flujo de información para poder determinar qué está sucediendo, estos casos pueden ser TCP, IP, Telnet, etc.

5.1. Problemas de red:

Estos problemas se generan en virtud del desconocimiento que el IDS tiene de la topología de la red, y de las ambigüedades que presentan los distintos sistemas operativos (SO) en las metodologías de aceptación y descarte de paquetes.

Datagramas: Existen varias formas de generar un paquete IP para que sea descartado o aceptado por un IDS y suceda lo contrario en el host destino. El encabezado IP es descrito en la RFC 791, y los distintos SO lo implementan de manera diferente, es decir un mismo datagrama puede ser aceptado por un determinado SO y descartado por otro.

Los campos más significativos para aprovechar estas ambigüedades son:

- Direcciones.
- Longitud total y/o de cabecera errónea.
- Versión diferente de la 4.
- Tamaño erróneo.
- Errores de checksum de cabecera.
- Errores de fragmentación
 - Un IDS que no siga las secuencias de fragmentación y reensamble de IP es vulnerable. También lo es si lo hace, pues puede ser desbordado al dejar fragmentos ausentes

- Fragmentos fuera de orden.
- Pequeños fragmentos, así como pueden en muchos casos evadir listas de control de accesos pues no poseen suficiente información para ser filtrados, también pueden hacerlo con los IDS.
- Sobreposición de fragmentos: Este problema ocurre cuando fragmentos de diferente tamaño arriban fuera de orden, y solapan las posiciones acorde al campo desplazamiento de fragmentación. Los distintos SO lo tratarán de forma diferente.
- Fragmentos repetidos: acorde al SO, descartará los primeros o los últimos.
- En los casos que un IDS se encuentre en el acceso de una red, que luego posea más router, existen dos campos que pueden presentar falencias:
 - Campo TTL que sólo llegue hasta el IDS y no al host.
 - Bit de Don't Fragment, que no permita llegar hasta el host.
 - Campo opciones al fragmentar: Al fragmentar IP, el problema radica en cómo debe ser tratado el campo opciones. Nuevamente la RFC 791 es clara respecto a qué campos opciones pueden aparecer en cada fragmento y cuales solo en el primero. Una vez más los distintos SO lo tratan de forma diferente presentando ambigüedades en el descarte o no de los mismos.
- Errores de campo opciones:
 - Errores de longitud de opciones.
 - Ruta fuente y ruta fuente estricta (Offset menor que 4, el host destino no se encuentra en la lista, host configurado para descartar esta opción, sin ruta al próximo salto).
 - Registro de ruta (Offset menor que 4, sin ruta al próximo salto).
 - Timestamp (Longitud muy corta, fallas de espacio de registro, errores de tipo).
- Fallas de autenticación: IP versión 4 no permite la autenticación, con ello es muy simple falsificar (IP Spoofing) una determinada dirección IP, haciendo muy difícil su posterior análisis (forensic). El único método más o menos eficaz es poder asociar siempre la Dirección IP con la MAC correspondiente (esto lo realiza por ejemplo el software ARPWATCH), pero se debe tener en cuenta que también se puede falsificar la dirección MAC (de hecho el comando ifconfig de linux lo permite), y además algunos IDS directamente no almacenan la trama Ethernet, haciendo imposible la detección de estos eventos.
- Usar IPX sobre IP, es muy probable que el IDS no entienda el contenido de esos datos.
- Probar con técnicas de encapsulamiento: IP sobre IP, MPLS, PPTP, IPSec.

5.2. Dirección MAC:

Este tipo de ataques se lleva a cabo desde la misma LAN, las posibilidades que ofrece son:

- Direccional tramas, directamente al IDS.
- Explotar el hecho que el IDS funciona en modo promiscuo, por lo tanto procesará todas las tramas de ese segmento de red.
- Alterar las tablas de caché ARP de los host y/o IDS, por medio de ataques ARP.
- Generar direcciones fuente multicast y/o broadcast.
- Errores del campo Ethertype y/o Length.
- Longitudes erróneas a nivel Ethernet.
- Errores de CRC.
- Modificar la dirección MAC real (MAC Spoofing).
- Si se puede tomar control del Switch de la red, se puede hacer “casi cualquier cosa”.

5.3. Segmentos TCP:

El protocolo TCP es quizás el más complejo de la familia TCP/IP. Los puntos fuertes que presenta son: Orientado a la conexión (De extremo a extremo), control de flujo y garantía de entrega. Como dice un viejo refrán *“solo lo simple promete éxito”*, la complejidad de este protocolo hace que la masa de los ataques a IDS se han detectado a través de este protocolo. La RFC 793 define los aspectos fundamentales de su funcionamiento. Uno de los temas de mayor interés es el tratamiento de los “ESTADOS” en que puede estar una conexión (established, closed, listen, etc). Los ataques que se han detectado se basan en:

- Sincronización y desincronización: Los números de secuencia de una conexión son los que permiten garantizar la entrega y recepción de los segmentos TCP (a través de la técnica de ventana deslizante). Si se logra que un IDS se desincronice, este no podrá reconstruir una sesión, con lo cual le será imposible determinar el estado de “confiabilidad” de ese flujo de información. Para tener idea de la magnitud de esta tarea, se pone de manifiesto aquí que por ejemplo: para el seguimiento de una sesión y la aceptación o no de un segmento los sistemas operativos Linux, emplean en el orden de 2000 líneas de código para procesar esta actividad. Otro problema en el desincronismo es que el IDS es un elemento pasivo, o sea que si pierde un determinado segmento no podrá pedir su retransmisión como sí lo hacen los extremos de la conexión, quedando el IDS totalmente desincronizado.
- Se debe tener en cuenta que si un IDS debe seguir la totalidad de las sesiones de la red, también puede presentar un potencial problema o debilidad pues es fácilmente desbordable. También puede incorporar información a partir del momento que la sesión se encuentra en estado ESTABLISHED, con lo que perderá todo tipo de información sobre sesiones no establecidas. Si lo que desea es comenzar el seguimiento de una sesión antes de que la misma se encuentre establecida (es decir antes de recibir el último ACK del triple handshake), el tema aquí radica en decidir en cuál de estos estados se inicia el IDS. Otro problema que se deriva del establecimiento de sesiones es que si el IDS no realiza el seguimiento de esta actividad y comienza a procesar segmentos una vez que la sesión está establecida, no puede determinar quién es el cliente y quién el servidor.
- Generar sesiones con diferentes números de secuencia pero idénticos parámetros.
- Es de suma importancia el conocimiento de la distribución de direcciones IP, para que un IDS pueda determinar el origen y destino de la información. Este concepto se hace de particular interés en el caso del establecimiento de sesiones TCP, pues si existen dispositivos de control de acceso (de los cuales también debería tener conocimiento), los mismos determinarán el “SENTIDO” en el que el establecimiento de las sesiones está autorizado o no (Esto se logra en base a dejar entrar o salir de la red local la combinación de los bit SYN y ACK en sentido saliente o entrante, en conjunto con la dirección IP del cliente o servidor). Sobre este punto se deberá decidir, si el IDS confía o no en la seguridad impuesta por el control de acceso, pues si confía, podrá descartar las reglas correspondientes a las direcciones externas y arriesgará sobre IP Spoofing, por el contrario si no confía en este dispositivo, deberá tener en cuenta la presencia en la red local de direcciones que no forman parte de la misma. Se remarca aquí nuevamente **el conocimiento que el IDS debe tener de la red**, caso contrario será totalmente ajeno a estos ataques.
- Falsas combinaciones de FLAGS:
 - Al establecer sesiones, triple handshake (SYN, SYN ACK, ACK).
 - Datos sin ACK.
 - Ambigüedad en el tratamiento de SYN con datos. Algunos SO lo aceptan y otros no.
 - Flag URG sin datos en el puntero de urgente.

- Flag PSH sin datos.
- Flag SYN con multicat o broadcast.
- Flag ACK fuera de secuencia.
- Flag ACK en secuencia repetida (¿Cuál se acepta?), el tratamiento de este también varía en los distintos SO.
- Diferentes combinaciones de estos seis bit
- Campo opciones. De manera similar a IP, esta campo presenta una serie de debilidades, algunas de estas opciones son definidas desde las primeras RFC y otras posteriormente, incrementando con esto la forma en que son implementadas por los distintos SO. Algunas de ellas son:
 - Posibilidad de emplear opciones con flag SYN.
 - Las opciones timestamp y windows scale, fueron creadas recientemente, por esta razón, algunos SO la soportan y otros no.
- Ambigüedades de tratamiento del segmento TCP por distintos SO:
 - Segmentos de longitud errónea.
 - Tratamiento de PMTU.
- Error en timestamp: Esta implementación se define a través de un concepto llamado PAWS (Protection against wrapped sequence numbers) en la RFC 1323, y permite determinar un umbral de tiempo para el seguimiento de los segmentos enviados y recibidos. Si el timestamp difiere de este umbral, el segmento es descartado. Sobre este tema el IDS no solo debe conocer el SO para determinar si este emplea o no PAWS, sino que debe tener en cuenta cual es el valor de ese umbral pues sobre esta base se tendrá en cuenta o no este segmento.
- Segmentos TCP fuera de ventana.
- Políticas de “Teardwon”: Las políticas de un IDS deben determinar a partir de qué momento se deja de registrar datos de una conexión. Es claro que el seguimiento de una conexión consume recursos, por lo tanto, un sistema que no libera los mismos en un momento dado, es fácilmente desbordable. El final de una conexión queda determinado a través de dos Flag de TCP (FIN o RST), si esto no sucede una conexión puede permanecer abierta por largos períodos de tiempo, debido a esto es que el IDS debe contemplar también algún tipo de “time out” en sus políticas para evitar ser atacado con estas técnicas.
- Probar iniciar sesiones sobre puertos poco usados.

5.4. Negación de servicio:

Un gran número de estos ataques aprovechan bugs de software del sistema operativo y otros, detalles particulares de los IDS. El objetivo final es el de evitar el procesamiento de la totalidad de las tramas que circulan por la red, dentro de las cuales podría estar la información del intruso. Algunos de estos ataques son

- Consumo de recursos: Se trata aquí de saturar algún recurso del IDS como ciclos de CPU, memoria, espacio en disco, o ancho de banda.
- Negación de almacenamiento o envío de logs.
- Inhibición de transmisión de los Event generators ("E-boxes") a los Event analyzers ("A-boxes"). Con este ataque se deja “ciego” al IDS.
- El ataque de pequeños fragmentos hacia varios host de la red, en un IDS es de particular impacto pues debe mantener en buffer la totalidad de los fragmentos, procesar cada uno de los que arriban hasta completar el reensamble. Nuevamente esta tarea está pensada par ser realizada por un solo host, y el IDS debe realizar la de todos los de la red.

- Consumo de ancho de banda: Este es el ataque más simple, sólo hace falta generar mucho tráfico en la red. Si en particular se aprovechan patrones de alarma conocidos, esta actividad consumirá más recursos aún. Si se trata de una red segmentada a nivel 2 (Switch), y se conoce parte de su topología, hasta se puede incrementar el tráfico en el dominio de colisión en el que se encuentra el IDS, dejando mayor libertad de acción en los segmentos restantes.
- Saturar el IDS con ecos ICMP o TCP.
- Generar un alto volumen de “falsos positivos”

5.5. Aprovechamiento de medidas reactivas:

En algunos casos, el IDS mismo es una herramienta para generar ataques de negación de servicio, se trata aquí de los que permiten adoptar medidas ante la detección de alarmas. Si se logra que el mismo reaccione ante una falsa alarma, se puede aprovechar la medida tomada por el IDS en provecho del atacante.

- En el caso de IP Spoofing, un intruso interno puede engañar un IDS, haciéndole creer que un determinado evento fue generado desde afuera de la red, obligando al IDS a cerrar este acceso, dejando aislada toda una red. Peor aún puede ser el caso si el IDS tiene autoridad para modificar rutas o filtros de acceso

5.6. HTTP:

- Empleo de Unicode.
- Empleo de más de una barra.
- Texto fragmentado por caracteres.
- Reemplazo de caracteres por su representación hexadecimal (o combinación de estos).
- Empleo de distintos códigos (ASCII, EBCDIC, Transcode, etc.).

5.7. Otros protocolos de nivel de aplicación:

- Inserción de caracteres extraños en varios protocolos de nivel de aplicación, pueden generar falsas alarmas o no generar cuando realmente deberían hacerlo.
- Empleo de Tab en vez de espacios en comandos. Puede causar que el IDS, acorde a sus reglas no interprete estos separadores de la misma forma. También puede funcionar con “,” en vez de “;”.
- Lanzar ataques desde uno o varios host “bounce”, es decir tomar posesión de una máquina intermedia y aprovechar esta para atacar. Si bien el IDS lo detectará, le será muy difícil realizar el seguimiento de ese evento.
- Desarrollar protocolos propietarios de nivel aplicación y atacar sobre estos.
- Desarrollar el ataque como una macro de Word o Power Point y enviar el documento a la víctima.

5.8. Otras técnicas:

- Reordenamiento de un ataque detectado: Modificar la secuencia de tramas que hizo saltar una alarma en el IDS.
- Lanzar un ataque conocido, pero a través de más de un usuario (o IP o MAC Spoofing con el mismo), es decir, cada usuario lanza parte de un ataque, entremezclándolo.
- Partir un mismo ataque a través de varias sesiones, es decir lanzar una primera parte, cerrar la conexión, abrirla nuevamente, lanzar la segunda, cerrarla, abrirla, lanzar la tercera, y así sucesivamente.
- Crear macros de ataques reales, definiendo variables que reemplacen a la secuencia conocida, y luego enviar las variables en vez del patrón real.

- Crear scripts en el shell que reemplacen a los comandos que se necesitan emplear, y emplear los nombres de estos scripts en lugar de los comandos.
- Emplear diferentes comandos para realizar la misma función. Por ejemplo “echo * es casi lo mismo que “ls” en la familia Unix.
- Cambiar los nombres en los ataques estándar.
- Codificar el ataque en EBCDIC y cambiar el tipo de terminal a terminal EBCDIC. Todo el conjunto de caracteres será diferente.
- Probar de criptografiar los ataques.
- Escribir todo al revés y emplear un programa que lo revierta.
- Escribir los comandos muy lentamente (tardando horas), es muy probable que el IDS no realice el seguimiento de conexiones tan largas.
- Cambiar las rutas hacia el host destino, tanto de envío como de recepción.
- Iniciar sesiones desde otra conexión (ADSL, RDSI, telefonía analógica, etc).
- Compilar el ataque (como un troyano) y enviarlo a la víctima para que lo ejecute.
- Recompilar el ataque en un lenguaje diferente al que fue publicado.

6. Conclusiones:

a. Disparidad en la detección de un mismo evento por distintos productos:

Se puede apreciar en las mediciones presentadas en el punto 4.a. (las cuáles son sólo una parte de la totalidad realizada en este trabajo) que ante una secuencia de eventos importantes, **los distintos IDS responden de manera totalmente distinta.**

Este detalle permite inferir que **los distintos fabricantes asignan prioridades totalmente diferentes** a lo que se puede considerar un evento.

Siguiendo esta línea de pensamiento, **es válido creer que aún no existe un consenso** acerca de lo que se considera una alarma o no, sino los porcentajes de detección serían mucho más cercanos.

Resultados tomados del punto 4.a.: **56, 142 y 128 eventos detectados respectivamente.**

Estos valores son claros indicadores que no se ha llegado a un nivel de madurez importante en lo que un usuario final espera de estos productos, que en definitiva es poseer una herramienta más del sistema de seguridad que permita generar alarmas: confiables, oportunas y veraces.

Si su madurez fuera mayor, se podría comparar a la de cualquier otro producto, como por ejemplo los Firewalls, con los cuales este mismo tipo de mediciones presenta una respuesta que coincide en un altísimo porcentaje, pues ante un conjunto de reglas similares, los mismos bloquean o dejan pasar patrones que llevan ya varios años puestos a prueba, lo mismo se podría decir de los router y muchos productos más que han alcanzado ya una cierta madurez en el mercado.

b. Ausencia de detección del no cumplimiento a lo establecido por las RFCs.

Si se trata de obtener conclusiones a las mediciones realizadas en el **punto 4.b.** realmente es alarmante como ante campos que se encuentran taxativamente prohibidos en las RFCs, los mismos no son tenidos en cuenta en el análisis realizado por los IDS.

Este fue el detalle que más sorprendió durante el trabajo (y el motivo fundamental para escribir este texto), pues de la cantidad de tramas generadas hasta el momento, como se

puede apreciar en la tabla resumen que se presenta a continuación, sólo el 3,22 % fue detectado en el mejor de los casos (producto B con 96 de detecciones).

| Nº | Protocolo | Cant. tramas generadas | <u>Prod. A</u> Cant. de tramas | <u>Prod. B</u> Cant. de tramas | <u>Prod. C</u> Cant. de tramas |
|---------|------------------------------------|------------------------|-----------------------------------|-----------------------------------|-----------------------------------|
| 1 | arp1.cap | 500 | 0 | 12 | 2 |
| 2 | Ethernet1.cap | 170 | 0 | 11 | 0 |
| 3 | icmp1.cap | 578 | 0 | 6 | 16 |
| 4 | dhcp1.cap | 230 | 0 | 0 | 0 |
| 5 | ip1.cap | 261 | 4 | 2 | 1 |
| 6 | udp1.cap | 119 | 0 | 30 | 1 |
| 7 | http1.cap | - | - | - | - |
| 8 | tcp1.cap | 757 | 9 | 18 | 2 |
| 9 | igmp1.cap | 232 | 0 | 0 | 0 |
| 10 | Snmp1.cap | - | - | - | - |
| 11 | telnet1.cap | 59 | - | - | 4 |
| 12 | ftp1.cap | - | - | - | - |
| 13 | smtp1.cap | - | - | - | - |
| 14 | ssh1.cap | - | - | - | - |
| 15 | dns.cap | - | - | - | - |
| 16 | Ataque http con vulnerabilidad CGI | 70 | 8 | 17 | 16 |
| TOTALES | | 2976 | 21 | 96 | 42 |

Sobre este cuadro se podrían hacer varias hipótesis:

- Los eventos se presentan sólo como una ocurrencia, aunque aparezcan varias veces.
- No todos los valores erróneos en los campos deben generar eventos.
- Algunos valores pueden no ser considerados como erróneos, debido a las ambigüedades planteadas anteriormente con los distintos diseñadores de SO y aplicaciones.
- Este tipo de tráfico anómalo no presenta ninguna falla en seguridad pues no sirve para fingerprinting, para acceder a host, ni para insertar información.

Cualquiera de estos planteos puede ser válido, pero lo que parece ser muy cierto es que **el nivel de detección es ínfimo y en algunos casos nulo.**

Se debería reflexionar aquí sobre la enorme cantidad de antecedentes reales que se tiene sobre estas violaciones a las RFC, o aprovechamiento de las ambigüedades sobre la interpretación de las mismas, sacando ventaja los intrusos de alguna u otra manera. La masa de los ataques ICMP, ARP, TCP, etc., justamente hacen uso ilegal de campos o combinaciones de ellos. Como estos se podrían citar muchos ejemplos más.

Aquí es donde nace la idea de comenzar a trabajar en forma “Proactiva. y no “Reactiva”. Teniendo como punto de partida lo que está permitido, no establecido y prohibido en las RFCs correspondientes

c. Faltas de desarrollos en el relevamiento del software y hardware de red.

En el estudio de las vulnerabilidades y su comparativa con las mediciones realizadas en laboratorio y en producción, se pudo comprobar la veracidad de la afirmación del

conocimiento que debe tener un IDS de la red que está vigilando. Sin un nivel de detalle en esta actividad, el rol del IDS es prácticamente un fraude, pues nunca podrá:

- Saber si un ataque es interno o externo.
- Personalizar reglas en detalle que minimicen los eventos a los verdaderamente importantes.
- Compartir Logs y tareas con los firewalls, proxies y routers de la red.
- Saber que nivel de confiabilidad le proporciona una determinada captura.
- Determinar IP y MAC spoofing.
- Sincronizar relojes con los host de mayor impacto en la red.
- Determinar los límites de su competencia.
- Lanzar contramedidas responsablemente.
- Realizar el seguimiento de una intrusión.
- Seguir el rastro inverso de un evento.
- Evitar ambigüedades de tratamiento de la información respecto a los distintos SO y aplicaciones.
- Etc., etc., etc., etc.....

Este tema es de vital importancia y hasta el momento no se ha tenido en cuenta por los fabricantes de IDS. No se aprecia que la solución pase por integrar todo en un solo producto en un mismo host, pero sí se debería plantear alternativas de sistemas que ejecutándose en distintas máquinas puedan reunir eventos en una misma base de datos o también en una distribuida, consolidando toda la información que se dispone de la política de seguridad de la red, y de la cual al analizar un evento determinado que pueda plantear dudas, permita obtener información al respecto.

Las funciones que se aprecian de vital importancia en cuanto a la red son:

- Reconocimiento de direcciones IP y planos de red (Tipo Tívoli, Open View, Trascend).
- La asociación de las mismas con direcciones MAC (Tipo ARPWatch).
- El conocimiento de los servidores de la red (Hardware, Software y aplicaciones)
- El conocimiento de los clientes habituales de esos servidores.
- Las listas de control de accesos en routers y las reglas de los firewalls.
- Permisos (Usuarios, host y direcciones IP) sobre el empleo de Telnet, ftp, SSH, SNMP, etc.
- Elementos activos de red.
- Elementos de monitoreo y alarmas.
- Vínculos de acceso.

d. Faltas de iniciativas sobre trabajo en reglas “Proactivas”.

Se considera que si no se comienza a estudiar e implementar este tipo de medidas “Proactivas. y no sólo “Reactiva”, se estará siempre un paso atrás, jamás se podrá detectar una vulnerabilidad antes que un intruso la emplee. No se trata de una tarea fácil, pero

teniendo como punto de partida lo que está permitido, no establecido y prohibido en las RFCs correspondientes, se puede dar comienzo a un trabajo que permita adelantarse a los acontecimientos o incrementar un poco más las alternativas de detección de los IDS, pues se está totalmente seguro que sobre estas falencias surgirán nuevos ataques como ya lo hicieron en su oportunidad.

- e. **Estudiar muy en detalle qué IDS se ajusta mejor a la red de la empresa** sobre la cual se debe instalar, pues cada producto en particular tiene sus ventajas y desventajas. Acorde a la magnitud de la red, a la cantidad de personal que se posea para administrarla, al nivel técnico que se posea, a la planificación de gastos de mantenimiento que se pretendan con el mismo, al nivel de hardware que se posea, al grado de exposición de recursos que tenga la organización, a la importancia que se le de a la seguridad, al apoyo que se posea de la más alta conducción de la empresa, a la experiencia en seguridad, qué tipo de análisis se prefiere, etc. Las distintas ofertas de IDS del mercado responderán de manera diferente, pues como se pudo apreciar en las mediciones, no todos trabajan igual.

7. Fuentes consultadas:

- a. D. Schanackenberg – H. Holliday – R. Smith – K. Djahandari - D. Sterne, *Cooperative Intrusion Traceback and Response Architecture (CITRA)*, Boeng Phantom Company – NAI Labs, 2000.
- b. D. Schanackenberg – K. Djahandari - D. Sterne, Boeng Phantom Company *Infraestructure for Intrusion Detection and Response*, – NAI Labs, 2000.
- c. M. Ranum, *Coverage in Intrusion Detection System*, mjr@nfr.com, <http://www.nfr.com>, 2001.
- d. M. Ranum, *Experiences Benchmarking Intrusion Detections System*, mjr@nfr.com, <http://www.nfr.com>, 2001.
- e. F. Cohen, *50 Ways to defeat Your Intrusion Detection System*, fc@all.net, 2001.
- f. NSS Group, *Intrusion Detection & Vulnerability Assesment*, <http://www.NSS.co.uk>. 2000.
- g. NSS Group, *Intrusion Detection System*, <http://www.NSS.co.uk>. 2000.
- h. K. Frederick, *Abnormal IP Packet*, mjr@nfr.com, <http://www.securityfocus.com/>, 2001.
- i. K. Frederick, *Studying Normal Network Traffic, Part One*, mjr@nfr.com, <http://www.securityfocus.com/>, 2001.
- j. K. Frederick, *Studying Normal Network Traffic, Part two: Studying FTP Traffic*, mjr@nfr.com, <http://www.securityfocus.com/>, 2001.
- k. K. Frederick, *Studying Normal Network Traffic, Part three: TCP Header*, mjr@nfr.com, <http://www.securityfocus.com/>, 2001.
- l. Karen Frederick, *Network Monitoring for Intrusion Detection*, <http://www.securityfocus.com/>, 2001.
- m. B. Smith, *Thinking about Security Monitoring and Event Correlation*, bsmith@lurhq.com, 2000.

-
- n. *Welcome to the Intrusion Detection Systems Product Survey* (<http://www.c3.lanl.gov/~reid/kaj/>).
 - o. T. Miller, *Analysis of the Torn Rootkit*, infowar@erols.com, 2000.
 - p. T. Miller, *Social Engineering: Techniques that can bypass Intrusion Detection System*, infowar@erols.com, 2000.
 - q. T. Miller, *ECN and it's impact on Intrusion detection*, infowar@erols.com, 2001.
 - r. T. Miller, *Intrusion Detection Level Analysis of Nmap and Queso*, infowar@erols.com, 2000.
 - s. T. Miller, *Hacker Tools and Their Signatures, Part One*, infowar@erols.com, 2001.
 - t. Ofir Arkin, *Identifying ICMP Hackery Tools Used In The Wild Today*, ofir@sys-security.com, 2000.
 - u. E. Hacker, *IDS Evation with Unicode*, ehacker@lucent.com, 2001.
 - v. E. Hacker, *Re-synchronizing a NIDS*, ehacker@lucent.com, 2000.
 - w. T. Ptacek - T. Newsham, *Intrusion, Detection, and Denial of Service: Eluding Network Intrusion Detection*, tqbf@securenetworks.com - newsham@securenetworks.com, Secure Networks Inc., 1998.
 - x. D. Elson, *Intrusion Detection, Theory and Practice (Introduction)*, del@babel.com.au, 2001.
 - y. D. Elson, *Intrusion Detection on Linux*, del@babel.com.au, 2000.
 - z. R. MacBride, *Intrusion Detection: Filling in the Gaps*, rob.macbride@capitalone.com, 2000.
 - aa. G. Hoglund – J. Gary, *Multiple Levels of De-synchronization and other concerns with testing an IDS system*, hoglund@ieway.com - jgary@skylab.org, 2000.
 - bb. E. Powell, *Network Intrusion Detection for the E-Commerce Environment*, epowell1@tampabay.rr.com, 2000.
 - cc. L. Spitzner, *Passive Fingerprinting*, lance@spitzner.net, 2000.
 - dd. R. Wiens, *Realistic Expectations for Intrusion Detection Systems*, richard.wiens@getronics.com, 2001.
 - ee. C. Jordan, *Analysing IDS Data*, endeavor@nexus.net, 2000.
 - ff. G. Schultz, *Interview with Three Top Intrusion Detection Experts*, Information Security Bulletin, 2000.
 - gg. Test Centre, *Security Magazine*, 2002.
 - hh. J. Forristal – G. Shipley, *Vulnerability Assessment Scanners*, <http://www.networkcomputing.com/1201/1201f1b1.html>, 2001.

Alejandro Corletti es militar Argentino, Ingeniero en Informática, cursó un postgrado en Administración y Conducción y en la actualidad se encuentra cursando el Doctorado en Ingeniería de Sistemas Telemáticos en la Universidad Politécnica de Madrid, siendo su tema de investigación “Seguridad en entornos TCP/IP”.

Fue Jefe de Redes del Ejército Argentino durante 3 años. Es docente la Universidad de Ejército Argentino y de la Universidad Tecnológica Nacional de ese País, es Subdirector del CISIar (Centro de Investigación en Seguridad Informática de Argentina), también se desempeñó como docente en Telefónica de Argentina, CISCO System, CYBSEC. S.A. y como asesor en temas de seguridad en varias Empresas.