JITEL 2017
XIII Jornadas de Ingeniería Telemática
27-29 Septiembre 2017, Valencia (España)

Actas de las XIII Jornadas de Ingeniería Telemática (JITEL 2017), Valencia (España), 27-29 de Septiembre de 2017.

ISBN: 978-84-9048-595-8
DOI: http://dx.doi.org/10.4995/JITEL2017.2017.6520

UGR'16: Un nuevo conjunto de datos para la evaluación de IDS de red

Gabriel Maciá Fernández*[1], José Camacho[1], Roberto Magán-Carrión[1], Marta Fuentes-García[1], Pedro García-Teodoro[1]
[1]Departamento de Teoría de la Señal, Telemática y Comunicaciones - CITIC, Universidad de Granada - España
C/ Periodista Daniel Saucedo Aranda, s/n 18071 Granada
{gmacia,josecamacho,rmagan,nmfuentes,pgteodor}@ugr.es

Roberto Theron[2]
[2]Universidad de Salamanca - España
theron@usal.es

***Resumen*—La evaluación de algoritmos y técnicas para implementar sistemas de detección de intrusiones depende en gran medida de la existencia de conjuntos de datos (*dataset*) bien diseñados. En los últimos años, se ha realizado un gran esfuerzo para construir estos *datasets*. En este trabajo se presenta un nuevo *dataset* que se construye a partir de tráfico real y donde se realizan ataques actualizados. La principal ventaja de este conjunto de datos sobre otros previos es su utilidad para la evaluación de IDSs donde se considera la evolución a largo plazo y la periodicidad del tráfico. También permite entrenar y evaluar modelos que contemplen las diferencias entre día/noche o entre días laborables/fines de semana.**

***Palabras Clave*—seguridad en redes, dataset, IDS, tráfico de red, netflow**

I. INTRODUCCIÓN

Los Sistemas de Detección de Intrusiones (IDS) aparecieron en la esfera de la seguridad como una solución al problema de identificar actividades maliciosas en redes y sistemas. En pocas palabras, un IDS consta de un módulo encargado de la obtención de datos, un módulo de pre-procesamiento que adapta esos datos para los siguientes pasos en el sistema, y un módulo de decisión capaz de determinar si un evento debe ser considerado malicioso o no.

Existen varios tipos de IDS [1]: los *IDSs basados en red* (NIDS) monitorizan eventos de red como flujos o logs de cortafuegos, entre otros, mientras que los *IDS basados en host* (HIDS) consideran eventos relacionados con el sistema, por ejemplo *syslog*, monitorización de sistemas de archivos, carga de la CPU, etc. Los IDS también se clasifican de acuerdo al proceso de detección. Así, los *IDS basados en firmas* (S-IDS) hacen uso de reglas para decidir si un comportamiento observado es malicioso o no, mientras que los *IDS basados en anomalías* (A-IDS) [2] construyen un modelo a partir de datos de entrenamiento y consideran que cualquier comportamiento que se desvía de este modelo es anómalo. Hay que destacar que, aunque existe una diferencia semántica entre un comportamiento anómalo y uno malicioso, un A-IDS los considera equivalentes.

Un problema esencial cuando se evalúan las capacidades de los IDS es la necesidad de un conjunto de datos representativo que permita la comparación entre distintas propuestas. En los años 90 DARPA llevó a cabo un proyecto para construir un conjunto de datos con este fin, generándose los *datasets* ***DARPA'98*** y ***DARPA'99*** del MIT. [3]. Después de ser utilizados y estudiados ampliamente por varios autores, se identificaron algunas limitaciones, como la existencia de registros duplicados, muestras no balanceadas entre ataques y conexiones normales, y otras limitaciones inherentes por considerar tráfico sintético. Desde entonces, muchos otros investigadores y proyectos han intentado proporcionar versiones mejoradas de estos conjuntos de datos, como NSL-KDD, o construir nuevos *datasets*.

Más recientemente se han propuesto otros conjuntos de datos. Por ejemplo, ***UNB ISCX 2012***, creado en 2012 por Shiravi *et al.* [4]. La contribución más relevante de este trabajo es el uso de perfiles para la generación de tráfico. Los autores definen ciertos perfiles α para el tráfico de ataque y perfiles β para el tráfico de *background*. Implementan su propuesta en una red con 17 estaciones de Windows XP y un único ordenador Windows 7, capturando datos durante 7 días. El principal inconveniente de este *dataset* en la actualidad son su duración reducida, el uso de algunos sistemas operativos anticuados (Windows XP), y el uso de

tráfico sintético. ***UNSW-NB15*** fue propuesto por Moustafa *et al.* [5] en 2015. Los autores utilizaron una herramienta de generación automático de ataques llamada *IXIA Perfect-Storm* para implementar nueve familias de ataques reales y actualizados contra varios servidores. Capturaron las trazas *tcpdump* del tráfico de red en una duración total de 31 horas a comienzos de 2015, obteniendo 2 millones de flujos. A partir de estas trazas, se construyó un *dataset* de 49 características para cada flujo. El principal problema de este conjunto de datos es la generación sintética de tráfico, que está asociada a comportamientos teóricos en lugar de realistas en Internet.

Además, existen distintos conjuntos de datos que son específicos de ciertas áreas. Por ejemplo, se están construyendo nuevos *datasets* para sistemas de control industrial (SCADA) [6] [7].

A pesar de este gran número de esfuerzos para contribuir con un conjunto de datos definido para la evaluación de IDS, y después de haber aprendido varias lecciones, es posible constatar que, hasta el momento, todas ellas son soluciones parciales. En una primera revisión, se puede comprobar que muchos de los conjuntos recientes carecen de tráfico real o estrategias de ataque actualizadas. Otra limitación importante está relacionada con la duración de las capturas de datos. Esto es, para hacer posible la evaluación de algoritmos de detección que consideran la evolución ciclo-estacionaria del tráfico, es decir, las diferencias en el tráfico entre día/noche o laborables/festivos, se necesita una traza de larga duración.

Un problema adicional es el del *nivel de dificultad* del *dataset* [8]: si algoritmos de detección simples proporcionan buenos resultados de detección, el nivel de dificultad del *dataset* es bajo. Se puede verificar que algunos de los algoritmos propuestos en los últimos años proporcionan tasas de detección próximas al 100% con índices de falsos positivos realmente bajos. Sin embargo, los sistemas de detección reales todavía están muy lejos de funcionar tan bien. Esto provoca la sospecha de que el problema podría no residir solamente en el diseño de los algoritmos, sino en los conjuntos de datos utilizados para la evaluación.

En este trabajo, se describe un nuevo *dataset* (el conjunto de datos UGR'16[1]) que contiene trazas reales de *netflow* anonimizadas capturadas en un ISP Tier-3 durante 4 meses. En este conjunto, se han incluido escenarios de ataque realistas y se ha llevado a cabo el etiquetado el tráfico. Además, se demuestra que el nivel de dificultad del *dataset* es suficientemente elevado para probar nuevos algoritmos de detección.

El documento está estructurado como sigue. En la Sección II se describe cómo se ha construido el *dataset* y la metodología seguida para insertar tráfico de ataque. En la Sección III se analiza el conjunto de datos, discutiendo y proporcionando una descripción global de la información contenida en él. A continuación, se discute el proceso de etiquetado y se realiza una evaluación con algoritmos de detección del estado del arte para comprobar el nivel de dificultad del *dataset*. Finalmente, se presenta la Sección V de conclusiones.

[1]Tomamos este nombre para el *dataset* del acrónimo de Universidad de Granada

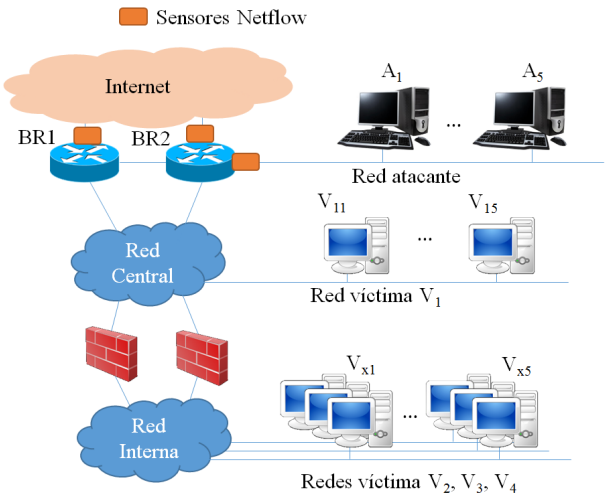


Fig. 1. Topología de la red.

II. METODOLOGÍA PARA LA GENERACIÓN DEL CONJUNTO DE DATOS

En lo que sigue, se describe la metodología seguida para producir el conjunto de datos. Además, se proporciona una extensa descripción de todos los detalles relevantes de cara a su utilización en la evaluación de un IDS.

A. *Infraestructura de red*

Los datos se obtienen de una red real de un ISP Tier-3. El ISP es un proveedor de servicios en la nube, por lo que algunos de los servicios típicos implementados en la red están virtualizados. Algunos de los servicios típicos de alojamiento que se encuentran son webs con configuraciones propietarias o estándares, por ejemplo Joomla o Wordpress, correo electrónico, servidores FTP y DNS, etc. Esta red se utiliza por muchas compañías que tienen tamaños dispares y que se centran en una gran variedad de mercados. Se espera así que el tráfico que atraviesa la red sea muy heterogéneo, pues incluye tanto accesos de clientes a Internet como recepción de tráfico por servidores típicos. Por tanto, una potencial ventaja de esta traza sobre otros conjuntos de datos es su representatividad de un amplio subconjunto de usuarios de Internet. Muchas otras bases de datos solamente recolectan tráfico de universidades o centros de investigación, donde sólo se presentan patrones de tráfico específico.

La topología esquemática de la red del ISP y la infraestructura usada para la recolección de datos se muestran en la Fig. 1. Los principales elementos son:

- Dos *routers* frontera redundantes, *BR1* y *BR2*, proporcionan acceso a Internet. En cada una de sus interfaces de red salientes, se configura un sensor de *netflow* que permite la recolección de todas las conexiones de entrada y salida. Nótese que, por razones de privacidad y volumen, no se proporciona información de carga útil, de modo que la información

se almacena con el formato de archivos *netflow* en lugar de *pcap* (*tcpdump*).
- El ISP tiene dos sub-redes distintas. Una denominada *red central*, donde se localizan los servicios que no están protegidos por cortafuegos. La segunda es la *red interna*, donde se proporcionan servicios de cortafuegos a los clientes.
- Una *red de atacantes* con 5 máquinas se despliega en el nivel superior. Nos referimos a estas máquinas como A_1-A_5.
- En la red central, se configuran 5 máquinas víctima que se utilizan solamente para la recogida de datos. Se colocan junto con otros clientes reales en una red existente que llamamos *red víctima* V_1. Estas máquinas se denominan V_{11}-V_{15}.
- En lo referente a la red interna, se emplazan un total de 15 máquinas víctima adicionales en tres redes existentes y distintas, cada una con 5 máquinas. Llamamos a esas redes: *red víctima* V_2 (V_{21}-V_{25}), *red víctima* V_3 (máquinas V_{31}-V_{35}) y *red víctima* V_4 (V_{41}-V_{45}).

B. Generación de tráfico de ataque

Algunos *datasets* existentes, como MAWILab [9] o CAIDA [10], solamente consideran el tráfico real capturado de ciertos sensores de red. Aunque es una ventaja para modelar el tráfico de *background*, esto conlleva ciertas limitaciones en la identificación de ataques. De hecho, el etiquetado de tráfico real implica la necesidad de confirmar que las conexiones señaladas como ataques son realmente tráfico malicioso. Por esta razón, se decide combinar tráfico de *background* real (que probablemente contenga instancias de ataque) con ataques que se generan intencionadamente para el experimento.

Con este objetivo, las 25 máquinas virtuales mencionadas se instalan con una configuración similar a las proporcionadas para los clientes ISP, es decir, se implementan servidores web, DNS, FTP y de correo electrónico. Las máquinas virtuales A_1 a A_5 se utilizan para lanzar un número de ataques específicos a lo largo del tiempo contra el resto ($V_{x1} - V_{x5}$, con $x = \{1 - 4\}$), que juegan el rol de víctimas de los ataques. Como se puede observar en la Fig. 1, tanto atacantes como víctimas se encuentran dentro de la infraestructura ISP para evitar la potencial detección y bloqueo de los ataques por otros ISP intermedios. Además, la red de atacantes se ubica en el *router* frontera para simular que el tráfico de ataque procede de Internet.

Implementación de ataques. Debido a que solamente se recoge tráfico de *netflow*, y por tanto, no se considera el *payload* de la información en la traza, no se incluyen tipos de ataque susceptibles de ser detectados mediante análisis del *payload*. Solamente se consideran ataques relacionados con la red. Los tipos de ataque implementados son:

- *DoS de baja tasa*: Se envían paquetes TCP SYN a las víctimas utilizando la herramienta `hping3`. El puerto destino es el 80, por lo que el tráfico se mezcla con el tráfico web de *background* real. El tamaño de cada paquete es de 1280 bits y la tasa es de 100 paquetes/s. Como puede comprobarse, la tasa de ataque es baja, por lo que no se afecta la operación normal de la red. Se consideran tres escenarios distintos de ataque:
  - *DoS11*: Ataque DoS uno-a-uno, donde el atacante A_1 ataca a la víctima V_{21}. La duración total de *DoS11* es de 3 minutos.
  - *DoS53s*: Los cinco atacantes A_1-A_5 atacan a tres de las víctimas, cada una en una red diferente, durante 3 minutos. En particular, estos ataques siguen esta estructura: (A_1, A_2) $\rightarrow$ V_{21}, (A_3, A_4) $\rightarrow$ V_{31} and A_5 $\rightarrow$ V_{41}. La letra '*s*' al final del nombre del ataque representa 'síncrono', lo que significa que los ataques se inician por todos los atacantes al mismo tiempo. Debido a esta sincronización, la duración de *DoS53s* es de 3 minutos también.
  - *DoS53a*: Los ataques se ejecutan como en *DoS53s*, pero ahora cada víctima es seleccionada secuencialmente, siendo atacada durante 3 minutos con un periodo de inactividad de 30 segundos entre los tres ataques. De esta forma, la duración total de *DoS53a* es de 10 minutos. En este caso, la letra '*a*' al final del nombre del ataque representa 'asíncrono'.
- *Escaneo de puertos*: Se ejecuta un escaneo SYN continuo a los puertos comunes de las víctimas durante 3 minutos, utilizando la herramienta `nmap`. Se implementan dos variantes para este ataque:
  - *Scan11*: Ataque de escaneo uno-a-uno, donde el atacante A_1 escanea a la víctima V_{41}.
  - *Scan44*: Ataque de escaneo cuatro-a-cuatro, donde los atacantes A_1, A_2, A_3 y A_4 inician un escaneo al mismo tiempo a las víctimas V_{21}, V_{11}, V_{31} y V_{41}, respectivamente. Como los ataques se llevan a cabo en paralelo (comienzan en el mismo instante), la duración total es de 3 minutos.
- *Actividad relacionada con una botnet*. Se simula tráfico de *botnet* mediante la exfiltración de datos desde algunas máquinas infectadas al puerto 80 de un *botmaster* localizado en A_1. Se consideran veinte *bots*, correspondientes a todas las máquinas víctima. Cada uno de los *bots* lleva a cabo estas variantes de exfiltración:
  - *Exf1KB*: Se envía un fragmento de información de 1KB al *botmaster*.
  - *Exf1MB*: Se envía un total de 1MB de información al *botmaster* en una única conexión.
  - *Exf1MBp*: El fragmento de información de 1MB a enviar al *botmaster* se divide en trozos de 1KB cada uno, y se envía al *botmaster* en conexiones distintas.

  De nuevo, la transmisión desde cada *bot* puede ser *síncrona* (sufijo *'s'*), lo que significa que todos ellos inician la transmisión de información al mismo tiempo, o *asíncrona* (sufijo *'a'*), donde cada uno de

Tabla I
PROGRAMACIÓN PLANIFICADA PARA CADA UNO DE LOS ATAQUES EN EL INTERVALO DE TIEMPO DE 2H COMENZANDO EN t_0.

Instante de inicio	Ataque	Duración
$t_0 + 0h00m$	DoS11	3m
$t_0 + 0h10m$	DoS53s	3m
$t_0 + 0h20m$	DoS53a	10m
$t_0 + 0h40m$	Scan11	3m
$t_0 + 0h50m$	Scan44	3m
$t_0 + 1h00m$	Exf1KBs	$\leq$10m
$t_0 + 1h10m$	Exf1MBs	$\leq$10m
$t_0 + 1h20m$	Exf1MBps	$\leq$10m
$t_0 + 1h30m$	Exf1KBa	$\leq$10m
$t_0 + 1h40m$	Exf1MBa	$\leq$10m
$t_0 + 1h50m$	Exf1MBpa	$\leq$10m

ellos selecciona un instante aleatorio en una ventana de 3 minutos para empezar el correspondiente procedimiento de exfiltración. En cualquier caso, es importante mencionar que los ataques relacionados con *botnet* en todas las variantes se restringieron a una duración menor de 10 minutos.

Programación de la ejecución de ataques. El tráfico de ataque se genera en lotes de 2 horas. En cada lote de ataque, se ejecutan todas las variantes de ataque siguiendo dos patrones posibles de programación:

1) *Programación planificada*: Se ejecuta cada ataque en el lote en un instante fijo y conocido dado por un desplazamiento desde el instante inicial de tiempo t_0. Los desplazamientos para los distintos ataques se muestran en la Tabla I. Nótese que no hay solapamiento en el tiempo para los distintos tipos de ataque.
2) *Programación aleatoria*: El instante inicial para la ejecución de cada uno de los ataques se selecciona aleatoriamente entre $t_0 + 00h00m$ y $t_0 + 01h50m$, por lo que se restringe la duración total del lote a un máximo de dos horas. En este caso, podría existir solapamiento temporal entre los ataques, lo que permitirá comprobar la adecuación de los detectores de anomalías cuando esta situación aparece.

Merece la pena destacar que los ataques se lanzaron mientras que el tráfico real de *background* atravesaba la red. De esta forma, el tráfico capturado por los sensores para la correspondiente ventana de monitorización incluirá instancias de tráfico relacionado tanto con ataques como con tráfico normal. Así, para permitir el estudio del tráfico de *background* para distintas horas del día junto con tráfico de ataque, se lanzan lotes de ataque durante 12 días consecutivos desde el comienzo del experimento de ataque, por lo que se cubren todas las horas posibles del día. Cada día en el experimento de ataque, se lanza un lote de programación planificada en el instante t_0, seguido de un lote de programación aleatoria que se inicia en t_0+12h. En cada día siguiente durante 12 días, t_0 se incrementa con un desplazamiento de $2h$. En la Tabla II se pueden ver los diferentes instantes de tiempo y fechas seleccionados para la ejecución de los lotes planificados y aleatorios.

Tabla II
FECHA Y HORA PARA LA EJECUCIÓN DE DISTINTOS LOTES DE ATAQUE EN EL CONJUNTO DE DATOS.

Fecha	Planificada	Aleatoria
Jue, 28/07/2016	00:00	12:00
Vie, 29/07/2016	02:00	14:00
Sab, 30/07/2016	04:00	16:00
Dom, 31/07/2016	06:00	18:00
Lun, 01/08/2016	08:00	20:00
Mar, 02/08/2016	10:00	22:00
Mie, 03/08/2016	12:00	N/A
Jue, 04/08/2016	14:00	00:00
Vie, 05/08/2016	16:00	02:00
Sab, 06/08/2016	18:00	04:00
Dom, 07/08/2016	20:00	06:00
Lun, 08/08/2016	22:00	08:00
Mar, 09/08/2016	N/A	10:00

C. Capturas del dataset

Los flujos se capturan y transfieren desde los sensores indicados en la red utilizando el formato Netflow v9 (ver Fig. 1). Los parámetros por defecto se mantuvieron durante la configuración *netflow* de los *routers* Cisco, esto es, el temporizador inactivo es de 15 segundos, mientras que el cronómetro activo para flujos es de 30 minutos.

El *dataset* completo contiene dos capturas distintas: un *conjunto de calibración* y un *conjunto de prueba*. El *conjunto de calibración* dura 100 días, y se extiende desde 10:52-18/03/2016 hasta 18:27-26/06/2016. Su principal propósito es ayudar a la construcción y calibración de modelos de normalidad, principalmente porque no se generaron ataques de forma artificial. Nótese que esto no implica que no haya presencia de ataques, como se discutirá a continuación en la Sección III.B.

Aunque la captura del *conjunto de calibración* se programó de forma automática para ser continua a lo largo del tiempo, se interrumpió dos veces por el ISP para llevar a cabo procedimientos de mantenimiento específicos de la red. Estos dos intervalos son:

1) [02:00 27/03/2016 — 03:00 27/03/2016]
2) [00:00 01/04/2016 — 17:20 06/04/2016]

El *conjunto de prueba* dura aproximadamente un mes, desde 13:43-27/07/2016 hasta 09:27-29/08/2016. Durante esta captura, los lotes de ataque se ejecutaron comenzando en 00:00-28/07/2016 y finalizando en 12:00-09/08/2016, como se muestra en la Tabla II. Esta captura pretende ser utilizada para la validación de los algoritmos de detección.

La Tabla III resume las diferentes características para los conjuntos tanto de *calibración* como de *prueba*. Se pueden ver las fechas para las dos capturas y el número y tamaño de los archivos incluidos en cada una. Adicionalmente, se muestra el número de conexiones incluidas en los dos conjuntos. Este número es mucho mayor que en los demás conjuntos de datos.

D. Pre-procesamiento y disponibilidad del dataset

Los archivos de formato binario `nfcapd` recolectados para los conjuntos de calibración y de prueba se agrupan

Tabla III
CARACTERÍSTICAS DE LOS CONJUNTOS DE CALIBRACIÓN Y PRUEBAS.

Característica	Calibración	Prueba
Inicio de la captura	10:47h 18/03/2016	13:38h 27/07/2016
Fin de la captura	18:27h 26/06/2016	09:27h 29/08/2016
Inicio de los ataques	N/A	00:00h 28/07/2016
Fin de los ataques	N/A	12:00h 09/08/2016
Número de archivos	17	6
Tamaño (comprimido)	181GB	55GB
# Conexiones	$\approx$ 13.000M	$\approx$ 3.900M

Tabla IV
CORRESPONDENCIA DE DIRECCIÓN IP ANONIMIZADA CON LAS MÁQUINAS EN LA CONFIGURACIÓN EXPERIMENTAL.

Máquinas/s	Direccion/es IP
A_1 - A_5	42.219.150.{*246,247,243,242,241*}
V_{11} - V_{15}	42.219.156.{*30,31,29,28,27*}
V_{21} - V_{25}	42.219.158.{*16,17,18,19,21*}
V_{31} - V_{35}	42.219.152.{*20,21,22,23,18*}
V_{41} - V_{45}	42.219.154.{*69,68,70,71,66*}

en un único archivo por semana para los dos periodos de captura. El tamaño medio de los diferentes archivos está en torno a $14GB$ en formato de compresión `tar`. En el conjunto de calibración hay 17 archivos, mientras que para el conjunto de prueba hay 6 archivos disponibles. Todos estos archivos están disponibles para descargar en nuestra página web: https://nesg.ugr.es/nesg-ugr16/.

La información disponible se corresponde con trazas *netflow* tanto en formato `nfcapd` como `csv`, éste último obtenido del posprocesado de los ficheros `nfcapd` mediante la herramienta `nfdump`. Las características que han sido seleccionadas para el formato `csv` son[2]: *timestamp* del final de un flujo (*te*), duración del flujo (*td*), dirección IP de origen (*sa*), dirección IP de destino (*da*), puerto origen (*sp*), puerto destino (*dp*), protocolo (*pr*), banderas (*flg*), estado de reenvío (*fwd*), tipo de servicio (*stos*), paquetes intercambiados en el flujo (*pkt*), y su correspondiente número de bytes (*byt*).

Las direcciones IP de las distintas máquinas en el *dataset* se han anonimizado utilizando CryptoPan, que proporciona anonimización preservando los prefijos de las direcciones IP [11], implementada en la herramienta `nfanon` [12]. Esta herramienta se ha utilizado tradicionalmente para la anonimización de las trazas en los conjuntos de datos CAIDA. La correspondencia entre direcciones de IP anonimizadas en las distintas redes víctimas y atacantes se muestra en la Tabla IV.

III. ETIQUETADO Y ANÁLISIS DEL CONJUNTO DE DATOS

A continuación se analizan los datos recogidos en las dos capturas de datos, Calibración y Prueba, cuyo fin es aportar ideas y resultados de cara a su evaluación por IDS. Después, se describe el proceso de etiquetado, indicando los procedimientos seguidos y sus limitaciones. Finalmente, se evalúa el nivel de dificultad asociado al *dataset*, mediante la utilización de tres algoritmos del estado del arte en detección de intrusiones basados en anomalías.

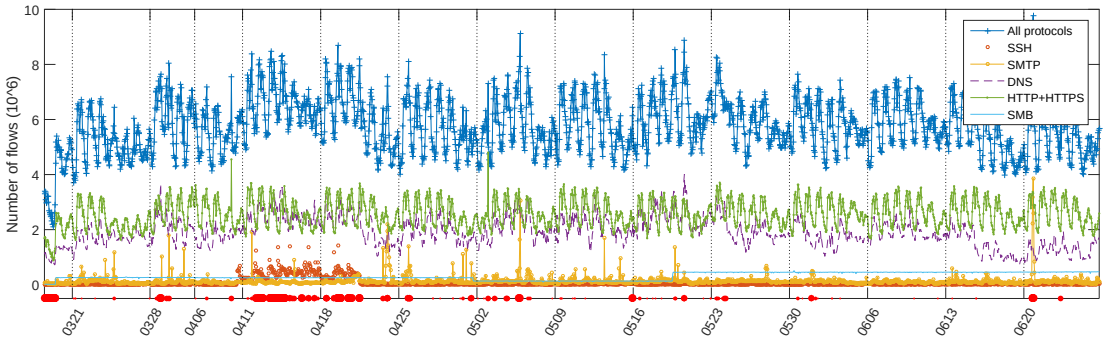
A. Cifras del dataset

Como se muestra en la Tabla III, el número de flujos total en el conjunto de datos es superior a 16.900 millones, y la duración de la traza es de más de 4 meses. Esto permite la evaluación de algoritmos de detección que hacen uso de la evolución ciclo-estacionaria del tráfico en diferentes patrones día/noche, así como fases días laborables/fin de semana. El número de IP externas observadas en la traza es mayor que 600 millones, correspondiente aproximadamente a 10 millones de sub-redes distintas.

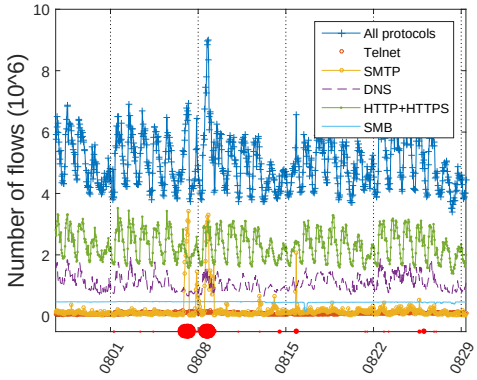
En la Fig. 2 se puede ver la evolución del número de flujos (en millones de flujos) para los conjuntos de calibración y de prueba. Las líneas punteadas verticales separan las distintas semanas (y por tanto archivos) en el conjunto de datos. Se señalan los protocolos con mayor cantidad de flujos. Los puntos rojos en el eje X muestran las anomalías encontradas por los tres detectores de anomalías utilizados. El tamaño del círculo está relacionado con el número de anomalías obtenidas en el intervalo (ver la discusión de este punto en la Sección III.A). A partir de estos resultados se pueden derivar algunas conclusiones interesantes:

- Se confirma que, tal y como se esperaba, predomina el protocolo HTTP/S y el tráfico DNS sigue un patrón coherente.
- Debido al hecho de que la mayoría de los clientes son compañías que alojan sus servidores en la red del ISP, el tráfico de BitTorrent o de otros servicios de P2P es casi residual.
- Nótese que hay un incremento de tráfico SSH en el intervalo 11/04/16 - 18/04/16. Se ha comprobado manualmente que este incremento de tráfico se debe a un ataque de escaneo SSH procedente de una única máquina alojada en el ISP. Las víctimas del ataque tienen un amplio rango de IP localizadas en un país de Sudamérica.
- Sorprendentemente, a pesar de su naturaleza insegura, se observa que muchas compañías todavía utilizan el servicio Telnet para gestionar sus equipos (ver Fig. 2(b)).
- Hay picos en el tráfico de SMTP dispersos a lo largo del tiempo en las dos capturas. Estos picos se refieren a veces a campañas de correo electrónico procedentes de compañías legítimas (bancos, servicios online, etc.), pero también se han encontrado campañas de *spam*. Por ejemplo, se identifica el pico en 20:39-06/08/16 - 05:59-07/08/16. Este se refiere a 12,5 millones de conexiones SMTP desde 5 IP públicas utilizando servidores de Yahoo. Un análisis minucioso de este tráfico lleva a concluir que sigue el patrón de una campaña de *spam*. Cada

[2]Se indica con paréntesis el nombre de las variables como viene dado por la herramienta `nfdump` para facilitar la identificación en el *dataset*.



(a) Conjunto de Calibración



(b) Conjunto de Prueba

Fig. 2. Evolución del número de flujos para (a) conjunto de Calibración y (b) conjunto de Prueba.

máquina generó alrededor de 2,5 millones de correos electrónicos. El equipo de IT del ISP confirmó que éste fue un cliente que alquiló máquinas virtuales durante 2 meses. Las IPs contratadas acabaron en las listas negras de Yahoo, y esta es probablemente la razón por la que el cliente dejó de alquilar las máquinas virtuales.

- Nótese el patrón constante para tráfico SMB. Este tráfico se corresponde con una sola empresa distribuidora con muchas tiendas. Cada establecimiento se comunica con un servidor central de forma periódica para comprobar el estado de algunos servicios o datos. Los picos en el tráfico de SMB se deben a la cancelación de la suscripción de ciertas tiendas a ese servicio.

B. Etiquetado

El etiquetado es un tema crítico en la producción de conjuntos de datos para la evaluación de IDS. Cuando se maneja tráfico real, es un reto decidir si un flujo dado corresponde a un ataque o no. Incluso cuando la naturaleza de un flujo es evaluada por un experto, hay ciertos casos en los que esto no está claro. Por esta razón, algunos autores deciden generar únicamente tráfico sintético, pues sólo en estos entornos se puede determinar de forma certera qué flujos se deben a un ataque y cuáles son legítimos. El problema con este enfoque es que el tráfico de *background* no es representativo del comportamiento de redes reales, por lo que se abre la posibilidad de que los algoritmos de IDS estén sesgados hacia la detección de falsos escenarios.

En nuestro caso, como se discutió anteriormente, se ha optado por utilizar tráfico de ataque artificial entrelazado con tráfico real de *background*. Este hecho supone un problema cuando se intenta etiquetar un conjunto de datos. El tráfico de *background* no está libre de ataques que podrían ser ejecutados por terceras partes durante la captura del *dataset* y, como ya se ha dicho, no está claro cómo identificarlos en algunos casos. Esto no ocurre cuando generamos ataques artificialmente. Incluso cuando se utilizan herramientas de generación de ataques reales y los escenarios de ataque son también reales, es posible etiquetar estos flujos, ya que se conocen las reglas para identificarlos (direcciones de IP y puertos, marcas de tiempo, etc.).

Algunos autores [13] que utilizan este mismo enfoque decidieron etiquetar flujos o paquetes utilizando tres etiquetas distintas: a) una etiqueta *attack* para los flujos que positivamente conocían que se correspondían con un ataque, b) una etiqueta *normal* para aquellos que se generaron de forma sintética con patrones normales, y c) una etiqueta *background* para aquellos que no sabían exactamente si eran ataques o no. Como en este caso no se está generando tráfico normal de forma sintética, finalmente se ha decidido etiquetar únicamente como *attack* aquellos flujos generados de forma artificial y dejar el resto etiquetados como *background*.

Tabla V
CORRESPONDENCIA ENTRE LAS ETIQUETAS Y ATAQUES EJECUTADOS.

Tipo de ataque	Etiqueta
DoS11	dos
DoS53s	dos
DoS53a	dos
Scan11	scan11
Scan44	scan44
Exf1KB	exf1KB
Exf1MB	exf1MB
Exf1MBp	exf1KB

Las etiquetas se aplican por flujo. Las conexiones son bidireccionales, por lo que una única conexión aparece como dos conexiones distintas unidireccionales. Las etiquetas de ataque se dividen en distintas sub-etiquetas, dependiendo del tipo de ataque que está siendo ejecutado. La Tabla V muestra la correspondencia entre las etiquetas utilizadas en el *dataset* y el tipo de ataque.

Nótese que no todos los ataques implementados (ver la Sección II.B) han sido etiquetados de forma individual. En su lugar, algunos de ellos han sido agrupados como uno solo, debido al hecho de que su patrón de tráfico es realmente el mismo. Tal es el caso de *DoS11*, *Dos53s* y *DoS53a*, todos ellos etiquetados como DoS. De manera similar, *Exf1KB* y *Exf1MBp* han sido etiquetados como *Exf1KB*, ya que realmente siguen el mismo patrón, esto es, los flujos *Exf1MBp* aparecen como muchas conexiones *Exf1KB* consecutivas.

IV. ANOMALÍAS EN EL CONJUNTO DE DATOS

A continuación se trata de ilustrar el aspecto de las 'anomalías' en el tráfico de *background* del *dataset* UGR'16. Nuestra única intención es motivar a otros investigadores a la identificación de estas anomalías utilizando sus propios métodos y algoritmos de detección. Para encontrar estas anomalías, se ha utilizado el detector de anomalías MSNM [14] y herramientas de análisis visual ad hoc con el propósito de describir, explorar y analizar los datos para descubrir el conocimiento subyacente de estos datos [15]. Adicionalmente, se ha contado con la ayuda del personal del ISP. Con fines ilustrativos nos centramos en la anomalía que tiene lugar en el intervalo de tiempo entre 04:10-01/08/16 hasta 04:14-01/08/16. Se observa un incremento de paquetes ACK y conexiones muy cortas que utilizan UDP. Inspeccionando los registros de Netflow para este periodo de tiempo se encuentra una única IP que crea 867.405 conexiones únicamente desde cuatro puertos origen (5061, 5062, 5066 y 5068) desde Alemania. Los destinos son 4.097 equipos distintos distribuidos en 16 sub-redes diferentes (máscara /24). Dependiendo del puerto origen de la conexión, cada *host* víctima se escanea en un rango específico de 60 puertos (por ejemplo, desde el puerto origen 5068 se escanearon los puertos 5000-5059). Debido a estos patrones de conexión, se concluye que parece tratarse de un ataque provocado por un *malware* orientado al escaneo de una vulnerabilidad específica.

A. Nivel de dificultad del dataset

Finalmente, estamos interesados en evaluar el nivel de dificultad del *dataset*. Recordemos que esta es una característica cualitativa que evalúa la probabilidad de que algoritmos de detección simple proporcionen muy buenos resultados de detección [8]. Un nivel de dificultad bajo no supondría un reto para el desarrollo de nuevos algoritmos que mejoren los existentes.

El *dataset* se ha evaluado utilizando tres detectores de anomalías del estado del arte. El primer par de detectores, MSNM_C y MSNM_S, se proponen en la metodología de monitorización de redes estadística multivariante basada en PCA dada en [14]. MSNM_C preprocesa los datos para centrarlos, mientras que MSNM_S los autoescala. Ambos esquemas muestran un rendimiento ligeramente distinto dependiendo de los tipos de ataque considerados [14]. Estos detectores han demostrado exhibir un comportamiento mejorado en varios escenarios distintos [16].

El tercer detector implementado es una máquina de vectores de soporte de una clase, OCSVM [17] [18]. OCSVM es un método de detección de anomalías de red basado en clasificación que se dice proporciona excelentes resultados según distintos estudios [19].

Para estos tres detectores se han obtenido las curvas ROC utilizando los datos de calibración obtenidos durante el mes de junio para la creación de los modelos de normalidad. Las ROC obtenidas se muestran en la Fig. 3. En la figura, se muestran dos tipos de conjuntos de curvas ROC distintas. Por una parte, las series MSNM_C, MSNM_S y OCSVM se corresponden con los resultados obtenidos cuando sólo se consideran ataques artificiales en la *ground truth*.

Los malos resultados relativos a las citadas ROC se deben principalmente al hecho de que no se han considerado todas las anomalías en el tráfico de *background*. Para resolver este problema, se ha identificado como anómalo todo el tráfico de *background* que activa una alarma en los tres detectores de anomalías al mismo tiempo. Para ello, primero se ha seleccionado el mejor punto de operación para cada detector de acuerdo con el criterio de Youven [20] (círculos rojos en la Fig. 3). Este criterio selecciona el umbral del detector de anomalías con la mayor distancia a la diagonal en la curva ROC.

Frente a esta experimentación, también se han añadido las anomalías detectadas por los tres detectores a la *ground truth* (observar estas anomalías como círculos rojos en el eje X de Fig. 2). Así se han vuelto a calcular las curvas ROC con esta nueva *ground truth*. Los resultados se muestran en la Fig. 3 como la serie $MSNM^*_C$, $MSNM^*_S$ and $OCSVM^*_C$. Se puede ver que incluso en el mejor de los casos, $MSNM^*_C$, los resultados son inferiores al 90% para la tasa de verdaderos positivos cuando el índice de falsos positivos está en torno al 10%, lo cual no es en absoluto excepcional. La principal razón es la dificultad para detectar ataques de exfiltración.

Aunque sería necesaria una evaluación más extensa de los algoritmos y sus resultados, éstos son suficientes para concluir que el nivel de dificultad del *dataset* resulta

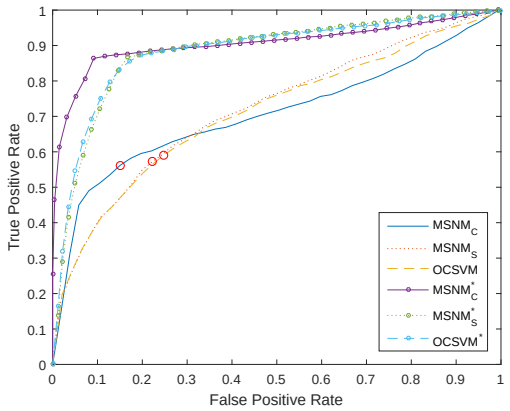


Fig. 3. Curvas ROC para los distintos detectores.

adecuado para la evaluación de nuevos algoritmos de detección.

V. CONCLUSIONES

La principal contribución de este trabajo es la generación de un nuevo *dataset* para la evaluación de algoritmos y sistemas IDS llamado UGR'16. El conjunto de datos se ha construido teniendo en cuenta lo aprendido sobre conjuntos de datos anteriores. UGR'16 es una colección de trazas *netflow* capturadas durante más de 4 meses de tráfico en una red real de un ISP Tier-3, junto con un conjunto de ataques de red de tipo real que se ha diseñado específicamente para entrenar y probar algoritmos IDS.

Las principales ventajas del *dataset* presentado frente a otros ya existentes se enumeran a continuación. Primero, el tráfico de *background* es muy representativo del tráfico de Internet, pues se captura de sensores de una red ISP donde se ubican perfiles muy diferentes de clientes. Esta es una diferencia principal con el resto de conjuntos de datos, en los cuales se recoge tráfico muy específico (como tráfico generado en una universidad o un centro de investigación). Segundo, el *dataset* tiene un nivel de dificultad que permite que nuevos algoritmos puedan ser comparados con otros ya existentes. Tercero, la duración del *dataset* lo hace adecuado para probar algoritmos que consideran la evolución ciclo-estacionaria del tráfico en día/noche y días laborables/fines de semana.

AGRADECIMIENTOS

Este trabajo ha sido parcialmente financiado por el Gobierno Español-MINECO (Ministerio de Economía y Competitividad) y fondos FEDER, a través del proyecto TIN2014-60346-R.

REFERENCIAS

[1] R. Di Pietro and L. V. Mancini, *Intrusion detection systems*. Springer Science & Business Media, 2008, vol. 38.

[2] P. García-Teodoro, J. Díaz-Verdejo, G. Maciá-Fernández, and E. Vázquez, "Anomaly-based network intrusion detection: Techniques, systems and challenges," *Computers & Security*, vol. 28, no. 1, pp. 18–28, 2009.

[3] DARPA'98 and DARPA'99 datasets. [Online]. Available: https://www.ll.mit.edu/ideval/docs/index.html

[4] A. Shiravi, H. Shiravi, M. Tavallaee, and A. A. Ghorbani, "Toward developing a systematic approach to generate benchmark datasets for intrusion detection," *Computers and Security*, vol. 31, no. 3, pp. 357–374, 2012.

[5] N. Moustafa and J. Slay, "UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," *2015 Military Communications and Information Systems Conference (MilCIS)*, pp. 1–6, 2015.

[6] A. Lemay and J. M. Fernandez, "Providing scada network data sets for intrusion detection research," in *9th Workshop on Cyber Security Experimentation and Test (CSET 16)*. USENIX Association, 2016.

[7] T. Morris and W. Gao, "Industrial control system traffic data sets for intrusion detection research," in *International Conference on Critical Infrastructure Protection*. Springer, 2014, pp. 65–78.

[8] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," *IEEE Symposium on Computational Intelligence for Security and Defense Applications, CISDA 2009*, no. Cisda, pp. 1–6, 2009.

[9] R. Fontugne, P. Borgnat, P. Abry, and K. Fukuda, "MAWILab: Combining Diverse Anomaly Detectors for Automated Anomaly Labeling and Performance Benchmarking," in *ACM CoNEXT '10*, Philadelphia, PA, December 2010.

[10] CAIDA. The cooperative association for internet data analysis. [Online]. Available: http://www.caida.org/

[11] J. Fan, J. Xu, M. H. Ammar, and S. B. Moon, "Prefix-preserving IP address anonymization: measurement-based security evaluation and a new cryptography-based scheme," *Computer Networks*, vol. 46, no. 2, pp. 253–272, 2004.

[12] P. Haag, "NFDUMP-NetFlow processing tools," *URL: http://nfdump.sourceforge.net*, 2011.

[13] S. Garcia, M. Grill, J. Stiborek, and A. Zunino, "An empirical comparison of botnet detection methods," *Computers & Security*, vol. 45, pp. 100 – 123, 2014.

[14] J. Camacho, A. Pérez-Villegas, P. García-Teodoro, and G. Maciá-Fernández, "PCA-based multivariate statistical network monitoring for anomaly detection," *Computers & Security*, vol. 59, pp. 118–137, 2016.

[15] Juan Alvarado-Pérez and Diego H. Peluffo-Ordóñez and Roberto Therón, "Bridging the gap between human knowledge and machine learning," *Advances in Distributed Computing and Artificial Intelligence Journal*, vol. 4, no. 1, 2015.

[16] G. Maciá-Fernández, J. Camacho, P. García-Teodoro, and R. A. Rodríguez-Gómez, "Hierarchical PCA-based multivariate statistical network monitoring for anomaly detection," in *Information Forensics and Security (WIFS), 2016 IEEE International Workshop on*. IEEE, 2016, pp. 1–6.

[17] B. Scholkopf, A. J. Smola, R. C. Williamson, and P. L. Bartlett, "New Support Vector Algorithms," *Neural Computation*, vol. 12, no. 5, pp. 1207–1245, 2000.

[18] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson, "Estimating the Support of a High-Dimensional Distribution," *Neural Computation*, vol. 13, no. 7, pp. 1443–1471, 2001.

[19] M. H. Bhuyan, D. K. Bhattacharyya, and J. K. Kalita, "Network Anomaly Detection: Methods, Systems and Tools," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 1, pp. 303–336, 2014.

[20] A. K. Akobeng, "Understanding diagnostic tests 3: Receiver operating characteristic curves." *Acta paediatrica*, vol. 96, no. 5, pp. 644–647, 2007.