TORPEDA: Una Especificación Abierta de Conjuntos de Datos para la Evaluación de Cortafuegos de Aplicaciones Web

Carmen Torrano-Gimenez
Instituto de Física Aplicada
Consejo Superior
de Investigaciones Científicas
Serrano 144 - 28006, Madrid, Spain
Email: carmen.torrano@iec.csic.es

Alejandro Perez-Villegas
Instituto de Física Aplicada
Consejo Superior
de Investigaciones Científicas
Serrano 144 - 28006, Madrid, Spain
Email: alejandro.perez@iec.csic.es

Gonzalo Alvarez
Instituto de Física Aplicada
Consejo Superior
de Investigaciones Científicas
Serrano 144 - 28006, Madrid, Spain
Email: gonzalo@iec.csic.es

***Resumen*—Los Cortafuegos de Aplicaciones Web (WAFs) han sido un importante objeto de estudio durante los últimos años. Se han propuesto numerosos sistemas que, utilizando una amplia variedad de técnicas, han intentado resolver el problema de la detección de ataques web. Uno de los problemas más comunes de estos trabajos es la dificultad de obtener los datos necesarios para evaluar la eficacia de la detección. Además, es habitual que estos datos no hagan públicos por cuestiones de confidencialidad u otros motivos, dificultando así la comparación entre sistemas diferentes. En el presente trabajo presentamos TORPEDA, una especificación abierta para la construcción de conjuntos de datos diseñada específicamente para la evaluación de los WAFs. En el conjunto de datos, que contiene exclusivamente tráfico HTTP, cada petición está etiquetada como normal, anómala o ataque, además de la categoría de ataque que contiene. Con TORPEDA pretendemos incorporar fundamentalmente dos aportaciones. En primer lugar, crear un conjunto de datos que sirva para la evaluación de cualquier sistema de detección de ataques web, ya sea basado en firmas, en anomalías o híbrido. En segundo lugar, proponer una especificación abierta con un formato común que permita la adición de nuevos conjuntos por parte de la comunidad. El objetivo es construir un *corpus* con numerosos conjuntos de datos específicos y generales que sirva como *benchmark* y facilite tanto la evaluación como la comparación de nuevas propuestas.**

I. Introducción

Las aplicaciones web constituyen hoy en día la manera más común de acceso a servicios y funcionalidades a través de la Red. Muchas aplicaciones tradicionales, como procesadores de textos, hojas de cálculo o clientes de correo, están transformándose en aplicaciones basadas en web debido a las enormes ventajas que incorpora, como la accesibilidad o la facilidad de mantenimiento. Las aplicaciones web han revolucionado nuestra forma de comprar, de visitar nuestro banco, de comunicarnos con los demas a través de redes sociales, de realizar transferencias monetarias y de operar con la Administración, entre otros muchos ejemplos. Asimismo, los ataques dirigidos a aplicaciones web han crecido en la misma medida, debido a la facilidad de acceso, el bajo coste requerido, el bajo nivel técnico que requieren muchos de estos ataques y sobre todo a la escasa conciencia de seguridad por parte de los desarrolladores.

Uno de los mecanismos de defensa más extendidos para defender los servidores web de los atacantes es el uso de Cortafuegos de Aplicación Web (*WAF - Web Application Firewall*). Estos sistemas operan en la capa de aplicación, analizando el tráfico HTTP para detectar acciones o comportamientos ilegítimos que puedan comprometer la seguridad de las aplicaciones web que protegen. El principal reto al que se enfrentan los WAFs es la distinción en tiempo real entre tráfico legítimo y tráfico ilegítimo. Este problema ha sido objeto de estudio por parte de la comunidad científica en los últimos años.

Los WAFs son un caso concreto de los Sistemas de Detección de Intrusiones (IDS), y como tales siguen dos enfoques principales. En primer lugar, los sistemas basados en firmas (*signature detection*), tratan de encontrar patrones de ataque o firmas en las peticiones analizadas. Estos sistemas siguen un enfoquen negativo, es decir, se centran en identificar el tráfico ilegítimo considerando el resto como normal. El enfoque basado en firmas permite alcanzar altas tasas de detección, pero presenta dificultades para detectar ataques desconocidos. Este enfoque es tradicionalmente el más utilizado por las herramientas comerciales, si bien también está presente en algunos trabajos científicos como [1] [2] y [3].

En segundo lugar, los sistemas basados en anomalías (*anomaly detection*), tratan de identificar desviaciones del comportamiento normal de la aplicación. La principal dificultad de estos sistemas radica precisamente en la definición del comportamiento normal de la aplicación. Estos sistemas siguen un enfoque positivo, es decir, definen e identifican el tráfico legítimo y consideran ilegítimo el resto. El enfoque basado en anomalías permite detectar ataques desconocidos, pero no suele alcanzar buenas tasas de detección y de falsos positivos. Aunque la presencia de este enfoque en herramientas comerciales es más moderada, existen numerosos trabajos de investigación al respecto, como [4] [5] [6] [7] y [8].

Uno de los problemas más comunes de los trabajos existen-

tes es la falta de datos de calidad y etiquetados que permitan una evaluación objetiva de los sistemas. Para dicha evaluación es necesario disponer de un conjunto de datos que contenga tanto peticiones normales como peticiones anómalas dirigidas contra una determinada aplicación web. La fiabilidad de las medidas de rendimiento del WAF (tasa de detección y tasa de falsos positivos) van a depender de la calidad del conjunto de datos elegido.

Por otra parte, los conjuntos de datos utilizados para la evaluación suelen ser heterogéneos y de distinta naturaleza, además de no estar disponibles para la comunidad. La utilización de distintos conjuntos de datos para la evaluación, así como el hecho de que estos datos no sean públicos dificultan en gran medida la comparación entre distintos sistemas de detección.

Existen varias alternativas a la hora de obtener los datos necesarios para la evaluación de un WAF.

- Captura de tráfico real. El uso de tráfico real es el más idóneo puesto que permite una evaluación más fiable y realista. Sin embargo presenta dos importantes inconvenientes. Por un lado, el tráfico considerado normal puede estar contaminado con ataques que circulen por la red. Por otro lado, la publicación de estos datos presenta problemas de privacidad.
- Generación artificial de tráfico. Esta alternatica resuelve los problemas de privacidad y de contaminación del tráfico. Sin embargo, este tráfico es menos realista y por tanto más predecible por el sistema de detección. Como consecuencia, la calidad de la evaluación disminuye.
- Uso de conjuntos de datos públicos. Esta alternativa ahorra el trabajo de construcción de los datos, además de facilitar la comparación de sistemas. Sin embargo, es muy difícil encontrar conjuntos de calidad y etiquetados adecuados a nuestro sistema.

Hasta donde conocemos, son muy pocos los conjuntos de datos públicos basados en tráfico HTTP. A continuación repasamos algunos de los existentes en la literatura.

El conjunto de datos DARPA [9] fue creado por el MIT en los años 1998 y 1999, y ha sido ampliamente utilizado en la evaluación de IDS, aunque también ha recibido duras críticas [10]. El conjunto DARPA fue diseñado para la evaluación de IDS, y sólo un pequeño porcentaje del tráfico es HTTP. Además, las tecnologías web han evoluciado enormemente en la última década y los datos aquí contenidos están muy desactualizados. Por estos motivos, hoy en día no es un conjunto apto para la evaluación de WAFs.

El conjunto de datos ECML/PKDD'07 [11] fue creado para el ECML/PKDD Discovery Challenge 2007. A diferencia del DARPA, este conjunto de datos contiene únicamente tráfico HTTP y está dividido en dos conjuntos, uno de entrenamiento y otro de prueba. El tráfico se presenta estructurado en formato XML, de tal manera que cada petición está etiquetada como normal o con una de las seis categorías de ataque definidas.

El conjunto ECML presenta sin embargo algunos inconvenientes. En primer lugar, se trata de un conjunto de datos diseñado para un concurso, y no con la idea de servir de *benchmark* de sistemas de detección. Aunque hemos tenido acceso a este conjunto de datos, su descarga está restringida bajo usuario y contraseña. En segundo lugar, por razones de privacidad, se han anonimizado los datos de tal manera que no se conoce la aplicación web a la que van dirigidas. Por este motivo, el conjunto no es apto para la evaluación de sistemas basados en anomalías, ya que el sistema no puede ser alimentado con tráfico normal. En [7] se discute este problema. En tercer lugar, el conjunto sólo contiene seis categorías de ataques, y no puede ser ampliado a otros tipos de ataque diferentes. Por último, cabe destacar que el ECML es el único conjunto de datos de ataques web del que tenemos conocimiento, lo cual pone de manifiesto la necesidad de construir otros conjuntos complementarios.

Aunque fue creado con el propósito de evaluar un WAF [6], en aras de la comparación cabe mencionar el conjunto de datos CSIC2010 [12]. Este conjunto de datos fue creado artificialmente por nuestro grupo y contiene peticiones web dirigidas a una aplicación de comercio electrónico.

La escasez de datos de calidad y etiquetados nos ha motivado a proponer TORPEDA (TORrano PErez DAtaset), una especificación abierta para la construcción de nuevos conjuntos de datos (*datasets*) de tráfico web. Esta especificación pretende establecer un formato común que permita el intercambio de conjuntos de datos entre investigadores y que facilite la comparación de diferentes sistemas de detección.

Para ilustrar la propuesta, se ha construído un conjunto de datos siguiendo la especificación de TORPEDA. Este conjunto contiene tráfico dirigido a una aplicación web vulnerable que se ha utilizado como banco de pruebas. En el conjunto se ha incluido una amplia variedad de ataques generados con ayuda de herramientas semiautomáticas.

Las contribuciones de este trabajo son las siguientes:

1. Hemos propuesto un formato general y estructurado para la construcción de *datasets* etiquetados de tráfico web, que sirva de *benchmark* para la evaluación de WAFs.
2. Hemos creado y publicado un conjunto de datos siguiendo esta especificación. Este conjunto contiene una amplia variedad de ataques y puede ser utilizado para la evaluación de sistemas basados en firmas o en anomalías.
3. Hemos mostrado una alternativa para publicar y compartir los conjuntos de datos utilizados por la comunidad científica, de tal manera que se puedan evaluar y comparar diferentes propuestas de manera transparente y con independencia de los datos.

El presente texto se organiza de la manera siguiente. En la sección II se describe la especificación TORPEDA, la estructura de los conjuntos de datos, los tipos de peticiones y el etiquetado. En la sección III se describe el *dataset* CSIC2012 y la metodología que hemos seguido para su construcción. Finalmente, en la sección IV se exponen las conclusiones de este trabajo y se discuten las tareas a realizar en el futuro inmediato.

II. Descripción General de TORPEDA

TORPEDA (TORrano-PErez DAtaset) es una especificación general para la construcción de nuevos *datasets* de tráfico web. Los conjuntos tienen un formato estructurado que permite extraer cada una de las partes de la petición HTTP, además de etiquetas que aportan la información necesaria para el evaluador. Las características principales de TORPEDA son las siguientes:

- Todos los *datasets* generados son públicos y están disponibles para la comunidad.
- Los conjuntos de datos contienen exclusivamente tráfico web, transmitido mediante el protocolo HTTP.
- Los conjuntos de datos pueden ser utilizados tanto para la evaluación de sistemas basados en firmas como para los basados en anomalías.
- Cada una de las peticiones HTTP están etiquetadas con el tipo de petición (normal, anómala o ataque) y con la categoría de ataque que contiene. La incorporación del tipo «anómalas» representa una novedad con respecto a los conjuntos arriba mencionados.
- El sistema es ampliable con otros conjuntos de datos, y está abierto a la aportación de nuevos conjuntos por parte de otros investigadores.

Estructura

Un conjunto de datos construidos bajo TORPEDA se compone de un número determinado de peticiones HTTP numeradas (*samples*). Cada petición tiene asignada un número identificador (*sample id*) de tal manera que pueda ser identificada entre el resto. De manera opcional, es posible (y recomendable) incluir información adicional como nombre del *dataset*, autor, URL del recurso donde se encuentra o cualquier otra información que se considere necesaria. En la Fig. 1 se muestra de manera esquemática la estructura de un *dataset* genérico.

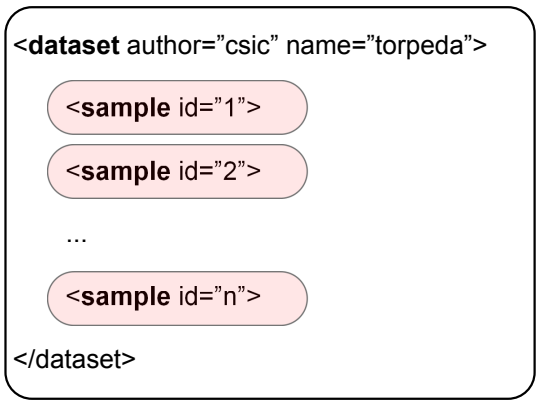

Figura 1. Estructura general de un dataset.

La información relativa a las peticiones se encuentra estructurada en dos partes.

- Información de la petición. En esta parte se incluye la información de la petición por sí misma, es decir, la información que puede obtenerse al capturar en la red el tráfico mediante un *sniffer*. Esta información se encuentra estructurada en los componentes existentes en el protocolo HTTP. A saber, método, protocolo, dirección URL, cabeceras, parámetros (en el caso de peticiones GET) y cuerpo (en el caso de peticiones POST).
- Etiquetado. En esta parte se incluye la información cualitativa de la petición, lo que podría aportar un experto sobre ella. Esta información es desconocida para un observador de la red, y es precisamente lo que un analizador automático debe predecir. En este apartado se incluye el tipo de petición y, en caso de constituir un ataque, el tipo de ataque presente en la petición.

En las Figuras 2 y 3 se pueden ver ejemplos genéricos de peticiones GET u POST incluidas en un *dataset*.

Para estructurar correctamente la información se ha elegido el formato XML, por ser un formato ampliamente extendido. Este formato, a diferencia de otros como PCAP, permite separar la información y etiquetar el tráfico de forma directa. La estructura elegida es similar a la utilizada en el ECML dataset [11] en lo que respecta a las peticiones, pero incorpora importantes diferencias en lo que respecta al etiquetado.

Tipos de peticiones

A diferencia de la visión tradicional dualista que distingue entre «tráfico malo» y «tráfico bueno», se han considerado tres tipos de peticiones.

- **Peticiones Normales**. Se trata de peticiones legítimas que representan el tráfico esperado por la aplicación web bajo un funcionamiento normal. Estas peticiones acceden a recursos válidos de la aplicación, contienen argumentos válidos, valores dentro de los rangos esperados y, por tanto, no representan ninguna amenaza. En un entorno real, la gran mayoría del tráfico generado en la red correspondería a esta categoría. Por ejemplo:

  `GET /comprar.jsp?cantidad=5`
- **Peticiones Anómalas**. Se trata de peticiones a priori legítimas pero que contienen datos no esperados por la aplicación, es decir, que se alejan del comportamiento normal. Aunque no representan ninguna amenaza por sí solas, pueden ser utilizadas por un atacante en la fase previa a un ataque. También puede tratarse de errores de usuarios legítimos al introducir los datos. El tráfico generado por las herramientas de *fuzzing* entraría dentro de esta categoría. Por ejemplo:

  `GET /comprar.jsp?cantidad=-10`
- **Ataques**. Se trata de peticiones que contienen alguna forma de ataque y que pretende explotar una vulnerabilidad de la aplicación (existente o no). Estas peticiones son potencialmente peligrosas para la aplicación web, y pueden haber sido generadas por un usuario o por un agente automático. Aquí entrarían todas las categorías de ataques consideradas. Por ejemplo:

  `GET /comprar.jsp?cantidad=1'SHUTDOWN--`

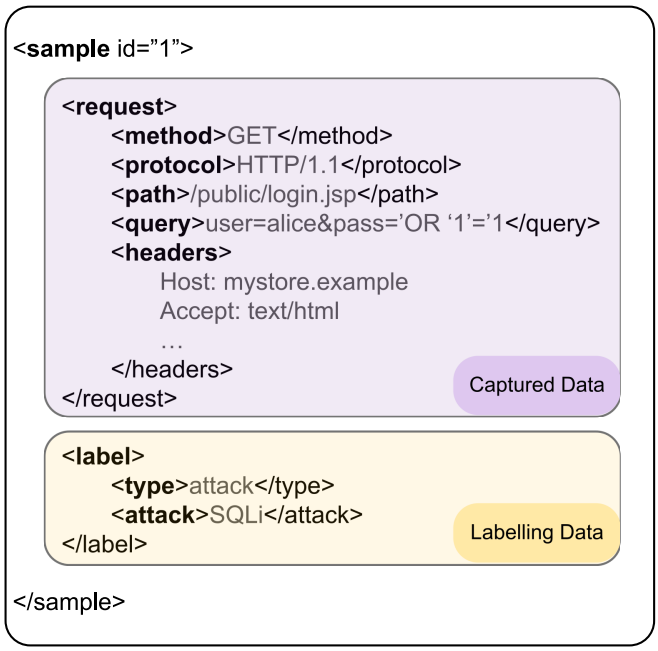

Figura 2. Ejemplo de petición GET que contiene un ataque.

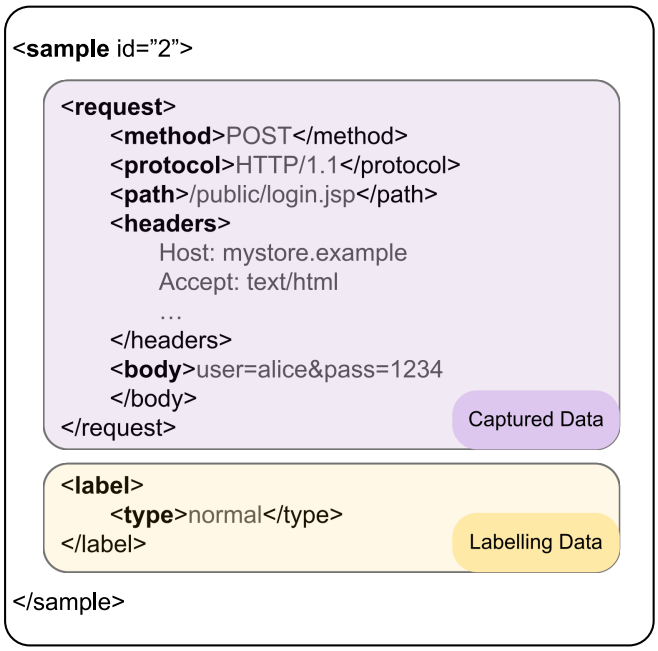

Figura 3. Ejemplo de petición POST legítima.

Tipos de ataques

Una de las características de TORPEDA es que los *datasets* pueden ser utilizados tanto para evaluar sistemas basados en anomalías como basados en firmas. Algunos enfoques basados en firmas son en realidad clasificadores que además de identificar la presencia de un ataque determinan el tipo del que se trata. La evaluación de la eficacia de estos clasificadores sólo es posible si se dispone de tráfico malicioso etiquetado por categorías de ataque. En general, a nuestro juicio el hecho de incluir información del tipo de ataque contenido en la petición mejora considerablemente la calidad del conjunto de datos.

La elaboración de una lista consistente de categorías de ataques no es un problema trivial. Aunque se han llevado a cabo trabajos sobre taxonomía web [13] o de listas de las vulnerabilidades más explotadas [14], no tenemos conocimiento de la existencia de una taxonomía ampliamente aceptada que englobe cualquier ataque. Por este motivo, se ha optado por una lista de ataques lo más amplia posible, pero siempre abierta a la adición o modificación de nuevas categorías. En la sección III se describen las categorías elegidas para el *dataset* construído.

Generación de tráfico y etiquetado

El método de generación de tráfico es uno de los puntos claves en la construcción del conjunto de datos. Si se opta por el uso de tráfico real, es importante que no contenga datos confidenciales. Una solución a este problema es la utilización de una aplicación web de pruebas, cuyos usuarios pueden ser colaboradores que generan tráfico con datos ficticios. La generación de tráfico artificial es menos costosa, pero disminuye la calidad del conjunto de datos. En el conjunto de datos generado hemos optado por un enfoque intermedio, utilizando una aplicación de pruebas y herramientas semiautomáticas.

El etiquetado de las peticiones es una cuestión clave, ya que va a determinar directamente la calidad del conjunto de datos. Un *dataset* mal etiquetado implica una evaluación errónea del sistema de detección, y por tanto tiene escasa utilidad. El etiquetado automático sólo puede conseguirse si se genera el tráfico en un entorno muy controlado, como es la simulación o la generación artificial. En caso de utilizar tráfico real, probablemente es necesaria la supervisión de uno o varios expertos de forma manual, lo cual es muy costoso. Sin embargo, este coste puede estar justificado si se construye un conjunto de datos suficientemente extenso y general.

Para facilitar la construcción de conjuntos de datos, se ha desarrollado un *script* que permite transformar capturas en formato «pcap» a documentos XML con el formato de TORPEDA. La herramienta permite definir las etiquetas (peticiones y ataques) para cada captura realizada. La correcta captura y etiquetado del tráfico es responsabilidad del autor del *dataset*. El *script* está escrito en Python y puede descargarse en [15].

III. Construcción de un dataset

Para ilustrar la especificación TORPEDA hemos construído un conjunto de datos de ejemplo, al que hemos llamado CSIC2012. Este conjunto de datos contiene tráfico artificial generado manualmente y con la ayuda de herramientas semiautomáticas.

Todas las peticiones incluídas en CSIC2012 están dirigidas a una aplicación web de comercio electrónico, desarollada en nuestro departamento para este propósito. La aplicación consiste en una tienda online en la que los usuarios pueden realizar acciones tales como añadir productos a su carrito de la compra o registrarse proporcionando sus datos personales.

Siguiendo la especificación descrita en el apartado anterior, el conjunto de datos contiene tres tipos de peticiones debidamente etiquetadas: peticiones normales, peticiones anómalas y ataques. En total, el conjunto de datos está formado por aproximadamente 8.500 peticiones normales, 15.000 peticiones anómalas y 55.000 ataques, aunque se puede ampliar con más muestras. Obviamente, la proporción de ataques presente en el conjunto no se corresponde con la de un entorno real. Téngase en cuenta que el objetivo del presente trabajo es la construcción de un *corpus* de datos, y no de la elección de las muestras usadas en la evaluación. Es responsabilidad del usuario del conjunto de datos la elección de muestras adecuadas a cada caso particular.

El conjunto de datos CSIC2012 incluye una gran variedad de ataques web, entre los que se incluyen muchos ataques de reciente creación que afectan a las aplicaciones web modernas:

- Inyección SQL, con variantes como inyección ciega, SQL *fingerprinting*, entre otras.
- *Cross-Site Scripting*, con variantes como inyección en las *cookies*, en las cabeceras, etc.
- Desbordamiento de *buffer*.
- Ataque de formateo (*Format String Attack*).
- Inyección LDAP.
- Inyección de comandos del SO.
- Partición de peticiones y respuestas (*HTTP Splitting*).
- *Local File Include*.
- *Server Side Include*.
- Inyección XPath.
- Inyección de retorno de carro (*CRLF Injection*).
- Adivinación de directorios (*Directory Browsing*).
- Falsificación de parámetros.

Nótese que el conjunto de datos incluye tanto ataques estáticos (aquellos que tratan de solicitar recursos ocultos o no existentes) como ataques dinámicos (los cuales modifican los argumentos válidos de la petición).

En el Cuadro I, se muestra una comparación de los ataques incluidos en los conjuntos de datos ECML/PKDD'07, HTTP CSIC2010 y CSIC2012. En él puede verse que en CSIC2012 se incluyen todos los ataques considerados en los otros dos conjuntos de datos y varios tipos de ataque más.

Cuadro I
TIPOS DE ATAQUE INCLUÍDOS EN CONJUNTOS DE DATOS ETIQUETADOS CONOCIDOS.

Ataque \ Dataset	ECML07	CSIC2010	CSIC2012
SQL injection	X	X	X
XSS	X	X	X
Buffer Overflow		X	X
Format String			X
LDAP injection	X		X
OSCommanding	X		X
Response Splitting			X
Local File Include			X
Server Side Include	X	X	X
XPath Injection	X		X
CRLF injection			X
Directory Browsing		X	X
Parameter Tampering		X	X

Metodología de generación de tráfico

La metodología seguida para la generación del tráfico ha sido distinta para cada tipo de peticiones.

- Tráfico normal. Las peticiones normales se han generado empleando tanto medios automáticos como manuales. Los medios manuales hacen referencia a la navegación, por parte de varios usuarios, por las diferentes páginas de la aplicación web. Estos usuarios hacen siempre un uso normal de la aplicación y no realizan ningún tipo de acción que pueda comprometer la seguridad de la aplicación web. Además del proceso manual, se ha utilizado la herramienta WebScarab [16] para ayudar en la generación de determinadas peticiones normales que utilizan parámetros. Los valores normales para cada uno de los parámetros (nombres, apellidos, direcciones, etc.) proceden de diccionarios genéricos, evitando así problemas de privacidad.
- Tráfico anómalo. En este caso también se ha generado el tráfico por medio de una combinación de medios automáticos y manuales. Se ha navegado por la aplicación manualmente introduciendo datos con formato erróneo, como ejemplo direcciones de correo mal construídas, DNIs sin números, caracteres especiales, etc. En general, se han generado datos que incumplen deliberadamente las reglas de la aplicación, modificando en ocasiones los valores normales de los parámetros. Alternativamente se ha usado nuevamente la herramienta WebScarab para automatizar el proceso.
- Ataques. Para la generación de los ataques se ha hecho uso principalmente de varias herramientas semiautomáticas: w3af [17], ZAProxy [18], sqlmap [19], XSSer [20] y Nikto [21].
  Algunos de los tipos de ataque y sus variantes se han generado con varias herramientas para tener una mayor variabilidad y por tanto, un conjunto de datos más completo. Todos los ataques incluídos en el conjunto involucran a una sóla petición, y por tanto no se han considerado sesiones de navegación. Quedarían excluídos ataques de fuerza bruta, *fuzzing*, y otros que sólo puedan ser detectados al observar una secuencia de peticiones. Hemos procurado que haya ataques y anomalías que afecten a todas las partes de la petición web (en el *path*, la *query*, las *cookies*, en otras cabeceras y en el cuerpo).

Todo el tráfico generado ha sido capturado con la ayuda de la herramienta TcpDump. Como resultado, se han generado diversas capturas en formato «pcap», para los diferentes tipos de peticiones y tipos de ataques. Esta separación ha facilitado enormemente el etiquetado de forma automática. Para transformar las capturas en documentos XML con el formato de TORPEDA se ha desarrollado un *script*, que además permite el etiquetado. En el caso del tráfico generado manualmente, el etiquetado ha de ser también manual.

Tanto los documentos XML del conjunto CSIC2012 como el *script* pueden ser descargados de la página web de TORPEDA [15].

IV. Conclusiones y Trabajo Futuro

La escasez de conjuntos de datos etiquetados se debe principalmente a la imposibilidad de etiquetar correctamente tráfico real de manera automática ya que, si esto fuera posible, el problema de la detección de ataques web ya estaría resuelto. Las únicas alternativas son la generación de tráfico artificial ya etiquetado, o bien el etiquetado de tráfico real por parte de un grupo de expertos, con el elevado coste que ello supone. Nuestra primera conclusión es que este alto coste puede merecer la pena si da como resultado un conjunto de datos de calidad que pueda ser tomado como referencia y pueda ser utilizado por un buen número de investigadores para evaluar y comparar sus propuestas.

La segunda conclusión que hemos sacado es que para extender el uso de conjuntos de datos de referencia es necesario disponer de una estructura común, que incluya una forma general de etiquetado de tráfico y que sirva para evaluar cualquier tipo de sistema. En el presente trabajo hemos propuesto una especificación que atiende a esta necesidad, y que puede servir como primer paso para la construcción de conjuntos de datos de referencia.

En el futuro próximo, vamos a trabajar en la ampliación de TORPEDA principalmente en dos aspectos. Por un lado, tenemos intención de construir una web de aprendizaje de seguridad en un entorno controlado que recoja los ataques realizados por usuarios colaboradores, lo cual aportaría una gran variabilidad a los ataques. Por otro lado, vamos a construir un marco general que permita evaluar y comparar automáticamente la eficacia de los WAFs utilizando uno o varios conjuntos de datos de referencia.

Agradecimientos

Este trabajo se ha realizado gracias al proyecto «Modelos de propagación de malware a través de redes sociales online» (TIN2011-29709-C0201) del Ministerio de Economía y Competitividad.

Referencias

[1] Mathieu Exbrayat. Analyzing Web Traffic: A Boundaries Approach. *Proceedings of the ECML/PKDD 2007 Discovery Challenge2*, pages 53–64, 2007.

[2] C Pachopoulos, D Valsamou, D Mavroeidis, and M Vazirgiannis. Feature Extraction from Web Traffic Data for the Application of Data Mining Algorithms in Attack Identification. *Proceedings of the ECML/PKDD 2007 Discovery Challenge2*, pages 65–70, 2007.

[3] B Gallagher and T Eliassi-Rad. Classification of HTTP Attacks: A Study on the ECML/PKDD 2007 Discovery Challenge. In *Center for Advanced Signal and Image Sciences (CASIS) Workshop*. Lawrence Livermore National Laboratory, 2008.

[4] C Kruegel, G Vigna, and W Robertson. A multi-model approach to the detection of web-based attacks. *Computer Networks*, 48(5):717–738, 2005.

[5] J Estevez-Tapiador, P Garcia-Teodoro, and J Diaz-Verdejo. Measuring normality in HTTP traffic for anomaly-based intrusion detection. *Computer Networks*, 45(2):175–193, 2004.

[6] C Torrano, A Perez-Villegas, and G Alvarez. An Anomaly-Based Approach for Intrusion Detection in Web Traffic. *Direct*, 5:446–454, 2010.

[7] M Hosseinkhani, E Tarameshloo, and B Sadeghiyan. A Two Dimensional Approach for Detecting Input Validation Attacks Based on HMM. In *IEEE 11th International Conference on Data Mining*, Vancouver, 2011.

[8] I Corona, D Ariu, and G Giacinto. HMM-Web: A Framework for the Detection of Attacks Against Web Applications. *2009 IEEE International Conference on Communications*, pages 1–6, 2009.

[9] R Lippmann, J Haines, D Fried, J Korba, and K Das. DARPA off-line intrusion detection evaluation. *Computer Networks*, 34(4):579–595, 1999.

[10] John Mchugh. Testing Intrusion Detection Systems : A Critique of the 1998 and 1999 DARPA Intrusion Detection System Evaluations as Performed by Lincoln Laboratory. *System*, 3(4):262–294, 2001.

[11] C Raïssi, J Brissaud, G Dray, P Poncelet, M Roche, and M Teisseire. Web Analyzing Traffic Challenge: Description and Results. *Proceedings of the ECML/PKDD 2007 Discovery Challenge*, pages 47–52, 2007.

[12] A Perez and C Torrano. HTTP Dataset CSIC 2010, 2010. http://iec.csic.es/dataset/.

[13] G Álvarez and S Petrovic. A new taxonomy of web attacks suitable for efficient encoding. *Computers Security*, 22(5):435–449, 2003.

[14] The Open Web Application Security Project Owasp. OWASP Top Ten Project, 2010. http://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project.

[15] C Torrano and A Perez. TORPEDA Dataset, 2012. http://iec.csic.es/torpeda/.

[16] OWASP. Web Scarab. https://www.owasp.org/index.php/Category:OWASP_WebScarab_Project.

[17] Andrés Riancho. Web Appliaction Attack and Audit Framework (w3af). http://w3af.sourceforge.net/.

[18] OWASP. Zed Attack Proxy (ZAP). http://code.google.com/p/zaproxy/.

[19] B Damele and M Stampar. Sqlmap. http://sqlmap.sourceforge.net/.

[20] Lord_Epsilon. XSSer. http://xsser.sourceforge.net/.

[21] David Lodge. Nikto. http://cirt.net/nikto2.