Detección de ataques a servidores web mediante detectores de intrusos gratuitos: estudio experimental

Javier Muñoz [1], Felipe Bueno [1], Rafael Estepa [1], Antonio Estepa [1], Jesús E. Díaz-Verdejo [2]
[1] Dpt. Ingeniería Telemática, Escuela Superior de Ingenieros, Univ. de Sevilla
fmjc@us.es, felipebuenocarranza@gmail.com, rafaestepa@us.es, aestepa@us.es
[2] Dpt. Teoría de Señal, Telemática y Comunicaciones, CITIC, Univ. de Granada,
jedv@ugr.es

Resumen- **Este trabajo pretende cuantificar de forma experimental el nivel de detección de ataques de algunos de los detectores de intrusiones basados en firmas (SIDS) disponibles de forma gratuita. Para su evaluación se ha utilizado un *dataset* de ataques generado a partir de la búsqueda y selección de herramientas de generación de ataques y de análisis de seguridad del servicio web. Se consideran dos escenarios de diferente complejidad en cuanto al servicio web proporcionado. Las peticiones HTTP registradas durante los ataques serán la entrada a tres SIDS gratuitos seleccionados por su amplio uso, de forma que se podrá determinar la capacidad de detección de los mismos. Este trabajo se encuentra aún en desarrollo, por lo que en esta contribución se muestran los primeros resultados relativos a la recolección y selección de herramientas para la generación de los ataques, la generación del dataset de ataques de forma que sea representativo de los ataques actuales y la evaluación preliminar de las capacidades de detección.**

***Index Terms*- sistemas de detección de intrusos basados en firmas, web application filters, ataques web**

Tipo de contribución: *Investigación en desarrollo*

I. INTRODUCCIÓN

Los servicios web han experimentado una gran expansión en los últimos años, tanto por el desarrollo de servicios cada vez más complejos y avanzados como por la fácil accesibilidad de los mismos mediante el uso de navegadores. Consecuentemente, los servidores web se han convertido en una de las dianas favoritas de los ciberataques. Entre otros escenarios, los servidores comprometidos se utilizan habitualmente para la distribución de malware, para realizar *phishing* o como puerta de acceso a la red de una empresa. El uso de conexiones HTTP cifradas ha provocado que los sistemas de detección de intrusiones basados en firmas (SIDS) [1] habitualmente desplegados se desplacen al propio servidor web, bien integrándose como parte del WAF (*Web Application Firewall*) o bien operando sobre los archivos de traza de los servidores. En este trabajo se pretende verificar el grado de detección de ataques basados en web que ofrecen algunos de los SIDS o WAF gratuitos más difundidos en la actualidad. En particular, en primera aproximación consideraremos tres de ellos: *Snort* [2], *ModSecurity* [3] y *Nemesida* [4] (con reglas gratuitas). Para ello, previamente, será necesario establecer *datasets* de ataques contra los que validar su capacidad de detección que sean representativos y se encuentren actualizados.

En este sentido, existen pocos datasets públicos disponibles para hacer experimentación sobre SIDS en general (e.g., KDD'99 [5], CAIDA [6], UNSW-NB15 [7]) y aún menos que sean específicos para HTTP (e.g. CICIDS2017 [8]). Los ataques que contienen aparecen en número reducido, son antiguos y mayoritariamente de sistemas simulados [9]. Esta carencia de cercanía a la realidad y la falta de representatividad los hacen poco aplicables a entornos de producción [10] y, consecuentemente, los invalidan para desarrollar y/o evaluar sistemas de detección de ataques válidos en escenarios reales modernos [11], especialmente porque los servicios web y los ataques evolucionan rápidamente con el tiempo. Si bien es posible encontrar más de 80 bases de datos públicas [12] con peticiones reales a servidores, un análisis de las mismas nos muestra que, o bien no están marcadas las peticiones de ataque, o bien se limitan a actividad recopilada mediante *honeypots*, lo que no garantiza la representatividad ni la presencia de todos los tipos de ataques. Para evitar estas carencias se ha generado un nuevo *dataset* con tráfico de ataques web (teniendo en cuenta sólo aquellos basados en el contenido de las URI) a partir de varias herramientas de ataque actualizadas seleccionadas al efecto. Se han considerado dos escenarios (una web estática y una web dinámica) y se ha seguido una metodología que permite su repetición y ampliación.

Este estudio se encuentra actualmente en elaboración y en este artículo presentaremos sus resultados preliminares. Las principales contribuciones son: a) elaboración de una lista actualizada de herramientas susceptibles de ser utilizadas para la realización de ataques web, con sus principales características, b) generación de un *dataset* con ataques, y c) la evaluación inicial de la capacidad de detección de distintos SIDS gratuitos sobre el *dataset* anterior. Estas contribuciones permitirán la selección de las herramientas SIDS más apropiadas y la ejecución de nuevas pruebas que utilicen el *dataset* proporcionado con distintos SIDS o con nuevas firmas. Por otra parte, el dataset podría también utilizarse para evaluar sistemas basados en detección de anomalías.

II. HERRAMIENTAS DE GENERACIÓN DE ATAQUES WEB

Para que el *dataset* de ataques sea representativo de los diversos tipos de ataques que pueden llevarse a cabo se han buscado las herramientas de ataques web disponibles en la actualidad. Para ello se han empleado referencias encontradas en tres fuentes de información básicas: a) OWASP: *Dynamic Application Security Testing* (DAST) [13], b) Mitre: técnicas *Exploit Public-Facing Application* [14] y Software [15] y, por último, c) Listado de Software del proyecto *Nmap* [16], que cuenta con una de las listas más actualizadas sobre herramientas de ciberseguridad. Se ha buscado en las categorías: *Web Vulnerabilitiy scanners* y *Vulnerability Exploitation tools.* Las distintas herramientas encontradas en las fuentes anteriores han sido revisadas de forma individual, encontrándose 22 de tipo *opensource* y 23 con licencia, para las que se ha solicitado al desarrollador una licencia de prueba gratuita que ha sido concedida en 7 casos, con limitaciones únicamente en tiempo de uso. Como resultado, han sido probadas e instaladas en el sistema operativo correspondiente las herramientas que se detallan en la Tabla I. En esta se muestra, para cada una, la licencia de uso, el sistema operativo, el tipo (SIDS genérico o específico de web), su funcionalidad (columna *Func*: sólo sondeo -S- o incluye explotación -E-) y los tipos de ataques que permite realizar (columna *Ataques*, donde los 10 primeros corresponden con la clasificación *OWASP Top10*).

Con independencia del sistema operativo empleado, es posible encontrar dos grandes categorías de herramientas: aquellas específicas de web, que permiten mayor granularidad en la especificación del ataque y suelen seguir los tipos de ataques web especificados en la clasificación de OWASP, y otras genéricas, orientadas a *pentesting,* que ejecutan en serie una lista de ataques ya preconfigurados y que permiten un menor control al usuario. Todas las herramientas tienen un funcionamiento similar, siendo preciso especificar la URL del sistema objetivo a atacar y, en su caso, el tipo de ataque correctamente parametrizado.

III. GENERACIÓN DEL *DATASET* DE ATAQUES WEB

A fin de que el *dataset* reproduzca la mayor diversidad de ataques posibles con las herramientas actuales, se han generado todos los tipos de ataques posibles para cada una de las 28 herramientas descritas en la Tabla I en dos escenarios web: una aplicación estática (*Apache* con un recurso html) y una aplicación dinámica (gestor de contenidos *Wordpress* con la instalación por defecto).

Para implementarlo, se ha desplegado un escenario con dos máquinas virtuales: una para el atacante y otra para el servidor web correspondiente. En la máquina del atacante se ha instalado el software de ataque actualizado a fecha de 10 abril de 2022, junto con el sistema operativo correspondiente. Desde dicha máquina se han lanzado todos los ataques posibles (ver Tabla I) contra los dos escenarios web previstos, capturado el tráfico recibido por el servidor *Apache* utilizando *tcpdump*. Una vez guardado el tráfico de ataque en formato *pcap,* se han extraído las peticiones web en formato

Tabla I
HERRAMIENTAS PARA ATAQUES WEB UTILIZADAS

ID	Nombre	Licencia	Sistema Operativo [1]	Tipo [2]	Func [3]	Ataques [4]											
						1	2	3	4	5	6	7	8	9	10	11	12
1	Havij	OpenSource	W	E	E												X
2	Wpscan	Comercial	L, M	E	S												X
3	Nuclei	OpenSource	W, M, L	E	E			X							X		
4	Sqlmap	OpenSource	W, M, L	E	E											X	X
5	OWASP-ZAP	OpenSource	W, M, L	E	S		X	X	X	X							
6	Grabber	OpenSource	W, M, L	E	S			X								X	
7	Openvas	OpenSource	L	G	S											X	X
8	Arachni	OpenSource	W, M, L	G	E			X									
9	Ironwasp	OpenSource	W, M, L	E	E	X	X	X								X	
10	W3af	OpenSource	L, M	E	S		X	X	X	X	X	X	X	X	X	X	
11	Nexpose	Comercial	W, L	E	S											X	X
12	SmartScanner	Comercial	W	E	S												X
13	Nessus	OpenSource	W	E	S												X
14	Golismero	OpenSource	W, M, L	E	S												X
15	Burpsuite	Comercial	W, M, L	E	S												X
16	Metasploit	OpenSource	W, M, L	G	E												X
17	Nikto	OpenSource	L	E	E	X	X	X									
18	Wapiti	OpenSource	W, M, L	E	S		X	X							X	X	
19	Grendel-Scan	OpenSource	W, M, L	E	S			X	X	X	X					X	
20	Webcruiser	Comercial	W	E	S											X	X
21	Nmap	OpenSource	W, M, L	G	E			X							X		
22	Nexploit	Comercial	SaaS	E	S											X	X
23	Xsser	OpenSource	W, M, L	E	E												X
24	Vega	OpenSource	W, M, L	E	S		X	X									
25	Skipfish	OpenSource	W, M, L	E	S											X	X
26	Watobo	OpenSource	W, M, L	E	S												X
27	Commix	OpenSource	W, M, L	E	E												X
28	Deepfence Threatmapper	OpenSource	L	G	S												X

[1] Sistema Operativo: W=Windows, M=MacOS, L=Linux, SaaS=Software as a service
[2] Tipo: G: Genérico, E: Específico web
[3] Función: S: sólo sondeo, E: incluye además explotación
[4] Ataques: 1: *Broken Access Control*, 2: *Cryptografic Failures*, 3: *Injection*, 4: *Insecure Design*, 5: *Security Misconfiguration*, 6: *Vulnerable and Outdated Components*, 7: *Identification and Authentication Failures*, 8: *Software and Data Integrity Failures*, 9: *Security Logging and Monitoring Failures*, 10: SSRF, 11: Otros, 12: Conjunto predefinido.

texto mediante la aplicación *tshark,* incluyendo la URI presente en las mismas. Adicionalmente, se han generado ficheros de resumen del tráfico en formato IPFIX con las aplicaciones: *ipt-netflow* y *nfcapd*. De esta forma, para cada instancia de ataque realizada por cada herramienta en los dos escenarios de aplicación, se han guardado los ficheros: *.pcap*, *.ipfix*, *.csv* y *.uri*. Estos ficheros conforman el cuerpo del *dataset* generado, que está disponible para la comunidad en https://github.com/fbuenoc97/TFG/tree/main/capturas. Los ficheros se encuentran organizados por escenario (web_dinámica, web_estática) y, dentro de cada escenario, podemos encontrar un directorio por cada herramienta que contiene los ficheros del *dataset* cuyo nombre se corresponde con el tipo de ataque realizado.

A modo de resumen, el *dataset* consta de un total de 148 ataques distintos realizados con las 28 herramientas especificadas en la Tabla 1 sobre cada escenario Web. Para el escenario estático se han generado ficheros de tráfico que ocupan 404 MB y contienen 317.433 peticiones web (URIs de sondeo o de explotación), mientras que para el escenario dinámico el tráfico almacenado asciende a 1.428 MB y el número de peticiones HTTP recopiladas es de 391.006.

Cabe señalar que, tras la captura del tráfico, se realiza una validación de la captura para evaluar si se ajusta al tráfico esperado, reajustando los parámetros del ataque en los casos en los que se consideró necesario. Esta inspección permitió verificar algunas peculiaridades, como que la mayoría de las herramientas realizan siempre una fase de escaneo previa al ataque en sí. Se observaron algunos casos muy residuales de comportamientos no esperados (con distintas tipologías de ataque, la herramienta lanzaba el mismo ataque –seguramente por un error de programación–) que fueron eliminados del *dataset*. La herramienta *wpscan* (ID=2) no se ha ejecutado sobre el escenario estático, ya que carece de sentido, y, como incidencias significativas, podemos señalar que los resultados de dos herramientas han sido eliminados del *dataset*: (1) *Deepfence Threatmapper* (ID=28), pues la versión usada (*free*) no permite generar tráfico HTTP de análisis de vulnerabilidades y las pocas URI existentes son intentos simples de acceso a "/web/"; y (2) en la herramienta *Commix* (ID=27) todas las URI existentes son "/web".

El *dataset* generado por las 26 herramientas restantes ha sido utilizado para verificar la capacidad de detección de distintos SIDS, como se detallará en el siguiente apartado.

IV. Capacidad de detección de SIDS gratuitos

Las peticiones web que contiene las URI de ataque incluidas en el *dataset* anterior han sido procesadas mediante diversos sistemas de detección. En este trabajo nos centraremos en tres sistemas SIDS ampliamente extendidos y de uso gratuito:

a) Snort [2], software IDS genérico de amplia implantación. Utilizaremos las reglas de Talos [17] así como las reglas ETopen [18] actualizadas a 24/03/2022 con todas las reglas activas. Previamente, se han seleccionado las reglas que afectan únicamente al URI de peticiones HTTP, siguiendo el procedimiento indicado en [19].

b) ModSecurity [3]: módulo WAF también de amplio uso por su fácil integración con *Apache*. Utilizaremos las reglas OWASP *Core Ruleset* (CRS) en su versión 3.3.2 y el nivel de paranoia 2.

c) Nemesida: es un WAF completo que incluye un conjunto de firmas de uso gratuito. Dicho conjunto proporcionaría una mayor tasa de falsos positivos que la versión de pago, lo que no afecta a este estudio.

El fichero de traza resultante en cada SIDS ha sido analizado a fin de correlar cuáles de las peticiones HTTP de entrada han sido detectadas como ataque por los distintos SIDS. Los resultados se muestran en la Tabla II, que indica, por cada herramienta de ataque, el número de peticiones (N.

Tabla II
Capacidad de detección de los S-IDS utilizados

			Web estática				Web dinámica			
ID	Herramienta	N. At.	N. URI	Snort	ModSec	Nemesida	N. URI	Snort	ModSec	Nemesida
			294	14%	99%	84%	138	95%	98%	100%
2	Wpscan	1	-	-	-	-	166	2%	67%	69%
3	Nuclei	14	8362	21%	49%	51%	2076	22%	52%	52%
4	Sqlmap	1	98	38%	86%	64%	995	29%	46%	18%
5	OWASP-ZAP	8	2109	0%	0%	0%	9061	7%	50%	35%
6	Grabber	5	2108	17%	95%	60%	22358	3%	46%	35%
7	Openvas	1	10492	6%	19%	33%	50081	13%	27%	36%
8	Arachni	10	3385	0%	0%	0%	17044	4%	38%	33%
9	Ironwasp	13	12301	0%	7%	7%	582	2%	37%	22%
10	W3af	26	7896	2%	0%	0%	8879	5%	11%	43%
11	Nexpose	1	5344	13%	20%	24%	5344	13%	20%	24%
12	SmartScanner	1	1187	0%	12%	19%	2280	4%	20%	25%
13	Nessus	1	49472	3%	27%	15%	43508	5%	23%	19%
14	Golismero	1	8890	6%	44%	26%	290	3%	18%	24%
15	Burpsuite	1	254	6%	9%	12%	2274	9%	15%	14%
16	Metasploit	1	48039	0%	14%	22%	48039	0%	14%	22%
17	Nikto	12	4306	1%	10%	33%	2841	0%	16%	17%
18	Wapiti	14	21783	0%	1%	4%	44963	2%	11%	10%
19	Grendel-Scan	8	17835	4%	3%	14%	19249	4%	3%	14%
20	Webcruiser	1	1548	0%	0%	0%	4827	0%	10%	9%
21	Nmap	9	2531	2%	4%	5%	2946	2%	4%	11%
22	Nexploit	1	7335	1%	7%	3%	17063	2%	9%	6%
23	Xsser	1	24	0%	0%	0%	61	0%	3%	11%
24	Vega	13	23342	1%	1%	1%	53276	1%	5%	5%
25	Skipfish	1	74763	1%	5%	8%	31813	0%	3%	8%
26	Watobo	1	3734	0%	0%	0%	852	0%	0%	0%

URI) maliciosas generadas por cada herramienta y el porcentaje de las mismas que han sido detectadas por cada SIDS. Cabe señalar que muchas de las peticiones enviadas forman parte de una fase de escaneo. Esta puede ser considerada una etapa temprana del ataque, según la taxonomía de Mitre ATT&CK [20], por lo que resulta de interés verificar si los SIDS son capaces de detectar este tipo de peticiones. Para el escenario web estático la tasa media de detección de *Snort* es del 2,2%, subiendo al 13% y 14% para *ModSecurity* y *Nemesida,* respectivamente. En el caso del escenario dinámico, los valores son similares, quedando en torno al 20% para *ModSecurity* y *Nemesida*. Podemos inferir que la eficiencia en la detección para los SIDS específicos de web (*ModSecurity* y *Nemesida*) es superior a la del SIDS genérico (*Snort*). Las tasas de detección para las herramientas que incluyen explotación (ver Tabla I), ofrecen resultados muy similares, incrementándose tan sólo un 2% adicional en el escenario dinámico, lo que resulta lógico, dado que estas herramientas también tienen fase de escaneo. Estos resultados sugieren que los SIDS están ajustados para un nivel de detección bajo en la fase de escaneo a fin de reducir la tasa de falsos positivos. Sería necesario, en cualquier caso, una revisión más a fondo de los resultados. Finalmente, en la Figura 1 se muestra, para cada escenario, la tasa media de detección obtenida por los 3 SIDS para los ataques generados por cada una de las herramientas.

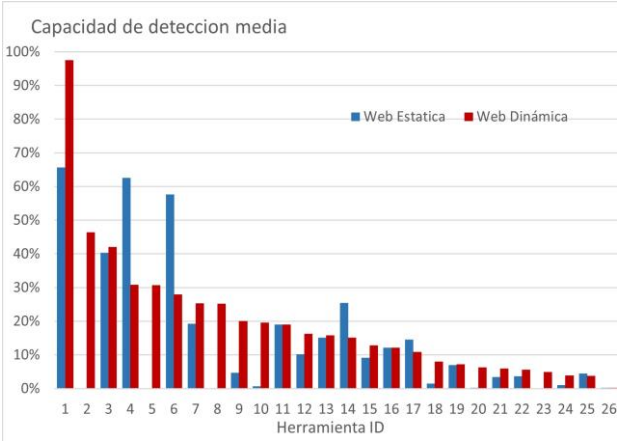

Fig. 1. Capacidad media de detección en ambos escenarios.

V. Conclusiones y líneas de avance

La disponibilidad de *datasets* adecuados es clave para acelerar la investigación en el campo de los IDS. En este trabajo se ha generado un *dataset* propio de ataques que se ha puesto a disposición de la comunidad investigadora. Este *dataset* se ha generado incorporando todos los posibles tipos de ataques que ofrecen las principales herramientas gratuitas disponibles contra dos escenarios web: uno estático y otro dinámico.

La capacidad de detección de los ataques web incluidos en el *dataset* por parte de los tres SIDS oscila entre un 5% y 20% de los ataques. Este porcentaje no implica necesariamente un bajo rendimiento de los sistemas de detección, y podría responder a que la fase de escaneo de un ataque no es suficientemente detectada. Estos resultados son preliminares y el trabajo continúa en curso.

En las siguientes fases del mismo se abordarán retos como: análisis de la eficiencia en la detección por tipo de ataque y por herramienta. También se pretende segregar en los resultados la capacidad de detección de la fase de escaneo. Por último, los resultados sobre la capacidad de detección de las distintas herramientas serán complementados con la tasa de falsos positivos que genera cada uno, lo que permitirá la estimación del rendimiento de manera fiable.

Como limitaciones de este trabajo pueden destacarse el uso de herramientas gratuitas y la limitación a dos escenarios de aplicación web sobre las que atacar, lo que restaría capacidad de generalización a los resultados. Aunque los dos escenarios utilizados son de amplia implantación, el uso de otros portales y aplicaciones/servicios web enriquecerían la aplicabilidad de los resultados. Por último, es importante reseñar que es posible que no todas las peticiones HTTP correspondan realmente a ataques, por lo que es necesario supervisar el dataset. Estos aspectos deberán ser tratados en posibles ampliaciones al trabajo.

Agradecimientos

Esta publicación es parte de los proyectos de I+D+i PID2020-115199RB-I00 financiado por MICIN/AEI/ 10.13039/501100011033 y PYC20-RE-087-USE financiado por FEDER/Junta de Andalucía - Consejería de Transformación Económica, Industria, Conocimiento.

Referencias

[1] H. Hindy, D. Brosset, E. Bayne, A. Seeam, C. Tachtatzis, R. Atkinson, X. Bellekens, *A taxonomy and survey of intrusion detection system design techniques, network threats and datasets*, arXiv:1806.03517, 2018.

[2] Snort, *Snort: an open source network intrusion prevention and detection system*, Sourcefire, disponible en http://www.snort.org.

[3] *Modsecurity Open Source Web Application Firewall*. Disponible en https://github.com/SpiderLabs/ModSecurity.

[4] *Nemesida Web Application Firewall.* Disponible en: https://nemesida-waf.com.

[5] S. Hettich, S.D. Bay, *The UCI KDD Archive.* Univ. of California, Dep. of Information & Computer Science, http://kdd.ics.uci.edu, 1999.

[6] *Cooperative Association for Internet Data Analysis* (CAIDA) datasets, 2008.

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

[8] P. Ranjit,, S. Borah, *A detailed analysis of CICIDS2017 dataset for designing Intrusion Detection Systems*, International Journal of Engineering & Technology 7.3.24:479-482, 2018.

[9] Riera, T.S., Higuera, J.R.B., Higuera, J.B., Herraiz, J.J.M., Montalvo, J.A.S; *Prevention and fighting against web attacks through anomaly detection technology. A systematic review*, Sustainability 12:1-45, 2020.

[10] Sharafaldin I., Gharib A., Lashkari A.H., Ghorbani A.A., *Towards a reliable intrusion detection benchmark dataset*, Softw. Netw., 1:177-200, 2018.

[11] R. Sommer, V. Paxson; *Outside the closed world: On using machine learning for network intrusion detection*, Proc. IEEE Symp. Secur. Privacy, 305–316, 2010.

[12] https://datasetsearch.research.google.com/

[13] https://owasp.org/www-community/Vulnerability_Scanning_Tools

[14] https://attack.mitre.org/techniques/T1190/

[15] https://attack.mitre.org/software/S0390/

[16] https://sectools.org/

[17] https://www.snort.org/talos

[18] https://doc.emergingthreats.net/

[19] J.E. Díaz-Verdejo, A. Estepa, R. Estepa et al. / Future Generation Computer Systems 109:67–82, 2020.

[20] Strom, Blake E., et al., *Finding cyber threats with ATT&CK-based analytics*, The MITRE Corporation, Technical Report No. MTR170202 (2017).