Opuntia Brava
ISSN: 2222-081X
RNPS: 2074
Volumen: 13 Número: 1
Recepción: 12/12/20. Aprobado: 11/01/21

Estrategia cognitiva para impedir ataques Ddos en servidores web

Cognitive strategy to prevent Ddos attacks on web servers

Maidel Arsenio de la Rosa Téllez [1] (maidel@ult.edu.cu) (https://orcid.org/0000-0003-3731-6028)

Mirelis Alcina Reyes [2] (mirelis@ult.edu.cu) (https://orcid.org/0000-0001-6089-1704)

Ariel Céspedez Pérez [3] (arces@ult.edu.cu) (https://orcid.org/0000-0002-9091-2462)

Resumen

La seguridad de las redes informáticas se ha convertido en uno de los temas fundamentales de todas las entidades, ya que de esto depende que la información este disponible, segura y confiable. Durante el desarrollo de esta investigación con el uso de herramientas se pudo constatar la existencia de la vulnerabilidad ante ataques tipo basado en Denegación de Servicio en el servidor Web con el gestor de contenidos Wordpress, una vez corregidas las configuraciones a nivel de Firewall y aplicaciones, se pudo comprobar que dicha vulnerabilidad había sido solucionada, lo que permite el funcionamiento del servicio bajo este tipo de ataques.

Palabras claves: DDoS, Sistema de Detección de Intrusos, Sistemas de Prevención de Intrusos, Ataques informáticos.

Abstract

The security of computer networks has become one of the fundamental issues of all entities, since it depends on this that the information is available, safe and reliable. During the development of this research with the use of tools it was possible to verify the existence of the vulnerability to attacks based on Denial of Service in the Web server with the Wordpress content manager, once corrected the configurations at Firewall and application level, it was possible to verify that this vulnerability had been solved, allowing the operation of the service under this type of attacks.

Key words: DDoS, Intrusion Detection System, Intrusion Prevention Systems, Computer Attacks.

[1] Máster en Tecnologías para la Educación. Profesor Auxiliar de la Facultad de Ciencias Técnicas y Agrícolas. Universidad de Las Tunas, Cuba.
[2] Máster en Tecnologías para la Educación. Profesor Auxiliar de la Facultad de Ciencias Técnicas y Agrícolas. Universidad de Las Tunas, Cuba.
[3] [3] Máster en Informática Aplicada. Profesor Auxiliar de la Facultad de Ciencias Técnicas y Agrícolas. Universidad de Las Tunas, Cuba.

Opuntia Brava
ISSN: 2222-081X
RNPS: 2074
Volumen: 13 Número: 1
Recepción: 12/12/20. Aprobado: 11/01/21

Maidel Arsenio de la Rosa Téllez
Mirelis Alcina Reyes
Ariel Céspedez Pérez

Seguridad en los servidores Web

Hoy en día prácticamente todas las empresas, centros educacionales o entidades públicas cuentan con sitios Web para ofrecer información acerca de sus propias entidades, tales como: servicios que se brindan, estructura organizacional de la entidad, personal de apoyo, entre otras. Lo cual se hace posible mediante un servidor Web que es un programa o aplicación para atender o responder las solicitudes provenientes de los navegadores, al suministrar los recursos que se le soliciten mediante los protocolos HTTP o HTTPS.

Sin embargo, hay un número creciente de agentes maliciosos que buscan brechas de seguridad en los servidores Web, lo que precisa que cada vez estén más protegidos. Los ataques dirigen su atención al aprovechamiento de las fallas de las aplicaciones Web. En su artículo Jia, Huang, Liu y Ma (2017) explican el funcionamiento de los tipos de ataques denegación de servicio (DoS) y los de distribuidos de denegación de servicios (DDoS). Estos ataques son ejecutados mediante la generación de una gran cantidad de peticiones ficticias al servidor con el objetivo de inhabilitar el uso de un sistema, lo que afecta la disponibilidad del servicio.

Varios autores han evaluado temas relacionados sobre ataques informáticos en servidores y algunas técnicas para mitigar su impacto. Entre ellos, Cheng y otros (2018) profundizan en los sistema de defensa DDoS para servicios Web en un entorno de nube. En cambio, autores como Jaafar, Abdullah y Ismail (2019) realizan la revisión de 12 detecciones de ataques DDoS en la capa de aplicación publicados entre enero de 2014 y diciembre de 2018.

Por su parte, Huang y otros (2019), en su estudio presenta un nuevo método para detectar ataques DDoS (Distributed Denial of Service) de Shrew y analiza las características de esos ataques periódicos para ser adecuado para protocolo TCP en el servidor; mientras que Han, Yang, Sun, Huang y Su (2018) desarrollan un mecanismo colaborativo de detección de ataques DDoS, que consiste en un algoritmo de monitoreo de flujo de grano grueso en el plano de datos y un algoritmo de clasificación de ataques basado en aprendizaje automático detallado en el plano de control. Por último, De Donno, Dragoni, Giaretta y Spognardi (2018) proponen una taxonomía actualizada y completa de los ataques DDoS, junto con una serie de ejemplos sobre cómo esta clasificación se relaciona con los ataques del mundo real; luego, describe la situación actual de los malwares habilitados para DDoS en las redes de IoT, y destaca cómo los datos recientes respaldan las preocupaciones sobre la creciente popularidad de estos malwares.

En las revisiones bibliográficas realizadas se relacionan disímiles eventos ocurridos en el pasado, lo que nos llevó a realizar el presente estudio para evaluar el comportamiento de los servidores Web alojados en el Ministerio de Educación Superior, Ciencia, Tecnología e Innovación en cuanto a estabilidad y disponibilidad, cuando están sometidos a posibles ataques con herramientas usadas por los atacantes, así como

Opuntia Brava
ISSN: 2222-081X
RNPS: 2074
Volumen: 13 Número: 1
Recepción: 12/12/20. Aprobado: 11/01/21

Maidel Arsenio de la Rosa Téllez
Mirelis Alcina Reyes
Ariel Céspedez Pérez

evidenciar las vulnerabilidades que presentan. Esto permitió trabajar en las medidas de contingencia y lograr la estabilidad deseada en los servidores.

Principales softwares para servidores WEB

Un servidor Web, también conocido como servidor HTTP o HTTPS no es más que un programa o aplicación informática que procesa del lado del servidor, crea conexiones que son enviadas y/o recibidas y sincronizadas o no con el usuario, otorga como respuesta los contenidos solicitados.

Existen varios servicios o aplicaciones que pueden ser usados como servidores Web entre los cuales se encuentran (Valarezo, Honores, Gómez y Vinces, 2018, pp. 28-30):

- Apache: Es el más popular en el mercado del Hosting Web (alojamiento Web), se considera como el más utilizado en el mundo ya que es libre y corre sobre Windows, Mac OS y Linux.
- Ngnix: Altamente recomendado por ser un servidor Web muy ligero, puede ser instalado sobre sistemas Unix y Windows, se distribuye bajo licencia BSD.
- Lighttp: Se encuentra diseñado para ser rápido, seguro, flexible. Está optimizado para rendimientos en velocidad, por lo que su consumo en CPU y memoria RAM es inferior a la de otros servidores.
- Microsoft IIS: Su funcionamiento es solo para sistema Windows. Se pueden ejecutar aplicaciones Web de ASP.NET, ASP clásico y PHP. No se recomienda su uso para correr aplicaciones nativas de otros sistemas operativos como PHP, Python, Perl o Ruby.
- Sun Java System Web Server: Este producto fue diseñado para correr aplicaciones javas, pero al coexistir con otras aplicaciones como Apache y Nginx, los cuales son multiplataforma Sun, decidió que su distribución se realice bajo licencias de código abierto (BSD concretamente).

La comunicación entre los servidores Web mencionados anteriormente y el cliente, que no es más que los navegadores Web como: Internet Explorer, Mozilla FireFox o Google Crome, se realiza mediante el protocolo de comunicación Hyper Text Transport Protocol (HTTP) y Hypertext Transfer Protocol Secure (HTTPS). Se trata de un protocolo del tipo petición-respuesta creado con el objetivo de definir y estandarizar las comunicaciones que se llevan a cabo entre los diferentes equipos que forman parte de una red.

Las brechas de seguridad son aprovechadas por los hackers mediante el uso de diferentes técnicas de ataques. Entre las más conocidas se encuentran las de denegación de servicio, que su objetivo principal es dejar fuera de funcionamiento los servicios y sistemas por periodos de tiempo que pueden ser indefinidos. Las acciones no están orientadas a recuperar o alterar informaciones, sino a impedir el acceso a las informaciones brindadas por las entidades.

Opuntia Brava
ISSN: 2222-081X
RNPS: 2074
Volumen: 13 Número: 1
Recepción: 12/12/20. Aprobado: 11/01/21

Maidel Arsenio de la Rosa Téllez
Mirelis Alcina Reyes
Ariel Céspedez Pérez

Estos tipos de ataques se dividen en dos partes:

- Las denegaciones de servicio por saturación: el atacante satura al equipo objetivo para que no sea capaz de responder las solicitudes reales realizadas por clientes que necesiten informaciones.
- Las denegaciones de servicio por explotación de vulnerabilidades: se aprovechan las vulnerabilidades existentes en los sistemas para volverlos inestables.

Para la realización de los ataques por denegación de servicios, se envían una gran cantidad de paquetes IP o datos de tamaños o formatos atípicos, logran la saturación de los equipos víctimas y los vuelven inestables. Cuando el atacante usa varios equipos para la denegación de servicio, el proceso se conoce como "sistema distribuido de denegación de servicio" (DDOS, Distributed Denial of Service), entre los que se encuentran los siguientes (Hernández y Mejia, 2015, p. 15):

- UDP Flood (Saturación UDP). Este tipo de ataque envía numerosos paquetes UDP a puertos aleatorios, lo que provoca que la víctima compruebe las peticiones realizadas a cada puerto.
- ICMP Flood (Saturación por Ping). Este tipo de ataque satura la víctima con solicitudes de paquetes “eco” ICMP (más conocido como ping), consiste en enviar paquetes de solicitud sin esperar respuesta, logran consumir tanto ancho de banda saliente y entrante como sea posible.
- Service Port Flood (Ataque sobre Puertos de Servicio). En este tipo de ataques se realizan peticiones tanto entrantes como salientes y dirigidas hacia los puertos estándar que pueden estar corriendo algún servicio (el puerto TCP 80, por ejemplo), es considerado de los más complejos para evitarlos o detenerlos.
- HTTP Flood (Saturación HTTP). Consiste en la realización de numerosas peticiones GET o POST en apariencia válidas para atacar servidores o aplicaciones Web.
- Fuerza Bruta. Se realiza al enviar a la víctima innumerables secuencias de ataques, con el objetivo de obtener usuario y contraseña de algún usuario y así acceder al sistema (smtp, ssh, http, entre otros).
- Inundación SYN (SYN Flood). Funciona mediante el envío de una serie de peticiones, donde la mayor parte con direcciones de origen falsificadas junto a la petición real, provoca que el servidor conteste todas las peticiones.

Para el desarrollo de la presente investigación se siguieron los procedimientos y las herramientas usadas por (Guijarro, Cevallos y Torres, 2018):

- Realización de escaneo de puertos al servidor WEB.

- Exposición de las posibles vulnerabilidades encontradas.
- Realización de test para explotar las vulnerabilidades.
- Incremento de la seguridad del servidor Web.
- Presentación de los resultados.

Realización de escaneo de puertos al servidor WEB

Para conocer los puertos abiertos y las diferentes vulnerabilidades que puedan tener, se realizó un escaneo de puertos al servidor Web, en el cual se encontraron algunas brechas que pueden ser usadas por personas maliciosas y realizar ataques externos. Para ello, se localizó y descargó la herramienta Zenmap, desde su sitio oficial https://nmap.org/dist/nmap-7.70-setup.exe, la sintaxis usada es nmap –T4, el cual es un temporizador, -A para realizar un ack scan y –v para obtener un informe detallado del escaneo. Con esta combinación de atributos se logra hacer un escaneo intenso o profundo al servidor, como se observa en la Figura 1, el servidor tiene abierto 3 puertos y se pueden ver, las versiones de los servicios que se ejecutan en ellos.

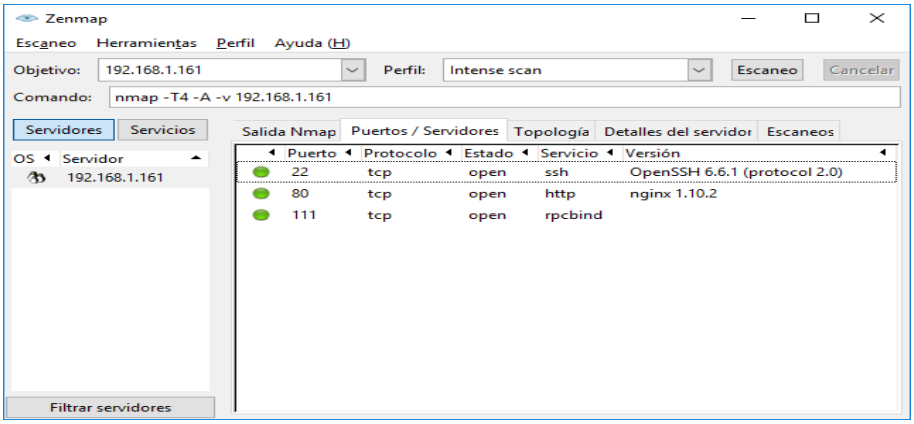

Figura 1: Realización del scaner de puerto al servidor WEB.

Como se puede observar en la Figura 1 muestra como resultado que el servidor posee el puerto ssh abierto y además muestra la versión de la aplicación que lo ejecuta (OpenSSH 6.6.1 protocolo 2.0), el puerto y la versión que está ejecutando el servidor http o servidor Web (nginx 1.10.2) además del puerto rpcbind.

Exposición de las posibles vulnerabilidades encontradas

Con los resultados anteriores se pueden detallar las posibles vulnerabilidades que sufre el servidor Web. Paralelo a ello como prevención se expone la posible solución a cada uno de los problemas detectados y así mitigar los resultados del posible atacante como se muestra en la tabla 1.

Opuntia Brava
ISSN: 2222-081X
RNPS: 2074
Volumen: 13 Número: 1
Recepción: 12/12/20. Aprobado: 11/01/21

Maidel Arsenio de la Rosa Téllez
Mirelis Alcina Reyes
Ariel Céspedez Pérez

Problemas detectados	Posible solución
Acceso al protocolo al servidor mediante protocolo SSH.	Ajustar las reglas del Firewall o Corta Fuego a solo las direcciones autorizadas
El servidor SSH muestra la versión en uso.	Ajustar la configuración del servidor SSH para que no muestre la versión
Se muestra la versión del servidor nginx 1.10.2	Ajustar la configuración del servidor Web para que no muestre la versión en uso.
Se muestra abierto el puerto 111 (rpcbind)	Ajustar las reglas del Firewall para cerrar el acceso a puertos innecesarios

Tabla 1. Problemas detectados y Posibles solución

Se pueden implementar otras medidas de prevención a nivel de Firewall en los router y switch para minimizar el tráfico de ips no autorizadas, así como el tráfico de tipo de ataques como: TCP/SYN Flood, TCP/DYN Flood y UDP Flood.

Realización de test para explotar las vulnerabilidades

Para comprobar si realmente existe la vulnerabilidad en el servidor Web se utilizó la herramienta goldeneye que es una herramienta en lenguaje Python. Se utilizó mediante la simulación de 20 clientes y 100 conexiones por cada cliente usando los métodos POST y GET aleatoriamente. Esta herramienta se descargó e instaló en Kali Linux diseñada principalmente para la auditoría y seguridad informática. Este sistema se configuró con una IP la cual tendría acceso al servidor Web para realizar las diferentes peticiones al mismo como se muestra en la figura 2.

```
root@kali:~/GoldenEye# ./goldeneye.py http://192.168.1.61
-s100 -w 20 -m random

GoldenEye v2.1 by Jan Seidl <jseidl@wroot.org>

Hitting webserver in mode 'random' with 20 workers running
100 connections each. Hit CTRL+C to cancel.
```

Figura 2: Uso de la herramienta goldeneye

Al iniciarse el ataque al servidor, la herramienta antes mencionada realiza 100 conexiones por cada cliente o trabajador, lo que provoca una sobrecarga en el servidor y comienza a afectar considerablemente los tiempos de respuesta del servidor hasta que dejz de responder a las peticiones realizadas por clientes reales.

Maidel Arsenio de la Rosa Téllez
Mirelis Alcina Reyes
Ariel Céspedez Pérez

Opuntia Brava
ISSN: 2222-081X
RNPS: 2074
Volumen: 13 Número: 1
Recepción: 12/12/20. Aprobado: 11/01/21

Para comprobar si el ataque tuvo éxito abrimos otra terminal y mediante el comando ping, que mediante protocolo ICMP, utilizado para medir la latencia o tiempo que tardan en comunicarse con el servidor, comprueba la disponibilidad del servidor como se muestra en la figura 3.

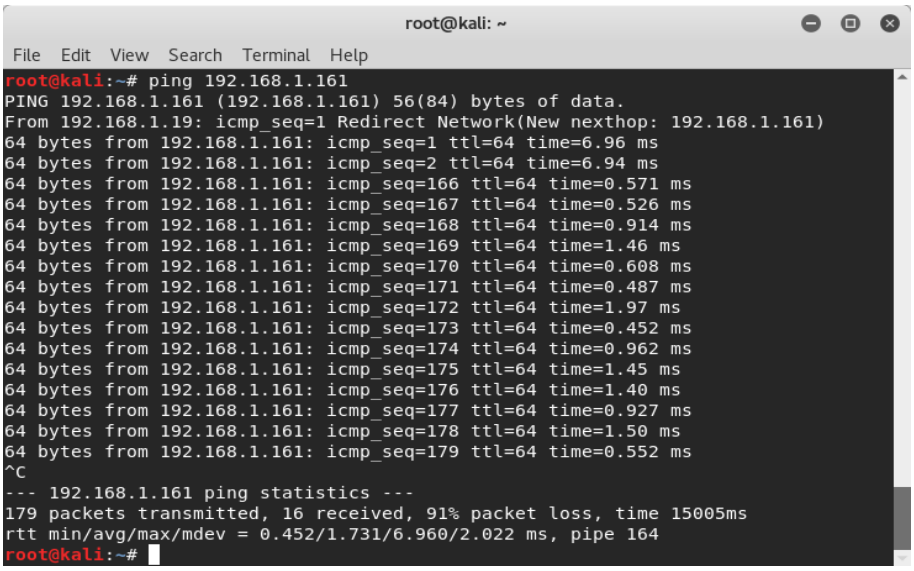

Figura 3: Uso del comando ping

Como se muestra en la figura anterior de 179 paquetes trasmitidos solo se logró respuesta de 16 de ellos y 163 paquetes perdidos, lo que representa un 91% del total de paquetes perdidos durante el ataque. Se consultó además la disponibilidad del sitio Web mediante el cliente Firefox como se muestra en la figura 4.

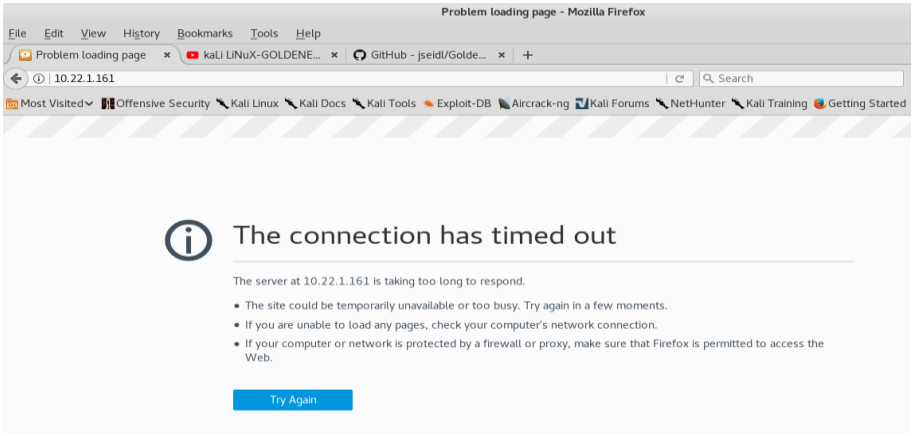

Figura 4: Caída del servidor Web.

Como se puede apreciar el servidor Web deja de estar en funcionamiento por el tiempo que tarda en dar respuesta a conexiones reales.

Opuntia Brava
ISSN: 2222-081X
RNPS: 2074
Volumen: 13 Número: 1
Recepción: 12/12/20. Aprobado: 11/01/21

Maidel Arsenio de la Rosa Téllez
Mirelis Alcina Reyes
Ariel Céspedez Pérez

Incremento de la seguridad del servidor

Inicialmente se configuró el servicio de administración remota ssh para que ocultara la versión, adicionando en su archivo de configuración ubicado en /etc/ssh/sshd_config

DebianBanner no

Para incrementar la seguridad del servidor Web se realizó en dos vertientes: la primera, al reforzar las reglas de filtrado del cortafuego o Firewall en este caso IPTABLES, y la segunda al colocar módulos de contención en la aplicación que se desempeña como servidor Web.

En la configuración del IPTABLES se adicionaron las reglas que aparecen en la figura 5 para proteger el servidor de varios tipos de ataques.

```
-A INPUT -p icmp -m limit --limit 1/sec --limit-burst 2 -j ACCEPT
-A INPUT -p icmp -m limit --limit 1/sec --limit-burst 2 -j LOG --log-prefix "PING-DROP:"
-A INPUT -p icmp -j DROP
-A INPUT -p icmp -f -j DROPLOG
-A INPUT -p icmp -m state --state ESTABLISHED -m limit --limit 3/sec --limit-burst 8 -j ACCEPT
-A INPUT -p icmp -m state --state RELATED -m limit --limit 3/sec --limit-burst 8 -j RELATED_ICMP
-A INPUT -p icmp -m icmp --icmp-type 8 -m limit --limit 3/sec --limit-burst 8 -j ACCEPT
-A INPUT -p icmp -j DROPLOG
-A INPUT -i lo -j ACCEPT
-A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
-A INPUT -p tcp -m multiport --dports 135,137,138,139,445,1433,1434 -j DROP
-A INPUT -p udp -m multiport --dports 135,137,138,139,445,1433,1434 -j DROP
-A INPUT -m state --state INVALID -j DROP
-A INPUT -p tcp -m state --state NEW -m tcp --tcp-flags FIN,SYN,RST,PSH,ACK,URG FIN,SYN,RST,PSH,ACK,URG -j DROP
-A INPUT -p tcp -m state --state NEW -m tcp --tcp-flags FIN,SYN,RST,PSH,ACK,URG NONE -j DROP
-A INPUT -p tcp -m tcp --tcp-flags FIN,SYN,RST,ACK SYN -j SYN_FLOOD
-A INPUT -p tcp -m state --state NEW -m tcp --dport 80 -j ACCEPT
-A INPUT -p tcp -s 192.168.2.154 -m state --state NEW -m tcp --dport 22 -j ACCEPT
```

Figura 5. Nuevas reglas en IPTABLES

Mediante las reglas adicionadas el servidor queda protegido de varios tipos de ataques tanto DOS como DDOS, además solo el puerto 80 que pertenece al servidor Web queda con acceso público y se cierra el acceso al puerto SSH para administración remota solo a la IP 192.168.2.154 perteneciente al administrador de la red.

El otro nivel de seguridad se realizó en las aplicaciones, tanto en la que brinda el servicio Web, en este caso el nginx oculta la versión usada como se muestra en la figura 6, como en el gestor de contenidos WordPress. En este último, instala los módulos de seguridad necesarios para contención de posibles atacantes.

```
events {
        worker_connections 768;
        # multi_accept on;
}
http {

        ##
        # Basic Settings
        ##

        sendfile on;
        tcp_nopush on;
        tcp_nodelay on;
        keepalive_timeout 65;
        types_hash_max_size 2048;
        server_tokens off;

        # server_names_hash_bucket_size 64;
        # server_name_in_redirect off;

        include /etc/nginx/mime.types;
-- INSERT --
```

Figura 6. Configuración nginx para ocultar la versión.

Opuntia Brava
ISSN: 2222-081X
RNPS: 2074
Volumen: 13 Número: 1
Recepción: 12/12/20. Aprobado: 11/01/21

Maidel Arsenio de la Rosa Téllez
Mirelis Alcina Reyes
Ariel Céspedez Pérez

En el gestor de contenidos wordpress se instalaron los plugin de seguridad para proteger al sistema tanto de posibles ataques DDOS como la intromisión de intrusos o hackers directamente a los contenidos. Entre los plugin que se instalaron se encuentran los siguientes: iThemes Security, WordFence, BulletProof, All In One WP Security & Firewall, SecuPress.

Presentación de los resultados

Una vez configurados los servicios y aplicaciones en el servidor Web, se procede a realizar el escaneo de puertos nuevamente al servidor Web como se muestra en la figura 7.

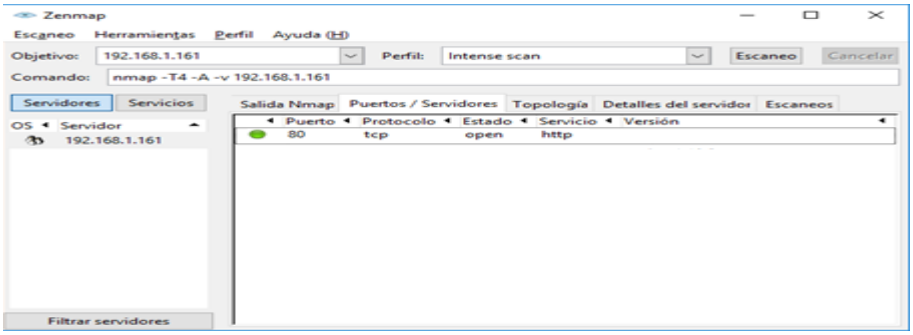

Figura 7. Escaneo de puerto con zenmap

Una vez realizado el scaneo de puertos con la herramienta Zendmap se puede apreciar que solo se encuentra abierto el puerto 80, que corresponde al servidor y además no muestra la versión usada como anteriormente.

Como paso adicional se realiza nuevamente el ataque al servidor, iden al realizado en la figura 2 para analizar el comportamiento del mismo después de las configuraciones de seguridad realizadas, los resultados del comando ping al servidor se muestran en la figura 8.

```
PING 192.168.1.161 (192.168.1.161) 56(84) bytes of data.
64 bytes from 192.168.1.161: icmp_seq=1 ttl=64 time=0.938 ms
64 bytes from 192.168.1.161: icmp_seq=2 ttl=64 time=1.40 ms
64 bytes from 192.168.1.161: icmp_seq=3 ttl=64 time=0.268 ms
64 bytes from 192.168.1.161: icmp_seq=4 ttl=64 time=0.273 ms
64 bytes from 192.168.1.161: icmp_seq=5 ttl=64 time=0.277 ms
64 bytes from 192.168.1.161: icmp_seq=6 ttl=64 time=1.32 ms
64 bytes from 192.168.1.161: icmp_seq=7 ttl=64 time=0.258 ms
64 bytes from 192.168.1.161: icmp_seq=8 ttl=64 time=0.286 ms
64 bytes from 192.168.1.161: icmp_seq=9 ttl=64 time=1.35 ms
64 bytes from 192.168.1.161: icmp_seq=10 ttl=64 time=0.293 ms
64 bytes from 192.168.1.161: icmp_seq=11 ttl=64 time=0.293 ms
64 bytes from 192.168.1.161: icmp_seq=12 ttl=64 time=0.298 ms
64 bytes from 192.168.1.161: icmp_seq=13 ttl=64 time=0.303 ms
64 bytes from 192.168.1.161: icmp_seq=14 ttl=64 time=0.531 ms
64 bytes from 192.168.1.161: icmp_seq=15 ttl=64 time=0.289 ms
^C
--- 192.168.1.161 ping statistics ---
15 packets transmitted, 15 received, 0% packet loss, time 14006ms
rtt min/avg/max/mdev = 0.258/0.559/1.406/0.436 ms
```

Figura 8. Uso del comando ping después de las configuraciones realizadas.

Maidel Arsenio de la Rosa Téllez
Mirelis Alcina Reyes
Ariel Céspedez Pérez

Opuntia Brava
ISSN: 2222-081X
RNPS: 2074
Volumen: 13 Número: 1
Recepción: 12/12/20. Aprobado: 11/01/21

En las estadísticas de la figura anterior se muestra que ya no existen pérdidas de paquetes, de 15 paquetes trasmitidos se reciben 15 para un 0% de pérdidas de paquetes. Por último, se accede al sitio mediante el navegador Mozilla Firefox como se muestra en la figura 9.

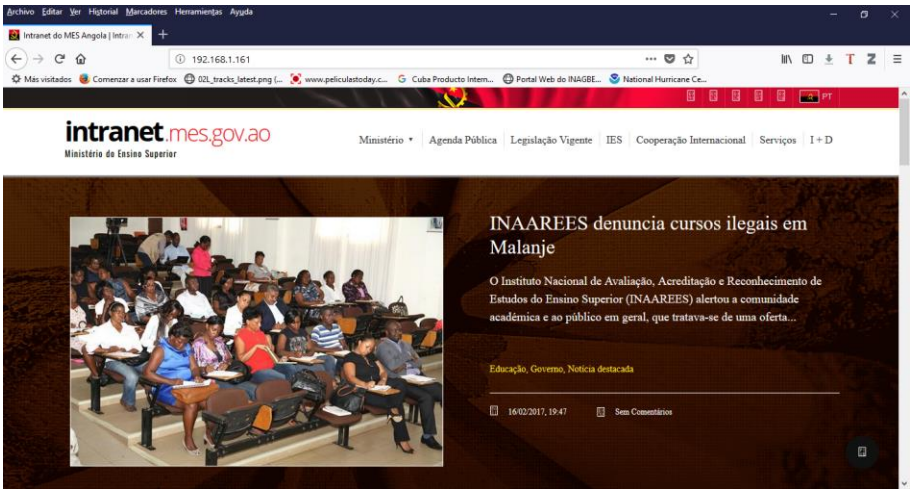

Figura 9. Disponibilidad del servicio Web

Como se muestra en la figura anterior el sitio Web se encuentra disponible encontrándose el servidor bajo ataques, lo que demuestra la efectividad de las configuraciones realizadas surtieron el efecto esperado.

Precisiones finales

Al realizar el primer escaneo de puertos al servidor se detectaron vulnerabilidades que atentan contra la seguridad y disponibilidad del sitio Web, lo que conllevó a realizar una serie de contramedidas en las configuraciones del servidor. Se realizó un ataque tipo DDOS que dejó el servidor fuera de servicio, lo que evidenció las vulnerabilidades existentes.

Una vez corregidas las configuraciones que tenían las fallas en la seguridad del servidor, se procedió a realizar nuevamente el ataque para comprobar los resultados, los cuales fueron del todo positivos y permitieron que el servidor funcionara todo el tiempo.

Referencias

Cheng, J., Zhang, C., Tang, X., Sheng, V. S., Dong, Z. y Li, J. (2018). Adaptive DDoS Attack Detection Method Based on Multiple-Kernel Learning. *Security and Communication Networks*. Recuperado de https://doi.org/10.1155/2018/5198685

De Donno, M., Dragoni, N., Giaretta, A. y Spognardi, A. (2018). DDoS-Capable IoT Malwares: Comparative Analysis and Mirai Investigation. *Security and Communication Networks*. Recuperado de https://doi.org/10.1155/2018/7178164

Opuntia Brava
ISSN: 2222-081X
RNPS: 2074
Volumen: 13 Número: 1
Recepción: 12/12/20. Aprobado: 11/01/21

Guijarro, A., Cevallos, L. y Torres, I. (2018). Estrategia cognitiva para la evaluación de una central Ip basada en Asterisk frente a los ataques DoS. *Opuntia Brava*, *8*(4). Recuperado de http://opuntiabrava.ult.edu.cu/index.php/opuntiabrava/article/view/271

Han, B., Yang, X., Sun, Z., Huang, J. y Su, J. (2018). OverWatch: A Cross-Plane DDoS Attack Defense Framework with Collaborative Intelligence in SDN. *Security and Communication Networks*. Recuperado de https://doi.org/10.1155/2018/9649643

Hernández, A. L. y Mejia, J. (2015). Guía de ataques, vulnerabilidades, técnicas y herramientas para aplicaciones web. *ReCIBE. Revista electrónica de Computación, Informática, Biomédica y Electrónica*, (1). Recuperado de https://www.redalyc.org/articulo.oa?id=512251501005

Huang, C., Yi, P., Zou, F., Yao, Y., Wang, W. y Zhu, T. (2019). CCID: Cross-Correlation Identity Distinction Method for Detecting Shrew DDoS. *Wireless Communications and Mobile Computing*. Recuperado de https://doi.org/10.1155/2019/6705347

Jaafar, G. A., Abdullah, S. M. y Ismail, S. (2019). Review of Recent Detection Methods for HTTP DDoS Attack. *Journal of Computer Networks and Communications*. Recuperado de https://doi.org/10.1155/2019/1283472

Jia, B., Huang, X., Liu, R. y Ma, Y. (2017). A DDoS Attack Detection Method Based on Hybrid Heterogeneous Multiclassifier Ensemble Learning. *Journal of Electrical and Computer Engineering*. Recuperado de https://doi.org/10.1155/2017/4975343

Valarezo, M., Honores, J., Gómez, A. y Vinces, L. (2018). Comparación de tendencias tecnológicas en aplicaciones web. *3C Tecnología. Glosas de innovación aplicadas a la pyme*, 28-49. Recuperado de http://ojs.3ciencias.com/index.php/3c-tecnologia/article/view/618