UNIVERSIDAD INTERNACIONAL DE LA RIOJA

Universidad Internacional de La Rioja
Máster universitario en Seguridad Informática

Diseño de prototipo de defensa para mitigación de ataques DDoS para PYMES

Trabajo Fin de Máster

Presentado por: Rosety Molins, Blanca

Director/a: Vázquez Poletti, Jose Luis

Ciudad: Cartagena

Fecha: 14 de Julio de 2016

Resumen

En la actualidad, muchas pequeñas empresas desarrollan parte de su negocio en Internet, pero no disponen de los recursos económicos necesarios para invertir en seguridad para sus Sistemas de Información. El objetivo de este trabajo es poder facilitar unas pautas de diseño y configuración basadas en software gratuito, que les permitan ofrecer un servicio estable y robusto, capaz de superar determinados ataques DDoS o minimizar su impacto en el negocio.

Las pruebas realizadas durante el estudio, demuestran que una buena configuración de elementos tales como firewalls, IPS y balanceadores de carga permiten detectar, mitigar o, incluso, evitar muchos de estos ataques y todo ello, sin coste de adquisición de hardware o software específico.

Palabras Clave: Denegación de Servicio, Ataque, Mecanismos de Defensa

Abstract

Nowadays, most small companies develop at least part of their business on the Internet. However, they do not have enough economic resources to invest on theirs information systems security. The aim of this paper is to provide guidelines for the design and configuration of a stable and robust service, based on free software, capable of overcoming certain DDoS attacks or, at the very least, minimize their impact of such attacks to a company's business.

Tests performed during the study, show that an appropriate configuration of elements such as firewalls, IPS and load balancers allow detection, mitigation, or even, avoiding many of these attacks, without the acquisition cost of specific hardware or software.

Key Words: Denial of Service, Attack, Defense Mechanisms

Tabla de Contenidos

Resumen....2

Abstract....2

1. Introducción....6

1.1. Motivación para el desarrollo del estudio....6

1.1.1. Causas de los ataques DDoS....7

1.1.2. Consecuencias de los ataques DDoS....7

1.1.3. Ejemplos de ataques de los últimos años....8

1.2. Propósito de la investigación....10

1.3. Estructura del documento....11

2. Ataques de Denegación de Servicio....13

2.1. Que es un DoS/DDoS....13

2.2. Clasificación de ataques de denegación de servicio....14

2.3. Mecanismos de defensa....15

2.3. Estadísticas....19

3. Metodología de Trabajo....22

3.1. Descripción de la metodología....22

3.2. Software básico de implementación de la arquitectura....22

3.3. Tipo de métrica definido para el estudio....23

3.4. Descripción del entorno de pruebas completo....24

3.4.1. Arquitectura de PYME protegida....24

3.4.2. Arquitectura de PYME sin protección....28

4. Pruebas y Análisis de Resultados....29

4.1. Estrategias de mitigación incorporadas....29

4.2. Mecanismo de medición de las pruebas 32

4.3. Tipo de ataque: TCP SYN Flood 32

4.3.1. Ejecución de la prueba 34

4.3.2. Resultados 35

4.4. Tipo de Ataque: TCP Connect Flood 37

4.4.1. Ejecución de la prueba 37

4.4.2. Resultados 38

4.5. Tipo de Ataque: UDP Flood 40

4.5.1. Ejecución de la prueba 41

4.5.2. Resultados 41

4.6. Tipo de Ataque: UDP Fragmentation Flood 43

4.6.1. Ejecución de la prueba 43

4.6.2. Resultados 44

4.7. Tipo de Ataque: ICMP Flood 45

4.7.1. Ejecución de la prueba 45

4.7.2. Resultados 46

4.8. Tipo de Ataque: Slow HTTP DDoS 46

4.8.1. Slow Headers 48

4.8.2. Slow Read 49

4.8.3. Apache killer 51

4.8.4. Slow http post body 53

5. Trabajo Futuro 55

6. Conclusiones 57

7. Referencias 59

8. Anexos 61

8.1. Anexo 1. Tipos de Ataques DDoS 61

8.2. Anexo 2. Glosario/Acrónimos 70

8.3. Anexo 3. Configuración de IPtables 71

8.4. Anexo 4. Reglas de configuración de ModSecurity 74

Índice de Figuras

Figura 1. Ejemplo de BOTNET 13

Figura 2. Ejemplo de arquitectura de red de PYME desprotegida 17

Figura 3. Ejemplo de arquitectura de PYME con todas las recomendaciones anti DDoS 18

Figura 4. Origen de los ataques en el cuarto trimestre de 2015 (Akamai) [12]....19

Figura 5. Pico máximo alcanzado por ataques DDoS en los últimos 12 años [13]....20

Figura 6. Estudio sobre herramientas de detección de amenazas DDoS de Arbor Networks[13] 21

Figura 8. Ejemplo de arquitectura de red de defensa de PYME con recomendaciones anti DDoS sin elementos comerciales....25

Figura 9. Arquitectura implementada para el estudio....26

Figura 10. Arquitectura física de la PYME protegida 27

Figura 11. Establecimiento de conexión TCP 33

1. Introducción

Los ataques de Denegación de Servicio (DoS) y los ataques de Denegación de Servicio Distribuidos (DDoS) tienen como objetivo conseguir que los usuarios legítimos de un servicio o recurso no puedan disponer de él. En ningún caso tratan de atentar contra la integridad o la confidencialidad de los sistemas de información, únicamente contra su disponibilidad.

Al tratarse de acciones relativamente fáciles de ejecutar y sin embargo, muy complicadas de contrarrestar, se han convertido en un recurso habitual para gran variedad de hackers, que se sienten seguros perpetrando ataques distribuidos debido a la complejidad de identificación del atacante. Su eficacia se debe por un lado, a la gran cantidad de dispositivos que pueden emplear simultáneamente para llevar a cabo esta acción, por ejemplo, equipos infectados con algún tipo de malware; y por otro, a que no es necesario superar las medidas de seguridad que protegen a los sistemas de información (SI), pues no se intenta penetrar en su interior, solo bloquearlo. Estas características, convierten a los ataques DoS y DDoS en una gran amenaza para cualquier negocio, gobierno o infraestructura crítica, provocando pérdidas multimillonarias cada año.

1.1. Motivación para el desarrollo del estudio

En los últimos años se han desarrollado numerosos mecanismos de defensa para combatir este tipo de ataque. Sin embargo, los piratas informáticos modifican sus herramientas continuamente para saltarse los sistemas de seguridad, obligando a los investigadores a tratar de optimizar y actualizar sus mecanismos de protección para gestionar los ataques de la mejor forma posible.

En el caso de una pequeña empresa, el capital disponible para invertir en la seguridad de sus SI suele ser muy limitado, en algunos casos, prácticamente inexistente. El objetivo de este trabajo es poder facilitar unas pautas de diseño y configuración que permitan a estas PYMES ofrecer un servicio a clientes, proveedores y empleados, lo más estable y robusto posible y que sea capaz de superar determinados ataques DDoS o minimizar su impacto en el negocio.

1.1.1. Causas de los ataques DDoS

Los motivos que pueden llevar a un individuo u organización a lanzar un ataque de Denegación de Servicio contra un determinado sistema son muy variados, pudiendo tratarse incluso, de acciones legítimas ejecutadas por el bien del objeto del ataque.

Las razones más comunes por las que se lanzan este tipo de acciones son las siguientes:

- Se desea bloquear los servicios de la víctima por diversos motivos:
  - Políticos, ideológicos o como arma de guerra cibernética.
  - Mercantiles, empresas que se atacan entre sí para mermar su volumen de negocio.
  - Chantaje, el atacante pide algo a cambio de detener los ataques.
  - Búsqueda de fama o reconocimiento de otros hackers.
  - Conflictos personales, venganza.
- Se ha instalado de forma ilegal algún tipo de malware y es necesario el reinicio del sistema para su correcto funcionamiento.
- Se desean cubrir acciones maliciosas o un uso abusivo de la CPU, provocando una caída del sistema que pueda ser asociada a algo pasajero y ocultando así la naturaleza real del ataque.
- Se quiere verificar la resistencia de un sistema ante este tipo de acciones o se necesita detener sus servicios.

Según el Global IT Security Risks Survey de Kaspersky Lab (2015) [1], las compañías encuestadas creen que el 12% de los ataques recibidos proceden de empresas de la competencia, que tratan de mejorar su posición en el mercado o acceder a información confidencial.

1.1.2. Consecuencias de los ataques DDoS

A pesar de que en apariencia, el único efecto directo de un ataque DoS es la pérdida de disponibilidad de los servicios ofrecidos por una empresa o entidad, las consecuencias que derivan de él son muy variadas. Las principales preocupaciones de las empresas en lo que respecta a este tema, se centran en la pérdida de reputación (27%), pérdida de oportunidades de negocio (27%), el coste derivado de la lucha contra el ataque y la restauración del servicio (16%), y la pérdida de clientes (15%).

Alfonso Ramírez, Director General de Kaspersky Lab Iberia hizo la siguiente declaración:

> Es importante tomarse en serio un ataque DDoS. Es un ciberataque relativamente fácil de perpetrar pero el efecto sobre la continuidad del negocio puede ser de largo alcance. Nuestro informe encontró que, además del tiempo de inactividad del sitio web, el daño a la reputación y los clientes insatisfechos, los ataques DDoS también pueden penetrar profundamente en los sistemas internos de la empresa. Y no importa lo pequeña que sea la empresa, o si tiene o no tiene un sitio web: si está online, es un objetivo potencial. Los sistemas operativos no protegidos son tan vulnerables a un ataque DDoS como el sitio web externo, y cualquier interrupción puede detener un negocio. (Rosa Martín, Newsbook.es, 2016) [2]

El abaratamiento del coste de lanzamiento de los ataques DDoS tiene como consecuencia un rápido aumento del volumen de este tipo de acción. Además, cada día son mucho más sofisticados y complejos, haciendo que una buena defensa contra ellos sea mucho más complicada, de acuerdo con el informe citado en el apartado anterior [1].

El coste medio de un ataque DDoS a una PYME es de unos 53.000$, en caso de una empresa multinacional, esta cifra asciende hasta los 417.000$. Además, es importante señalar que el 26% de las empresas que sufrieron un ataque DDoS admitieron la pérdida de información reservada, y un 31%, de información no-sensible para el desarrollo del negocio (Kaspersky Lab, 2015)[3] Sin embargo, a pesar de que la gran mayoría de las compañías conocen la importancia de instalar sistemas de protección y prevención de ataques de denegación de servicio, solo un 31% de las empresas españolas cuentan con un sistema completo de protección. (Patricia Bachmaier, Computer World, 2016). [22]

1.1.3. Ejemplos de ataques de los últimos años

Como se puede ver en algunos de los ejemplos que se incluyen a continuación, los ataques de denegación de servicio son empleados por activistas, gobiernos, empresas, ciber-delincuentes que ofrecen sus servicios a cambio de dinero, o individuos que realizan este tipo de acciones simplemente por diversión o para comprobar si son capaces de conseguir su objetivo.

Este tipo de ataque es cada vez más demandado y puede ser utilizado para fines muy diversos: censura, defensa propia, motivos políticos, competencia desleal, ciberguerra o,

simplemente, para desviar la atención de un ataque de mayor repercusión. Cualquier tipo de organización de cualquier país del mundo, puede convertirse en objeto de un ataque DDoS en cualquier momento:

- A principios de este mismo año, la cadena británica BBC y la web oficial de Donald Trump, fueron objeto del mayor ataque DDoS de la historia, alcanzando los 602 Gbps. La organización New World Hacking se ha atribuido ambos ataques, en los que se utilizaron dos servidores de Amazon Web Services fuertemente protegidos por técnicas de detección, mitigación y prevención de mal uso de sus servicios. Según un portavoz de New World Hacking, han desarrollado una nueva herramienta (BangStresser) cuyo objetivo es desenmascarar al Estado Islámico y terminar con su propaganda en la red y comenta que estos dos ataques no son más que una demostración de su verdadero poder. (David Bisson, 2016) [4]

- En marzo de este 2015, China lanzó durante días un ataque DDoS contra GitHub dirigido en particular contra los proyectos GreatFire y CN-NYTimes (versión en chino del periódico The New York Times, cuyo principal objetivo es mermar la censura del gobierno de dicho país). (Mohit Kumar, 2015) [5]

- En noviembre de 2014, un ataque atribuido a Corea del Norte dejó sin servicio a Sony Pictures, que tardó una semana en recuperar el control total de sus sistemas informáticos. Las consecuencias de este ataque fueron más allá que un simple bloqueo de los sistemas de la empresa, ya que unos días más tarde se comenzó a filtrar información confidencial de todo tipo, empezando por la subida a varios sitios de descarga, de archivos de cinco estrenos próximos. Unos días después, el servicio de Internet de Corea del Norte sufrió una serie de cortes inicialmente intermitentes, pero que terminaron deteniendo completamente el servicio. Hay dudas sobre el origen de este contraataque, aunque se barajan principalmente dos posibilidades:
    - Que haya sido lanzado por el gobierno de EEUU como represalia por la ofensiva a Sony Pictures.
    - Que haya sido el grupo Anonymous, que había anunciado medidas contra Corea del Norte por la censura a la película de Sony Pictures “The Interview”.

  Por otro lado, Sony Pictures después del ciberataque del que fue víctima, decidió lanzar un ataque DDoS desde los centros de datos de Singapur y Tokio contra los servidores que almacenaban o compartían las películas pirateadas, con la intención de colapsar la transferencia de datos. (Dann Albright, 2015) [6] y (John Kennedy, 2014) [7]

- El 29 de julio de 2014, hackers de Cyber Berkut, un grupo de activistas pro-rusos creado después de la disolución de las fuerzas especiales ucranianas Berkut, bloquearon con un ataque DDoS durante casi 24 horas la página web del presidente de Ucrania, Petró Poroshenko, a quien acusaron de "genocidio de su propio pueblo". Unos meses antes, este mismo grupo había dirigido un ataque DDoS de amplificación de DNS y reflexión del protocolo NTP que alcanzó un pico máximo de 400 Gbps, contra algunos servidores de la OTAN, debido a su desacuerdo con la presencia de la OTAN en suelo ucraniano. (Sputniknews, 2014) [8]

- En marzo de 2013 se efectuó uno de los mayores ataques DDoS conocidos hasta aquel momento, basado en la amplificación de DNS que alcanzó un pico máximo de 300Gbps, y que ralentizó el funcionamiento de Internet en varios países europeos. La página de la organización Spamhaus que lucha contra correos basura fue el principal objetivo de los piratas informáticos que lanzaron el ataque a modo de protesta. (Securitybydefault, 2013) [9]

- En junio de 2011, el grupo activista LulzSec dejó fuera de servicio durante un par de horas el portal de la Agencia Central de Inteligancia (CIA) y el Senado estadounidenses, mediante sendos ataques distribuidos de denegación de servicio. (Graham Cluley, 2011) [10]

- A finales de 2010 Anonymous lanzó una serie de ataques DDoS contra PayPal, Visa y MasterCard por oponerse a la actividad de Wikileaks. (Lindsay Fortado, 2012) [11]

1.2. Propósito de la investigación

Esta investigación tiene como objetivo mejorar la resistencia de los Sistemas de Información de las pequeñas empresas, a los ataques de denegación de servicio a los que puedan verse expuestas, a partir de la incorporación de elementos software libres que permitan por un lado, combatir de la forma más efectiva dichos los ataques, y por otro, minimizar el tiempo de recuperación y el impacto tanto en sus sistemas como en el desarrollo del negocio, después de un ataque.

1.3. Estructura del documento

El contenido del presente trabajo ha sido organizado de forma que el lector conozca el alcance y las consecuencias que pueden llegar a tener un ataque de denegación de servicio sobre cualquier organización a nivel mundial, antes de profundizar en las características, clasificación o tipo de ataques existentes:

- Capítulo 1: Introducción
  Este apartado introductorio pretende por un lado, mostrar los motivos que pueden llevar a un individuo o grupo organizado a ejecutar un ataque de denegación de servicio contra cualquier sistema de información expuesto a Internet, y por otro, recalcar las graves consecuencias que pueden llegar a ocasionar los ataques de denegación de servicio sobre cualquier empresa, organización o gobierno; así como el perjuicio económico que supone la pérdida de oportunidad de negocio, la degradación de la imagen pública, y los gastos de recuperación de los sistemas.

- Capítulo 2: Ataques de Denegación de Servicio
  Descripción, análisis y clasificación de ataques DoS y DDoS. A continuación se explican los tipos de mecanismos de defensa existentes para evitar o mitigar las consecuencias de estos ataques. Algunas de estas herramientas hardware y software tienen un alto precio, por ello, este estudio trata de ofrecer a las PYMEs protección al mínimo coste. El capítulo termina con un resumen basado en estadísticas de la duración, frecuencia, amplitud y otro tipo de información que permite al lector hacerse una idea general y, sobre todo, muy actual, de la importancia y el alcance de estas acciones.

- Capítulo 3: Metodología de Trabajo
  En este capítulo se explica el proceso continuo de diseño, instalación, configuración y pruebas del entorno; las herramientas que han sido necesarias para desarrollar la investigación y las métricas empleadas para el análisis de la respuesta ofrecida por el sistema durante los ataques. Antes de proceder a explicar las pruebas realizadas sobre el sistema en el siguiente apartado, se expone la arquitectura final implementada para la protección de la PYME y el diseño del entorno de pruebas (elementos hardware y software empleados).

- Capítulo 4: Pruebas y Análisis de Resultados
  Antes de proceder a la descripción de cada una de las pruebas realizadas, el tipo de ataque al que hacen referencia y los resultados obtenidos, se ha incluido en este

capítulo un apartado en el que se explican cada una de las herramientas gratuitas que se han utilizado para la investigación y el método de medición empleado en las pruebas, para saber si ha sido posible mejorar la respuesta del sistema con la integración de los elementos de protección.

- Capítulo 5: Trabajo Futuro
  Durante el desarrollo de la investigación, se han encontrado algunas limitaciones que no han permitido incluir todos los elementos propuestos para mejorar la resistencia de los sistemas a ataques DDoS ni realizar pruebas distribuidas basadas en la utilización de una BOTNET. Sin embargo, sería interesante en caso de poder superar estas limitaciones, ampliar el trabajo de investigación realizado hasta el momento.

- Capítulo 6: Conclusiones
  En este apartado se trata de responder a preguntas del tipo: ¿Es posible mejorar la disponibilidad de los sistemas de información de una empresa pequeña, expuesta a Internet, con una arquitectura de software libre? ¿Se pueden bloquear ataques DDoS, minimizar su impacto en el negocio o sencillamente, detectar estas acciones con alguna herramienta que permita a un administrador con conocimientos técnicos limitados, frenar temporalmente la embestida?

2. Ataques de Denegación de Servicio

2.1. Que es un DoS/DDoS

Un ataque de Denegación de Servicio (DoS) o ataque Distribuido de Denegación de Servicio (DDoS) es un tipo de ciberataque en el que uno o varios ordenadores envían paquetes, datos o transacciones a través de la red en un intento de inhabilitar un servidor, servicio o infraestructura acaparando sus recursos o sobrecargando su ancho de banda.

En ocasiones, los equipos que ejecutan el ataque pertenecen a un conjunto de cibercriminales cuyo objetivo es detener total o parcialmente el funcionamiento de un sitio web; sin embargo, en otros muchos, casos los ordenadores que lanzan el ataque no son más que equipos infectados por algún tipo de malware, que forman parte de una BOTNET sin tener conocimiento de ello.

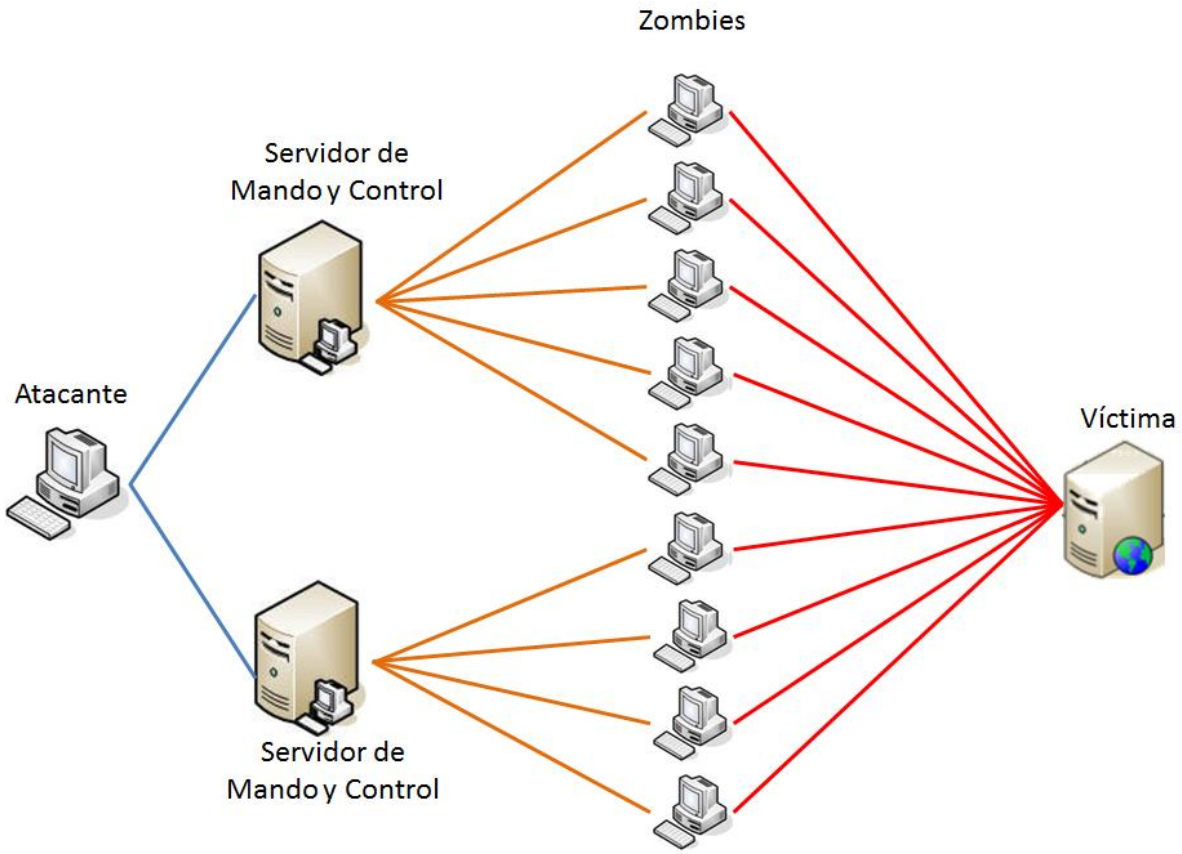

Figura 1. Ejemplo de BOTNET

Los ataques DDoS se caracterizan por ser muy complicados de evitar, ya que el tráfico malicioso es muy difícil de detectar y de parar, son fácilmente escalables, en muchas ocasiones intervienen gran cantidad de máquinas, y rara vez se puede localizar su origen.

2.2. Clasificación de ataques de denegación de servicio

La primera y más habitual clasificación que se puede hacer de este tipo de ataques es la siguiente:

- Ataques volumétricos: Su finalidad es saturar el ancho de banda de un sitio web con la idea de causar congestión. UDP flood e ICMP flood son ejemplos de este tipo de ataque, que en los últimos años han empezado a emplear técnicas de amplificación/reflexión para aumentar aún más su impacto. Se suelen medir en bits por segundo.
- Ataques de protocolo: Consumen recursos del servidor o de algún intermediario, firewalls o sistemas de balanceo de carga. Como ejemplo de ataques de este tipo están SYN flood o ataques de fragmentación de paquetes. En este caso, su medición se hace en paquetes por segundo.
- Ataques de capa de aplicación: Se envían peticiones en apariencia legítimas con la finalidad de agotar ciertos recursos del servidor para que así deje de funcionar. Los ataques de tipo Low and Slow son un buen ejemplo, y se suelen medir en peticiones por segundo.

Además, se puede decir que existen tres estrategias que permiten inhabilitar un sitio web:

- Por saturación: Su objetivo es agotar alguno de los recursos clave de un sistema con una cantidad demasiado alta de solicitudes:
    - Ancho de banda: Ataque que consiste en saturar la capacidad de la red del servidor, haciendo que sea imposible acceder a él.
    - Recursos: Ataque que trata de agotar los recursos computacionales del sistema, tales como espacio en disco o tiempo de procesador, impidiendo que este pueda responder a sus usuarios.

- Por explotación de vulnerabilidades: Categoría de ataque que explota fallos en el software que inhabilitan el equipo, toman su control, o lo vuelven inestable:
    - Modificación de la configuración, por ejemplo rutas de encaminamiento.
    - Alteración de información de estado, tal como interrupción de sesiones TCP (TCP reset).
- Por interrupción de componentes físicos de red, destrucción o alteración física de los equipos.

En el Anexo 1. Tipos de Ataques DDoS se puede ver una tabla en la que se describen distintos tipos ataques DDoS, la capa del modelo OSI a la que van dirigidos y los mecanismos de defensa que se podrían emplear para evitar su impacto.

2.3. Mecanismos de defensa

A pesar de haber muchas recomendaciones, configuraciones y herramientas hardware y software para combatir o resistir los posibles ataques de Denegación de Servicio de los que pueda ser objeto una empresa, el paso del tiempo ha demostrado que ninguna de ellas es suficiente por sí sola. Para conseguir tener un sistema seguro y confiable ante este tipo de ataques, se recomienda diseñar un mecanismo de defensa en profundidad que contenga los siguientes elementos:

- Contratar con el proveedor de servicio de Internet (ISP) un servicio de monitorización y limpieza (scrubbing) de paquetes maliciosos, que permita el manejo de ataques volumétricos a las capas 3 y 4 del modelo OSI. Este servicio, lo que hace es redirigir a sus sistemas el tráfico atacante para filtrarlo y reenviar únicamente los paquetes legítimos. De esta forma se consigue una primera mitigación de los ataques DDoS antes de que accedan a la infraestructura propia de la empresa.
- Ajustar los firewalls de la empresa para que sean capaces de gestionar grandes tasas de conexión, reconocer y manejar ataques volumétricos (bloquear paquetes DDoS) y de nivel de aplicación, y mejorar su rendimiento en circunstancias de ataque. Se recomienda, además, instalar una herramienta firewall de aplicación web (WAF) que protegerá al entorno de ataques contra la capa de aplicación (XSS, inyección SQL, etc)

- Instalación de un sistema de prevención de intrusiones (IPS), que se centrará en la mitigación de ataques en la capa de red. Es fundamental conocer el funcionamiento normal de la red y sus sistemas, ya que esto facilitará la detección de un uso anormal del ancho de banda, anomalías en la carga del sistema, etc.

- Desarrollar una estrategia de protección de las aplicaciones frente a ataques DDoS:
    - Ajuste y configuración segura de los servidores web.
    - Instalación de servicios auxiliares o redundantes y balanceadores de carga.
    - Introducción de mecanismos que eviten múltiples intentos de login y que limiten el número de conexiones nuevas al servidor.
    - Eliminación de actividades automatizadas.
    - Análisis del contenido de los servidores (es fundamental evitar almacenar ficheros de gran tamaño)
    - Instalación de parches y actualizaciones en los sistemas para limitar al máximo el número de vulnerabilidades explotables.
    - Instalación de herramientas antivirus/antimalware y de sistemas de monitorización de integridad del contenido (CIMS), que alertan sobre cambios en el sistema de archivos.

- Instalación de algún sistema de gestión de registros, que pueda generar alertas en tiempo real. El más recomendado sería el SIEM (Security Information Event Management) que proporciona análisis inteligente y mitigación de amenazas.

- Tener un plan de emergencia previsto, que trate de eliminar la incertidumbre tras la detección de un ataque.

Además de todo esto, es recomendable hacer una estimación del pico máximo de tráfico legítimo esperado, para contratar un ancho de banda suficientemente superior a este como para poder absorber ataques DDoS basados en volumen.

A continuación se muestra un esquema de arquitectura de red muy sencilla de una empresa pequeña en la que no se ha tenido en cuenta para nada la seguridad.

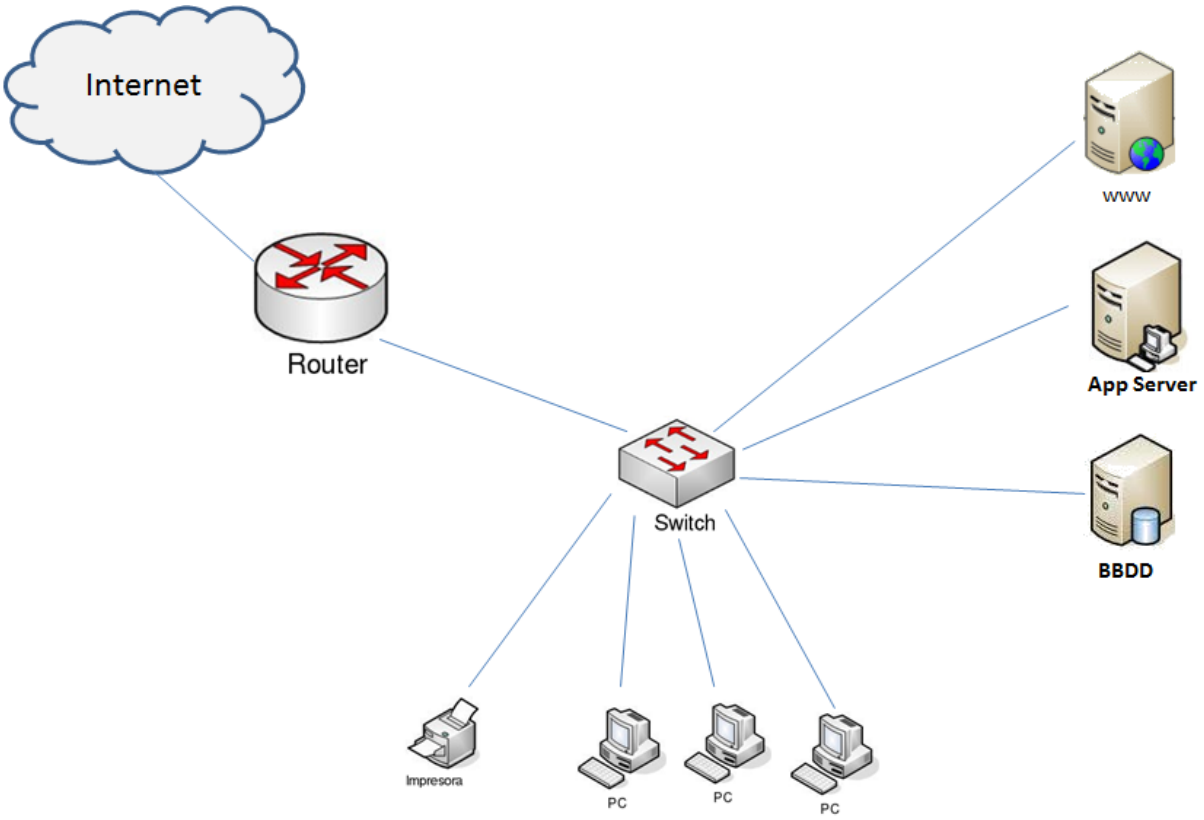

Figura 2. Ejemplo de arquitectura de red de PYME desprotegida

La siguiente figura, incorpora todos los mecanismos de seguridad recomendados anteriormente a la arquitectura de PYME desprotegida.

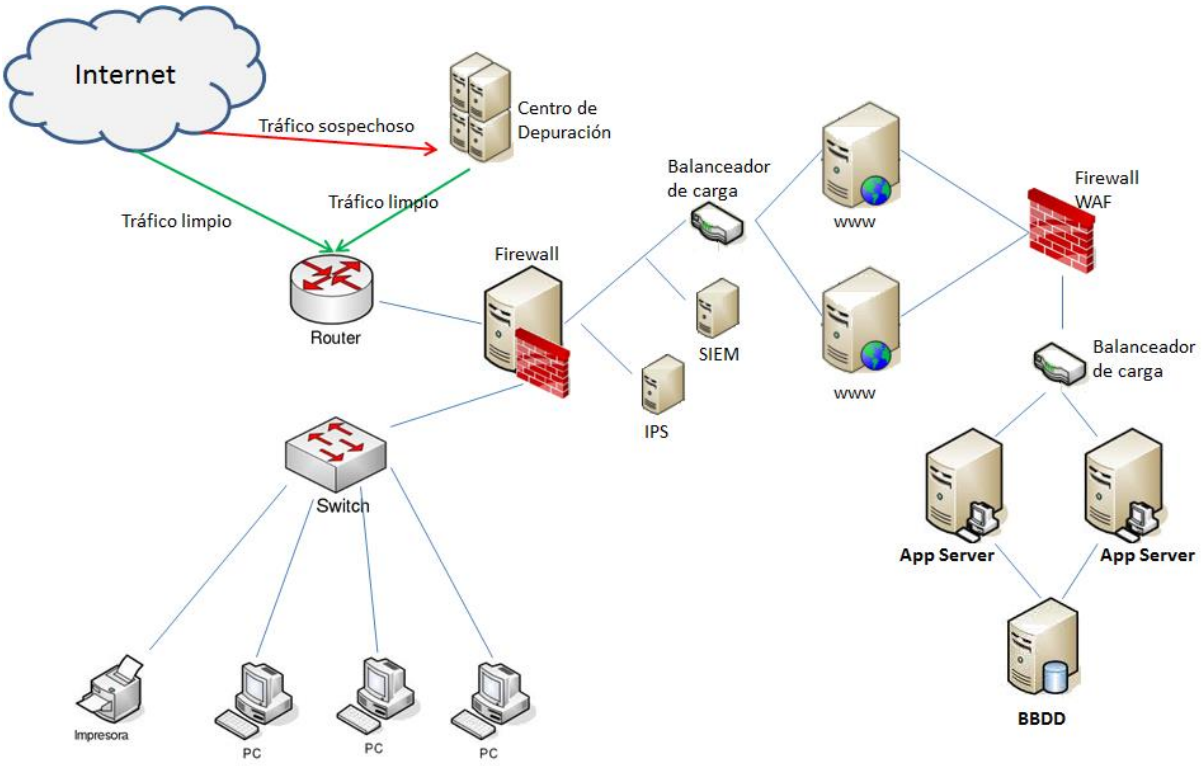

Figura 3. Ejemplo de arquitectura de PYME con todas las recomendaciones anti DDoS

Sin embargo, para conseguir una protección básica contra ataques de denegación de servicio no es necesario implantar todos estos mecanismos de seguridad. El objetivo, en el caso de una PYME, sería contar con un sistema suficientemente robusto en el que la inversión en mecanismos de seguridad sea asequible por la empresa; a ser posible a coste cero. Para conseguir esto, se va a tratar de diseñar una arquitectura basada en software libre que permita optimizar la seguridad de una pequeña empresa sin coste alguno.

En cualquier caso, cabe pensar que los recursos invertidos en un ataque de denegación de servicio contra una empresa serán normalmente proporcionales a su tamaño. Es decir, una multinacional será objeto de mayor cantidad de ataques y mucho más sofisticados que una pequeña empresa que ofrece un determinado servicio local.

2.3. Estadísticas

Según el estudio realizado por Akamai's State of the Internet [12] del cuarto trimestre del pasado año, China y Turquía fueron los países desde los que se originaron mayor cantidad de ataques DDoS:

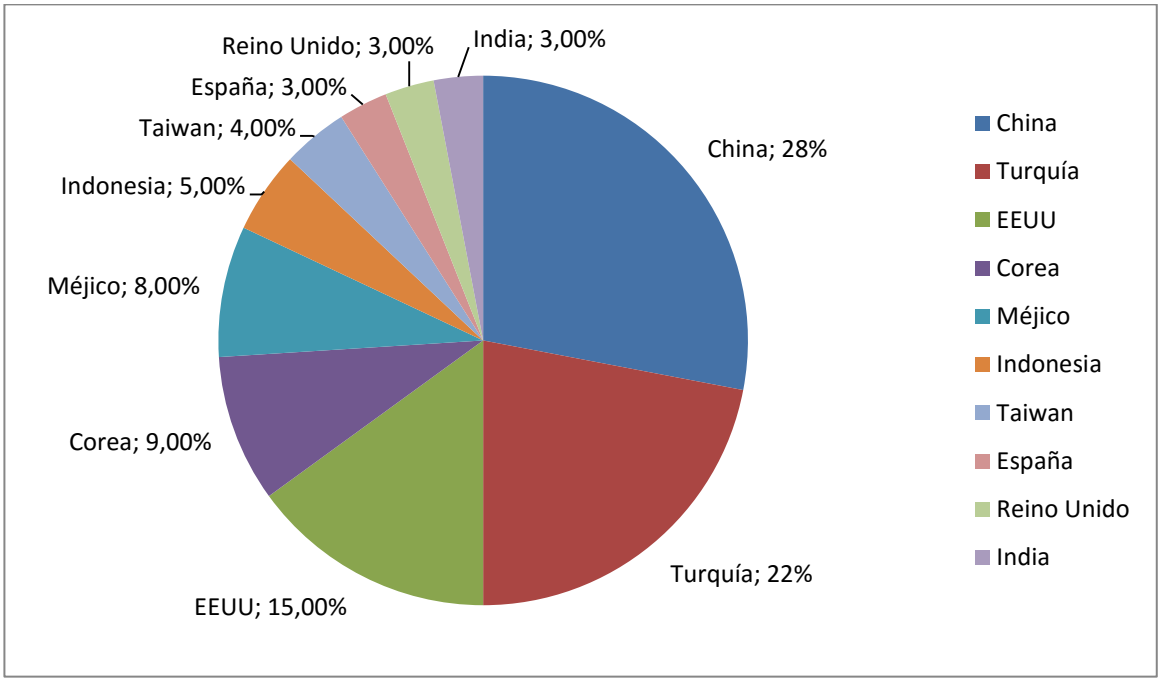

Figura 4. Origen de los ataques en el cuarto trimestre de 2015 (Akamai) [12]

Sin embargo, echando un vistazo a los cuatro informes trimestrales de 2015 realizados por esta misma empresa, se puede observar que el único dato que no varía en ningún caso es que a la cabeza de los ataques siempre se encuentran China y Estados Unidos. Otros países como España y Reino Unido aparecen también en todos los informes, pero con una tasa de generación de ataques muy inferior a los anteriores. En cuanto a Turquía, aparece en el informe del último trimestre de 2015 por primera vez.

El estudio del cuatro trimestre de 2015 realizado por Akamai [12] concluye lo siguiente:

- Hubo un aumento de un 149% en los ataques DDoS en el último trimestre del año.
- Los ataques son de menor duración y mayor frecuencia.
- Normalmente repetidos contra el mismo objetivo.

- Se estima que hubo un 46% de ataques de reflexión y un 56% de ataques multivectoriales.
- Hubo 5 ataques por encima de los 100 Gbps.
- Los ataques más frecuentes a la capa de infraestructura en el cuatro trimestre del año fueron: Fragmentación UDP, NTP, SYN y DNS, alcanzando conjuntamente casi el 60% del total. Sin embargo, teniendo en cuenta el resultado de los estudios realizados durante todo el año, se concluye que los vectores de ataque más frecuentes fueron: Fragmentación UDP, SSDP (que despuntó en el primer trimestre), SYN, NTP, UDP Flood y DNS.
- Las empresas dedicadas al juego y las tecnológicas fueron objeto de unas tres cuartas partes de los ataques DDoS en el segundo semestre de 2015.

La figura siguiente, basada en el Worldwide Infrastructure Security Report (Volume XI)[13] de Arbor Networks, muestra como en los últimos doce años ha habido un incremento realmente importante en el volumen de los ataques DDoS, que ha pasado de ser de un máximo de 8Gbps en el año 2004 a 602 Gbps a comienzos de 2016 (dato no proporcionado por el informe). El motivo es el reciente uso de las técnicas de amplificación/reflexión utilizando protocolos como NTP, DNS SNMP, CharGEN o SSDS que permiten aumentar la cantidad de tráfico en el ataque.

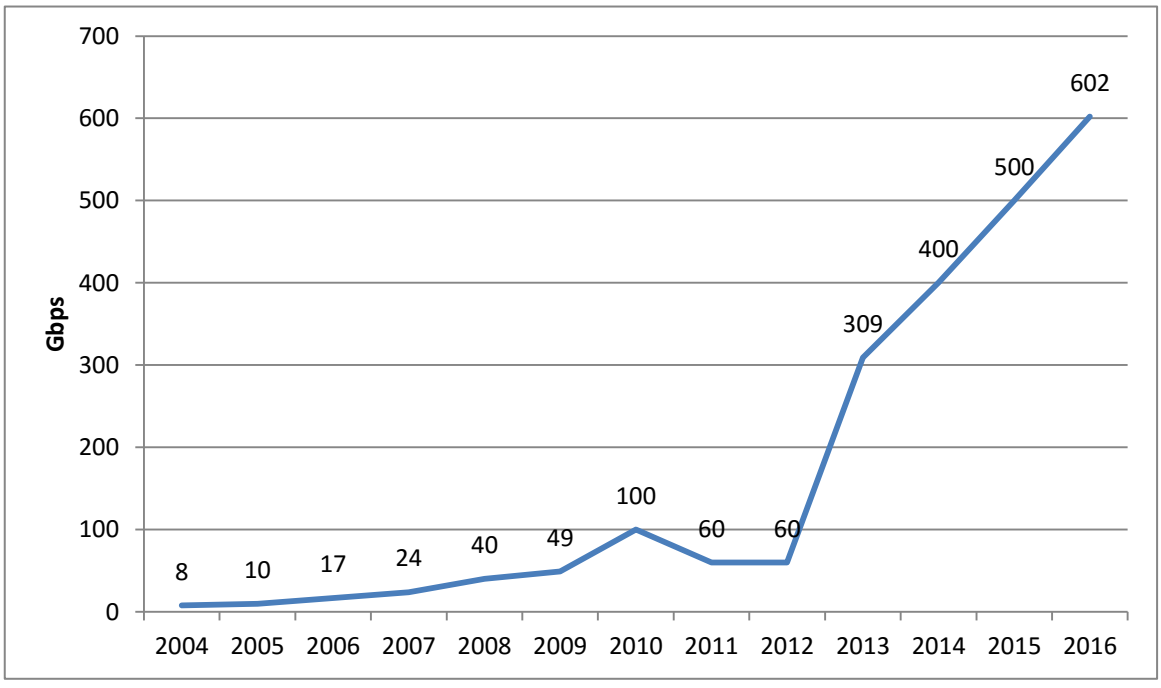

Figura 5. Pico máximo alcanzado por ataques DDoS en los últimos 12 años [13]

Según el mismo estudio de Arbor Networks [13], las herramientas más utilizadas en la detección de amenazas DDoS, son, en primer lugar, los analizadores basados en Netflow y en segundo, los logs de los firewalls. Sin embargo, estos últimos ocupan el sexto puesto en efectividad, siendo los sistemas in-line de detección/mitigación el segundo elemento más efectivo para la prevención y erradicación de los ataques:

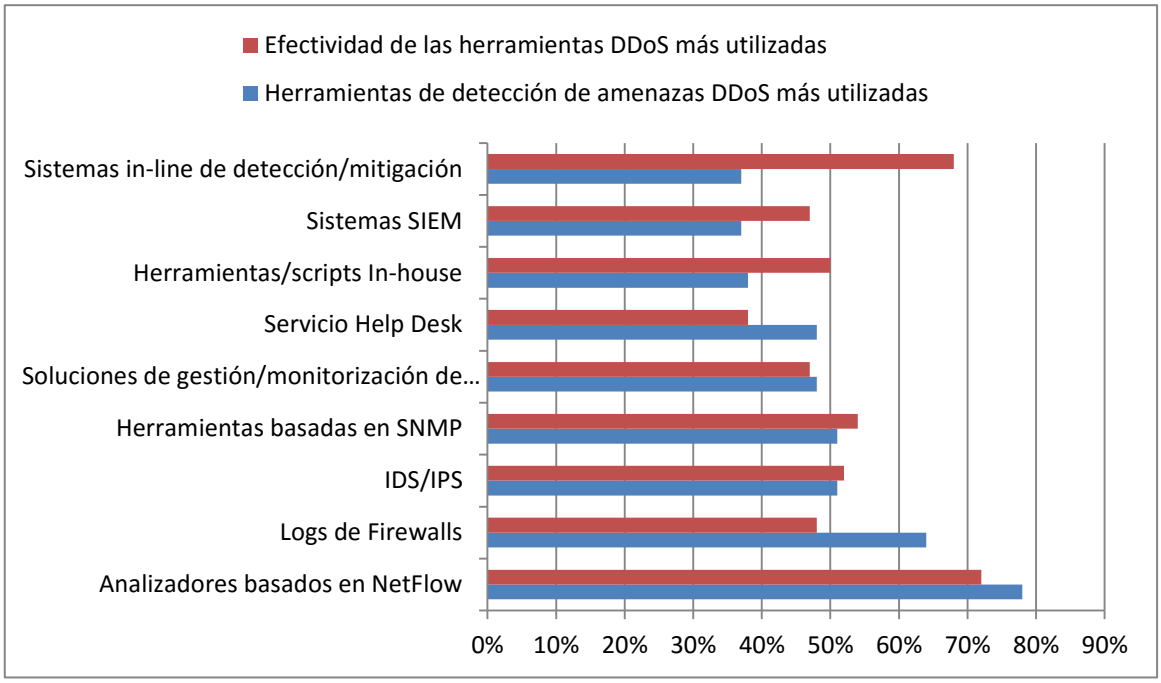

Figura 6. Estudio sobre herramientas de detección de amenazas DDoS de Arbor Networks[13]

3. Metodología de Trabajo

3.1. Descripción de la metodología

La arquitectura propuesta en los siguientes apartados es el resultado de un proceso continuo de diseño, instalación, configuración y pruebas de conexión y rendimiento de máquinas virtuales y software. Las limitaciones económicas, y más en particular las restricciones hardware, a las que se ha visto sometido el desarrollo del trabajo, han dificultado en gran medida el diseño del entorno y han provocado continuos cambios y reajustes del mismo.

Una vez alcanzado el mejor diseño posible para la PYME, se ha procedido a introducir uno a uno los elementos de protección recomendados y disponibles de manera gratuita, para comprobar su eficacia ante ataques DDoS.

Las pruebas realizadas sobre el entorno protegido se han ido ejecutando a medida que se iban incluyendo herramientas de protección, de forma que se han ido manteniendo aquellos elementos que proporcionaban una mejora en la respuesta del sistema ante un ataque, y se han eliminado los que de alguna manera empeoraban la resistencia o el rendimiento del sistema, no aportaban ningún beneficio o no se encontraban en un estado suficientemente maduro para su integración en el estudio.

3.2. Software básico de implementación de la arquitectura

El sistema de virtualización empleado para toda la arquitectura se basa en VMWare, que permite la creación de diversos tipos de redes y la simulación de tarjetas de red ofreciendo una gran flexibilidad de diseño.

Los Sistemas Operativos (SO) instalados son los siguientes:

- Ubuntu Server v14.04 para todas las máquinas virtuales de las PYMEs y para el único host no virtualizado de la arquitectura.
- Kali Linux v2016.1 para las MV de los atacantes.

La pequeña empresa que se simula en el estudio consta de dos tipos distintos de servidores:

- Servidor web Apache2
- Servidor de aplicaciones Wildfly

Las herramientas de ataque que se han utilizado para estresar de alguna manera los entornos con y sin protección son:

- Hping3: Ataques de tipo UDP Fragmentation Flood, UDP Flood, ICMP Flood y TCP SYN Flood con IP spoofing.
- Nping: Ataque del tipo TCP Connect Flood.
- SlowHttpTest: Ataques a la capa de aplicación del tipo Low and Slow: Slow Headers, Slow Read, Apache Killer, Slow HTTP Post Body

Es importante mencionar que se han descargado e instalado otras aplicaciones de Internet, publicitadas como herramientas para hacer pruebas de denegación de servicio, pero que no se han podido utilizar por contener malware, o simplemente, porque para su uso era necesario conectar los equipos a Internet de forma que el ataque se realizara a partir de una BOTNET cuya legalidad es de carácter dudoso. Antes de descartar definitivamente dichas herramientas, ha sido probado su funcionamiento en la red local, sin conseguir si quiera una ralentización de la PYME desprotegida.

Para el análisis de tráfico se han utilizado:

- Wireshark, sobre todo durante el diseño, instalación y configuración de los entornos.
- Tcpdump, como herramienta de observación de la transferencia de datos durante las pruebas realizadas en ambas arquitecturas de negocio.

3.3. Tipo de métrica definido para el estudio

El método empleado para medir la resistencia o mejora en la disponibilidad del sistema ante un ataque DDoS es bastante simple. Se ha desarrollado un script en el que, cada segundo, se hace una petición legítima al servidor web y se almacena en un registro sus tiempos de respuesta y descarga de datos.

Los resultados ofrecidos por este estudio indicarán si la PYME con sistemas de protección ha sido capaz de resistir a los ataques y en qué medida. Es decir, si durante el ataque no ha

habido perjuicio alguno para el usuario legítimo, si se ha podido recibir el contenido con pequeños retardos, o, en caso de caída inevitable del sistema, cuánto tiempo ha sido capaz de aguantar el ataque en comparación con la PYME sin protección.

3.4. Descripción del entorno de pruebas completo

El entorno de pruebas ha sido diseñado en función del hardware disponible:

- Un router Orange Livebox configurado para funcionar en modo switch.
- Cuatro ordenadores portátiles, de características y capacidades diferentes.

En cuanto a la configuración de los equipos, se ha realizado el diseño de la PYME en función de la capacidad de cada uno de ellos. En tres de los cuatro ordenadores, se utilizan máquinas virtuales para simular los servidores, enrutadores, balanceadores de carga o cualquier otro dispositivo que fuera necesario en el diseño. El ordenador restante, ha sido plataformado con Ubuntu Server y configurado para alojar al IDS Snort y para balancear la carga dirigida a los servidores web.

Al no disponer de ningún equipo con doble tarjeta de red que permitiera separar físicamente a los atacantes de la red interna de la PYME, se ha seleccionado el ordenador más potente de todos, en el que se puede ejecutar mayor número de máquinas virtuales simultáneamente, para la instalación de los atacantes, y del primer firewall-enrutador de acceso a la red interna (este será, por tanto, el punto de acceso al exterior). Las máquinas diseñadas para el ataque pertenecerán a una subred distinta al resto de equipos de la PYME, simulando así una separación física de los elementos.

3.4.1. Arquitectura de PYME protegida

Si se parte de la arquitectura propuesta en la *Figura 3. Ejemplo de arquitectura de PYME con todas las recomendaciones anti DDoS*, y se eliminan por un lado, los elementos comerciales de mayor coste, y por otro, la subred correspondiente a la intranet que en principio no será objeto de estudio en lo relativo a ataques DDoS, quedaría el siguiente esquema:

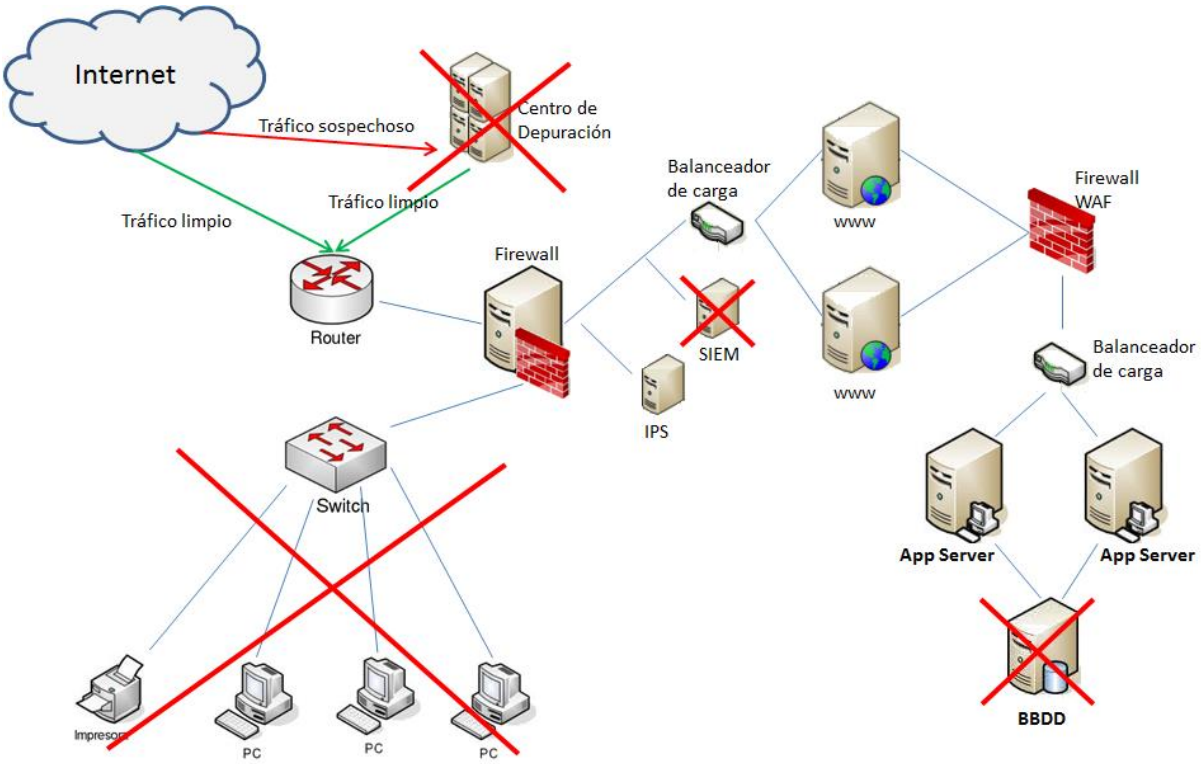

Figura 7. Ejemplo de arquitectura de red de defensa de PYME con recomendaciones anti DDoS sin elementos comerciales

Resumiendo, la arquitectura con sistemas de protección que se ha implementado para realizar el estudio, y que se basa en software de código abierto queda de la siguiente manera:

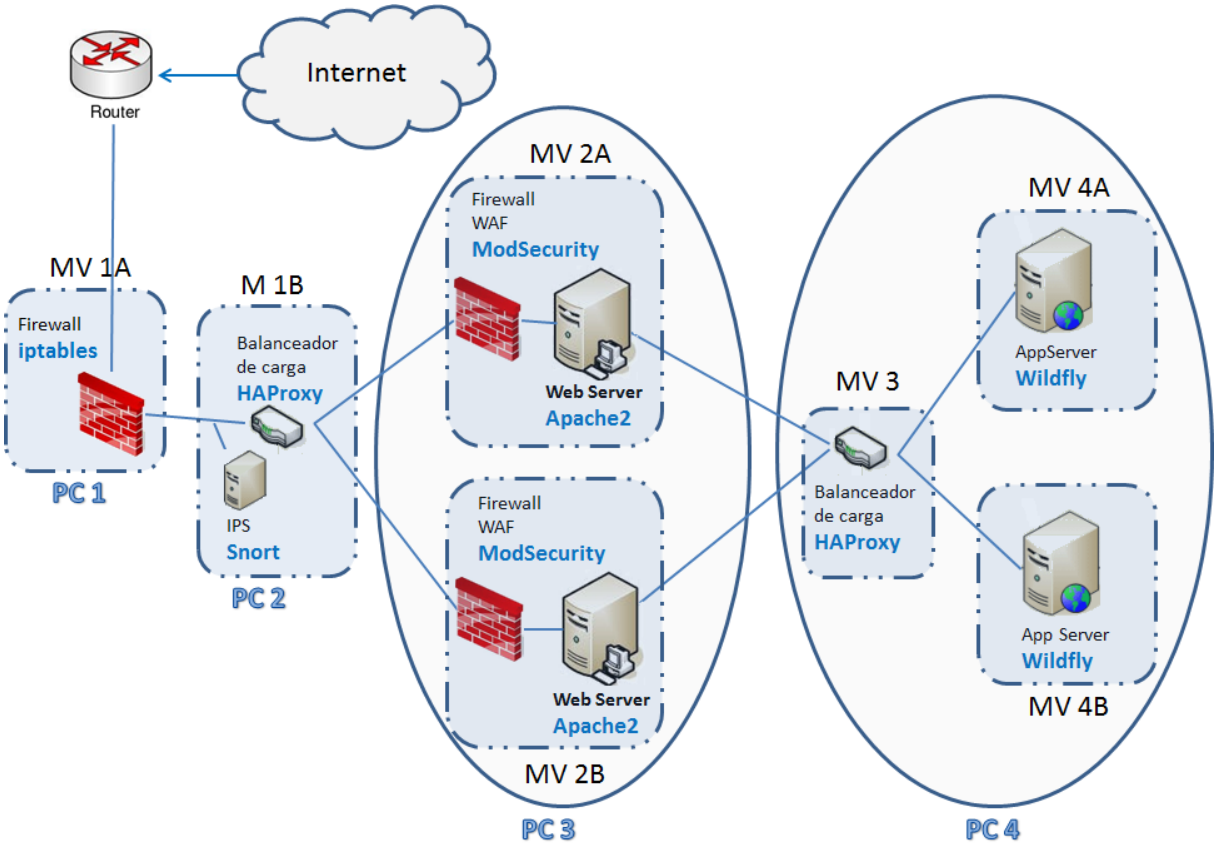

Figura 8. Arquitectura implementada para el estudio

Como ya se ha comentado, el entorno de pruebas se basa en una arquitectura con una mínima inversión en hardware y una gran cantidad de elementos software gratuitos. Analizando bien las dos figuras anteriores se puede observar que se ha producido un cambio en la ubicación de los firewall WAF, que han subido un nivel en el diseño en profundidad del sistema.

En ambas arquitecturas, la idea es proporcionar únicamente contenido estático en los servidores web, y meter en un nivel más profundo los servidores de aplicaciones y las bases de datos. Los ataques DDoS a los que se puede ver sometido un servidor de contenido estático son mucho más limitados que los que se pueden producir contra un servidor de aplicaciones. De hecho, un firewall como IPtables con unas reglas bien definidas, junto con una adecuada configuración del balanceador de carga, pueden evitar ataques HTTP del tipo Low and Slow contra el servidor web. Situar el WAF delante del servidor web puede reducir el rendimiento del sistema convirtiéndose en un cuello de botella, ya que deberá controlar no sólo el acceso a las aplicaciones sino también los accesos a contenido estático. Este es el motivo por el cual en el diseño comercial, los WAF se situaron justo delante de los servidores de aplicaciones.

Sin embargo, el software libre utilizado para el desarrollo de la investigación, limita en algunos aspectos su arquitectura. En este sentido, por ejemplo, la elección del servidor web ha sido Apache, que tiene un módulo WAF que se integra directamente con él y es muy popular por sus prestaciones. Dicho módulo, se encontraba en fase experimental para los servidores JBoss, Tomcat y Weblogic en diciembre de 2014[1], pero no está disponible en Internet en estos momentos.

En la figura siguiente se puede observar cómo se han introducido todos estos elementos en el diseño anterior, sin disponer, por ejemplo, de equipos con doble tarjeta de red.

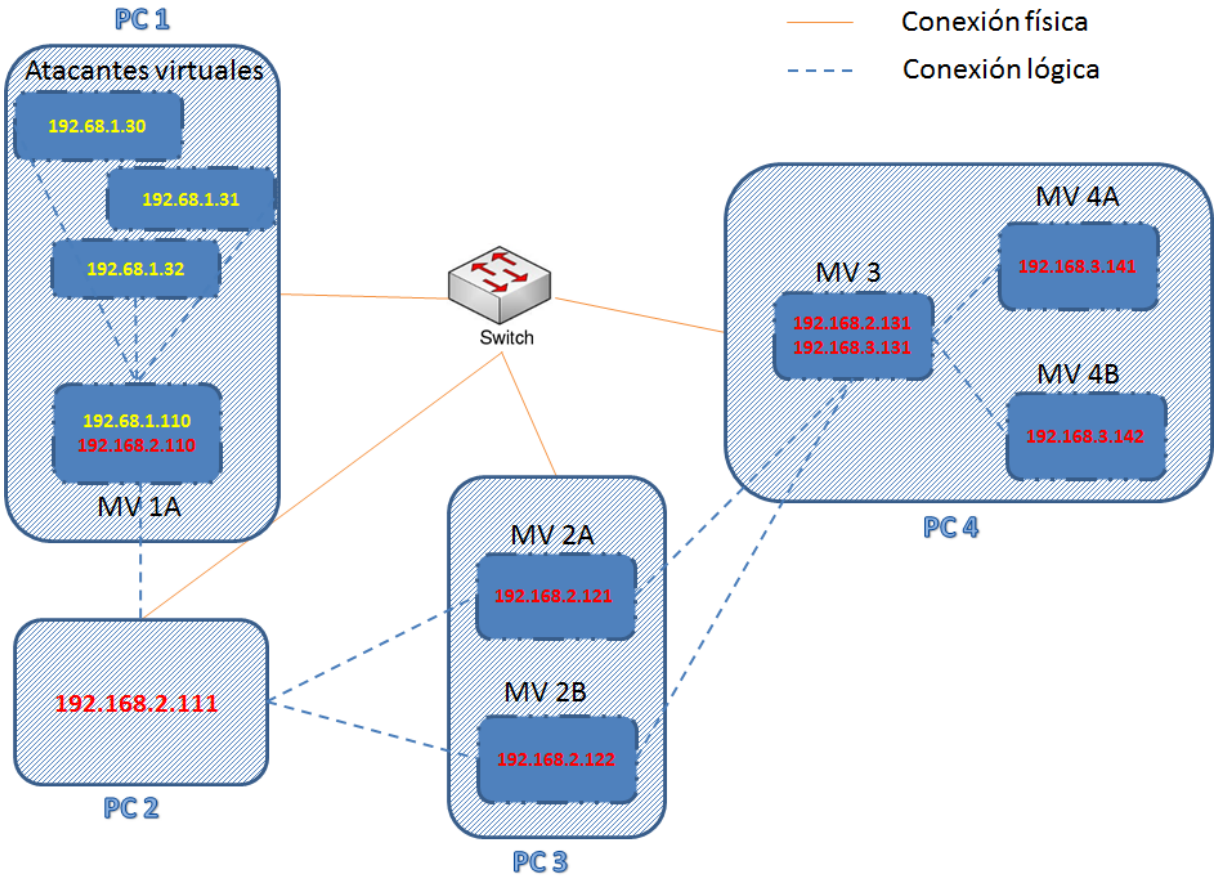

Figura 9. Arquitectura física de la PYME protegida

- MV1A – Firewall enrutador (IPtables), punto de acceso a la red interna de la empresa
- PC2 – Balanceador de carga HAProxy
- MV2A y MV2B – Servidores Web Apache2
- MV3 – Balanceador de carga HAProxy
- MV4A y MV4B – Servidores de aplicaciones Wildfly

[1] http://www.elarraydejota.com/dos-modulos-de-seguridad-esenciales-para-apache-mod_security-y-mos_qos/

3.4.2. Arquitectura de PYME sin protección

Su instalación en el entorno de pruebas se basa en la *Figura 2. Ejemplo de arquitectura de red de PYME desprotegida*, en la que se puede observar que los servidores web y de aplicaciones se conectan directamente al punto de acceso al exterior. Para simular este diseño, se ha utilizado PC3 en el que se han instalado un servidor Apache2 y un Wildfly en dos máquinas virtuales diferentes. De este modo, los recursos disponibles para ella en el momento de los ataques son los similares a los recursos empleados para el funcionamiento simultáneo de los dos servidores Apache2 de la arquitectura de PYME protegida.

4. Pruebas y Análisis de Resultados

4.1. Estrategias de mitigación incorporadas

Existen herramientas comerciales, tanto HW como SW que ofrecen una protección muy eficaz contra ataques de denegación de servicio, pero su coste muchas veces no puede ser asumido por empresas pequeñas. Por este motivo, el presente estudio se centra en herramientas gratuitas/open source que les permitan conseguir una protección básica anti DDoS tratando de incurrir en un gasto mínimo o incluso, prácticamente inexistente.

El software utilizado por este estudio para proteger la PYME simulada es el siguiente:

- **Iptables**, utilizado en este caso como firewall y enrutador:
  Herramienta cortafuegos, disponible en el núcleo de Linux, que permite entre otras cosas, el filtrado de tráfico, la redirección de paquetes con traducción de direcciones de red (NAT) para IPv4, y la configuración del nivel de verbosidad que se desea aplicar a su log.
  La configuración de IPtables aplicada en este estudio se puede ver en el *Anexo 3.* Configuración de IPtables Un análisis en profundidad de las reglas incluidas para la protección del sistema, mostrará que se ha aplicado en todo momento una política muy restrictiva. Esto es así por varios motivos:
    - Las limitaciones HW y de rendimiento hacen que el límite máximo de conexiones permitidas para las pruebas, deba ser mucho menor que el de un entorno real de producción. En el caso de estudio, por ejemplo, se han limitado a 20 conexiones por IP.
    - Al haber eliminado la red de empleados de la arquitectura de la PYME, por no formar parte del estudio, se han podido incluir también reglas muy restrictivas en cuanto a las conexiones ICMP y UDP, por ejemplo. Por no ser necesarias para nada, por aquellos usuarios externos de la empresa que se conectan a su web para operar con ella.

  En el archivo de configuración, también se puede observar que se han incluido la gran mayoría de las reglas de filtrado en la tabla MANGLE, en lugar de utilizar la tabla FILTER como suele ser habitual. Esto es por una cuestión de rendimiento, muy importante si se pone este software justo a la entrada del sistema. Si no se optimiza

la velocidad de procesamiento del cortafuegos, se podrá convertir en un cuello de botella y, por tanto, será el primero en bloquearse durante un ataque DDoS.

La cuestión es, ¿Por qué este cambio de tabla mejora la respuesta del firewall ante un ataque? La respuesta es sencilla: La cadena INPUT de la tabla FILTER, comúnmente utilizada para todo este tipo de filtrado de paquetes, se procesa después de las cadenas PREROUTING y FORWARD, y sus reglas sólo se aplican en caso de que no hayan coincidido previamente con alguna de ellas dos. Esto produce un retraso en el filtrado de los paquetes y, por tanto, un mayor consumo de recursos.

Es muy importante filtrar los paquetes maliciosos lo antes posible y esto se hace desde la cadena PREROUTING, no soportada por la tabla INPUT pero sí por la tabla MANGLE. (Constantin Oesterling, JavaPipe, 2016) [14]

Como último comentario respecto a este software, parece interesante señalar que actualmente existe un proyecto de Netfilter, para la sustitución de iptables por nftables, actualmente en desarrollo, que en principio será más completa, más sencilla e intuitiva y con tablas y cadenas totalmente configurables.[2]

- **Haproxy**, instalado para balancear la carga de los servidores Apache y Wildfly:

  Los balanceadores de carga son dispositivos hardware o software que se encargan de repartir el volumen de datos a procesar entre varios sistemas semejantes, de manera que todos ellos puedan ofrecer el mismo servicio de la forma más eficiente posible, mejorando el rendimiento y la respuesta al usuario, y todo esto, de manera completamente transparente a él. Además, en caso de producirse un bloqueo o caída de alguno de los equipos o servidores, estos dispositivos permitirán enviar toda la carga a aquél o aquellos que se encuentren funcionando correctamente hasta que se produzca la recuperación del que ha fallado, ofreciendo de esta forma la capacidad de redundancia de los sistemas.

  Entre las opciones de configuración que ofrece, se encuentra el tiempo máximo de recepción de mensajes completos HTTP, de forma que el software descarta automáticamente aquellos mensajes que sobrepasan el límite establecido. Así el sistema tiene una protección más contra ataques DDoS del tipo Low and Slow. [15].

  Este tipo de herramientas suelen ser de coste bastante elevado. Sin embargo, HAProxy, es un software gratuito, de código abierto, muy rápido y fiable, especialmente recomendado para sitios web con una alta carga de procesamiento y volumen de tráfico.

[2] https://es.wikipedia.org/wiki/Nftables#Diferencias_entre_nftables_e_iptables

- **ModSecurity** Web Application Firewall (WAF)

  Los WAF son un tipo de firewall diseñados para controlar el acceso a aplicaciones o servicios web. La diferencia fundamental con los firewall tradicionales es que estos últimos operan sobre las capas de red y transporte del modelo OSI (3 y 4), mientras que los primeros, lo hacen sobre la capa de aplicación (7).

  Es sabido por todos que es imposible diseñar y codificar aplicaciones 100% seguras y libres de vulnerabilidades. Errores de diseño o fallos de implementación en el software son habituales por muy controlado que esté el proceso de desarrollo. En algunos casos, las vulnerabilidades pueden ser detectadas y eliminadas antes de subir el sistema a producción; sin embargo, en muchas ocasiones, no se detecta la vulnerabilidad o no es posible arreglarla sin hacer un cambio en el diseño, haciendo necesario su uso a pesar de las deficiencias de seguridad encontradas. Por este motivo es muy importante el uso de un WAF, que permita configurar reglas para evitar que las vulnerabilidades del sistema puedan ser explotadas.

  ModSecurity es un firewall de aplicaciones web de código abierto que se ejecuta como un módulo más del servidor Apache, para proteger al sistema contra ataques del tipo XSS, SQL Injection, LFI, etc. En realidad se trata de un motor de detección y prevención de intrusiones para aplicaciones web que, instalado junto con **ModEvasive**, permite además proteger a los servidores web contra ataques DDoS de fuerza bruta.

  El funcionamiento de ModSecurity es a través de reglas que se establecen a partir de tres parámetros de configuración:

  - Variables: Se indican los parámetros de la aplicación que se van a controlar.
  - Operadores: Expresiones regulares que determinan las reglas de filtrado.
  - Acciones: Respuesta del sistema en caso de que se cumpla alguna de las condiciones previamente establecidas.

  Este módulo de Apache contiene un conjunto de reglas precargadas por defecto, que ofrecen protección básica contra ataques comunes. [16]

  OWASP ha diseñado un conjunto de reglas que se encuentran en continuo desarrollo y actualización para adaptarse a las nuevas amenazas.[3] La lista de reglas aplicadas para el filtrado de paquetes se encuentra en el Anexo 4. Reglas de configuración de ModSecurity.

[3] http://www.elarraydejota.com/dos-modulos-de-seguridad-esenciales-para-apache-mod_security-y-mos_qos/

- **IDS Snort**

  Sistema de Detección de Intrusos (IDS) basado en red, de código abierto, para plataformas Unix y Windows. Este tipo de sistemas trata de detectar cualquier intento de comprometer la seguridad de una red, en base a un conjunto de reglas con las que se definen los patrones de monitorización del sistema y que pueden ser predefinidas, o determinadas por el administrador de la herramienta. Estos patrones de ataque, almacenados en una base de datos, son los que permiten la detección de actividades anómalas en el sistema.

  Se ha desarrollado el pluging Snortsam permite la utilización de Snort como IPS (Sistema de Prevención de Intrusos).

4.2. Mecanismo de medición de las pruebas

El método de medición de la resistencia de los sistemas (des)protegido ante ataques de denegación de servicio está basado en un script, que se mantiene en ejecución durante todo el ataque, y que lanza peticiones *wget* al servidor web cada segundo. Dicho script, que se ejecuta en una máquina virtual ubicada en la misma red que los atacantes para simular que se trata de un cliente legítimo conectado a través de Internet, permite comprobar el tiempo de descarga del contenido web durante todo el transcurso de la prueba. El archivo de log generado muestra si ha habido retrasos en la transferencia de datos o denegaciones de servicio al cliente.

4.3. Tipo de ataque: TCP SYN Flood

Se trata de un tipo de ataque basado en explotar vulnerabilidades del mecanismo de establecimiento de conexión TCP de negociación en tres pasos (three way handshake), en el que el cliente envía un mensaje TCP SYN al servidor para indicarle que quiere iniciar una conexión con él. El servidor, responde con un paquete TCP SYN/ACK y queda a la espera de la confirmación por parte del cliente mediante la recepción de un mensaje TCP ACK. A partir de este momento, la conexión entre ambas partes ha sido establecida correctamente.

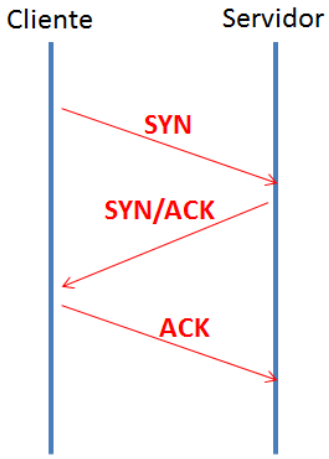

Figura 10. Establecimiento de conexión TCP

Realizar un ataque del tipo TCP SYN Flood, es tan sencillo como enviar solicitudes de conexión TCP SYN a un servidor y no responder nunca a los mensajes TCP SYN/ACK enviados por él. De esta forma, el atacante consigue consumir los recursos de la víctima, alcanzando el número máximo de conexiones TCP pendientes.

En la actualidad, este tipo de ataque se realiza falsificando la dirección IP de la máquina que origina la petición de establecimiento de conexión. De esta forma, el atacante se asegura de que la víctima enviará su paquete TCP SYN/ACK a máquinas que no habrán tratado de establecer ninguna comunicación con ellas, y que, por tanto, descartarán el mensaje sin enviar ninguna respuesta a la víctima. La gran variedad de direcciones IP empleadas en el ataque, impide basar la defensa de la víctima en el bloqueo de direcciones a través de listas negras.

El grupo activista Anonymous, conocido por sus ataques de denegación de servicio a nivel mundial, ha utilizado, y sigue empleando herramientas basadas en TCP SYN Flood para lanzar ataques contra gobiernos, empresas, etc.

Entre los mecanismos de defensa que se pueden emplear para evitar o mitigar su impacto se encuentran los siguientes:

- Filtrado de paquetes.
- Reducción del tiempo de espera del TCP/ACK.
- Sobre escritura de conexiones Half Opened.
- SYN Cache/SYN Cookies.

- Bloquear la IP que produce la inundación (sólo en caso de no haber IP Spoofing).

De los mecanismos de protección/prevención mencionados, la arquitectura de PYME protegida incluye el filtrado de paquetes con iptables

4.3.1. Ejecución de la prueba

Para lanzar un ataque de este tipo se ha utilizado el comando hping3 que permite la falsificación de direcciones IP y el envío masivo de paquetes a la dirección IP y puerto establecidos como argumentos:

```
Hping3 –S IP ---flood --rand-source –d 5000 –p 80
```

Lista de argumentos empleados en el ataque:

- S: Envío de paquetes SYN.
- IP: Dirección IP sobre la que se lanza el ataque.
- flood: Se produce una inundación mediante el envío continuo de paquetes.
- rand-source: Falsificación de dirección IP origen mediante datos aleatorios.
- d: Tamaño del campo de datos de los paquetes.
- p: Puerto al que se dirige el ataque.

Los ataques han sido ejecutados desde máquinas virtuales con Kali Linux:

- En el caso de la PYME sin protección, se han realizado dos ataques con configuraciones diferentes:
  - En el primero, el atacante se encontraba en un host distinto al de los servidores Apache y Wildfly.
  - El segundo ataque, se ha realizado desde una MV situada en el mismo host que los servidores desprotegidos.
- El ataque a la PYME protegida se ha realizado siguiendo el diseño de la Figura 9. Arquitectura física de la PYME protegida; es decir, las MV encargadas de lanzar el ataque, se encontraban físicamente alojadas en el mismo host que el firewall-enrutador, nexo de unión entre la red interna y el exterior.

4.3.2. Resultados

4.3.2.1. Mediciones anteriores a la inclusión de las estrategias de mitigación

Como se ha comentado en el apartado anterior, en este caso se han hecho dos pruebas distintas: La primera con el atacante y el script de control en un host distinto a los servidores web y de aplicaciones; y la segunda, con el atacante en el mismo host que los servidores y el script de control en uno diferente.

El motivo de lanzar dos ataques en diferentes configuraciones contra la PYME desprotegida es que en el primer caso, se ha producido una caída total y reinicio automático de la máquina que hospedaba a los servidores. Además, en el fichero de salida del script de control se observa que, desde el mismo momento en el que se lanzó el ataque, el servidor web dejó de responder a las peticiones realizadas por el cliente legítimo. A continuación, unos segundos más tarde, las peticiones comenzaron a ser rechazadas con dos mensajes de error diferentes:

- No route to host.
- Command terminated by signal 2.

Con estos datos, es más que evidente que el ataque ha sido satisfactorio y el sistema no ha ofrecido ninguna resistencia. Sin embargo, el mensaje de error en el que se indica que no hay ruta al host, puede indicar que el motivo por el que la denegación ha sido tan rápida, puede ser la sobrecarga de la tarjeta de red del host alojador de los servidores, que no ha sido capaz de gestionar tantos paquetes entrantes. Por este motivo, se lanza el segundo ataque con cambio en la disposición de las máquinas virtuales.

En esta segunda prueba, se ubica el atacante en la misma máquina que los servidores web y de aplicaciones. Como dicha máquina ha sido seleccionada en la arquitectura para alojar, en principio solo dos MV, y el servidor Wildfly no tiene ninguna implicación en esta prueba, se elimina temporalmente del escenario para mejorar las condiciones de rendimiento y procesamiento del host. En resumen, se tiene el script de control lanzando peticiones cada segundo en un host distinto a aquél que aloja el servidor web y al atacante.

Los resultados obtenidos en este caso no son tan alarmantes como en el caso anterior, porque, al menos, no existe reinicio de la máquina que hospeda a los servidores. Sin embargo, también en este caso, se observa que una vez lanzado el ataque el servidor web no es capaz de responder a ninguna de las peticiones del script de control.

Esta segunda prueba se mantiene durante unos minutos, y cuando se para el ataque, el servidor se recupera y empieza a responder a las peticiones enviadas por el script de control. Se observa en el servidor web el siguiente mensaje de error:

- CPU stuck for 22s

4.3.2.2. Mediciones posteriores a la inclusión de las estrategias de mitigación

Para esta prueba se tiene completamente configurada la arquitectura de PYME basada en el diseño de la Figura 9. Arquitectura física de la PYME protegida. La única diferencia, que será igual en el resto de las pruebas, es que de las tres máquinas virtuales configuradas fuera de la red interna, dos de ellas se utilizarán para lanzar los ataques y una siempre se encargará del script de control.

En este caso se lanza el ataque de forma simultánea desde dos MV diferentes, es decir, se duplica el volumen de paquetes enviados a la PYME. Los resultados obtenidos muestran que las protecciones incluidas son capaces de frenar el ataque en todo su alcance, es decir, durante los cinco minutos que se ha mantenido, no se ha observado ninguna denegación de servicio y, mucho más importante, ningún retardo de respuesta al cliente legítimo.

Se comprueba en los logs que IPtables está frenando el ataque, el HAproxy está balanceando la carga correctamente entre los dos servidores Apache, y ninguno de los servidores web muestra mensajes de error. Por tanto, la arquitectura diseñada para el estudio funciona y consigue frenar, al menos, este tipo de ataques.

4.3.2.3. Análisis Comparativo

De las pruebas realizadas anteriormente se puede resumir lo siguiente:

- Un ataque TCP SYN Flood contra un servidor desprotegido no sólo es muy fácil de ejecutar, sino que además consigue agotar los recursos del sistema prácticamente desde el mismo momento en el que se lanza, consiguiendo muy rápidamente el objetivo de denegación de servicio.
- Las protecciones diseñadas para la PYME consiguen frenar el ataque al 100%, sin perjuicio alguno en el rendimiento del sistema.

4.4. Tipo de Ataque: TCP Connect Flood

Este tipo de ataque trata de establecer una gran cantidad de conexiones con la víctima para conseguir superar su límite máximo de conexiones o su ancho de banda. Es un tipo de ataque que no permite hacer IP spoof, es muy ineficiente y deja mucha información en los logs. Sin embargo, es importante estar protegido contra él porque es el método por defecto utilizado por herramientas como LOIC.

A pesar de ser un tipo de ataque con tantos inconvenientes, si el hacker que quiere lanzarlo consigue diseñar una BOTNET, infectar una gran cantidad de equipos y lanzar el ataque de forma simultánea desde todos ellos, es muy probable que consiga su objetivo.

Para prevenir o evitar ataques de este tipo, se debe limitar el número máximo de conexiones permitidas por una misma dirección IP, además, para aumentar un poco más la resistencia del sistema, se puede limitar la cantidad máxima de conexiones establecidas por segundo.

4.4.1. Ejecución de la prueba

Para lanzar este tipo de ataque, la primera idea fue utilizar herramientas como LOIC. Sin embargo, para que LOIC pueda ser ejecutada utilizando una red de bots, es necesario conectarla a Internet (o disponer de una red propia con gran cantidad de equipos) de forma que se pueda hacer uso de las máquinas “voluntarias” que forman parte de su red. Por otra parte, para realizar un ataque de denegación de servicio de este tipo, sería necesario pedir permiso al ISP al que se encuentra conectada la arquitectura, ya que podría llegar a verse afectado por el ataque según su alcance. Por último, parece que a pesar de lanzar el ataque contra mi mismo sistema y suponiendo que las máquinas que forman parte de la red de LOIC lo hacen de manera voluntaria, la legalidad de su uso es dudosa, por lo que se ha optado por descartar esta y otras herramientas que se ofertan en Internet para realizar pruebas de “estrés” a los sistemas (HOIC, Ufonet…)

El comando empleado para la ejecución de esta prueba es nping:

```
Nping --tcp-connect –rate=90000 –c 900000 –q IP
```

Los argumentos empleados para la ejecución del comando son los siguientes:

- tcp-connect: Para indicar el tipo de prueba que se va a realizar.
- rate: número de paquetes por segundo.
- c: número de conexiones que se abren contra la víctima.
- q: Para reducir la verbosidad (la cantidad de información que se muestra).
- IP: dirección IP de la víctima (o url).

Los recursos utilizados para realizar esta prueba son los siguientes:

- En el caso de la PYME desprotegida:
  - Se lanzan en el mismo host dos MV, una de ellas con el servidor Apache, y la otra, con un único atacante. La máquina Wildfly, en principio no es necesaria.
  - En un host diferente, se lanza una segunda MV Kali Linux en la que se ejecuta el script de control de estado del servidor.
- Para el ataque a la PYME protegida, cuyo diseño se basa en la Figura 9. Arquitectura física de la PYME protegida, se han lanzado tres máquinas virtuales en el host en el que se encuentra el enrutador que da acceso a la red interna:
  - Dos de ellas son MV Kali Linux, encargadas de lanzar el ataque.
  - Una tercera MV Kali Linux con el script de control de estado del servidor.

En la prueba a la PYME sin protección, se decide ubicar al atacante en el mismo host que la víctima para evitar las limitaciones impuestas por el hardware de los equipos (el switch y los adaptadores de red). Como se puede ver en el párrafo anterior, en el caso de la PYME protegida, los atacantes y el primer punto de acceso al entorno de la empresa se encuentran ubicados en el mismo host, por lo que las condiciones de ataque serán similares.

4.4.2. Resultados

4.4.2.1. Mediciones anteriores a la inclusión de las estrategias de mitigación

En esta prueba, se ha mantenido el script de ataque durante casi 5 minutos. En este tiempo no se han podido enviar los 900.000 paquetes para los que se había configurado la prueba. El número máximo de peticiones de conexión enviadas ha sido de 390.757. El resultado, en este caso ha sido el siguiente:

- Atacante:
  - Conexiones establecidas: 134.192
  - Conexiones fallidas: 256.565

- Porcentaje total de error de conexión: 65,66%
- RTT medio: 0,008ms

- Script de control:
  - En cuanto se lanza el ataque, se empiezan a observar retardos en la descarga de la página web. Unos pocos segundos más tarde, se muestra el siguiente mensajes de error:
    - Read error (Connection reset by peer)… **DoS**
  - El script no consigue descargar el contenido web hasta que el ataque ha finalizado.

4.4.2.2. Mediciones posteriores a la inclusión de las estrategias de mitigación

El ataque lanzado de forma simultánea desde dos MV diferentes (duplicando así su potencia), ha enviado 900.000 intentos de conexión desde cada una de ellas, y ha tardado en ejecutarse 58 segundos en la máquina desde la que se ha lanzado primero y 63 segundos en la otra MV (al ser un lanzamiento manual siempre una comienza el ataque antes que la otra, teniendo más recursos disponibles al principio). Los resultados obtenidos son los siguientes:

- Atacante 1:
  - Conexiones establecidas: 3227
  - Conexiones fallidas: 896.773
  - Porcentaje total de error de conexión: 99,64%
  - RTT medio: 0,542ms
- Atacante 2:
  - Conexiones establecidas: 3427
  - Conexiones fallidas: 896.573
  - Porcentaje total de error de conexión: 99,62%
  - RTT medio: 0,551ms
- Script de control:
  - Cuando ambos ataques se encuentran activos, se pueden observar algunos pequeños retardos (de un máximo de cuatro segundos para esta prueba) en las peticiones legítimas.
  - No se observa ninguna denegación de servicio por parte de la PYME.

4.4.2.3. Análisis Comparativo

Analizando los resultados obtenidos, se puede concluir que las protecciones incluidas en la arquitectura minimizan el impacto del ataque, evitando al usuario legítimo las denegaciones de servicio a las que se ha visto expuesto el cliente de la prueba sin protección.

Sin embargo, como se comentó durante la descripción de este tipo de ataque, para que la prueba fuera completa sería necesario diseñar una BOTNET con la que se lanzara este mismo ataque de forma simultánea, desde muchas más máquinas que las utilizadas en el estudio.

Por tanto, la conclusión es que la configuración de las herramientas incluidas permiten resistir ataques TCP Connect en el que se implica un número pequeño de atacantes, pero se desconoce el impacto que tendría un ataque de este tipo lanzado desde herramientas como LOIC, que emplean mayor cantidad de máquinas.

4.5. Tipo de Ataque: UDP Flood

En este caso, la víctima recibe paquetes UDP con un rango muy grande de direcciones IP origen falsificadas, o reales (con la dirección IP del atacante) y a una velocidad muy alta. El ataque consume el ancho de banda disponible y los recursos de la red (routers, firewalls, IPS/IDS, etc), hasta conseguir su caída. Como UDP es un protocolo no orientado a conexión, los ataques UDP son difíciles de detectar y extremadamente efectivos, y pueden sobrecargar una red utilizando direcciones IP origen fijas o aleatorias.

Para evitar estos ataques se pueden incluir varios tipos de medidas defensivas:

- Configuración del firewall para rechazar cualquier conexión UDP desde el exterior.
- Configuración del firewall para que se rechacen las conexiones UDP a todos los puertos excepto a los estrictamente necesarios. Para aquellos puertos que se mantienen abiertos, se configura el firewall para que se rechacen los paquetes si la tasa de transferencia es superior a la establecida en la configuración. (configuración muy restrictiva, utilizada para el entorno de pruebas)
- Otro método es la identificación de las direcciones IP que están ejecutando el ataque, para proceder a bloquearlas, en caso de no haber falsificación de dirección IP.

4.5.1. Ejecución de la prueba

Para la realización de esta prueba, se ha vuelto a utilizar el comando hping3:

```
hping3 IP --udp --flood --rand-source
```

Lista de argumentos empleados en el ataque:

- IP: Dirección IP a la que se envía el ataque.
- udp: Tipo de paquetes que se van a enviar.
- flood: Se produce una inundación mediante el envío continuo de paquetes.
- rand-source: Falsificación de dirección IP origen mediante datos aleatorios.

Los recursos utilizados para realizar esta prueba son los siguientes:

- En el caso de la PYME desprotegida:
  - Se lanzan en el mismo host dos MV, una de ellas con el servidor Apache, y la otra, con un único atacante
  - En un host diferente, se lanza una MV con el script de control de estado del servidor.
- Para el ataque a la PYME protegida, se han utilizado tres máquinas virtuales en el host en el que se encuentra el enrutador que da acceso a la red interna:
  - Dos de ellas son MV Kali Linux, encargadas de lanzar el ataque.
  - Una tercera MV Kali Linux con el script de control de estado del servidor.

4.5.2. Resultados

4.5.2.1. Mediciones anteriores a la inclusión de las estrategias de mitigación

El ataque se ha mantenido durante más de cuatro minutos. En todo este tiempo no se ha observado ninguna denegación de servicio del lado del cliente legítimo. Es el propio kernel del servidor web el que filtra los paquetes producidos por la inundación. Es una configuración segura por defecto, con los valores predeterminados del servidor Ubuntu 14.04 utilizado.

Para no sobrecargar al servidor web, se ha lanzado *tcpdump* para verificar si los paquetes UDP estaban o no entrando en el sistema. La información ofrecida por la herramienta indica que el **96,63%** de los paquetes UDP fueron descartados por el filtro.

Como se ha comentado en la descripción de este tipo de ataque, su propósito es agotar los recursos de red, y teniendo ambas máquinas alojadas en el mismo host, conseguir este objetivo es muy complicado. Se podría reproducir la prueba lanzando el ataque desde otro host, pero el resultado no podría ser comparado con un ataque similar ejecutado contra la PYME protegida, por las limitaciones HW del entorno de pruebas.

4.5.2.2. Mediciones posteriores a la inclusión de las estrategias de mitigación

Se realiza una prueba similar a la anterior, en la que se lanza únicamente un atacante en una MV ubicada en el mismo host que el punto de acceso a la red interna. En este caso, la información ofrecida por *tcpdump* muestra que el kernel ha descartado el **98,19%** de los paquetes UDP. Una cifra algo superior a la obtenida en la prueba anterior.

Se repite la prueba lanzando el ataque desde dos MV corriendo en el mismo host. El resultado, es algo mejor que el anterior. El resumen de paquetes recibidos y descartados por el kernel mostrado por la herramienta *tcpdump* indica que se han descartado casi el **99.4%** de los paquetes UDP.

Igual que en el caso de la PYME sin protección, para concluir con la prueba, se lanza el ataque desde dos máquinas y se mantiene durante 5 minutos. Para evitar sobrecargar el punto de acceso a la red interna, no se ejecutará *tcpdump.* Lo que se pretende en esta prueba es comprobar si el sistema aguanta el ataque durante este tiempo y si se producen o no retardos relevantes en la descarga web del cliente conectado al servidor. Con esto se observa que en algunos momentos la velocidad de transferencia de datos disminuye un poco, pero no se producen retrasos relevantes en la descarga del contenido web y en ningún caso se produce denegación de servicio.

4.5.2.3. Análisis Comparativo

La primera conclusión que se puede sacar de esta prueba es que, a pesar de poder simular un ataque distribuido, gracias a la opción de falsear las direcciones IP origen que ofrece la

herramienta hping3, el hardware disponible no permite un análisis en profundidad de los resultados.

Sin embargo, a pesar de las limitaciones, los resultados de las pruebas muestran una mejora de casi un 3% en los resultados obtenidos al aplicar los sistemas de protección al servidor web, debido a las reglas incluidas en IPtables para el filtrado de este tipo de inundación.

4.6. Tipo de Ataque: UDP Fragmentation Flood

Variante de UDP flood que consiste en el envío de paquetes muy grandes (de 1500 Bytes) con los que se consume mayor ancho de banda con menor cantidad de mensajes; los paquetes enviados remiten voluntariamente a otros paquetes que nunca se envían, por lo que no pueden ser re-ensamblados por la víctima, que consume todos los recursos de su CPU tratando de hacerlo. Con esto se consigue una sobrecarga de los procesos y el reinicio del servidor.

La mejor medida para evitar este tipo de ataque es el filtrado de paquetes, incluido dentro de las reglas de configuración de IPtables para la PYME protegida.

4.6.1. Ejecución de la prueba

Para la realización de esta prueba, se ha vuelto a utilizar el comando hping3:

```
hping3 IP -f --udp --flood --rand-source
```

Lista de argumentos empleados en el ataque:

- IP: Dirección IP a la que se envía el ataque.
- f: Envío de paquetes fragmentados.
- udp: Tipo de paquetes que se van a enviar.
- flood: Se produce una inundación mediante el envío continuo de paquetes.
- rand-source: Falsificación de dirección IP origen mediante datos aleatorios.

Los recursos utilizados para realizar esta prueba son los siguientes:

- En el caso de la PYME desprotegida:
  - Se lanzan en el mismo host dos MV, una de ellas con el servidor Apache, y la otra, con un único atacante
  - En un host diferente, se lanza una MV con el script de control de estado del servidor.
- Para el ataque a la PYME protegida, se han utilizado tres máquinas virtuales en el host en el que se encuentra el enrutador que da acceso a la red interna:
  - Dos de ellas son MV Kali Linux, encargadas de lanzar el ataque.
  - Una tercera MV Kali Linux con el script de control de estado del servidor.

4.6.2. Resultados

4.6.2.1. Mediciones anteriores a la inclusión de las estrategias de mitigación

Como este ataque es muy similar al anterior, la prueba se ha realizado lanzando el ataque durante menos de un minuto y controlando con *tcpdump* los paquetes UDP recibidos y procesados por el servidor web. Igual que en el caso anterior, a pesar de no haberse incluido protección específica en el servidor, el propio kernel se ha encargado del filtrado de paquetes, descartando el **96,07%** de los paquetes UDP recibidos.

4.6.2.2. Mediciones posteriores a la inclusión de las estrategias de mitigación

La primera prueba se ejecuta con un único atacante de la misma forma que las pruebas realizadas con ataques del tipo UDP flood. El resultado es que en este primer caso, el **98,38%** de los paquetes UDP enviados al punto de acceso de la PYME fueron descartados.

Se ejecuta una segunda prueba con dos atacantes y se observa que se han descartado el **99,62%** de los paquetes.

La tasa de paquetes descartados por el kernel aumenta precisamente porque el volumen del ataque es mayor. Es decir, el sistema está continuamente recibiendo peticiones UDP procedentes del switch y de otras máquinas. Estos paquetes legítimos, son también contabilizados en las estadísticas, por tanto, cuanto más fuerte sea el ataque menor será la cantidad proporcional de paquetes lícitos que llegan al sistema y por tanto, estadísticamente la cantidad de paquetes eliminados será mayor.

4.6.2.3. Análisis Comparativo

Se observa una leve mejoría de la respuesta de la PYME protegida respecto de la que se encuentra configurada sin protección alguna. Pero igual que en el caso anterior, sería necesario un escenario en el que los atacantes se alojaran en máquinas distintas a los servidores y se conectaran a través de otros equipos físicos para poder comparar con mayor fiabilidad la respuesta de ambos entornos ante ataques de este tipo.

4.7. Tipo de Ataque: ICMP Flood

En este tipo de ataque, la víctima recibe gran cantidad de paquetes ICMP (ping) falsificados con una gran variedad de direcciones IP origen; en un intento de responder a todas estas peticiones (pong), la víctima consume los recursos de la red y su ancho de banda.

Para evitar este tipo de ataque la principal medida de protección que se puede aplicar es el filtrado de paquetes:

- Configuración del firewall para rechazar conexiones ICMP.

Las peticiones ping, sirven para conocer si los sistemas se encuentran o no en funcionamiento, por ello, se emplean para el control de la red interna de la empresa. Sin embargo, de puertas a fuera, es una funcionalidad innecesaria para los clientes, y que por tanto, puede ser eliminada mediante la configuración del firewall. Esta es la configuración que se ha implantado en la PYME protegida.

4.7.1. Ejecución de la prueba

En este caso, de nuevo se utiliza el comando hping3:

```
hping3 IP --icmp --flood --rand-source
```

Lista de argumentos empleados en el ataque:

- IP: Dirección IP a la que se envía el ataque.
- icmp: Tipo de paquetes que se van a enviar durante el ataque.
- flood: Se produce una inundación mediante el envío continuo de paquetes.

- rand-source: Falsificación de dirección IP origen mediante datos aleatorios.

Los recursos utilizados para realizar esta prueba son los mismos que en las pruebas realizadas para los tipos de ataque de los apartados anteriores

4.7.2. Resultados

4.7.2.1. Mediciones anteriores a la inclusión de las estrategias de mitigación

En cuanto se lanza el ataque, la tasa de transferencia de datos al cliente legítimo cae rápidamente, aumentando un poco el tiempo de espera hasta que, unos segundos más tarde, se deja de recibir respuesta. No sólo se produce una denegación de servicio sino que, además, el sistema tarda mucho en recuperarse una vez el ataque ha finalizado.

4.7.2.2. Mediciones posteriores a la inclusión de las estrategias de mitigación

El ataque se lanza simultáneamente desde dos MV y se mantiene en ejecución durante unos minutos. El script de control del servidor web no detecta ninguna anomalía en la descarga del contenido. No se producen retardos ni denegaciones de servicio.

4.7.2.3. Análisis Comparativo

En este caso, el sistema pasa de denegar el servicio en cuestión de segundos tras el inicio del ataque, a mantener un funcionamiento normal como si no estuviera pasando nada. La configuración establecida hace que todos los paquetes ICMP enviados para atacar al sistema, sean descartados evitando perjuicio alguno para los usuarios reales del sistema.

4.8. Tipo de Ataque: Slow HTTP DDoS

Es un conjunto de ataques a la capa de aplicación a los que precisamente el servidor Apache es muy vulnerable. Básicamente se trata de establecer conexiones HTTP con el servidor web y mantenerlas abiertas durante el mayor tiempo posible, enviándole peticiones incompletas.

El protocolo HTTP ha sido diseñado para que el servidor no procese ninguna petición recibida hasta que esté completa. Por este motivo, si nunca se terminan las peticiones HTTP

o la tasa de transferencia es muy baja, el servidor va llenando poco a poco el pool de conexiones, hasta que llega a su punto de saturación y no puede aceptar ninguna petición más.

Dentro de este tipo de ataque existen algunas variantes, que se pueden lanzar desde una misma herramienta, slowhttptest[4]:

- Slow Headers, tipo de ataque en el que se envían las cabeceras HTTP incompletas (sin el CRLF final), lo que impide al servidor terminar de establecer la conexión con el cliente, manteniendo una gran cantidad de conexiones abiertas.
- Slow HTTP Post Body, basado en el envío de peticiones POST con la cabecera HTTP completa, indicando un content-length con el tamaño en bytes del POST DATA a enviar. A continuación se envían menos bytes de los indicados, dejando al servidor en espera.
- Apache Killer, en este ataque se envían muchas peticiones superponiendo rangos de bytes de la cabecera HTTP, agotando recursos de memoria y CPU del servidor
- Slow Read, tiene la particularidad de que los atacantes establecen sesiones TCP válidas con la víctima y solicitan la descarga de algún documento u otro objeto de gran tamaño. Cuando el servidor comienza a enviar los paquetes, los atacantes retardan al máximo el envío de los ACKs, de forma que parezca que el servidor trabaja en una red muy lenta.

Durante el proceso de configuración del entorno protegido, se hicieron pruebas antes de establecer las reglas de filtrado de IPtables. Con estas pruebas se pudo observar que la resistencia del sistema ante estos ataques había mejorado notablemente únicamente con la instalación del balanceador de carga y la duplicidad de los servidores web.

Por tanto, entre los mecanismos de protección que se pueden incluir en un sistema para evitar ataques de este tipo están:

- Reglas de filtrado para limitar por un lado el máximo número de conexiones abiertas por IP, y por otro el número máximo de conexiones establecidas por segundo.
- Duplicado de servidores y balanceo de carga.
- Configuración del balanceador de carga para descartar los paquetes que no se han completado en un tiempo determinado. En este caso, 5 segundos.

[4] Hackplayers.com/2016/06/ataques-dos-slow-http-mediante-SlowHTTPTest.html

4.8.1. Slow Headers

4.8.1.1. Ejecución de la prueba

El ataque se ha realizado ejecutando el siguiente comando, desde una o varias MVs Kali Linux:

```
slowhttptest –c 5000 –i 10 –r 200 –t GET –u http://IP –x 24 –p 3
```

Los argumentos empleados para la ejecución del ataque son los siguientes:

- c: Número de conexiones
- i: Intervalo en segundos
- r: Número de conexiones por segundo
- t: Método HTTP (GET/POST/HEAD)
- u: Dirección IP o URL a la que se dirige el ataque
- x: tamaño máximo de los datos de seguimiento
- p: timeout por conexión

Los ataques se han lanzado siguiendo la misma configuración que en las pruebas anteriores.

4.8.1.2. Resultados

4.8.1.2.1. Mediciones anteriores a la inclusión de las estrategias de mitigación

Según la información mostrada por la herramienta de ataque, 10 segundos después de lanzarlo, el servicio deja de estar disponible. Este dato coincide con el ofrecido por el script de control de la conexión en el que se observa, al mismo tiempo, que la siguiente petición realizada por el cliente legítimo, no obtiene respuesta del servidor hasta que no finaliza el ataque.

4.8.1.2.2. Mediciones posteriores a la inclusión de las estrategias de mitigación

Se lanzan contra el sistema protegido dos ataques: El primero con un único atacante y el segundo con dos. Los resultados son los siguientes:

- A los 5 segundos de lanzar el ataque, el atacante ya muestra que el servicio no se encuentra disponible, porque ha alcanzado el máximo de 20 conexiones establecidas para una única IP. El script de control funciona correctamente durante todo el ataque, sin retardos ni denegaciones de servicio.
- En el segundo caso, el resultado es el mismo. Cada uno de los atacantes mantiene 20 conexiones abiertas, pero el pool de conexiones del servidor es mucho mayor, por lo que su rendimiento no se ve afectado por el ataque.

4.8.1.2.3. Análisis Comparativo

No es necesario realizar un análisis en profundidad para determinar que las reglas de filtrado de IPtables protegen al sistema de ataques de tipo *low and slow.* Se ha visto fácilmente que en el caso de la PYME desprotegida en unos pocos segundos, el servidor deja de dar servicio al cliente.

Además de la protección de IPtables, el sistema incluye una configuración en HAproxy que descarta las peticiones que no han sido completadas en 5 segundos.

Los datos de configuración establecidos (20 conexiones por IP y 5 segundos para la recepción del paquete completo) pueden ser demasiado restrictivos en la práctica. Sin embargo, se han elegido estas cifras debido a las limitaciones físicas del entorno de pruebas.

4.8.2. Slow Read

4.8.2.1. Ejecución de la prueba

El ataque se ha realizado ejecutando el siguiente comando, desde una o varias MVs Kali Linux:

Slowhttptest –c 4090 –X –r 200 –w 512 –y 1024 –n 5 –z 32 –k 3 –u http://IP –p 3

Los argumentos empleados para la ejecución del ataque son los siguientes:

- c: Número de conexiones.
- X: El modo de ejecución de la herramienta es Slow Read
- r: Número de conexiones por segundo
- w: Especifica el inicio del tamaño de la ventana
- y: Especifica el fin del tamaño de la ventana para la prueba
- n: Intervalo en segundos para operaciones de lectura.
- z: Cantidad de bytes leídos por el buffer en cada operación read()
- k: Número de veces que el recurso va a ser solicitado por cada socket.
- u: Dirección IP o URL a la que se dirige el ataque
- p: Intervalo de espera de respuestas del serviro HTTP antes de declararlo fuera de servicio.

Los ataques se han lanzado siguiendo la misma configuración que en las pruebas anteriores.

4.8.2.2. Resultados

4.8.2.2.1. Mediciones anteriores a la inclusión de las estrategias de mitigación

Tras lanzar el ataque, el servidor deja de responder al script de control; el servicio no deja de estar disponible para el atacante y para el usuario normal. Al principio se observan retrasos en la respuesta del servidor web, pero unos segundos más tarde el servidor deja de enviar cualquier tipo de respuesta hasta que finaliza el ataque.

En cuanto finaliza el ataque, el sistema se recupera rápidamente, y comienza a responder a las peticiones de descarga del script de control del sistema.

4.8.2.2.2. Mediciones posteriores a la inclusión de las estrategias de mitigación

El sistema de protección funciona en este caso de la misma forma que en el apartado anterior para la PYME protegida. A pesar de haber realizado el ataque no desde un MV sino desde dos, y por tanto, duplicando su volumen, el cliente legítimo en ningún momento se entera de que se está produciendo un ataque.

Los atacantes consiguen establecer 20 conexiones cada uno, y a partir de ese momento perciben que el servicio deja de estar disponible.

4.8.2.2.3. Análisis Comparativo

En este caso, igual que en la prueba Slow Headers la diferencia en cuanto al comportamiento de ambos sistemas es más que evidente. La PYME desprotegida no resiste el ataque, perdiendo la disponibilidad del sistema hasta la finalización del mismo. En caso de la arquitectura con protecciones, los usuarios legítimos no se enteran en ningún momento de lo que está sucediendo.

4.8.3. Apache killer

4.8.3.1. Ejecución de la prueba

El ataque se ha realizado ejecutando el siguiente comando, desde una o varias MVs Kali Linux:

```
Slowhttptest –R –u http://IP –t HEAD –c 5000 –a 10 –b 3000 –r 500
```

Los argumentos empleados para la ejecución del ataque son los siguientes:

- R: Modo de ejecución Range Header.
- u: Dirección IP o URL a la que se dirige el ataque.
- t: Método HTTP (GET/POST/HEAD).
- c: Número de conexiones.
- a: Valor de inicio del rango de ataque.

- b: límite en bytes del rango de ataque especificado.
- r: Número de conexiones por segundo

Los ataques se han lanzado siguiendo la misma configuración que en las pruebas anteriores.

4.8.3.2. Resultados

4.8.3.2.1. Mediciones anteriores a la inclusión de las estrategias de mitigación

Se lanza el escenario y se observa que a los 28 segundos finaliza con indicación de que no quedan conexiones disponibles. Sin embargo, a pesar de lo que se pudiera pensar inicialmente, el cliente legítimo observa pequeños retrasos en la respuesta del servidor web, pero en ningún momento se produce una denegación de servicio.

4.8.3.2.2. Mediciones posteriores a la inclusión de las estrategias de mitigación

Se lanzan contra el sistema protegido dos ataques: El primero con un único atacante y el segundo con dos. En ambos casos, la herramienta de ataque informa de que la prueba ha finalizado en el segundo 2, porque no quedan conexiones abiertas, es decir, el servidor no le permite abrir nuevas conexiones porque el atacante ya ha alcanzado su máximo. Por tanto, la protección del sistema contra este ataque es un éxito, y el cliente legítimo no se ve afectado en modo alguno por el ataque.

4.8.3.2.3. Análisis Comparativo

En ninguno de los casos se ha producido una denegación de servicio. En el caso de la PYME desprotegida se han observado retrasos prácticamente inapreciables, con lo que se puede concluir que este ataque no ha funcionado contra ninguno de los dos entornos.

4.8.4. Slow http post body

4.8.4.1. Ejecución de la prueba

El ataque se ha realizado ejecutando el siguiente comando, desde una o varias MVs Kali Linux:

```
Slowhttptest –c 1000 –B –i 110 –r 200 –s 8192 –t FAKEVERB –u http://IP –x 10 –p 3
```

Los argumentos empleados para la ejecución del ataque son los siguientes:

- c: Número de conexiones
- B: Tipo de ataque modo POST
- i: Intervalo en segundos
- r: Número de conexiones por segundo
- s: Tamaño de la cabecera
- t: Método HTTP
- u: Dirección IP o URL a la que se dirige el ataque
- x: Tamaño máximo de los datos de seguimiento
- p: Timeout del servidor

Los ataques se han lanzado siguiendo la misma configuración que en las pruebas anteriores.

4.8.4.2. Resultados

4.8.4.2.1. Mediciones anteriores a la inclusión de las estrategias de mitigación

Se lanza el escenario, al principio se empiezan a observar pequeños retrasos en la descarga del contenido del script de control. Unos segundos después, el cliente deja de obtener respuesta del servidor hasta que finaliza el ataque y el servidor se recupera. Se ha conseguido llegar al estado de denegación de servicio.

4.8.4.2.2. Mediciones posteriores a la inclusión de las estrategias de mitigación

Igual que en pruebas anteriores, la herramienta de ataque muestra un mensaje de error que indica, a los 5 segundos de ser lanzada, que no quedan conexiones abiertas. Cinco segundos más tarde, finaliza la ejecución del supuesto atacante, sin afectar, en absoluto al rendimiento obtenido por el script de control en la descarga de contenidos.

4.8.4.2.3. Análisis Comparativo

En este caso, igual que en la prueba Slow Headers y Slow Read la diferencia en cuanto al comportamiento de ambos sistemas vuelve a hacerse evidente. De nuevo el entorno desprotegido no resiste el ataque, perdiendo la disponibilidad del sistema hasta la finalización del mismo. En caso de la arquitectura con protecciones, los usuarios legítimos no se enteran en ningún momento de lo que está sucediendo.

5. Trabajo Futuro

Respecto al diseño planteado para la protección de la PYME, se ha dejado sin instalar el IDS Snort que debería ir ubicado en la misma máquina que el balanceador que se encuentra conectado por un lado al firewall y por otro a los servidores Apache. De todos los elementos del diseño, Snort es un sistema de protección pasivo, ya que se encarga de la detección de comportamientos anómalos, y requiere de un profundo conocimiento sobre el funcionamiento normal del entorno, y en particular de su red, que permita al administrador ajustar las reglas lo suficientemente bien como para evitar tener un número demasiado elevado de falsos positivos.

Como se ha comentado en este mismo documento, Snort cuenta ahora con un módulo llamado SnortSam que lo convierte en IPS, por lo que pasa de ser una herramienta pasiva de detección a un sistema activo que permite el filtrado y bloqueo de paquetes en la red. Sin embargo, a pesar de ser una herramienta muy popular y muy útil para proteger redes privadas, el resto de herramientas incorporadas a la arquitectura son suficientes para el desarrollo del trabajo.

Una de las limitaciones encontradas durante el estudio, ha sido la incompatibilidad de los módulos de apache QoS y ReqTimeOut con la versión actual del servidor. De hecho, ambos módulos fueron instalados y probados, pero no sólo no mejoraban la respuesta del sistema ante los ataques sino que, además, empeoraban su rendimiento. Uno de los objetivos de estos módulos es precisamente proteger al servidor web de ataques de denegación de servicio dejando de aceptar conexiones cuando su pool se encuentra en el límite establecido por el administrador, permitiendo ajustar dinámicamente el ancho de banda dedicado a cada conexión o establecimiento de Keep Alive, permitiendo así al sistema aguantar mejor un ataque distribuido.

Por otro lado, sería muy interesante disponer de un laboratorio de pruebas real (como el de una universidad, por ejemplo) para poder crear una BOTNET que permita ejecutar ataques distribuidos en lugar de simularlos, permitiendo así probar de forma más exhaustiva la configuración establecida para la protección de PYME. Además, si se dispusiera de un entorno con mayor cantidad de máquinas, se podría realizar también la siguiente prueba:

- Ataque amplificado smurf/fraffle para los protocolos ICMP y UDP con la herramienta hping y el siguiente comando:

- *hping3 BcastIP –a IPvictima –(icmp/udp) --flood*

  Lo que se conseguiría ejecutando la herramienta hping3 con esta configuración sería enviar a la dirección broadcast de la red peticiones icmp/udp falsificando la dirección IP origen, de forma que el resto de máquinas de la red recibieran su supuesta solicitud de ping y enviaran sus respuestas pong (en caso de ICMP).

En el tiempo que ha durado el estudio se ha podido encontrar una vulnerabilidad explotable a Wildfly para un ataque DDoS. Para conseguir la denegación de servicio de este servidor sería necesario instalar una aplicación con una vulnerabilidad explotable en sí misma que permitiera hacer pruebas en profundidad al firewall WAF, que se ha instalado en el sistema.

6. Conclusiones

Con las pruebas realizadas se ha demostrado que con una correcta configuración de IPtables se pueden bloquear gran cantidad de ataques distribuidos. La duplicidad de los servidores aumenta la resistencia del sistema ante un ataque, dando tiempo a los administradores para detectarlo y frenarlo, mediante la incorporación de nuevas reglas al firewall o simplemente bloqueando las direcciones IP desde las que se envían los ataques.

Una buena definición de reglas, lo más ajustada posible a las necesidades del entorno, mantener los equipos siempre actualizados, duplicar o incluso triplicar los servidores que almacenan mucho contenido y que van a estar expuestos a una alta carga de conexiones, balancear su carga, controlar las conexiones al sistema y tener firewalls de aplicación que impidan la explotabilidad de las vulnerabilidades, son fundamentales para tener un sistema de información lo más robusto posible ante intentos de ataque de denegación de servicio.

Sin embargo, todas las medidas incorporadas en el sistema no servirán de nada en caso de recibir un ataque de reflexión o amplificado. Por muy buena que sea la configuración del firewall de acceso al sistema, un ataque de este tipo agotará sus recursos casi de manera inmediata. Para evitar esto, el único método existente es dirigir el tráfico a un centro de depuración que impida que dichos ataques lleguen a la empresa.

Hasta la fecha, las empresas pequeñas no han sido nunca objeto de ataques amplificados, ya que son bastante complejos y no los puede llevar a cabo cualquiera. Este tipo de ataques los lanzan normalmente organizaciones cuyos componentes tienen grandes conocimientos técnicos y que no se sienten atraídos por pequeñas organizaciones que no les van a proporcionar beneficio o reconocimiento alguno.

En cuanto a las pruebas realizadas en el estudio, en el caso de los ataques Low and Slow, no está completamente demostrado que con la configuración actual ante el ataque de una BOTNET, el sistema pudiera resistir sin perjuicio alguno para sus usuarios reales. Se ha demostrado que si la red de atacantes no es muy grande, el sistema resistirá con las herramientas empleadas, pero ¿Qué pasaría en caso de ser un ataque mayor? En principio, los mejores elementos para proteger al servidor Apache en estos casos son los módulos ReqTimeOut y QoS, pero actualmente no funcionan con la última versión estable de Apache2. La ventaja que ofrecen estos módulos es que permiten a los administrados definir la cantidad máxima de conexiones abiertas permitidas por el sistema, de forma que cuando

se alcance el valor establecido, se vayan cerrando las más antiguas para permitir la entrada de las nuevas. De esta forma el pool de conexiones del sistema nunca se vería sobrepasado. Al no disponer de estos dos módulos, se ha incluido en el balanceador de carga, una regla que descarta las conexiones de paquetes que no se han completado en 5 segundos, aumentando así la resistencia del sistema.

Con todo esto, se puede concluir que existen herramientas gratuitas, a disposición de todo el mundo, fácilmente configurables, que permiten mejorar la respuesta de los sistemas de información de una PYME expuesta a internet con una inversión mínima (la del hardware necesario para alojar el software gratuito). Con la configuración propuesta, se pueden bloquear muchos de los ataques de denegación de servicio existentes, o mantener en pie al sistema, mientras se lucha contra ellos.

7. Referencias

[1] Kaspersky Lab (2015). Global IT Security Risks Survey 2015. Recuperado de www.kaspersky.com

[2] Rosa Martín (2016). Los ataques DDoS, una amenaza para la empresa española. Recuperado de newsbook.portalinformatico.com

[3] Kaspersky Lab (2015). Denial of Service: How Businesses Evaluate the Threat of DDOS Attacks. Recuperado de www.kaspersky.com

[4] David Bisson (2016). DDoS gang takes down BBC websites, Donald Trump's campaign site over holiday weekend. Recuperado de www.grahamcluley.com

[5] Mohit Kumar (2015). GitHub hit by Massive DDoS Attack From China. Recuperado de thehackernews.com

[6] Dann Albright (2015). 2014's Final Controversy: Sony Hack, The Interview & North Korea. Recuperado de www.makeuseof.com

[7] John Kennedy (2014). The Sony empire strikes back – unleashes DDoS attacks against pirates. Recuperado de www.siliconrepublic.com

[8] Sputnik News (2014). Los ciberataques más famosos del siglo XXI. Recuperado de mundo.sputniknews.com

[9] Contribuciones (2013). Cómo CyberBunker atacó a Spamhaus y casi se llevó a medio Internet por delante. Recuperado de www.securitybydefault.com

[10] Graham Cluley (2011). CIA website brought down by DDoS attack, LulzSec hackers claim responsibility. Recuperado de nakedsecurity.sophos.com

[11] Lindsay Fortado (2012). 'Anonymous' Hackers Convicted of Targeting PayPal, MasterCard. Recuperado de www.bloomberg.com/europe

[12] Security Team (2016). Akamai's state of the internet / security Q4 2015 Report. Recuperado de www.akamai.com

[13] Arbor Networks (2016). Worldwide Infrastructure Security Report (Volume XI). Recuperado de www.arbornetworks.com

[14] Constantin Oesterling (2016). IPtables DDoS Protection: The Best Rules to Mitigate DDoS Attacks. Recuperado de javapipe.com

[15] Baptiste Assmann (2016). What is a slow POST Attack and how to turn HAProxy into your first line of Defense? Recuperado de blog.haproxy.com

[16] Fernando Catoira (2013). Web application firewall: Cómo proteger su aplicación web con ModSecurity. Recuperado de www.welivesecurity.com/la-es/

[17] Sam Hobbs (2016). Getting Started with Apache ModSecurity on Debian and Ubuntu. Recuperado de samhobbs.co.uk

[18] Alvin Wan (2015) How To Set Up mod_rewrite for Apache on Ubuntu 14.04. Recuperado de www.digitalocean.com

[19] John R. Vacca (2013) Computer and Information Security Handbook, 2nd Edition. Morgan Kaufmann.

[20] Saravanan kumarasamy y Dr.R.Asokan (2011). Distributed Denial Of Service (DSoS) Attacks Detection Mechanism

[21] Saman Taghavi Zargar, James Joshi y David Tipper (2013). A Survey of Defense Mechanisms Against Distributed Denial of Service (DDoS) FloodingAttacks

[22] Patricia Bachmaier (2016). El 68% de las empresas españolas expuestas a ataques DDoS. Recuperado de cso.computerworld.es

8. Anexos

8.1. Anexo 1. Tipos de Ataques DDoS

A continuación se puede ver una tabla en la que se muestra una gran variedad de ataques de Denegación de Servicio y para cada uno de ellos se indica el tipo de ataque del que se trata, la capa del modelo OSI a la que va dirigido, si existe o no IP Spoofing, a velocidad a la que se deben transmitir los datos, recomendaciones de defensa para evitar o minimizar el impacto del ataque, y por último, una descripción de mismo.

Nombre del Ataque	Capa OSI	Tipo de Ataque	IP Origen	Rango de IPs Origen	Velocidad de Transmisión	Tipo de Defensa	Explicación
ICMP Flood	L3	Ancho de banda	Falsa	Muy Grande	Muy Alta	Filtrado de paquetes	La víctima recibe gran cantidad de paquetes ICMP falsificados con una gran variedad de direcciones IP origen, consumiendo los recursos de la red y su ancho de banda.
ICMP Fragmentation Flood	L3	Recursos	Falsa	Media	Muy Alta	Filtrado de paquetes	Los atacantes envían paquetes ICMP falsificados, de gran tamaño (1500 Bytes), que no pueden ser reensamblados y a una velocidad muy alta. El servidor víctima agota los recursos de su CPU tratando de recomponer los paquetes recibidos.
ICMP Echo Request Flood / Ping Flood	L3	Ancho de Banda/ Recursos de red	Falsa	Muy Grande	Muy Alta	Filtrado de paquetes	Adaptación del ataque ICMP flood específico de aplicación, en el que la víctima recibe solicitudes de ping falsas (ICMP echo request) a gran velocidad con una gran variedad de direcciones IP origen. La víctima, en un intento de responder a todos los mensajes con paquetes (pong) con el mismo contenido que los recibidos, consume los recursos de la red y su ancho de banda.
Smurf / Fraggle	L3 / L4	Ancho de banda	Falsa	Media	Muy Alta	Filtrado de paquetes: - Configurar los equipos de la LAN y los routers para que no respondan a peticiones ICMP o broadcast. - Configurar los router para que no redireccionen paquetes enviados a la dirección broadcast.	Ataque amplificado que consiste en enviar una serie de ICMP Echo Requests (Ping) / paquetes UDP con una dirección origen IP falsa (la de la víctima) a la dirección broadcast de la red. De esta forma todos los equipos de la LAN responden enviando mensajes a la víctima.

Nombre del Ataque	Capa OSI	Tipo de Ataque	IP Origen	Rango de IPs Origen	Velocidad de Transmisión	Tipo de Defensa	Explicación
TCP Connect Flood	L4	Recursos / Ancho de banda	Real	Grande	Alta	- Limitar el número máximo de conexiones por IP - Limitar la cantidad máxima de conexiones por segundo	Establece una gran cantidad de conexiones con la víctima para agotar sus recursos o su ancho de banda. La forma en la que este ataque puede ser efectivo es a través de una red de BOTs que operen de forma simultánea contra el blanco
TCP Spoofed SYN flood	L4	Recursos	Falsa	Grande	Alta	- Filtrado de paquetes - Reducción del tiempo de espera del TCP/ACK - Sobre escritura de conexiones Half Opened - SYN Cache/SYN Cookies - Dispositivos de categorización de direcciones IP.	El atacante envía solicitudes de conexión TCP a la víctima con la dirección IP origen modificada, de forma que la víctima responda TCP/SYN-ACK a la dirección indicada y quede a la espera del paquete TCP/ACK de respuesta. Como el equipo con la dirección IP origen no ha solicitado la conexión, nunca responde a la víctima. De esta forma, se consumen los recursos del servidor que queda a la espera del paquete ACK, llegando al número máximo de conexiones pendientes.
TCP SYN-ACK flood	L4	Ancho de Banda / Recursos	Falsa	Grande	Alta	Filtrado de paquetes	Envío masivo de solicitudes de conexión TCP a un gran número de máquinas indicando la dirección de la víctima como origen de la conexión. Las respuestas TCP/SYN-ACK son enviadas a la víctima saturando su ancho de banda o sus recursos (memoria, CPU, etc).
TCP ACK & PUSH ACK flood	L4	Recursos	Falsa	Grande	Alta	Filtrado de paquetes	Envío masivo de paquetes ACK a la víctima que no pertenecen a ninguna de las conexiones de la lista del servidor. El servidor tratando de gestionar los paquetes consume sus recursos (memoria, CPU, etc) produciéndose una degradación o caída del sistema

Nombre del Ataque	Capa OSI	Tipo de Ataque	IP Origen	Rango de IPs Origen	Velocidad de Transmisión	Tipo de Defensa	Explicación
TCP Fragmented ACK	L4	Ancho de Banda	Falsa	Grande	Media	Filtrado de paquetes	Es una variación de TCP ACK & PUSH ACK Flood, que consiste en el envío de paquetes ACK de 1500 Bytes de tamaño para consumir el ancho de banda de la víctima. Como muchos sistemas de defensa no recomponen los paquetes en la capa IP, estos paquetes pueden atravesar routers, ACLs, firewalls, IPS e IDS afectando a todos los servidores de la red de la víctima.
RST / FIN flood	L4	Recursos	Falsa	Grande	Alta	Filtrado de paquetes	El servidor recibe gran cantidad de paquetes RST o FIN que no corresponden a ninguna sesión en curso. Los recursos de la víctima se agotan tratando de localizar las conexiones asociadas a dichos paquetes.
Synonymous IP	L4	Recursos	Falsa	Única	Alta	Filtrado de paquetes	La víctima recibe muchos paquetes TCP/SYN falsos con su dirección como IP origen e IP destino. Este ataque consume los recursos de la víctima al tratar de procesar esta irregularidad.
Fake Session	L4	Recursos	Falsa	Grande	Baja	Filtrado de paquetes	En este ataque se generan paquetes SYN falsificados, múltiples paquetes ACK y uno o varios paquetes FIN/RST por cada SYN. De esta forma, los paquetes aparentan formar parte de una misma conexión para que así se puedan saltar las herramientas de defensa que monitorizan únicamente el tráfico entrante. La baja tasa de paquetes SYN ACK hace que este ataque sea más difícil de detectar que el típico SYN flood, consiguiendo el mismo resultado.

Nombre del Ataque	Capa OSI	Tipo de Ataque	IP Origen	Rango de IPs Origen	Velocidad de Transmisión	Tipo de Defensa	Explicación
Session Attack	L4	Recursos	Real	Pequeña	Baja	Establecimiento de límite máximo de conexiones abiertas.	Se envían paquetes legítimos TCP/SYN desde una BOTNET a su víctima, cuando la víctima responde SYN-ACK los equipos de la BOTNET retardan el envío del ACK dejando la sesión abierta y lo mandan justo antes de que se ejecute en el servidor el time out de la sesión. De esta forma se consigue agotar los recursos de la víctima.
UDP flood	L4	Ancho de Banda	Falsa	Muy Grande	Muy Alta	- Deshabilitar la respuesta “ICMP Host Unreachable” - Filtrado de paquetes mediante firewall	La víctima recibe paquetes UDP falsificados, con un rango muy grande de direcciones IP origen, a una velocidad muy alta. El ataque consume el ancho de banda disponible y los recursos de la red (routers, firewalls, IPS/IDS, etc), hasta conseguir su caída. Como UDP es un protocolo no orientado a conexión, los ataques UDP son difíciles de detectar y extremadamente efectivos, y pueden sobrecargar una red utilizando direcciones IP origen fijas o aleatorias y se pueden diseñar contra un servidor específico.
Non spoofed UDP flood	L4	Ancho de Banda	Real	Pequeño	Muy Alta	- Deshabilitar la respuesta “ICMP Host Unreachable” - Filtrado de paquetes mediante firewall	En este ataque se envían de forma masiva paquetes UDP no falsificados a la víctima, consumiendo los recursos de su red y su ancho de banda hasta que se cae. En este caso, la dirección IP origen de los paquetes es la real, y habrá tantas como equipos formando parte del ataque.

Nombre del Ataque	Capa OSI	Tipo de Ataque	IP Origen	Rango de IPs Origen	Velocidad de Transmisión	Tipo de Defensa	Explicación
UDP fragmentation flood	L4	Recursos	Falsa	Moderada	Muy Alta	Filtrado de paquetes	Variante de UDP flood que consiste en el envío de paquetes muy grandes (de 1500 Bytes) con los que se consume mayor ancho de banda con menor cantidad de mensajes; los paquetes enviados remiten voluntariamente a otros paquetes que nunca se envían, por lo que no pueden ser reensamblados por la víctima, que consume todos los recursos de su CPU tratando de hacerlo. Con esto se consigue una sobrecarga de los procesos y el reinicio del servidor.
Misused Application	L7	Recursos	Real	Pequeña	Variable	Limitar el número máximo de conexiones al sistema	El atacante redirige clientes legítimos de una aplicación con mucho tráfico, por ejemplo, un servicio peer-to-peer, al servidor de la víctima. La víctima, incapaz de gestionar todo el tráfico entrante, agota sus recursos lo que deriva en una degradación o caída del servidor.
HTTP Fragmentation	L7	Recursos	Real	Pequeña	Muy Baja	- Limitar el número máximo de conexiones por IP - Limitar el tiempo máximo en que tarda en completarse el paquete	El atacante establece conexiones HTTP válidas con el servidor web víctima. Los paquetes HTTP legítimos son fragmentados en unidades muy pequeñas que son enviadas lo más despacio posible, justo antes de la expiración del timeout del servidor, manteniendo abiertas las conexiones durante mucho tiempo sin hacer saltar ninguna alarma y consiguiendo, finalmente, agotar los recursos disponibles del servidor.

Nombre del Ataque	Capa OSI	Tipo de Ataque	IP Origen	Rango de IPs Origen	Velocidad de Transmisión	Tipo de Defensa	Explicación
Excessive VERB (HTTP GET Flood)	L7	Recursos	Real	Pequeña	Alta	- Request/Response Behaviour Analysis - Client Request Limiting	Envío masivo de peticiones HTTP a un servidor web. Normalmente se envían peticiones GET y se dirigen a un proceso del servidor que produzca una gran carga en su CPU. Este ataque se puede realizar también con peticiones POST, HEAD, PUT, OPTIONS, etc.
Excessive VERB – Single Session	L7	Recursos	Real	Pequeña	Baja	- Request/Response Behaviour Analysis - Client Request Limiting	Este ataque se aprovecha de la característica de HTTP 1.1 que permite múltiples peticiones dentro de la misma sesión HTTP. El atacante puede limitar el número de sesiones y de esta forma saltarse las defensas de muchos sistemas de seguridad. Salvo por esta particularidad, el ataque es similar al anterior.
Multiple VERB – Single Request	L7	Recursos	Real	Pequeña	Muy Baja	- Request/Response Behaviour Analysis - Client Request Limiting	Variante del ataque Excessive VERB en la que los equipos atacantes envían múltiples peticiones HTTP dentro de un único paquete. Se produce una alta carga de CPU en la víctima a partir de una cantidad muy pequeña de paquetes.
Recursive GET	L7	Recursos	Real	Pequeña	Baja	- Request/Response Behaviour Analysis - Client Request Limiting	Otra variante del ataque Excessive VERB. El atacante identifica múltiples páginas y/o imágenes y genera peticiones HTTP GET que simulan navegar a través de ellas. Este ataque se puede combinar con otros métodos de ataque tipo VERB para hacerlo más difícil de detectar.

Nombre del Ataque	Capa OSI	Tipo de Ataque	IP Origen	Rango de IPs Origen	Velocidad de Transmisión	Tipo de Defensa	Explicación
Random Recursive GET	L7	Recursos	Real	Pequeña	Baja	- Request/Response Behaviour Analysis - Client Request Limiting	Variante de Recursive GET diseñado para foros y sitios de noticias que indexan numéricamente sus páginas, normalmente de forma secuencial. El ataque solicitará páginas diferentes utilizando números aleatorios dentro del rango válido de referencia.
Faulty Application / Specially Crafted packet	L7	Recursos	Real	Pequeña	Baja	WAF: - Overflows and Injections - Attack & Vulnerability Signatures	Los atacantes se aprovechan de los fallos de diseño e implementación de las aplicaciones web y de las bases de datos, explotando vulnerabilidades SQLI, HTTP, DNS, Buffer Overflow, Inyección de Código, etc, para provocar la caída del servidor o de la base de datos.
Slow Read	L7	Recursos	Real	Pequeña	Baja	- Request/Response Behaviour Analysis - Client Request limiting	Los atacantes establecen sesiones TCP válidas con la víctima y solicitan la descarga de algún documento u otro objeto de gran tamaño. Cuando el servidor comienza a enviar los paquetes, los atacantes retardan al máximo el envío de los ACKs, de forma que parezca que el servidor trabaja en una red muy lenta. De esta forma, se consumen los recursos del servidor.
DNS Amplificado	L7	Ancho de banda	Falsa	Única	Muy Alta	Centro de depuración	Se envían muchas consultas a diferentes DNS falsificando la dirección IP origen con la dirección de la víctima, consiguiendo, de esta forma, que todas las respuestas sean redirigidas a ella.

Nombre del Ataque	Capa OSI	Tipo de Ataque	IP Origen	Rango de IPs Origen	Velocidad de Transmisión	Tipo de Defensa	Explicación
DNS flood	L7	Ancho de banda / Recursos de red	Falsa	Muy Grande	Muy Alta	Filtrado de paquetes	Surge como una variante del ataque UDP flood. En este caso, la víctima es un servidor DNS que recibe solicitudes DNS falsas pero válidas de forma masiva y con una gran variedad de IPs origen. La víctima no puede determinar si una petición es real o no, por lo que responde a todas. De esta forma se consumen los recursos de la red y su ancho de banda.
VoIP flood	L7	Ancho de Banda / Recursos	Falsa	Muy Grande	Muy Alta	Filtrado de paquetes	También es una variante específica de aplicación del ataque UDP flood. La víctima es un servidor VoIP que recibe de forma masiva paquetes falsificados VoIP con una gran variedad de direcciones origen, lo que le obliga a consumir muchos recursos tratando de diferenciar conexiones reales y falsas.
NTP Amplificado	L7	Ancho de Banda	Falsa	Única	Muy Alta	Centro de depuración	El atacante envía peticiones NTP a diversos servidores suplantando la IP origen por la de la víctima, de forma que todos ellos le envíen sus respuestas
NTP Flood	L7	Ancho de Banda	Falsa	Muy Grande	Muy Alta	Filtrado de Paquetes	Otra variante del ataque UDP flood, en la que los atacantes envían paquetes válidos pero falsos de peticiones NTP a gran velocidad y desde una gran variedad de direcciones IP. Al parecer peticiones válidas, el servidor NTP trata de responder a todas ellas consumiendo gran cantidad de recursos de la red que agotan la infraestructura NTP hasta que se cae.

Tabla 1. Tipo de Ataques DDoS

8.2. Anexo 2. Glosario/Acrónimos

Acrónimo	Descripción
BOTNET	Red de robots
CIMS	Sistemas de Monitorización de Integridad del Contenido
DDoS	Denegación de Servicio Distribuido
DoS	Denegación de Servicio
IDS	Sistema de Detección de Intrusiones
IP	Internet Protocol
IPS	Sistema de Prevención de Intrusiones
ISP	Proveedor de Servicio de Internet
MV	Máquina Virtual
OSI	Open System Interconnection
OWASP	Open Web Application Security Project
PYME	Pequeña y Mediana Empresa
SI	Sistema de Información
SIEM	Security Information and Event Management
SO	Sistema Operativo
URL	Uniform Resource Locator
WAF	Web Application Firewall
XSS	Cross Site Scripting

8.3. Anexo 3. Configuración de IPtables

```
*filter
:INPUT DROP [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
:ICMP - [0:0]
:TCP - [0:0]
:UDP - [0:0]

# En la tabla mangle se prohíben todas las peticiones ICMP que vienen de eth1, (interfaz con Internet)
# y aquí, se permiten las procedente de eth0 (la red interna)
-A INPUT -p icmp -i eth0 -j ACCEPT

# Se permiten las conexiones a la cadena INPUT siempre y cuando vengan de la red interna
-A INPUT -i eth0 -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
-A INPUT -i lo -j ACCEPT
-A INPUT -m conntrack --ctstate INVALID -j DROP
-A INPUT -p tcp -m tcp --tcp-flags FIN,SYN,RST,ACK SYN -m conntrack --ctstate NEW -j TCP
-A INPUT -p icmp -m conntrack --ctstate NEW -j ICMP

# Por defecto, las conexiones que no han sido específicamente permitidas en el archivo de
# configuración, son descartadas. A continuación, se indica el modo en el que van a ser
# rechazados determinados intentos de conexión

# Se rechazan las conexiones UDP que no van a puertos abiertos
-A INPUT -p udp -j REJECT --reject-with icmp-port-unreachable

# Se rechazan las conexiones TCP a puertos cerrados
-A INPUT -p tcp -j REJECT --reject-with tcp-reset

# Se rechazan las conexiones a otros protocolos
-A INPUT -j REJECT --reject-with icmp-proto-unreachable

# Se aceptan las conexiones UDP limitando a un máximo de 10 conexiones por segundo
# y un total de 20 (por host origen)
-A INPUT -p udp -m conntrack --ctstate NEW -m limit --limit 10/s --limit-burst 20 -j ACCEPT
-A INPUT -p udp -m conntrack --ctstate NEW -j DROP

# Se establece un máximo de 20 conexiones abiertas por IP. Este valor deberá ser ajustado
# según las necesidades de la PYME y de sus clientes o usuarios. Sirve para proteger
# a la empresa contra ataques slow http
# En principio, se especifica el puerto 80 por ser el único abierto al exterior
-A FORWARD -p tcp --syn --dport 80 -m connlimit --connlimit-above 20 -j REJECT --reject-with tcp-reset
```

```
# Las peticiones TCP entrantes a través de la interfaz publica son redirigidas a la interfaz
# privada por el puerto 80, permitiendo también la salida de los paquetes de respuesta
-A FORWARD -i eth1 -o eth0 -p tcp -m tcp --dport 80 --tcp-flags FIN,SYN,RST,ACK SYN -m conntrack
      --ctstate NEW -j  ACCEPT
-A FORWARD -i eth1 -o eth0 -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
-A FORWARD -i eth0 -o eth1 -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
COMMIT

*raw
:PREROUTING ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
COMMIT

*nat
:PREROUTING ACCEPT [0:0]
:INPUT ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
:POSTROUTING ACCEPT [0:0]

# Aplicación de NAT a las direcciones entrantes/salientes
-A PREROUTING -i eth1 -p tcp -m tcp --dport 80 -j DNAT --to-destination 192.168.2.111
-A POSTROUTING -p tcp -m tcp -o eth0 --dport 80 -j SNAT --to-source 192.168.2.110
COMMIT

*security
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
COMMIT

*mangle
:PREROUTING ACCEPT [0:0]
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
:POSTROUTING ACCEPT [0:0]

# Bloquear todos los paquetes TCP que no son SYN y no pertenecen a ninguna conexión TCP abierta
-A PREROUTING -m conntrack --ctstate INVALID -j DROP

# Bloquear paquetes que se han saltado la regla anterior. Lo que se hace aquí en realidad es
# bloquear paquetes nuevos, que no pertenecen a una conexión abierta y no tienen el flag SYN
-A PREROUTING -p tcp ! --syn -m conntrack --ctstate NEW -j DROP

# Bloquear todos los paquetes que utilizan valores MSS (Maximum Segment Size) fuera del rango
# normal para paquetes SYN
-A PREROUTING -p tcp -m conntrack --ctstate NEW -m tcpmss ! --mss 536:65535 -j DROP
```

```
# Bloquear paquetes TCP que utilizan flags inadecuadas, que se encuentran fuera de las
# especificaciones normales de los sistemas.
# El primer bloque de flags indica los que se deben comprobar, y el segundo bloque, los que
# deben estar activos para que se cumpla la condición. Se explican como ejemplo las primeras dos reglas:
# En el siguiente caso, se descartan los paquetes que no tengan ningún flag activo
-A PREROUTING -p tcp --tcp-flags FIN,SYN,RST,PSH,ACK,URG NONE -j DROP
# A continuación se descartan los paquetes que vengan marcados como FIN y SYN
-A PREROUTING -p tcp --tcp-flags FIN,SYN FIN,SYN -j DROP
-A PREROUTING -p tcp --tcp-flags SYN,RST SYN,RST -j DROP
-A PREROUTING -p tcp --tcp-flags SYN,FIN SYN,FIN -j DROP
-A PREROUTING -p tcp --tcp-flags FIN,RST FIN,RST -j DROP
-A PREROUTING -p tcp --tcp-flags FIN,ACK FIN -j DROP
-A PREROUTING -p tcp --tcp-flags ACK,URG URG -j DROP
-A PREROUTING -p tcp --tcp-flags ACK,FIN FIN -j DROP
-A PREROUTING -p tcp --tcp-flags ACK,PSH PSH -j DROP
-A PREROUTING -p tcp --tcp-flags ALL ALL -j DROP
-A PREROUTING -p tcp --tcp-flags ALL NONE -j DROP
-A PREROUTING -p tcp --tcp-flags ALL FIN,PSH,URG -j DROP
-A PREROUTING -p tcp --tcp-flags ALL SYN,FIN,PSH,URG -j DROP
-A PREROUTING -p tcp --tcp-flags ALL SYN,RST,ACK,FIN,URG -j DROP

# Bloquear paquetes de subredes privadas. A la interfaz publica de la PYME solo le
# deben llegar peticiones con IPs públicas. Las demás son falsas (spoofed)
-A PREROUTING -s 224.0.0.0/3 -j DROP
-A PREROUTING -s 169.254.0.0/16 -j DROP
-A PREROUTING -s 172.16.0.0/12 -j DROP
-A PREROUTING -s 192.0.2.0/24 -j DROP
-A PREROUTING -s 10.0.0.0/8 -j DROP
-A PREROUTING -s 0.0.0.0/8 -j DROP
-A PREROUTING -s 240.0.0.0/5 -j DROP
-A PREROUTING -s 127.0.0.0/8 ! -i lo -j DROP

# Quedan exceptuadas las peticiones a través de la red interna, es decir, se permiten
# únicamente las peticiones procedentes de eth0 (red privada) y subred 192.168.0.0/16.
-A PREROUTING -s 192.168.0.0/16 -i eth1 -j DROP

# Bloquear ICMP, por ser irrelevante para los usuarios externos de la empresa. Se hace
# únicamente para las peticiones que entran desde la red pública, de forma que los
# equipos internos puedan verificar el correcto funcionamiento del resto de sistemas de la red
# En lugar de hacer reject y enviar un port unreachable, se hace drop para consumir menor
# cantidad de recursos del sistema
-A PREROUTING -p icmp -i eth1 -j DROP

# Bloquear paquetes fragmentados para evitar UDP fragmentation flood. Esto sólo será
# útil en caso de abrir algún puerto UDP para permitir conexiones de este tipo por algún
# motivo. Actualmente sería innecesario
-A PREROUTING -f -p udp -j DROP
COMMIT
```

8.4. Anexo 4. Reglas de configuración de ModSecurity

Se han incorporado las siguientes reglas de mod_security, configuradas por defecto:

```
modsecurity_35_bad_robots.data                  modsecurity_crs_23_request_limits.conf
modsecurity_35_scanners.data                    modsecurity_crs_25_cc_known.conf
modsecurity_40_generic_attacks.data             modsecurity_crs_30_http_policy.conf
modsecurity_42_comment_spam.data                modsecurity_crs_35_bad_robots.conf
modsecurity_50_outbound.data                    modsecurity_crs_40_generic_attacks.conf
modsecurity_50_outbound_malware.data            modsecurity_crs_41_sql_injection_attacks.conf
modsecurity.conf                                modsecurity_crs_41_xss_attacks.conf
modsecurity.conf-recommended                    modsecurity_crs_42_tight_security.conf
modsecurity_crs_10_ignore_static.conf           modsecurity_crs_43_csrf_protection.conf
modsecurity_crs_10_setup.conf                   modsecurity_crs_45_trojans.conf
modsecurity_crs_11_avs_traffic.conf             modsecurity_crs_47_common_exceptions.conf
modsecurity_crs_11_brute_force.conf             modsecurity_crs_47_skip_outbound_checks.conf
modsecurity_crs_11_dos_protection.conf          modsecurity_crs_48_local_exceptions.conf.example
modsecurity_crs_13_xml_enabler.conf             modsecurity_crs_49_header_tagging.conf
modsecurity_crs_16_authentication_tracking.conf modsecurity_crs_49_inbound_blocking.conf
modsecurity_crs_16_session_hijacking.conf       modsecurity_crs_50_outbound.conf
modsecurity_crs_16_username_tracking.conf       modsecurity_crs_55_application_defects.conf
modsecurity_crs_20_protocol_violations.conf     unicode.mapping
modsecurity_crs_21_protocol_anomalies.conf
pymeprot@pymemv2a:/etc/modsecurity$
```

Las reglas incorporadas pertenecen a los paquetes OWASP básicas y opcionales, además de las específicas anti denegación de servicio.

Se han eliminado de la configuración algunas de las reglas preestablecidas que provocaban falsos positivos durante el funcionamiento normal del sistema.