ISSN 0121–1129

Evaluación de ataques UDP Flood utilizando escenarios virtuales como plataforma experimental

UDP Inundation Attacks' Evaluation, Using Virtual Environment as an Experimental Platform

Fecha de recepción: 14 de agosto de 2011
Fecha de aprobación: 19 de octubre de 2011

Walter Fuertes*, Fernando Rodas**, Deyci Toscano**

Resumen

Los ataques por denegación de servicio (DoS) tienen como propósito imposibilitar el acceso a los servicios de una organización durante un periodo indefinido; por lo general, están dirigidos a los servidores de una empresa, para que no puedan ser accedidos por usuarios autorizados. El presente trabajo se enfoca en la evaluación de ataques DoS tipo UDP Flood, utilizando como plataforma de experimentación un entorno virtual de red que permite identificar cómo actúan dichos ataques en la saturación del ancho de banda; para llevarlo a cabo se diseñó e implementó una red híbrida con segmentación WAN, LAN y DMZ que inhabilita el acceso interno y externo a un servicio Web expuesto. Las herramientas evaluadas fueron UPD Unicorn, Longcat Flooder y UDPl.pl Script de Perl; las dos primeras instaladas sobre Windows, y la última, sobre Linux. Para validar está investigación se desarrolló un mecanismo de

Abstract

The Denial of Dervice Attacks (DoS) is used for precluding access to services and resources from an organization during an indefinite period of time. Usually, corporations apply DoS in their company's servers, so they cannot be accessed by authorized users. This project focuses on the evaluation of UDP Flood DoS attacks, using as an experimental platform a virtual network environment, which enables to identify how these attacks cause the Broadband saturation. In order to disable the internal and external access to an exposed Web service, a WAN, LAN, and DMZ segmentation network was designed and implemented.

The analyzed tools were: the UPD Unicorn, Longcat Flooder, and the UDPl.pl Perl Script. The first two were installed in a Windows environment and the third one was installed in a Linux. To validate this

* Ph.D. en Ingeniería Informática y de Telecomunicación, Ingeniero de Sistemas e Informática, Director de Postgrados, Escuela Politécnica del Ejército, Sangolquí, Ecuador.wfuertesd@espe.edu.ec
** Facultadde Ingeniería de Sistemas, Escuela Politécnica Nacional, Quito, Ecuador.frodaso@hotmail.com,deycitoscano@hotmail.com

detección y mitigación de los ataques a nivel del firewall e IDS/IPS, evitando de este modo la saturación de la red. Finalmente, se evaluó el consumo de memoria, CPU y ancho de banda durante el ataque, la detección y la evasión, con el fin de determinar cuál genera mayor impacto. Los resultados demuestran que el mecanismo detecta, controla y mitiga los ataques.

Palabras clave: Ataques de seguridad, Virtualización, Denegación de servicios, UDP Flood.

research a mechanism for detecting and mitigating attacks a firewall and IDS/IPS levels was designed and implemented, thus avoiding the network saturation. Finally, the memory, CPU, and broadband consumption during the attack, detection, and break out, were calculated in order to determine which generates the greatest impact. The results showed that the implemented mechanism detects, monitors, and mitigates this type of attacks.

Keywords: Security Attacks, Virtualization, Denial of Services, UDP Flood.

I. Introducción

Un ataque de Denegación de Servicio (Denial of Service -DoS-) tiene el propósito de evitar que el usuario legítimo haga uso de un recurso o servicio específico de red o un host. Entre las variantes de este tipo de ataque se pueden citar la inundación de la red mediante la inyección de paquetes, consumiendo el ancho de banda; la inanición de recursos, saturando la memoria; los errores de programación, para colapsar el procesador, y los ataques DNS y enrutamiento, para convencer mediante direcciones falsas y suplantación de identidad. Para esta investigación se seleccionó el ataque DoS UDP Flood, que provoca la pérdida de la conectividad de la red por la saturación del ancho de banda; este ataque ocurre cuando un atacante envía paquetes IP con datagramas UDP, con el propósito de colapsar el procesamiento de la máquina víctima, hasta el punto de no permitir manejar conexiones legítimas; debido a la naturaleza sin conexión del protocolo UDP, este tipo de ataques suele venir acompañado de técnicas de suplantación de identidad.

La comunidad científica ha investigado e implementado mecanismos que permitan disminuir y mitigar estos ataques de seguridad, empleando tecnologías de virtualización, cuya aplicación permite disminuir el riesgo a equipos y redes en producción, precautelando la información y servicios de las organizaciones. Los trabajos propuestos por Keller & Naues [1] y Ruíz *et al.* [2] presentan prototipos y modelos de laboratorios virtuales que permiten disponer de múltiples máquinas virtuales, para poder dimensionar la cantidad y técnicas de ataques. El trabajo propuesto por Jianga *et al.* [3] muestra el diseño, la implementación y la evaluación de Collapsar, que es una arquitectura basada en máquinas virtuales para el centro de detención de ataques de red. Abbasi & Harris [4], Fernández *et. al.* [5] y Galán *et al.* [6] presentan una metodología para establecer una *honeynet,* con el objetivo de identificar las técnicas utilizadas por los atacantes. El trabajo propuesto por Moore *et al.* [7] expone la técnica *análisis de backscatter,* que proporciona un estimado de la actividad de ataques de Denegación de Servicio (DoS). Otros trabajos [8, 9, 10] proponen la integración de las tecnologías de virtualización para el aseguramiento de una red, a través de la implementación de un Sistema de Detección de Intrusos (IDS). Otros investigadores [11, 12, 13] han utilizado las plataformas de virtualización para implementar técnicas de Recuperación de Desastres (DR).

En relación con la generación de varios ataques y con los mecanismos de mitigación, Fuertes *et al.* [14] presentan una investigación donde evalúan diversos ataques reales de redes IP, con el fin de establecer mecanismos de seguridad para mitigarlos. En un ámbito similar, Yaar, Perrig & Song [15] presentan un filtro de flujo de Internet (Siff) que permite detener selectivamente flujos individuales al llegar a la red. Finalmente, el trabajo presentado por Mirkovic & Reiher [16] propone D-WARD, un sistema de defensa contra ataques DDoS, cuyo objetivo es la detección autónoma de estos.

El presente trabajo se enfoca en la evaluación de ataques DoS tipo UDP Flood, utilizando como plataforma de experimentación un entorno virtual de red que permita identificar cómo actúan dichos ataques en la saturación del ancho de banda y cuál sería su impacto. Para llevarlo a cabo se diseñó e implementó una red con segmentación WAN, LAN y DMZ, con el propósito de inhabilitar el acceso interno y externo a un servicio Web expuesto. Las herramientas evaluadas fueron UPD Unicorn [17] , Longcat Flooder [18] y UDPl.pl Script de Perl [19]; las dos primeras instaladas sobre un ambiente Windows, y la última, sobre un ambiente Linux. Para validar está investigación se desarrolló un mecanismo de detección y mitigación de los ataques a nivel del firewall e IDS/IPS, evitando de este modo la saturación de la red.

Entre las principales contribuciones de esta investigación cabe mencionar: i) la evaluación de ataques UDP Flood mediante tres herramientas generadoras, de cara a determinar cuál provoca mayor impacto en la red, y ii) creación de reglas a nivel de Firewall e IPS que permitan detectar y mitigar estos ataques.

El documento ha sido organizado como sigue: el capítulo 2 presenta el fundamento teórico; en el 3 se describe el diseño y la configuración de la topología de la red experimental utilizada para la evaluación de las herramientas de generación de ataques UDP Flood; en el 4 se presentan, analizan y evalúan los resultados; en el 5 se discuten los resultados y, finalmente, en el capítulo 6 se establecen las conclusiones y se señala el trabajo futuro.

II. Fundamento teórico

A. Virtualización

Es la forma de particionamiento lógico de un equipo físico en diversas máquinas virtuales, para compartir recursos de hardware, como CPU, memoria, disco duro y dispositivos de entrada y salida [20]; esto implica hacer que un recurso físico, como un servidor, aparezca como si fuera varios recursos lógicos a la vez.

B. Escenario virtual de red

Puede ser definido como un conjunto de equipos virtuales (tanto sistemas finales como elementos de red –enrutadores y conmutadores–) conectados entre sí en una determinada topología desplegada sobre uno o múltiples equipos físicos, el cual emula un sistema equivalente y cuyo entorno deberá ser percibido como si fuera real [21]. El escenario virtual de red encapsula un conjunto de aplicaciones dentro de una red lógica, que permite configuraciones de los servicios de red de manera realista. En el caso de esta investigación, se ha utilizado este concepto porque la virtualización es una tecnología potencial para reproducir una topología de red real.

C. Denegación del servicio

De acuerdo con la World Wide Web Security FAQ [22], un ataque de Denegación de Servicio es diseñado para que una computadora o una red no sean capaces de proveer los servicios normales. Los ataques comunes de DoS tienen como objetivo el ancho de banda de la red o su conectividad. En los ataques de ancho de banda, la red es inundada por un gran volumen de tráfico que conduce al agotamiento de los recursos de red disponibles, de tal forma que usuarios legítimos no pueden acceder a ellos. En los ataques de conectividad, un computador es inundado por un gran volumen de solicitudes de conexión que conduce a un agotamiento de todos los recursos del sistema operativo, por tanto, el computador no es capaz de procesar las solicitudes de usuarios legítimos.

D. Tipos de ataques de denegación de servicios

Los ataques DoS pueden ser divididos en cinco categorías, basados en los niveles de protocolo atacados:

- Ataques a nivel de dispositivo de red: causados mediante el aprovechamiento de los errores o debilidades en el software o hardware, agotando recursos de los dispositivos de red [23];
- Ataques a nivel de Sistema Operativo: toman ventaja de la forma de los protocolos que se implementan en ellos; este ataca el Internet Control Message Protocol (ICMP) [24];
- Ataques a nivel de aplicaciones: aprovechan los errores en las aplicaciones de red sobre los host objetivos o mediante el uso de aplicaciones para consumir los recursos de su víctima [25];
- Ataques de inundación de datos: utilizan el ancho de banda disponible para una red, host o un dispositivo en su mayor extensión mediante el envío de cantidades masivas de datos por procesar;
- Ataques basados en las características de los protocolos: toman ventajas de las características del protocolo.

E. Ataque UDP Flood

Un ataque UDP Flood es posible cuando un atacante envía paquetes UDP a un puerto randómico de un equipo víctima; cuando este recibe un paquete UDP, determina la aplicación que está esperando en el puerto destino; si no existe ninguna aplicación esperando en el puerto mencionado, entonces genera un paquete ICMP de destino inalcanzable al origen. El envío excesivo de paquetes UDP puede producir la caída del sistema [26].

III. Configuración del experimento

A. Herramientas

Para implementar este experimento se utilizaron herramientas de código abierto y de libre distribución; a continuación se detallan:

1) Sistema de virtualización: Como plataforma de virtualización se utilizó VMware Workstation [28] sobre Windows 7; su objetivo fue configurar múltiples computadoras virtuales interconectadas entre sí, implementando un escenario virtual de red.

2) Firewall: Como firewall se utilizó PfSense [29], que es una distribución basada en FreeBSD; su objetivo es disponer de un cortafuego que permita establecer seguridad entre zonas de confianza como LAN y la DMZ. El PfSense permite proteger una red de accesos ilícitos, redirigir paquetes hacia máquinas de la red interna, otorgar accesos solo desde sitios conocidos, etc.; las conexiones SSH y HTTP fueron habilitadas para permitir la administración de este desde una interfaz Web.

3) Sistema de Detección de Intrusos (IDS): Como IDS se utilizó Snort [30], que está disponible dentro de la distribución de PfSense; su objetivo fue vigilar el tráfico de la red, examinar los paquetes en busca de datos sospechosos y detectar las primeras fases de cualquier ataque. Las reglas básicas de Snort fueron descargadas y actualizadas desde su sitio oficial, utilizando el oinkcode obtenido después del registro como usuario.

4) Web Server: Como servidor Web se utilizó Apache2 [31] sobre la versión estándar de Ubuntu Server; su objetivo fue servir páginas Web solicitadas por equipos cliente mediante el uso de navegadores Web. El módulo de seguridad mode_evasive fue instalado para prevenir o anular ataques de DoS. Además, se integró el servidor Web con el servidor de aplicaciones por medio del mod_jk de Apache.

5) App Server: Como servidor de aplicaciones se utilizó Tomcat 6 [32] sobre la versión estándar de Ubuntu Server; su objetivo fue proporcionar servicios de aplicación a las computadoras clientes. La versión de jdk 1.6 fue instalada para habilitar el servidor de aplicaciones.

6) Herramientas para inyección de paquetes UDP: Como herramientas para generación y envío de paquetes UDP se utilizó Longcat, UDP Unicorn y un script en Perl (UDPl.pl); las dos primeras funcionaron sobre plataformas Windows, y la última, sobre cualquier distribución de Linux o Windows con soporte Perl. El Firewall de Windows de las máquinas atacantes fue desactivado, y una regla a nivel del Firewall fue agregada, con el objetivo de permitir pasar un gran volumen de paquetes UDP.

7) Captura de tráfico: Como herramientas para captura de tráfico se utilizó Tcpdump [33] y Wireshark [34] sobre el Firewall y el Web Server respectivamente; su objetivo fue identificar y analizar el tráfico que circula por la red, analizar los paquetes de datos en una red activa desde un archivo de lectura previamente generado.

B. Diseño de la topología experimental

La generación de ataques de Denegación de Servicios, DoS, utilizando UDP Flood, y su mecanismo de mitigación requirieron de la creación de una infraestructura de red similar a la utilizada por cualquier red en producción. Es así que para el diseño e implementación de la topología de prueba se requirió de un enrutador que posibilitó la salida a Internet, un computador con Windows7 y un equipo anfitrión de Virtualización que permitió crear los diferentes componentes de la implementación, convirtiéndola en una plataforma híbrida, tal como se muestra en la Fig. 1.

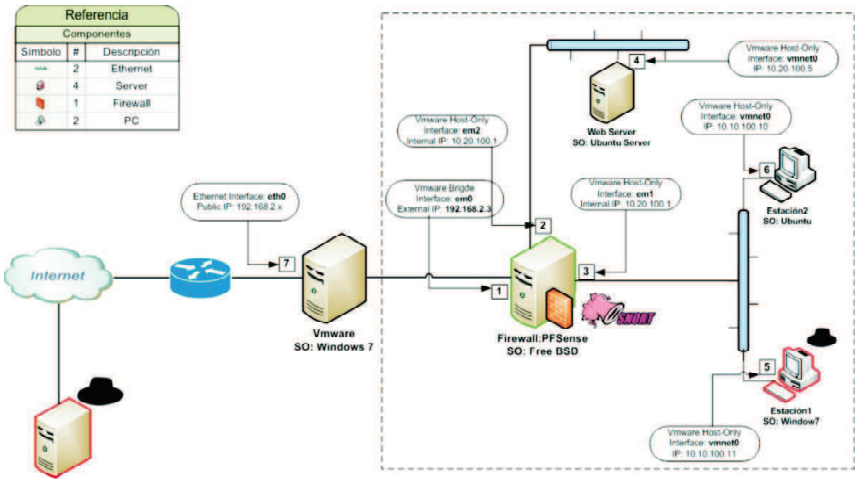

Fig. 1. Diseño para la generación y mitigación de ataques UDP Flood

C. Implementación de la plataforma experimental

Las pruebas se ejecutaron en el equipo anfitrión VMware, bajo Windows7, con procesador Core5, memoria 4Gb y almacenamiento 500 Gb. En las máquinas virtuales se instaló pfSense FreeBSD como firewall, Ubuntu Server 10.10 como Web Server, Ubuntu Server 10.10 y Windows XP como estaciones de trabajo.

El siguiente procedimiento ha sido utilizado para implementar el diseño propuesto en un entorno virtual: i) En primer lugar, se ha sincronizado el reloj mediante el protocolo de temporización de red (NTP) en el equipo anfitrión; ii) Luego se ha creado la primera máquina virtual VMware, en la cual se ha instalado el sistema operativo FreeBSD con Pfsense como firewall y con Snort como IDS/IPS, configurando tres interfaces de red (WAN, LAN y DMZ); iii) Posteriormente, se ha creado una segunda máquina virtual VMware, en la cual se ha instalado el sistema operativo Ubuntu Server con Tomcat 6 como servidor de aplicaciones, y Apache2 como Web server, configurando una sola interface de red en la DMZ; iv) A continuación, se ha clonado la segunda máquina virtual, con el fin de reducir el tiempo de instalación y con la finalidad de administrar el firewall desde la LAN. Adicionalmente se instaló el programa Udpl.pl Script de Perl para generar ataques desde la LAN; v) Luego, se ha creado una tercera máquina virtual VMware, en la cual se ha instalado el sistema operativo Windows XP y los programas LongCat Flooder y UDP Unicorn de forma que permitan generar ataques desde la LAN, y, finalmente, vi) Se ha conectado con una estación de trabajo externa, en la cual se ha instalado el sistema operativo Windows XP y los programas LongCat Flooder y UDP Unicorn de forma tal que se puedan generar ataques desde la WAN.

En este punto cabe señalar que el enrutador de la Fig. 1 es un dispositivo físico que conecta al equipo anfitrión tanto hacia el Internet como a las máquinas virtuales.

D. Configuración del firewall

Para la conexión de los segmentos de red detallados en la Fig. 1 se requirió la creación de un mecanismo de mitigación basado en reglas a nivel de firewall que permitieron redirigir los paquetes de LAN a la WAN y viceversa (ver Fig. 2).

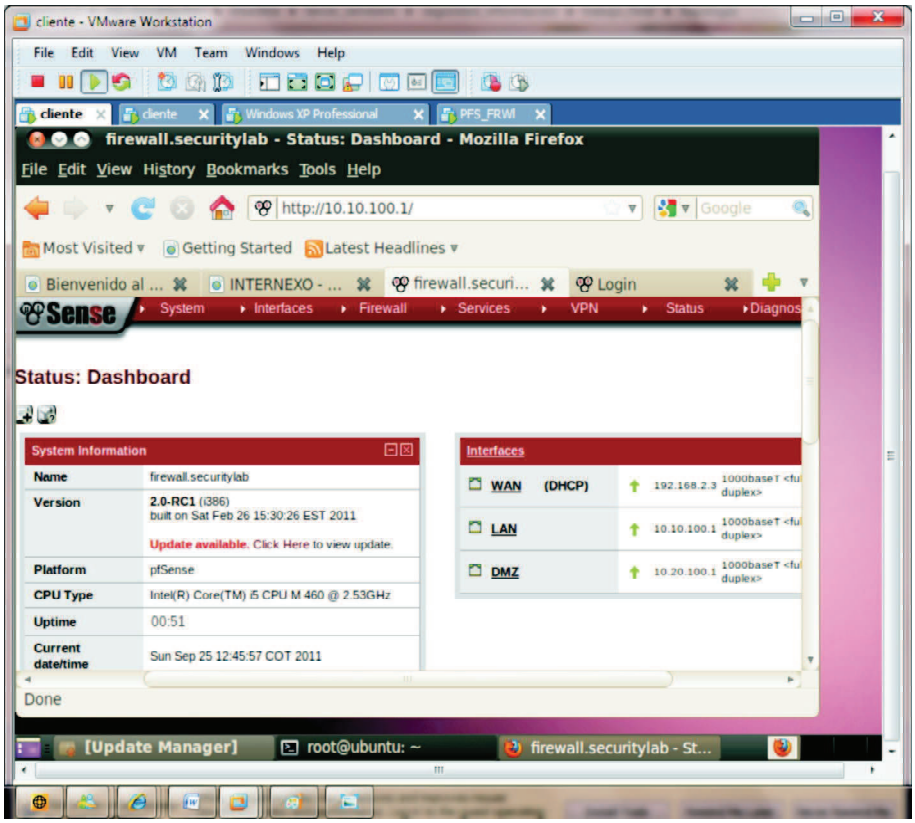

Fig. 2. Configuración de firewall

Para la publicación del servicio Web se requirió crear una regla de traducción de direcciones de red (NAT, Network Address Translation) dentro del firewall, la cual permitió redirigir los paquetes de la DMZ a la WAN y LAN, habilitando el acceso al servicio desde la red interna y red externa. Además, se requirió crear dos reglas dentro del Snort que permitieron filtrar todos los paquetes provenientes de la WAN y LAN a la DMZ, bloqueando la IP origen y generando alertas en la consola administrativa del *Snort*.

Debido a que los servidores y estaciones de trabajo se agrupan en segmentos de redes diferentes dentro de la LAN, DMZ y WAN fue necesario definir 3 interfaces de red virtuales, que se detallan a continuación (ver Fig. 2):

- Interface "em0" (en eth0): es una interface VWware tipo puente (brigde) apuntando a la red externa WAN y conectada al enrutador para permitir el acceso a Internet;
- Interface "em1" (en eth1): es una interface VMware tipo bridge apuntando al segmento de red interna LAN;
- Interface "em2" (en eth2): es una interface VMware tipo bridge apuntando al segmento de red de la zona desmilitarizada DMZ.

E. Generación de ataques

La generación de ataques UDP Flood se realizó ejecutando los programas UDP Unicorn y LongCat Flooder en una máquina virtual con Windows XP desde la LAN, para un primer caso (interno), y en una estación de trabajo con Windows XP desde la WAN, para un segundo caso (externo). Además, se realizó un tercer caso ejecutando el script de perl Udpl.pl en una máquina virtual con Ubuntu Server desde la LAN. Para todos estos ataques fue necesario configurar los siguientes parámetros: dirección IP

de la máquina víctima (Web Server), tamaño del paquete y número de hilos.

Estos ataques se caracterizaron por generar un considerable volumen de tráfico en la red y por comprometer la disponibilidad del servicio Web expuesto para usuarios legítimos. Este volumen se evidenció comparando la cantidad de paquetes recibidos utilizando Tcpdump y la disponibilidad al observar un consumo de recursos de CPU, RAM, memoria virtual a través de las herramientas del sistema operativo.

IV. Resultados experimentales

Para el monitoreo de los paquetes inyectados generados por las herramientas de ataques UDP Flood se tomaron algunas mediciones de cada una de ellas, considerando, en un primer caso, un ataque generado desde la red interna, y, en un segundo caso, generado desde la red externa.

Se realizaron pruebas para cada herramienta y cada caso con 15, 30 y 45 hilos. A continuación se tomaron los datos de consumo de CPU, memoria, memoria virtual, paquetes enviados y paquetes recibidos utilizando el Monitor del Sistema y Tcpdump [33]; estas lecturas fueron tomadas cada minuto durante un período de 15 minutos. Una vez realizadas las 5 pruebas de cada herramienta se calcularon los promedios para cada minuto, y con esos valores se generaron las figuras que se analizan a continuación.

A. Ataque de DoS con UDP Unicorn

Para realizar este ataque de DoS se utilizó como herramienta UDP Unicorn, que es un utilitario de código abierto escrito en lenguaje C, para win32 de multihilos que utiliza librerías de WinSock para crear sockets UDP e inundar la máquina víctima [17].

La Fig. 3 muestra el consumo de CPU, memoria física y memoria virtual para ataques realizados con 15, 30 y 45 hilos. Como se puede observar, el CPU se vio saturado y no podía procesar peticiones al realizar un ataque con 15 hilos a los 13 minutos, con 30 hilos a los 9 minutos y con 45 hilos a los 7 minutos. Así mismo, la memoria física se vio saturada al realizar un ataque con 15 hilos a los 8 minutos, con 30 hilos a los 5 minutos y con 45 hilos a los 3 minutos. Por último, la memoria virtual se vio saturada al realizar un ataque con 15 hilos a los 16 minutos, con 30 hilos a los 9 minutos y con 45 hilos a los 7 minutos.

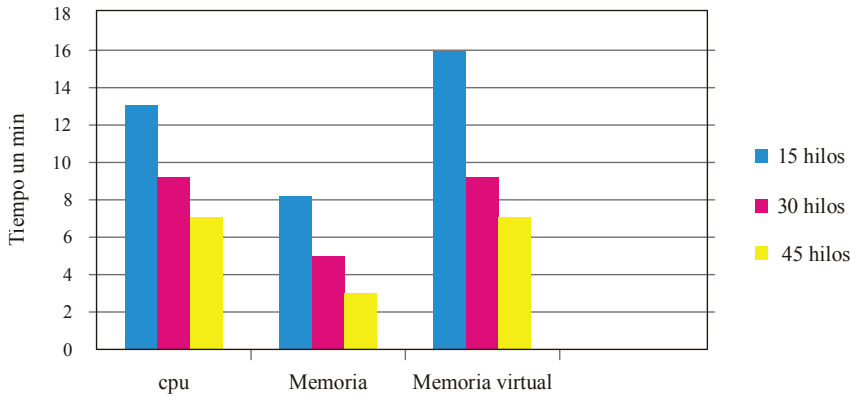

Fig. 3. Tiempo de saturación de recursos para UDP Unicorn

Así mismo, la memoria física se vio saturada al realizar un ataque con 15 hilos a los 8 minutos, con 30 hilos a los 5 minutos y con 45 hilos a los 3 minutos. Por último, la memoria virtual se vio saturada al realizar un ataque con 15 hilos a los 16 minutos, con 30 hilos a los 9 minutos y con 45 hilos a los 7 minutos.

Al aplicar el mecanismo de mitigación (ver Fig. 4) se puede apreciar que las cantidades de paquetes capturados y borrados son similares, mientras que los capturados sean de 15, 30 y 45 hilos, prácticamente son iguales; esto indica que con UDP Unicorn el firewall acepta cierta cantidad y luego lo bloquea y no acepta ningún paquete más que venga de la máquina atacante.

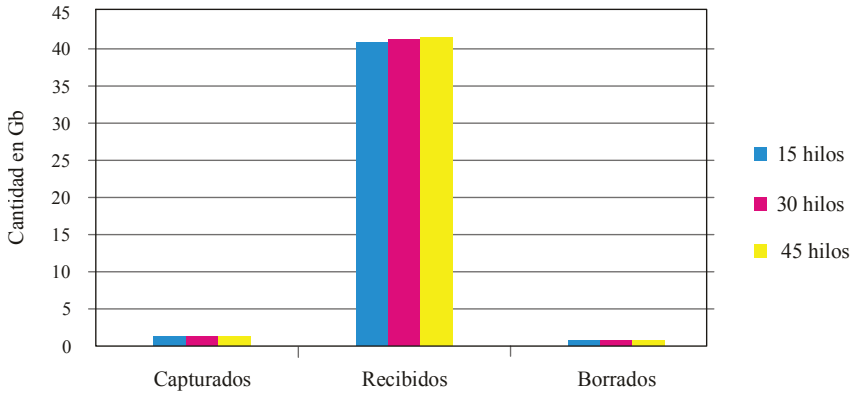

Fig. 4. Cantidad de paquetes procesados con Tcpdump para UDP Unicorn

B. Ataque de DoS con Longcat Flooder

Para un ataque tipo DoS también se puede utilizar Longcat Flooder, herramienta de inundación multiprotocolo que ha llegado a ser popular debido a su simplicidad de uso y sus características. Los protocolos soportados son TCP (SYN flooding), UDP y HTTP [18].

La Fig. 5 muestra el consumo de CPU, memoria física y memoria virtual para ataques realizados con 15, 30 y 45 hilos. Como se puede apreciar, el CPU se vio saturado y no podía procesar peticiones al realizar un ataque con 15 hilos a los 16 minutos, con 30 hilos a los 12 minutos y con 45 hilos a los 8 minutos. La memoria física se ve saturada al realizar un ataque con 15 hilos a los 8 minutos, con 30 hilos a los 5 minutos y con 45 hilos a los 3 minutos. Por último, la memoria virtual se ve saturada al realizar un ataque con 15 hilos en 16 minutos, con 30 hilos en 9 minutos y con 45 hilos a los 7 minutos.

Con los datos capturados por Tcpdump, se pudo determinar que la cantidad de paquetes fue directamente proporcional al número de hilos, es decir, a mayor número de hilos, mayor será la cantidad de los paquetes capturados y recibidos, mientras no existen paquetes borrados, como se puede ver en la Fig. 6.

C. Ataque de DoS con script en Perl

La tercera herramienta utilizada para un ataque tipo DoS en este trabajo de investigación fue la realización de un script tomado desde flood.pl. Para la instalación se realizó la copia del archivo fuente en flood.pl, y como requisitos previos, la distribución de PERL 5.8, por lo general preinstalado en un sistema Unix / Linux [19].

La Fig. 7 muestra el consumo de CPU, RAM y memoria virtual para ataques realizados con 15, 30 y 45 hilos. Tal como se puede observar, el CPU se vio saturado y no podía procesar peticiones al realizar un ataque con 15 hilos a los 9 minutos, con 30 hilos a los 6 minutos y con 45 hilos a los 6 minutos. La RAM se ve saturada al realizar un ataque con 15, 30 o 45 hilos a los 4 minutos.

Al aplicar el mecanismo de mitigación y con los datos capturados por Tcpdump, se pudo determinar que la cantidad de paquetes durante los 15 minutos es solo de recepción, ya que los datos capturados y borrados son casi nulos, como se puede ver en la Fig. 8.

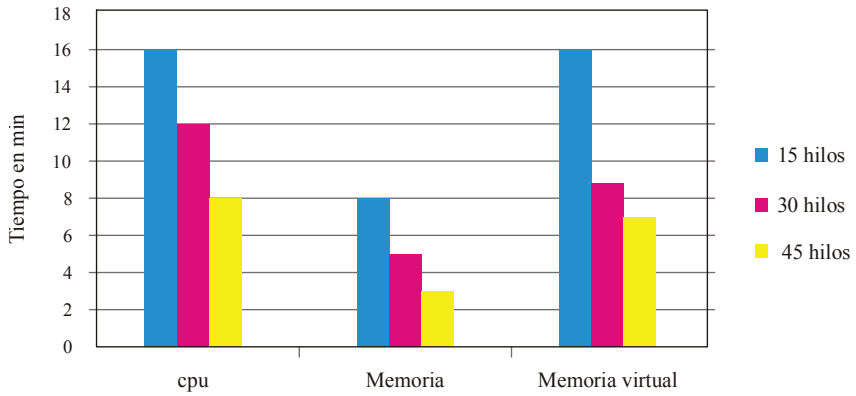

Fig. 5. Tiempo de saturación de recursos para Longcat Flooder

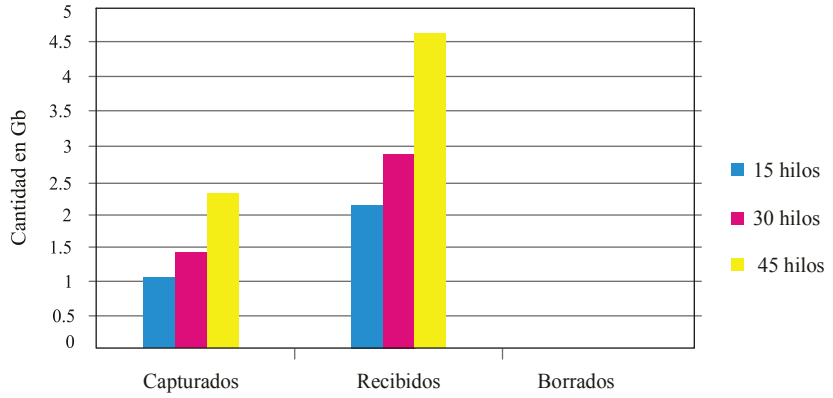

Fig. 6. Cantidad de paquetes procesados para Longcat Flooder

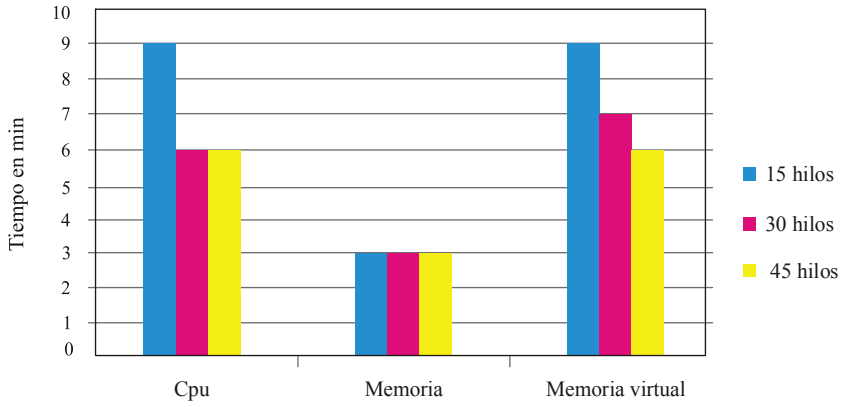

Fig. 7. Tiempo de saturación de recursos para UDPl_perl

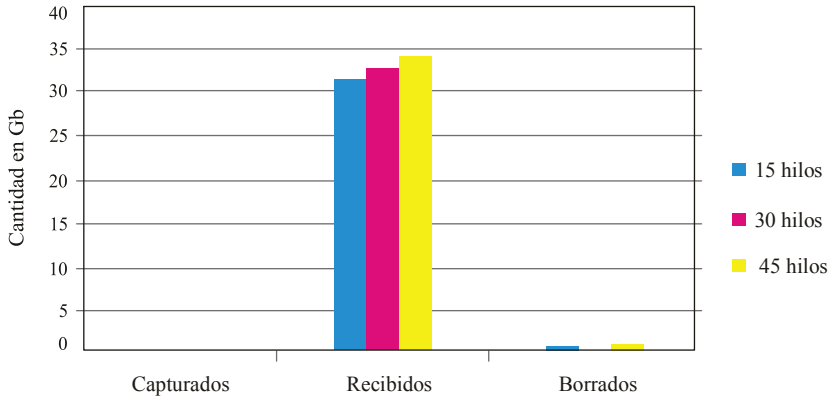

Fig. 8. Cantidad de paquetes procesados para Script UDPl_perl

D. Análisis comparativo de las tres herramientas utilizadas

Luego de analizar el comportamiento de las tres herramientas fue necesario compararlas para determinar la que mayor riesgo e impacto genera a la red de una organización:

1) Comparación del consumo del CPU: Se observó que Longcat Flooder fue la herramienta que en mayor porcentaje hizo uso del CPU, y fue constante durante el ataque. La herramienta Udpl_perl saturó el uso de CPU en el momento inicial del ataque y fue disminuyendo rápidamente. Finalmente, UDP Unicorn tuvo un consumo intermedio entre las dos herramientas anteriores, como se puede observar en las Figs. 9, 10 y 11. Nótese además que al minuto 8 el porcentaje de uso del CPU es muy bajo, esto se debió a que el firewall filtró los paquetes, bloqueó la IP de la máquina atacante y generó alertas en la consola administrativa del IDS.

2) Comparación del uso de memoria real: Al comparar el consumo de memoria, se observó que es la más afectada para este tipo de ataque; es así como en un promedio de 6 a 7 minutos la máquina víctima ya tenía saturada su RAM (consumo al 100%), como se puede ver en las Figs. 12, 13, y 14.

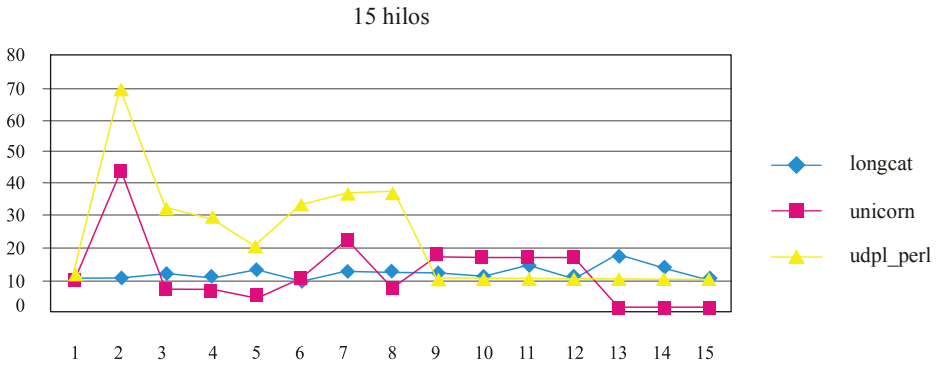

Fig. 9. Comparación del consumo de CPU con 15 hilos

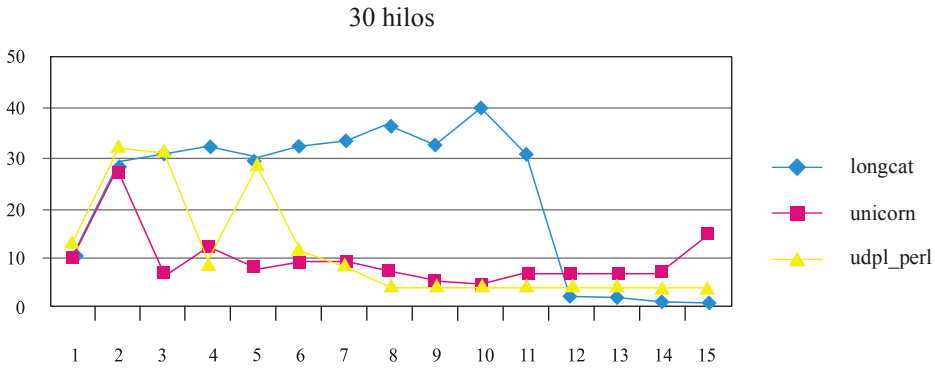

Fig. 10. Comparación del consumo de CPU con 15 hilos

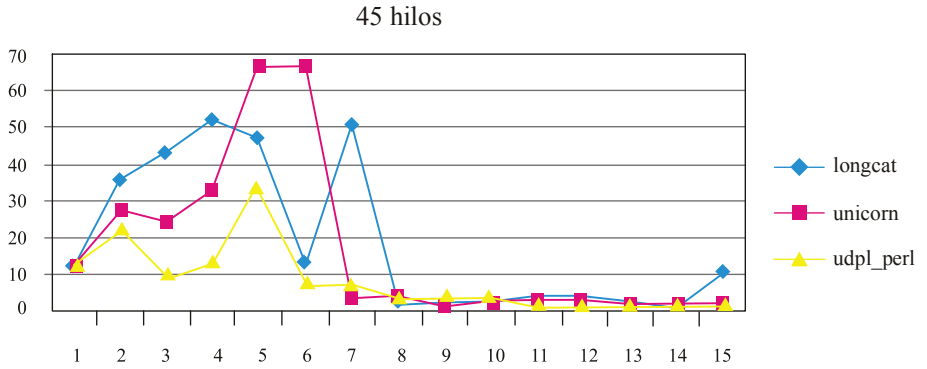

Fig. 11. Comparación del consumo de CPU de las tres herramientas

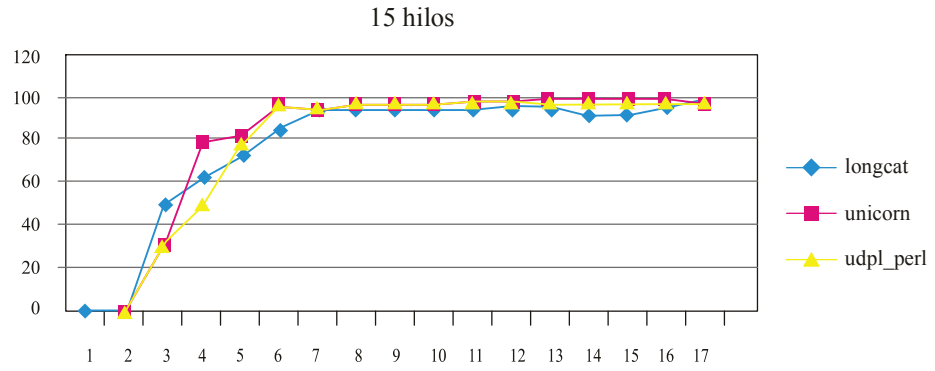

Fig. 12. Comparación del consumo de la memoria con 15 hilos

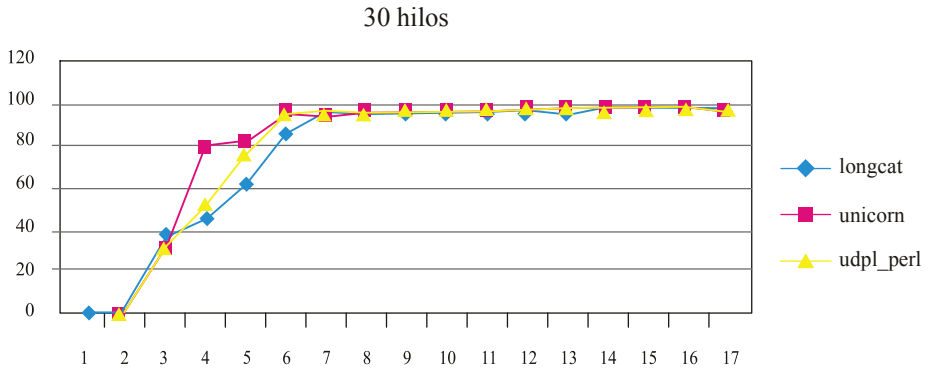

Fig. 13. Comparación del uso de la memoria con 30 hilos

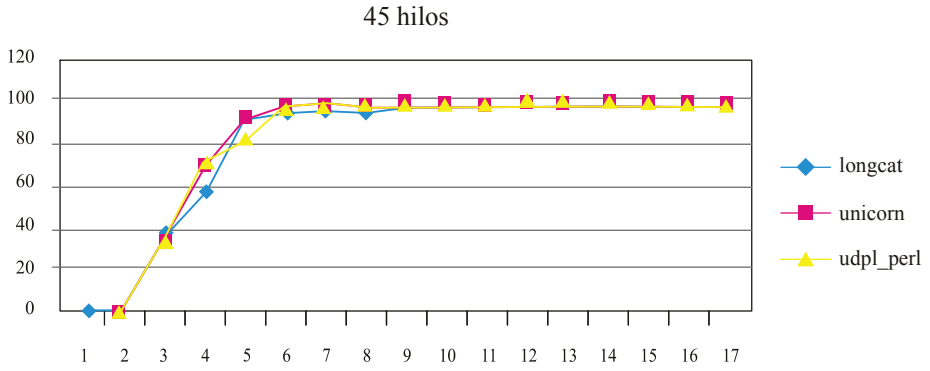

Fig. 14. Comparación uso de memoria con 45 hilos

Es necesario mencionar que a pesar de que el IDS detuvo el ataque evitando el paso de paquetes hacia la máquina víctima, el uso de memoria se mantiene en los niveles más elevados, y el proceso de restauración (niveles estables de uso de CPU y memoria) tomó un período de aproximadamente 1 hora.

3) Comparación de la memoria virtual: Al analizar el uso de la memoria virtual, se observó que los tiempos para la saturación fueron menores a medida que aumentaba el número de hilos, como se puede observar en las Figs. 15, 16 y 17. A medida que se incrementa el número de hilos para cada herramienta, el comportamiento tiende a ser muy similar, así, al tener 45 hilos, las 3 herramientas llegan al 100% de uso de memoria al minuto 7.

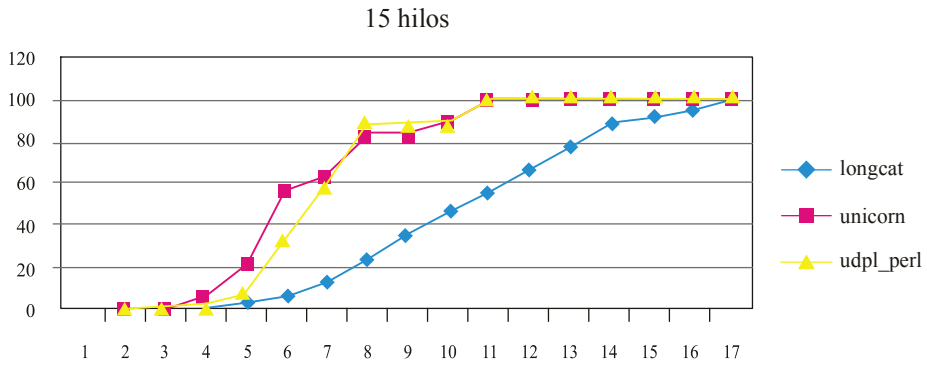

Fig. 15. Comparación uso de memoria virtual con 15 hilos

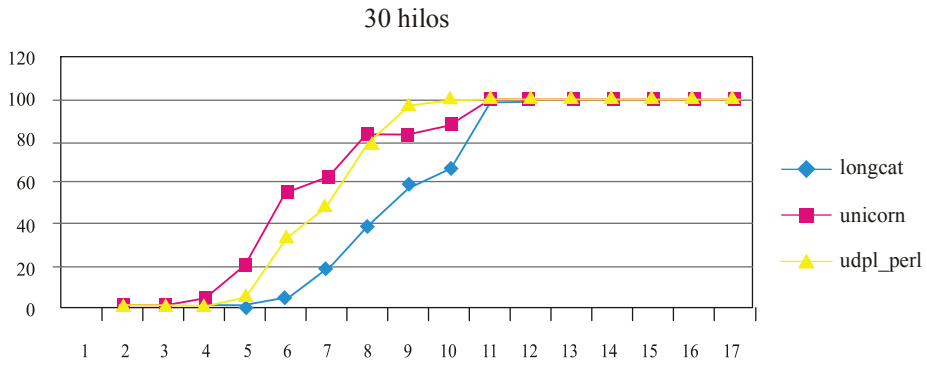

Fig. 16. Comparación uso de memoria virtual con 30 hilos

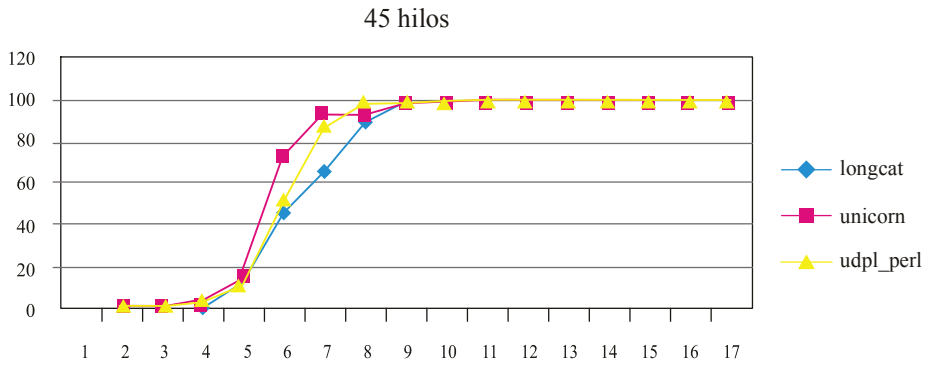

Fig. 17. Comparación uso de memoria virtual con 45 hilos

V. Discusión

Como se muestra en las figuras del capítulo de Resultados, la generación de ataques de denegación de servicios utilizando Longccat, UDP Flooder y Script de Perl con procesamiento multihilo provoca la inundación de la red y el máximo consumo de recursos de la máquina víctima, es así como en un lapso menor a 10 minutos, y teniendo inactivas las reglas del IDS, la máquina víctima no es capaz de procesar peticiones de usuarios legítimos. El problema de saturación de la red puede agudizarse, aún más, a medida que se incremente el número de hilos concurrentes y se elimine el tiempo de espera entre hilos de ejecución. La evaluación de estas herramientas para detectar y medir ataques UDP Flood traerá beneficios a los administradores y encargados de la red debido a que permitirá alertar y dimensionar el tráfico y tipo de paquetes generados por este tipo de ataques.

Las estadísticas obtenidas en este trabajo de investigación permitieron diseñar un esquema efectivo de mitigación, que fue configurado a nivel de Firewall e IDS. La aplicación de este diseño de mitigación traerá beneficios inmediatos a los administradores y encargados de la red, debido a la facilidad del establecimiento de reglas a nivel de Firewall e IDS.

Como limitante, existe la necesidad de mantener actualizada la base de reglas del IDS, dado que se pueden presentar más tipos de ataques o variaciones de estos. Esto se puede resolver con la subscripción al sitio oficial de Snort [30], de donde, por un pago especial, se pueden descargar mensualmente las reglas actualizadas, facilitando la administración y creando un mecanismo de protección seguro para la red de la organización.

Los valores presentados en este trabajo de investigación son referenciales y dependen de las capacidades de la máquina anfitrión en donde se levantaron todos los componentes del ambiente de experimentación.

Una de las mayores ventajas que presenta el presente trabajo fue la creación de un entorno virtual donde se puede reproducir la funcionalidad de una red de servicios telemáticos; esto facilitó la evaluación de algunos escenarios de experimentación o pruebas y la validación de algunas herramientas relacionadas con un ataque de Denegación de Servicios. El uso de entornos virtuales permite la ejecución de pruebas de seguridad informática, el ahorro de tiempo y la disminución de costos de experimentación, en comparación con pruebas en un escenario con equipos reales.

Finalmente, después de realizado el análisis comparativo entre las herramientas que generan ataques UDP Flood se identifica que el Script de Perl es la herramienta que mayor consumo de ancho de banda genera y, por tanto, mayor saturación al equipo y a la red dentro de nuestro ambiente de experimentación; esto se debe a que establece una conexión y escritura nativa de bytes que, junto al procesamiento multihilo, tamaño máximo de paquete y no definición de tiempo de espera entre hilos de ejecución, consigue el mayor consumo de recursos.

VI. Conclusiones y trabajo futuro

El presente trabajo se enfocó en el análisis y evaluación de herramientas que generan ataques DoS de tipo UDP Flood dentro de un ambiente virtualizado, con el fin de mejorar las técnicas de seguridades ante este tipo de ataques. Para implementar este trabajo se creó una topología de red virtual segmentando los accesos por la LAN, WAN y DMZ y obteniendo el ambiente virtual necesario para generar ataques de UDP Flood con herramientas como UDP Unicorn, Longcat Flooder y UDPl.pl Script desarrollado en Perl. Para validar esta investigación se desarrolló un mecanismo de detección y mitigación de los ataques a nivel del firewall e IDS/IPS, evitando de este modo la saturación de la red. Sobre la máquina víctima se realizó el análisis de consumo de CPU, memoria y ancho de banda. Los resultados demuestran que el mecanismo detecta, controla y mitiga estos ataques.

Como trabajo futuro se plantea realizar el análisis para herramientas de DoS de tipo SYN Flood, HTTP

Flood e ICMP Flood, con el fin de identificar el tipo de inundación que mayores daños provoque al ancho de banda de una red corporativa.

Agradecimientos

Los autores de este proyecto desean agradecer a la Escuela Politécnica Nacional, Facultad de Ingeniería de Sistemas y a la Dirección de Posgrados de la Escuela Politécnica del Ejército por las facilidades prestadas durante el desarrollo de esta investigación.

Referencias

[1] J. Keller, R. Naues. *Design of a virtual computer security lab*. Fern Universititat in Hagen-Germany. Available: http://www.fernuniversitaet-hagen.de/imperia/md/content/fakultaetfuermathematikundinformatik/pv/97-08/547-045_1_.pdf

[2] J. Ruiz, D. Fernández, F. Galán, L. Bellido. *Modelo de Laboratorio Docente de Telemática basado en Virtualización Distribuida*. Universidad Politécnica de Madrid.

[3] X. Jianga, D. Xua, Y. Wang. "Collapsar: AVM-based honeyfarm and reverse honeyfarm architecture for network attack capture and detention". Journal of Parallel and Distributed Computing (2006) Volume 66, Issue 9, pages 1165-1180. Available: http://www.mendeley.com/research/collapsar-a-vmbased-honeyfarm-and-reverse-honeyfarm-architecture-for-network-attack-capture-and-detention/

[4] F. H. Abbasi, R. J. Harris. *Experiences with a Generation III Virtual Honeynet*. School of Engineering and Advanced Technology (SEAT), Massey University, New Zealand.

[5] H. Fernández, J. Sznek, E. Grosclaude. *Detección y limitaciones de ataques clásicos con Honeynets virtuales*. Publicado en el V Congreso de Seguridad Informática 2009, (CIBSI'09), realizado del 16 al 18 de noviembre de 2009, Montevideo, Uruguay.

[6] F. Galán, D. Fernández. *Use of VNUML in Virtual Honeynets Deployment*. IX Reunión española sobre criptología y seguridad de la información (RECSI), Barcelona, pp. 600-615, Sep. 2006. ISBN: 84-9788-502-3.

[7] D. Moore, G. Voelker and S. Savage. *Inferring Internet Denial-of-Service* Activity. Department of Computer Science and Engineering University of California, San Diego.

[8] P. Li, T. Mohammed. *Integration of Virtualization Technology into Network Security Laboratory*. In Proc. 38th ASEE/IEEE Frontiers in Education Conference, Saratoga, NY, 10/2008.

[9] T. Garfinkel and M. Rosenblum. *A Virtual Machine Introspection Based Architecture for Intrusion Detection*. In Proc. Network and Distributed Systems Security Symposium, pp: {191-206}, 2003.

[10] K. Ali. *Algorizmi: A Configurable Virtual Testbed to Generate Datasets for Offline Evaluation of IDS*. Electronic Theses and Dissertations, University of Waterloo, 2010.

[11] E. Damiani, F. Frati, D. Rebeccani. *The open source virtual lab: a case study*. In proceedings of the workshop on free and open source learning environments and tools, hosted by: FOSLET 2006; pp. 5-12, Italy nel, 2006.

[12] Co-innovation lab Tokyo. *Disaster Recovery Solution Using Virtualization Technology*, White paper. Available: http://www.cisco.com/en/US/prod/collateral/ps4159/ps6409/ps5990/N037_COIL_en.pdf.

[13] P. Ferrie P. "*Attacks on Virtual Machine Emulators*", Symantec White Paper, 2008.

[14] W. Fuertes, P. Zapata, L. Ayala y M. Mejía. *Plataforma de Experimentación de Ataques Reales a Redes IP utilizando Tecnologías de Virtualización*, Memorias del Tercer Congreso

de Software Libre CONASOL-2010. Talara, Perú, dic. 2010.

[15] A. Yaar, A. Perrig, D. Song. SIFF: "*A Stateless Internet Flow Filter to Mitigate DDoS Flooding Attacks*", C. Mellon University.

[16] J. Mirkovic, P. Reiher. *D-WARD: A Source-End Defense Against Flooding Denial-of-Service Attacks*. IEEE.

[17] UDP Unicorn. Available: http://sourceforge.net/projects/UDPunicorn/

[18] Longcat Flooder. Available: http://partyvan.info/wiki/Longcat_Flooder

[19] Perl flood script. Available: http://sourceforge.net/projects/freeman015/

[20] F. Galán, D. Fernández, W. Fuertes, M. Gómez and J. E. López de Vergara. *Scenario-based virtual network infrastructure management in research and educational test beds with VNUML*. Annals of Telecommunications, Vol. 64, No. 5, pp. 305-323, May 2009.

[21] W. Fuertes and J. E. López de Vergara. *An emulation of VoD services using virtual network environments*. In Proc.GI/ITG Workshop on Overlay and Network Virtualization, Kassel-Germany, March 2009.

[22] S. Lincoln D. & J. N. Stewart. *The World Wide Web Security FAQ*, Available: http://www.w3.org/Security/Faq/

[23] D. Karig and R. Lee. *Remote denial of service attacks and countermeasures*. Technical Report CE-L2001-002, Department of Electrical Engineering, Princeton University, Princeton, NJ, Oct. 2001.

[24] M. Kenney. Malachi, "*ping of death*". Available: http://www.insecure.org/sploits/ping-o-death.html, Jan. 1997.

[25] Finger bomb recursive request, http://xforce.iss.net/static/47.php.

[26] L. D. Stein and J. N. Stewart. *The World Wide Web Security FAQ*, versión 3.1.2. Available: http://www.w3.org/Security/Faq, Feb. 2002.

[27] NetworkDicctionary. [Online]. Available: http://www. networkdictionary.com/security/u.php, Nov. 2005.

[28] VMWare: http://www.vmware.com/. Última comprobación, junio 2011.

[29] Pfsense: http://www.pfsense.org/. Última comprobación, junio 2011.

[30] Snort: https://www.snort.org/. Última comprobación, junio 2011.

[31] Apache: http://apache.org/. Última comprobación, junio 2011.

[32] Apache Tomcat: http://tomcat.apache.org/download-60.cgi. Última comprobación junio 2011.

[33] Tcpdump: http://www.tcpdump.org/. Última comprobación, jun. 2011.

[34] Wireshark: http://www.wireshark.org/. Última comprobación, jun. 2011.