Plataforma de Experimentación de Ataques Reales a Redes IP utilizando Tecnologías de Virtualización

Dirección de Postgrado, Escuela Politécnica del Ejército, Sangolquí - Ecuador
Walter Fuertes, Patricia Zapata, Luis Ayala y Miguel Mejía
wfuertesd@espe.edu.ec, lzapata@ups.edu.ec, mike_m78@hotmail.com

RESUMEN:

Los ataques a redes IP pueden colapsar la continuidad de los servicios de las empresas afectando su imagen y causando graves pérdidas económicas. La presente investigación se centra en la evaluación de diversos ataques reales de redes IP utilizando plataformas de virtualización con el fin de establecer mecanismos de seguridad para mitigarlos. Para llevarlo a cabo, se diseñó e implementó varias topologías de experimentación utilizando entornos virtuales de red, dentro de las cuales se probaron el escaneo de puertos, fuerza bruta, suplantación de identidad y denegación de servicios, tanto en una red de área local como en una extendida. Para cada topología, se utilizó diferente software libre tanto para producir el ataque como para obtener el flujo de tráfico, evaluándose las consecuencias del ataque. Para contrarrestar dichos ataques, se desarrolló un demonio en Shell script que sea capaz de detectar, controlar y mitigar los ataques mencionados de manera programable y constante. Los resultados muestran la funcionalidad de esta investigación que reduce las amenazas y vulnerabilidades de las redes en producción.

Palabras clave: Ataques de seguridad, evaluación, mitigación, tecnologías de virtualización

ABSTRACT

IP networks Attacks can collapse the continuity of business services affecting its image and causing economic losses. This research focuses on the evaluation of several IP networking real attacks using virtualization platforms to provide security mechanisms to mitigate them. To carry out this work, we designed and implemented several experimentation topologies using virtual network environments, within which were tested port scans, brute force, spoofing and denial of services, both on a local area network as wide area network. For each topology, different free open source software was used both to produce the attack and to obtain the traffic flow, evaluating the consequences of these attacks. To deal with such attacks, we developed a demon program that is able to prevent, detect and mitigate these attacks mentioned. The results show the functionality of this research that reduces threats and vulnerabilities in production networks.

Key words: security attacks, virtualization technology

1. INTRODUCCION

Las redes teleinformáticas están expuestas a ataques e intrusiones que pueden dejar inoperativos los recursos y causar pérdidas de la imagen, productividad, credibilidad y competitividad, provocando perjuicios económicos que podrían comprometer la continuidad del negocio [1.]. Esta incertidumbre sigue agravándose, pues continúan apareciendo diversas amenazas, vulnerabilidades y tipos de ataques que implican hurto, modificación, espionaje, interrupción, falsificación, denegación de servicios etc., perjudicando directamente a los negocios que son altamente dependientes de sus sistemas y redes de información [2.].

Para prevenir y contrarrestar una amplia gama de amenazas a las seguridades de las redes, es necesario conocer las vulnerabilidades de las empresas e identificar diversos tipos de ataques. Para manejar esta situación se propone crear un ambiente de red controlado con los componentes necesarios que detecten ataques maliciosos, para analizarlos y contrarrestarlos.

Una primera alternativa sería mediante equipos reales, sin embargo esto encarecería la solución y pondría en riesgo la red en producción. Otra alternativa sería utilizar máquinas virtuales, con las cuales es posible reducir costos de inversión de hardware, costos de mantenimiento, costo y tiempo de experimentación y sobre todo reduciría el riesgo del colapso de la red en producción [3.].

En este contexto, la comunidad científica ha mostrado un creciente interés en investigar e implementar soluciones para disminuir los ataques de seguridad a la redes aprovechado las tecnologías de virtualización. De acuerdo con la guía de Seguridad para Tecnologías de Virtualización, del Instituto Nacional de Estándares y Tecnología (NIST) la virtualización podría reducir el impacto de esta explotación [4.]. Bajo este precepto, el trabajo propuesto por Keller y Naues [5.], formula la implementación de un laboratorio colaborativo de seguridad utilizando máquinas virtuales. Li y Mohammed [6.], proponen la integración de las tecnologías de virtualización para la instrucción de seguridad en redes implementando un laboratorio remoto de detección de intrusiones. Otros investigadores [7.][8.], han utilizado el concepto de *Honeynet* basada en máquinas virtuales, como una herramienta de seguridad cuyo propósito es el estudio de las técnicas y motivaciones de los atacantes al romper los sistemas de seguridad. En este mismo ámbito [9.][10.][11.], han utilizado las plataformas de virtualización para recuperación de desastres y mitigación de ataques reales a redes IP.

El presente trabajo tiene como objetivo diseñar e implementar una plataforma de experimentación para evaluar ataques reales de redes IP utilizando plataformas de virtualización de libre distribución, e implementar mecanismos de control y mitigación para contrarrestarlos. Para llevarlo a cabo, se diseñó e implementó diferentes escenarios de experimentación utilizado VMware Player y VirtualBox. Luego se aplicó diversos tipos los ataques a cada escenario creado. Posteriormente se evaluó el impacto que provocan los diversos ataques analizando la información de las trazas. Finalmente se proponen mecanismos de mitigación de cada uno de estos ataques. Todo esto utilizando diversas herramientas de código abierto y de libre distribución.

Entre las principales contribuciones de esta investigación cabe mencionar: *i)* Evaluación de diversos ataques utilizando tecnologías de virtualización; y *ii)* diseñar e implementar un demonio que permita identificar el tipo de ataque, analizarlo y mitigarlo.

El resto del artículo ha sido organizado de la siguiente manera: La sección 2 presenta el marco conceptual que fundamenta esta investigación. En la sección 3 se describe el entorno en el que se desarrollaron los ataques, la configuración de la topología de pruebas y los diversos tipos de ataques evaluados. La sección 4 analiza, evalúa y discute los resultados. En la sección 5, se resume los trabajos relacionados. Finalmente en la sección 6 se establecen las conclusiones sobre la base de los resultados obtenidos y se delimita el trabajo futuro.

2. PLATAFORMA DE EXPERIMENTACION PARA ATAQUES REALES DE REDES IP

En esta sección se describen los fundamentos teóricos de este proyecto. Inicia con una conceptualización de virtualización, escenarios virtuales de red y las herramientas utilizadas para implementar la plataforma de experimentación. Posteriormente realiza un análisis y síntesis de los principales ataques que fueron evaluados durante esta investigación y de los mecanismos diseñados para contrarrestar dicho ataques.

2.1 Virtualización y Escenarios Virtuales de Red como plataforma de experimentación

La *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 [12.]. Esta tecnología permite la ejecución de múltiples máquinas virtuales y sus aplicaciones simultáneamente, siendo una gran alternativa para la implementación de escenarios virtuales de red que permiten la reproducción de la funcionalidad

de redes reales, facilitando la evaluación de múltiples ambientes de experimentación y validación de software [13.].

Un *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 determina topología desplegada sobre uno o múltiples equipos físicos, que emula un sistema equivalente y cuyo entorno deberá ser percibido como si fuera real [14.].

Para implementar los escenarios virtuales para esta investigación, se ha elegido *VMware Player 3.0 y VirtualBox* que son plataformas de libre distribución basadas en tecnología de virtualización completa que permiten la creación de máquinas virtuales X86 de 32 y 64 bits y que son muy utilizadas en la industria [15.]. En el caso de *VMware Pleyer*, porque es capaz de repartir un servidor físico en múltiples máquinas virtuales, de tal forma que múltiples sistemas operativos pueden ejecutarse sin modificación y al mismo tiempo. VMware funciona bajo Microsoft Windows, Linux, NetWare y Solaris. Con VMware se facilita el proceso de creación de máquinas virtuales en razón de la existencia de un sistema de gestión propio de máquinas virtuales [16.]. En el caso de *VirtualBox*, porque es un software que dispone de una interfaz gráfica denominada Virtual Box Manage, la misma que permite crear máquinas virtuales, definiendo sus características virtuales de memoria, disco, teclado, mouse y CDROM, así como la respectiva configuración de red. Cabe destacar que permite la ejecución de máquinas virtuales de forma remota a través del Protocolo de escritorio virtual (VRDP) [17.]

2.2 Tipos de ataques reales de redes IP evaluados

A través de los años se han desarrollado formas cada vez más sofisticadas de ataque a redes IP. Entre los más comunes y que han sido evaluados a lo largo de esta investigación se pueden describir los siguientes:

Escaneo de Puertos, que consiste en el envío de una serie de señales (paquetes), que llegan a la máquina atacada, y ésta responde reenviando otra determinada cantidad de paquetes, que el escaneador decodificará y traducirá. Dicha información consta esencialmente del número IP de la máquina atacada y datos sobre el o los puertos que se encuentran en ese momento abiertos. Suele ser la última actividad previa a la realización de un ataque, y con su ejecución el atacante consigue: el descubrimiento de direcciones IP activas, exploración de puertos TCP activos, exploración de puertos UDP activos, reconocimiento del tipo de sistema operativo del equipo como elemento de una red. La aplicación por excelencia para realizar exploración de puertos es *Nmap (Network Mapper)*[18.]

Fuerza Bruta, que es la forma de recuperar una clave probando todas las combinaciones posibles hasta encontrar aquella que permite el acceso. Consiste en generar el diccionario (hash) de todas las posibles combinaciones y compararlas con el patrón (hash) que permita el acceso [19.]. Técnicamente, el término Hash se refiere a una función o método para generar claves que representen de manera casi unívoca a un documento, registro, archivo, etc., [20.]. El objetivo de este ataque es ingresar al sistema de la víctima con credenciales (nombre de usuario y contraseña) y haciendo uso de una conexión remota (i.e., ssh, telnet, etc.), acceder a máquinas a través de una red. Para ello este tipo de ataque bombardea al servidor con nombres de usuarios y contraseñas aleatoriamente generados. Una manera eficiente de realizar ataque de fuerza bruta es mediante el uso de diccionarios de contraseñas. Los ataques tradicionales más conocidos de fuerza bruta son *Jhon the Ripper [21.] e Hydra*.

Suplantación de Identidad (Spoofing), que consiste en aplicar técnicas de suplantación de identidad generalmente con usos maliciosos o de investigación [22.]. Existen diferentes tipos como el IP spoofing, ARP spoofing, DNS spoofing, Web spoofing o e-mail spoofing. Para efecto del presente estudio nos hemos enfocado al ARP spoofing. ARP Spoofing hace referencia a la construcción de tramas de solicitud y respuesta ARP falseadas, de forma que en una red local se puede forzar a una determinada máquina a que envíe los paquetes a un host atacante en lugar de hacerlo a su destino legítimo.

Denegación de servicios (DoS), que son ataques que provocan que un servicio, equipo o recurso sea inaccesible para usuarios legítimos. Un ataque DoS puede ser perpetrado en varias formas. En esta investigación nos hemos centrado en el tradicional ataque denominado SYN Flood, que consiste en enviar mensajes TCP de petición de conexión por parte del cliente, pero sin enviar su confirmación lo cual provoca colapsos en equipos y consumo de recursos en forma desproporcionada. SYN Flood envía un flujo de paquetes TCP/SYN, muchas veces con la dirección de origen falsificada. Cada uno de los paquetes recibidos es tratado por el destino como una petición de conexión, causando que el servidor intente establecer una conexión al responder con un paquete TCP/SYN-ACK y esperando el paquete de respuesta TCP/ACK (i.e., parte del proceso de establecimiento de conexión TCP de 3 vías). Sin embargo, debido a que la dirección de origen es falsa o la dirección IP real no ha solicitado la conexión, nunca llega la respuesta [23.].

Los métodos de ataque descritos serán evaluados en los escenarios virtuales de red cuya topología será descrita en la siguiente sección.

3. CONFIGURACION DEL EXPERIMENTO

3.1. Diseño y configuración del escenario

Ante la necesidad de crear una plataforma de experimentación con diferentes escenarios para llevar a cabo los ataques reales a redes IP descritos en el apartado 2.2, se ha diseñado una topología de prueba tanto con VMware Server como con VirtualBox, aplicando las mismas condiciones y parámetros de configuración para ambas herramientas, tomado como modelos aquellos escenarios de uso más común en pequeña y mediana organización. A continuación, se ha implementado dicha topología donde los equipos involucrados, ya sean virtuales o físicos, comparten un mismo espacio de direcciones IP. La Fig. 1 representa el caso real en el cual una red LAN/WAN es sometida a ataques IP y los atacantes son usuarios de la Intranet o del Internet.

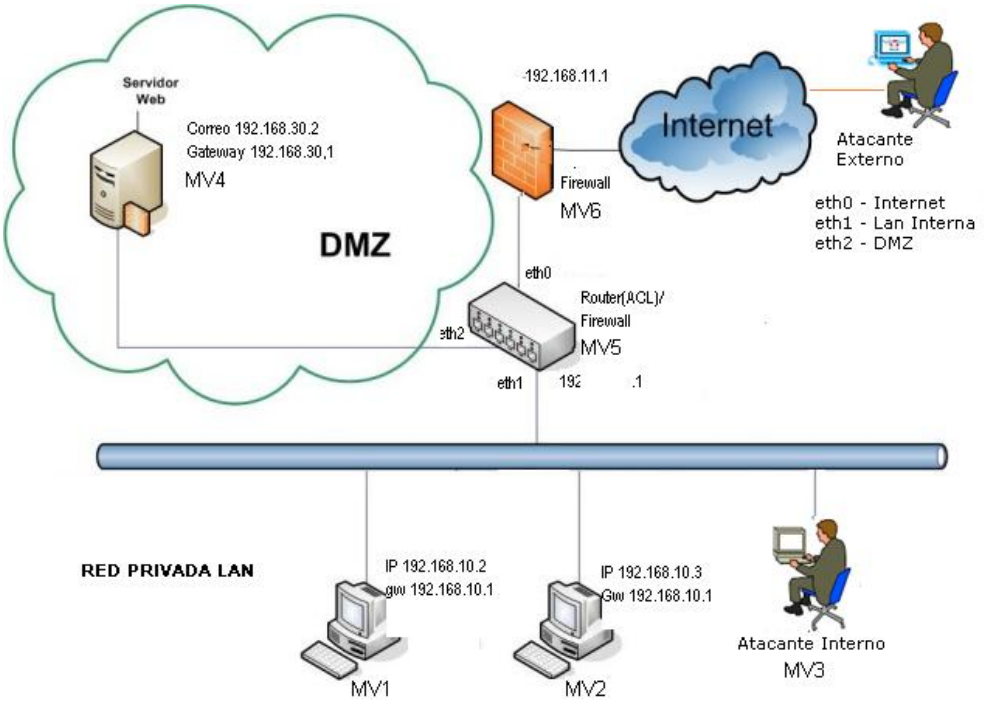

Figura 1 Diseño de la topología de prueba

3.2. Implementación del escenario

Todas las pruebas se desarrollaron sobre Linux Ubuntu Server -i386, en un computador Pentium Intel core duo, RAM de 4GB y una partición Ext3 de 120 GB en disco duro. En todas las VMs se instaló el mismo sistema de ficheros y el mismo kernel.

El procedimiento utilizado para implementar el experimento consistió en los siguientes pasos: instalación de VMware Player 3.1 y VirtualBox, creación de máquinas virtuales en cada herramienta de virtualización, direccionamiento IP, configuración de servicios Web y SSh, creación y aplicación de algoritmos para arranque automático en shell script del escenario, sincronización de reloj con NTP (Network Time Protocol), configuración del ataque y aplicación del algoritmo o aplicación de software para análisis de tráfico.

Para la captura de tráfico de los dos experimentos que se exponen a continuación, se utilizó Wireshark [24.], deshabilitando el modo promiscuo en las interfaces correspondientes. Tcpdump es una herramienta en línea de comandos cuya utilidad principal es analizar el tráfico que circula por la red. Los registros obtenidos se visualizaron mediante Wireshark [25.], que es un analizador de protocolos de red.

En las dos plataformas virtuales creadas con VMware y VirtualBox, el servidor Virtual funciona sobre Ubuntu 9.04 y constituye la máquina anfitriona que alberga a seis máquinas virtuales (MV) (Ver figura 1), en donde la MV5 con sistema operativo Ubuntu 9.10 cumple la función de enrutador IP y Firewall interno de la red local (LAN), que tiene como fin establecer los parámetros de conexión entre los dispositivos de la red privada (LAN), el servidor web y cortafuegos externo (Firewall). La MV4 hace la función de Servidor Web, DNS y Correo, el mismo que se incluye dentro de una zona desmilitarizada (DMZ), debido a que este servidor podría estar conectado a Internet maximizando los riesgos de estar expuesto a un ataque de terceros, logrando así proteger a la red interna privada (LAN). La MV6 cumple la función de cortafuego (Firewall externo), cuya función es proporcionar conexiones válidas y seguras entre los equipos de la red privada y el exterior (Internet) y viceversa. El atacante externo (físico) representa a la red mundial Internet, la misma que puede ser un simple usuario que visita la página web de la red interna (LAN) o que utiliza un servicio de correo electrónico, o simplemente un intruso que tal vez busca cambiar el contenido de una web, incluso en el caso más grave intentar hacer un ataque de denegación de servicio (DoS) o Spoofing.

El resto de máquinas virtuales, con S.O. Windows o Ubuntu, son los equipos que forman la Red Privada (LAN).

3.3. Implementación de los ataques

Para la implementación de cada uno de los ataques, fue necesario instalar algunas herramientas de libre distribución que han permitido generar los diversos ataques y que también han facilitado la captura de tráfico, tanto para Linux como para Windows. La Tabla 1 describe el tipo de ataque y las diversas herramientas utilizadas en esta plataforma de experimentación.

TABLA 1. Resumen de herramientas utilizadas para la ejecución de los ataques.

Nro. Ataque	*Descripción*	*Sistema Operativo*	*Software para el ataque*	*Software para obtener el Flujo de tráfico*
1	Rastreo de Sistemas o Escaneo de Puertos	Ubuntu	Nmap	Ettercap Wireshark
		Windows	Zenmap	Ettercap, Wireshark
2	Fuerza Bruta	Ubuntu	Medusa	
		Windows	John the Ripper	
3	Suplantación de Identidad	Ubuntu	Hping3	Wireshark
		Windows	Nemesis	Wireshark
4	Denegación de Servicio	Ubuntu	Nemesis	Ettercap,
		Windows	Ettercap	Wireshark

3.3.1 Rastreo de Sistemas (Escaneo de puertos)

La herramienta utilizada para este tipo de ataque es Nmap, la misma que permite hacer un barrido a las redes informáticas y a ordenadores, a fin de determinar que puertos tienen activos, servicios y aplicaciones en ejecución, el tipo de sistema operativo, entre otros. Adicionalmente se emplearon algunas técnicas de escaneo, entre las más conocidas son: TCP connect(), TCP SYN, TCP FIN, UDP scan y ACK scan.

Las pruebas se realizaron desde un ordenador (MV), con sistema operativo (S.O.) Linux-Ubuntu, como atacante hacia cuatro (MV) víctimas, dos con S.O. Linux y las otras con Windows XP. Igualmente se hicieron pruebas con un equipo atacante con S.O.. Windows XP, con el mismo número de víctimas (MV) y condiciones al caso anterior.

3.3.2 Ataque de fuerza bruta

Para la generación de ataques a contraseña se emplearon dos herramientas, Medusa y John The Ripper, que tienen como objetivo común obtener usuarios y contraseñas inseguras dentro de un servidor.

La aplicación John The Ripper fue utilizada en máquinas locales con S.O. Windows, requiriendo previamente la generación de un fichero de contraseñas cifradas que se obtiene para el caso de Ubuntu con el comando *unshadow* y para Windows a través del programa *pwdump*. Posteriormente el programa *Jhon the Ripper* empieza a trabajar y automáticamente va mostrando en pantalla las posibles contraseñas descifradas sobre el archivo de contraseñas. Según la robustez de las contraseñas *Jhon The Ripper* puede llegar a tardar segundos, minutos, horas, días, semanas o incluso meses en encontrar las mismas.

El programa Medusa fue utilizada para realizar ataques, a equipos remotos en una red LAN, a través de equipos con S.O. Linux-Ubuntu. Adicionalmente Medusa trabaja con un archivo diccionario que contiene un gran número de palabras que son comparadas con la contraseña de la víctima; el tiempo que se demore Medusa en crackear una contraseña, depende del tamaño del diccionario y la ubicación de la clave dentro del mismo.

3.3.3 Ataque de Suplantación de Identidad

El objetivo de esta técnica spoofing consiste en suplantar la identidad a fin de alcanzar nuestra confianza y hacernos pasar por otra máquina. Entre las técnicas de spoofing, para la realización de este tipo de ataque, se consideraron el IP Spoofing y ARP Spoofing, por ser consideradas las que más daño pueden causar a su víctima.

Para el IP Spoofing se utilizó la herramienta HPing3 a fin de lograr suplantar la IP de la máquina atacante por otra distinta.

En el caso del ARP Spoofing se hizo uso de la herramienta *Némesis* [27.], la misma que permite modificar las direcciones MAC de los equipos de una red. El objetivo de este ataque consiste en lograr por ejemplo que la MaqA capture el tráfico generado entre MaqB y MaqC, para ello realiza el envenenamiento de las tablas ARP de ambos. Para hacerlo envía un paquete ARP a MaqB diciéndole que a la IP de MaqC le corresponde la dirección MAC de MaqA y a su vez envía otro a MaqC indicando que a la IP de MaqB le corresponde la dirección MAC de MaqA.

Este tipo de ataque genera también ataques de denegación de servicio (DoS).

3.3.4 Ataque de denegación de Servicio

Como se mencionó en el apartado anterior, un ataque de suplantación de identidad del tipo IP y ARP Spoofing, generan también un ataque de denegación de servicio (DoS). Por esta razón

los datos obtenidos en el anterior ataque, del tipo ARP Spoofing, se utilizaron para el análisis de un ataque DoS,

Para la generación de este ataque se aplicó un bucle repetitivo, de inyección de paquetes maliciosos a través del programa *Nemesis*, en el equipo atacante con S.O. Windows. El propósito, de esta forma de atacar, es mantener demasiado ocupado al equipo víctima y no pueda responder a las peticiones legítimas o para denegar a los usuarios válidos el acceso a un equipo. Además se utilizó la herramienta Ettercap, desde un atacante con S.O. Ubuntu, para realizar igualmente un ataque ARP Spoofing.

3.4 Mecanismos para contrarrestar los ataques definidos

3.4.1 Demonio o administrador regular de procesos

Es un programa en segundo plano que ejecuta comandos programados en Shell scripts y tiene como objetivo programar cada cierto tiempo, a través de la configuración de crontab, la ejecución de los scripts que mitigue tanto un ataque por fuerza bruta como el de suplantación de identidad o denegación de servicios.

La configuración del contrab consiste en incluir la línea de comando que indica la ejecución de los respectivos scripts cada minuto, de todos los días, de todos los meses y durante todo el año.

3.4.2 Script para contrarrestar un ataque de fuerza bruta

Consiste en un mecanismo de autenticación script que realiza una revisión sobre el fichero auth.log de las últimas validación de autentificaciones, considerando solamente las inválidas (failure) o no válidas, las mismas que cuando supera el límite definido en el script se procede a la apertura del archivo de denegación de host (host.denny) para registrar la IP de la máquina que está intentando realizar conexiones no válidas. Una vez que se cumple éstas condiciones y se activan, se cierra la conexión a las IP capturas y por lo tanto se detiene el ataque. En la figura 2 se describe el diagrama de secuencia sobre el demonio para detectar el ataque de fuerza bruta y el bloque del mismo.

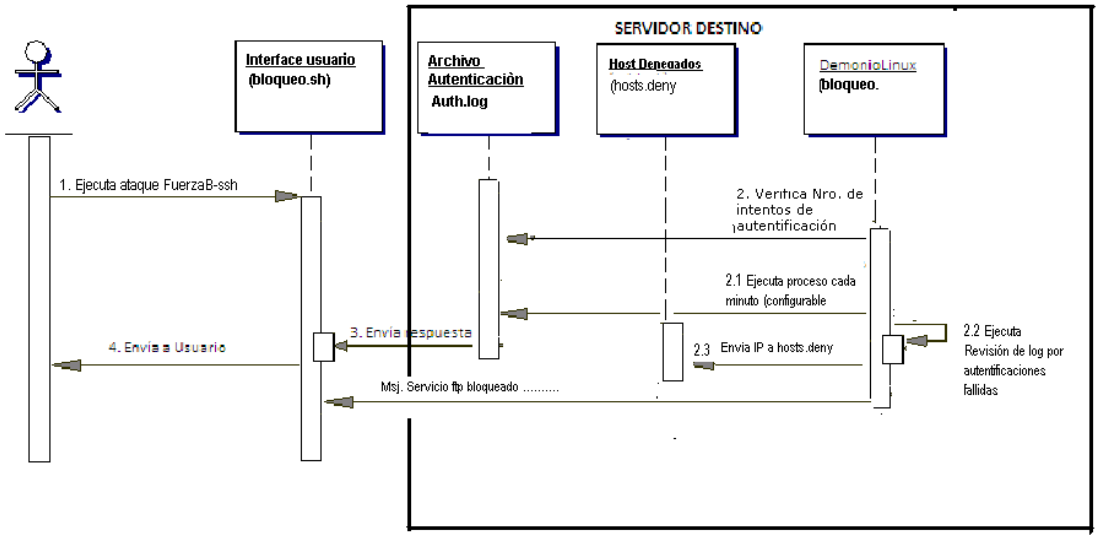

Figura. 2 Diagrama de secuencias del proceso de mitigación a un ataque de fuerza bruta.

3.4.2 Script que modifica la configuración del firewall en Ubuntu

La configuración del firewall consiste en filtrar el tráfico TCP/UDP/ICMP/IP y decidir que paquete pasa, se modifica se convierte o se descarta, todo esto se logra haciendo uso de iptables, que son cadenas formadas por agrupación de REGLAS encargadas de decir qué destino tiene un paquete. La lógica de funcionamiento optada para el cortafuego es la siguiente:

Lo primero que se hace es borrar las reglas que pudiera haber. Se ha considerado también la tabla de nat, en vista de que se usa 2 cadenas (PREROUTING y POSTROUTING) para hacer redirecciones y enmascarar la red local que pertenecen a dicha tabla. Luego se establece las políticas por defecto. Se ha puesto DROP a todo. Una vez dada las políticas, lo primero que se pone son las redirecciones, es decir, las conexiones que permitimos desde el exterior a la red privada local. En este caso se redirige al PC en cuestión las peticiones que van al puerto 80 (servidor web). Seguidamente se filtra el acceso al propio firewall permitiendo explícitamente las conexiones que se crea oportunas. Una vez realizado todo esto, se filtra en la cadena FORWARD aquellas conexiones que permitidas desde la LAN. Por el momento sólo son peticiones WEB y DNS, para que sólo se pueda navegar. Luego deniega el resto.

Teniendo todo esto configurado se procede a enmascarar la red local y habilitar el forwarding. Como los paquetes que salen de una LAN tienen una IP privada que no puede usarse en internet, es necesario algún mecanismo que le cambie por una dirección válida. Esto se lo hace con MASQUERADE.

3.4.3 Implementación del Enrutador basado en software

La implementación del enrutador, basado en software, contempla la instalación del sistema operativo Linux e instalación y configuración del paquete de enrutamiento Quagga (versión 0.99.13) para la implementación del protocolo RIP de enrutamiento dinámico.

Para alcanzar el enrutamiento dinámico, utilizando el protocolo RIP, se planteó un escenario donde se analiza la configuración del demonio ripd; el escenario se lo puede observar en la Figura 4, en donde se busca establecer la comunicación entre el usuario 1 y el usuario 2, ubicados en redes diferentes. Lla Red A representa la red LAN y la red B representa Internet, esto en relación con el escenario de la figura1.

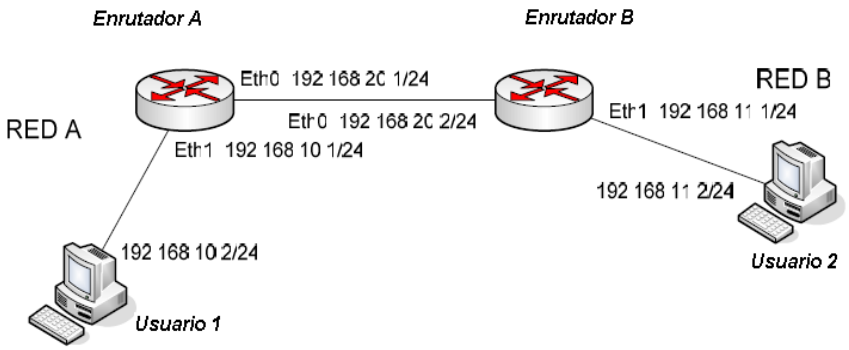

Figura 4 Escenario para el enrutamiento dinámico con RIP

4 RESULTADOS EXPERIMENTALES

4.1 Rastreo de Sistemas (Escaneo de puertos)

En la Tabla 2, tenemos un resumen de las muestras tomadas en la realización de un ataque de rastreo de sistemas (escaneo de puertos) que permitieron generar gráficas estadísticas para una mejor interpretación de los resultados.

En la gráfica de la figura 5-a, se observa que el tipo de escaneo UDP Scan, realizado desde un equipo con Linux, supera aproximadamente en 1000%, el tiempo en segundos, al resto de escaneos realizados. Esta diferencia significativa se debe a que UDP Scan utiliza paquetes UDP y no TCP como los otros tipos, generando mensajes ICMP de error que ocasiona lentitud en el ataque. En la Figura 5-b) se observa en un equipo con Ubuntu ocupa más recursos de red que un equipo con Windows, seguramente por la cantidad de paquetes y mensajes ICM que se trasmite en la conversación establecida entre el atacante y su víctima. Adicionalmente en la figura 5-c. tenemos la gráfica referente al número de paquetes capturados (enviados/recibidos) entre el atacante y su víctima, existiendo una correspondencia con el tiempo en segundos marcados por los tres primeros tipos de ataques de escaneo, lo que no ocurre con el escaneo UDP Scan que a pesar de ser lento trasmite menos paquetes que los anteriores debido a que los puertos cerrados no están obligados a responder con el envío de paquetes.

La figura 5-d muestra el histograma y la frecuencia acumulada, en donde se puede apreciar que la mayoría de equipos toma un promedio de 1,58 segundos en realizar un escaneo de puertos TCP connect,TCP SYN o TCP FIN, se excluyó el tipo de escaneo UDP debido a que el tiempo es demasiado elevado en relación a los otros tipos, impidiendo tener una mejor apreciación del tiempo que se demoran los mismos.

Tabla2. Muestra de datos referente al tiempo, recurso de red y número de paquetes E/R ocupados por un atacante de Rastreo de Sistemas.

Descripción Ataque Escaneo de Puertos		***Software para el ataque*** Nmap , Zenmap	***Muestra de Datos***											
Tipo de escaneo			**Atacante Windows**						**Atacante Ubuntu**					
Nro Tipo	**Descripción**	**Comando**	**Victima Windows**			**Victima Ubuntu**			**Victima Windows**			**Victima Ubuntu**		
			Tmp/s	**Rec/ Red (MiB/s)**	**paq. Captu.**	**Tmp/s**	**Rec/ Red (MiB/s)**	**paq. Captu.**	**Tmp/s**	**Rec/ Red (MiB/s)**	**paq. Captu.**	**Tmp/s**	**Rec/ Red (MiB/s)**	**paq. Captu.**
1	**TCP connect**	**nmap -vv -P0 -sT x.x.x.x**	**238,9**	**15,0**	**6444,0**	**213,0**	**20,0**	**6045,0**	**1,6**	**55,0**	**2025,0**	**1,4**	**57,5**	**2015,0**
2	**TCP SYN**	**nmap -vv -P0 -sS x.x.x.x**	**1,4**	**42,0**	**2084,0**	**1,4**	**39,0**	**1667,0**	**1,4**	**42,8**	**2056,0**	**1,1**	**46,8**	**2005,0**
3	**TCP FIN**	**nmap -vv -P0 -sF x.x.x.x**	**1,4**	**50,0**	**2096,0**	**1,5**	**40,0**	**1798,0**	**1,5**	**43,0**	**2087,0**	**2,3**	**26,8**	**2024,0**
4	**UDP scan**	**nmap -vv -P0 -sU x.x.x.x**	**20,6**	**11,0**	**2188,0**	**1078,7**	**10,0**	**3598,0**	**2,2**	**38,3**	**2002,0**	**1070,9**	**10,5**	**2026,0**

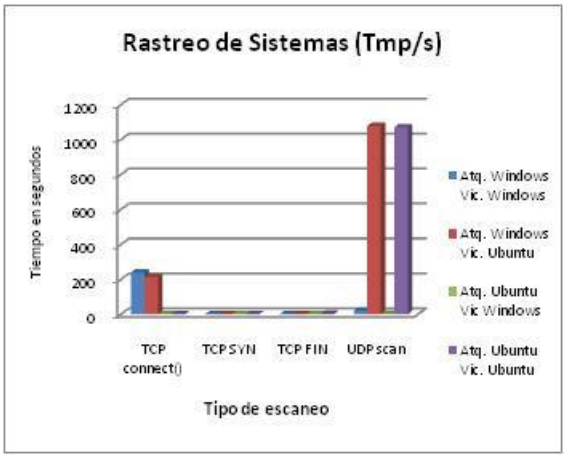

Figura 5-a) Tiempo en segundos que demora un ataque de rastreo de sistemas.

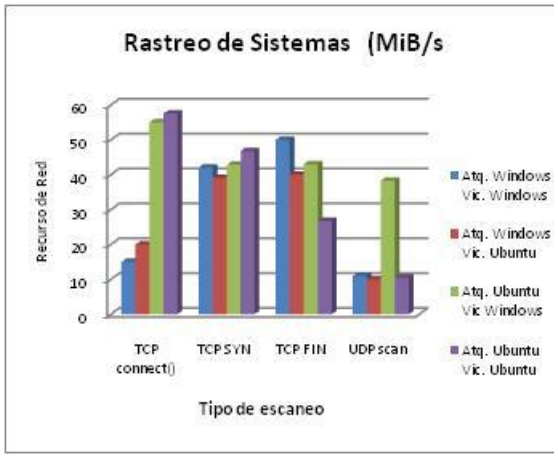

Figura 5-b) Recurso de red que ocupa un ataque de rastreo de sistemas

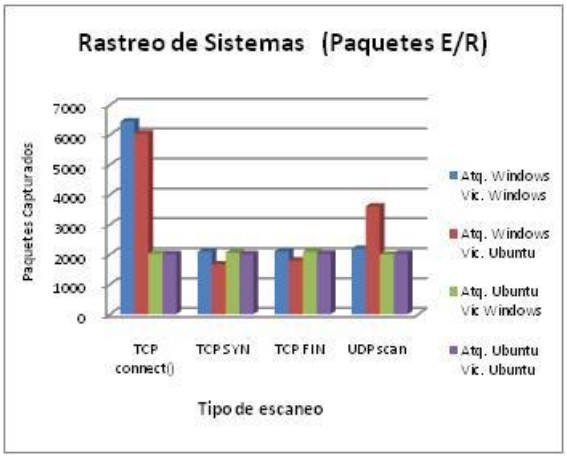

Figura 5-c) Número de paquetes capturados ante un ataque de rastreo de sistemas

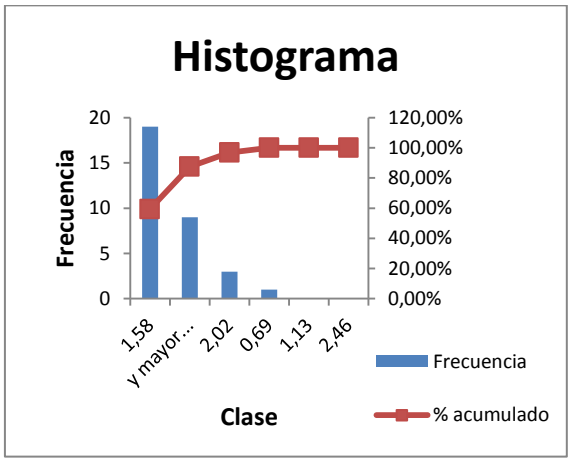

Figura 5-d) Función de distribución de probabilidad acumulada

Figura 5, Recursos que ocupa un atacante de Rastreo de Sistemas.

4.2 Fuerza Bruta

En la Tabla 3 se tiene los datos obtenidas en relación al tiempo en segundos que se toma un equipo atacante, con Windows XP, en descifrar una contraseña con John the Ripper. Las contraseñas asignadas a los equipos víctimas fueron de longitud variante entre tres y ocho caracteres alfanuméricos (igual número de letras y números). Para el caso de Medusa las pruebas fueron tomadas en función al tiempo que se demora en localizar la contraseña dentro de un fichero diccionario (inicio, mitad y final).

En la gráfica de la figura 6-a se observa que descifrar una clave con John de Ripper resulta fácil cuando la contraseña es pequeña, y toma mayor tiempo cuando la misma es más extensa y consta de caracteres alfanuméricos. Para el caso de Medusa se tomó un diccionario de contraseñas con 3500 palabras, ver figura 6-b, cuando mayor sea el archivo diccionario utilizado por el programa medusa, más posibilidades se tendrá de encontrar la contraseña.

TABLA 3: Tiempo que se tarda Jhon the Ripper y Medusa en descifrar una contraseña.

Descripción Ataque	*Software para el ataque*	***Muestra de Datos***	*Sniffer*

Fuerza Bruta		Medusa John the Ripper					wireshark
Nro.	**Ubicación Clave en diccionario**	**Comando**	Ubuntu Tmp/s	Ubuntu Rec/Red (MiB/s)	Ubuntu CPU (%)	Acierta	Detecta
Medusa							
1	Inicio	medusa -h xxx -u <user> -P <claves> -M ssh	0,5	3	90	√	√
2	Mitad	medusa -h xxx -u <user> -P <claves> -M ssh	3567	3	70	√	√
3	Final	medusa -h xxx -u <user> -P <claves> -M ssh	7332,6	3	83	√	√
John te Ripper							
Nro.	Tamaño contraseña	**Comando/clave**	Windows Tmp/s	Windows Rec/Red (MiB/s)	Windows CPU (%)	Acierta	Detecta
1	Tres	# ./john mispasswords	20	24,57	55	√	√
2	Cuatro	# ./john mispasswords	35	10,31	81	X	√
3	Cinco	# ./john mispasswords	65	15,46	72	X	√
4	Seis	# ./john mispasswords	120	25,42	90	√	√
5	Siete	# ./john mispasswords	1252	25,17	71	√	√
6	Ocho	# ./john mispasswords	10212	25,12	78	X	√

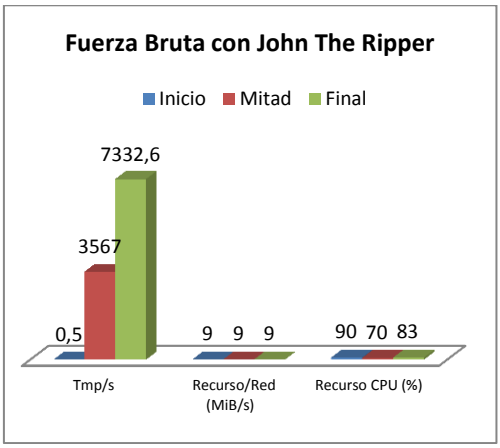

Figura 6-a) Recursos consumidos en un ataque de Fuerza Bruta con Medusa

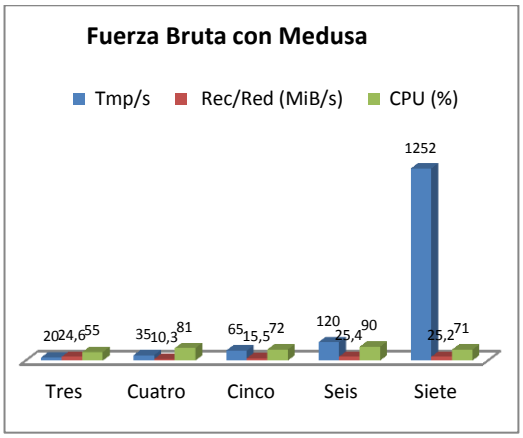

Figura 6-b) Recursos consumidos en un ataque de Fuerza Bruta con John Te Ripper

Figura 6. Tiempo que utiliza un atacante en descifrar una contraseña con Medusa y John the Ripper.

4.3 Ataque de suplantación de Identidad

La tabla 4, describe los recursos de red y de CPU que ocupa un ataque de suplantación de identidad, en vista de la rapidez con que se ejecutan el comando para este tipo de ataque, fue necesario programar el envenenamiento en la tabla ARP por 60 segundos, a fin de poder tomas las muestras correspondientes.

Durante el ataque IP Spoofing, realizado desde Ubuntu, se observó a través del programa Wireshark que la víctima recibía paquetes por parte de su atacante con dirección IP diferente a la real y que la víctima no envía respuestas (paquetes) cuando es desconocida la IP suplantada.

Durante el ataque ARP Spoofing, realizado desde Windows, ocurrió algo parecido a la suplantación de IP pero con direcciones MAC. Cabe mencionar que estos tipos de ataque spoofing generan también un ataque de denegación de Servicio.

En la Figura 7 tenemos un resume de las mediciones obtenidas sobre los recursos de red y CPU que ocupa un ataque spoofing. En la gráfica 7-a y 7-b) observamos una similitud en datos obtenidos en los dos tipos de spoofing realizados y que los recurso de red se ocupa muy poco en relación a los anteriores ataques.

TABLA 4: Datos relacionados a los recursos de Red y CPU que utiliza un ataque de Fuerza Bruta (Spoofing).

Descripción Ataque **Suplantación de Identidad** IP Spoofing		***Software para el ataque*** hping Nemesis	***Muestra de Datos***			***Sniffer*** Wireshark
Nro.	**Tiempo/s**	**Comando**	Recurso Red (KiB/s)	Recurso CPU (%)	Acierta	Detecta
IPSpoofing						
1	0	hping -1 -a x.x.x.x x.x.x.x	0,2	25	√	√
2	10	hping -1 -a x.x.x.x x.x.x.x	0.2	50	√	√
3	20	hping -1 -a x.x.x.x x.x.x.x	0,2	10	√	√
4	30	hping -1 -a x.x.x.x x.x.x.x	0,1	15	√	√
5	40	hping -1 -a x.x.x.x x.x.x.x	0,3	25	√	√
6	50	hping -1 -a x.x.x.x x.x.x.x	0.2	30	√	√
7	60	hping -1 -a x.x.x.x x.x.x.x	0.3	55	√	√
ARP Spoofing						
1	0	nemesis arp -v -d eth0 -H <MAC> -S <IP> -D <IP>	0,2	20	√	√
2	10	nemesis arp -v -d eth0 -H <MAC> -S <IP> -D <IP>	0,3	30	√	√
3	20	nemesis arp -v -d eth0 -H <MAC> -S <IP> -D <IP>	0,1	10	√	√
4	30	nemesis arp -v -d eth0 -H <MAC> -S <IP> -D <IP>	0,2	10	√	√
5	40	nemesis arp -v -d eth0 -H <MAC> -S <IP> -D <IP>	0,2	20	√	√
6	50	nemesis arp -v -d eth0 -H <MAC> -S <IP> -D <IP>	0,1	25	√	√
7	60	nemesis arp -v -d eth0 -H <MAC> -S <IP> -D <IP>	0,2	30	√	√

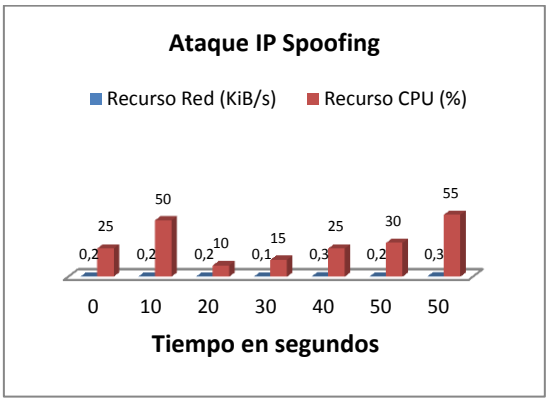

Figura 7-a) Recursos consumidos por un ataque IP Spoofing con HPing

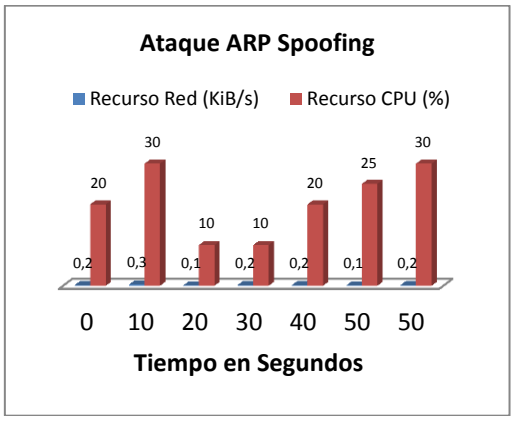

Figura 7-b) Recursos consumidos po un ataque ARP Spoofing con Némesis..

Figura 7. Recursos utilizados por un atacante se suplantación de identidad.

4.4 Ataque de Denegación de servicios

Los resultados obtenidos durante la ejecución de un ataque ARP Spoofing, a través de Ettercap, fueron que todos los host detectados quedaron sin servicio interno de Web y Correo Electrónico, y sin acceso a Internet. Algo similar se obtuvo al hacer un ARP Spoofing con Némesis en Windows, con la diferencia que se bloquea los servicios a un único equipo víctima y no a todos como ocurrió con Ettercap.

A fin de determinar el consumo de ancho de banda ante este tipo de ataque, se realizaron varias pruebas donde los equipos víctimas fueron máquinas virtuales y se tomaron 35 muestras. La figura 8, muestra los resultados obtenidos y como se puede apreciar en la gráfica los equipos víctimas ocupan un ancho de banda de 963 kbps a los 60 segundos y que el consumo de

ancho de banda no es continuo, esto se debe a que los ataques fueron realizados a varios equipos por separado.

clase	*min*	*min+tam interv*	*Frec. Acum*	*Frec*	*Frec Relatina*	*% acum*	*Clase*	*Frec*	*% acum*
1	120,00	240,43	3	4	0,03	11,43%	963,01	19	54,29%
2	240,43	360,86	8	5	0,04	25,71%	360,86	5	68,57%
3	360,86	481,29	9	1	0,01	28,57%	842,58	5	82,86%
4	481,29	601,72	9	0	0,00	28,57%	240,43	4	94,29%
5	601,72	722,15	11	2	0,02	31,43%	481,29	1	97,14%
6	722,15	842,58	16	5	0,04	45,71%	722,15	1	100,00%
7	842,58	963,01	35	19	0,16	100,00%	601,72	0	100,00%
						100,00%	y mayor	0	100,00%

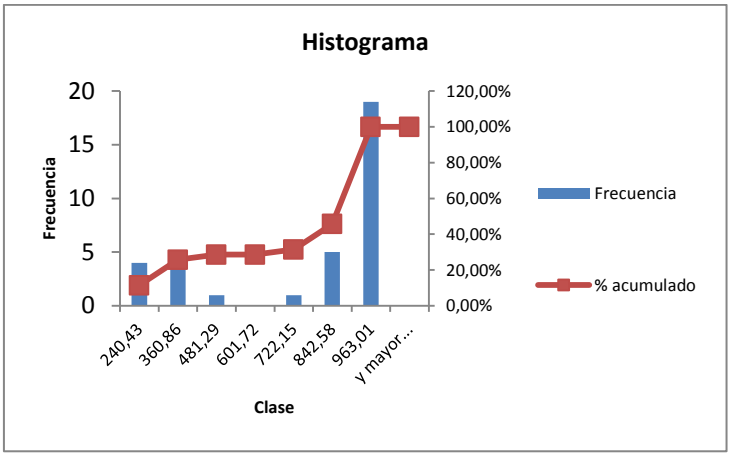

Figura 8. Histograma y % acumulado del consumo de ancho de banda ante un ataque DoS.

4.5 Evaluación de los mecanismos de detección, control y mitigación.

4.5.1 Pruebas del enrutador basado en software

La configuración de cada enrutador se realizó de forma independiente y en función del esquema de direccionamiento presentado en la figura 4, además se hicieron las pruebas tanto de conectividad como de ruta en los usuarios finales, utilizando el protocolo ICMP (Protocolo de Mensajes de Control de Internet) a través del utilitario ping, y verificando el camino que siguen los paquetes empleando el utilitario traceroute que se basa de igual manera en el protocolo ICMP. Los resultados de las pruebas se los puede observar en la Tabla 5, en donde el esquema de enrutamiento para la conectividad de extremo a extremo y la forma de direccionamiento se ejecutó con éxito en las pruebas realizadas.

Enrutamiento	**Protocolo**	**Demonio**	**Direccionamiento**	**Prueba Ping**	**Prueba Traceroute**
Dinámico	RIP	Ripd	IPv4	OK	OK

Tabla 5. Resultados de las pruebas de conectividad y de ruta

5.2.1 Ataques a una red una vez implementado los mecanismos de mitigación

En la Tabla 6, se observa el resultado de realizar ataques de fuerza bruta, escaneo de puertos, suplantación y denegación de servicio, dentro del esquema descrito en la Tabla 3. Hay que mencionar que en el escenario creado se instaló y se configuró un firewall Interno (dentro de la red privada LAN), un demonio para detectar y terminar con un ataque de fuerza bruta y un segundo firewall externo (entre la red LAN e Internet), todos estos como mecanismo para prevenir, detectar y contrarrestar un ataque a la red informática. Una vez revisado y analizados los resultados obtenidos se logra mitigar la mayoría de los ataques considerados para la realización del presente trabajo.

Nro. Ataque	***Descripción***	**Aciertos**		**Tiempo**		**Recurso de Red**		***Mecanismo para contrarestar el ataque***
		Obuntu	***Window***	***Sistema Operativo***		***Sistema Operativo***		
1	**Escaneo de Puertos**	***Nmap***	***Zenmap***	***Obuntu***	***Window***	***Obuntu***	***Window***	Firewall
		Fallido	Fallido	supera el minuto	supera el minuto	No hay variaciòn	No hay variaciòn	
2	**Fuerza Bruta**	***Medusa***	***John the Ripper***	***Obuntu***	***Window***	***Ubuntu***	***Windows***	Demonio (Script)
		Fallido	Éxito	supera el minuto	variable en función del tamaño del password	No hay variaciòn	No hay variaciòn	
3	**Denegación de Servicio**	***Nemesis***	***Ettercap***	***Obuntu***	***Window***	***Ubuntu***	***Windows***	Firewall
		Fallido	Fallido	0,1 segundos	0,15 segundos	No hay variaciòn	No hay variaciòn	
4	**Spoofing**	***HPING***	***Nemesis***	***Obuntu***	***Window***	***Ubuntu***	***Windows***	Firewall
		Fallido	Fallido	0,1 segundos	0,15 segundos	No hay variaciòn	No hay variaciòn	Firewall

TABLA 6.-Ataques de Rastreo de Sistemas, Fuerza Bruta, Suplantación de Identidad y Denegación de Servicios, sobre un escenario con mecanismos para prevenir y mitigar dichos ataques.

5 DISCUSION

Una de las mayores ventajas presentadas en el desarrollo de presente trabajo fue la creación de un escenario virtual, a través de la tecnología de virtualización, en donde se pudo reproducir la funcionalidad de una red teleinformática real y facilitó la evaluación de los ambientes de experimentación y validación de software relacionados con los ataques a una red como son: el Analizador de Sistemas (escaneo de puertos), Fuerza Bruta, Suplantación de Identidad y el ataque de Denegación de Servicios. Con este escenario virtual se pudo hacer todas las pruebas del caso, ahorrando tiempo y espacio, en comparación con un escenario de red con equipos reales.

Una de las desventajas que se tuvo fue la adquisición de aplicaciones de libre distribución para las MV con Windows, que brinden las mismas facilidades que las aplicaciones de libre distribución para Ubuntu. Ante este inconveniente casi todos los escenarios creados funcionan sobre Ubuntu y las MV con Windows simplemente fueron consideradas como un equipo más de

la red sobre la cual se realizaron los respectivos ataques.

Entre una de las dificultades que se presentó durante el desarrollo de éste trabajo fue la poca experiencia que se tuvo sobre el tema, las herramientas y aplicaciones utilizadas, lo que ocasionó que se vuelva a realizar, más de una vez, los experimentos por la descubrimiento de nuevas y mejores herramientas.

De todo lo realizado en el presente trabajo de investigación se puede decir, con la seguridad del caso, que todo el esfuerzo y dedicación sirvió para conocer que tan vulnerable es una red a ciertos ataques y que mecanismos se puede implementar para prevenir y mitigar dichos ataques. Sobre todo fue una gran oportunidad para conocer más en detalle el sistema operativo Ubuntu y comprender el por qué? se están inclinando las organizaciones por utilizar Linux.

6 TRABAJOS RELACIONADOS.

Aunque exista una diversidad de trabajos relacionados, en esta sección se han incluido los más relevantes, que se han encontrado durante la investigación:

En lo que se refiere al ámbito educativo, el trabajo desarrollado por Keller y Naues [5.], expone la implementación de un laboratorio colaborativo de seguridad utilizando máquinas virtuales. Esta investigación permite realizar las tareas de administración de seguridad mediante un Shell remoto, además cuenta con otra interfaz web que permite saber los resultados de su práctica de laboratorio, y tareas pendientes. En este mismo ámbito, Li y Mohammed [6.], proponen la integración de las tecnologías de virtualización para la instrucción de seguridad en redes implementando un laboratorio remoto de detección de intrusiones. Adicionalmente, [7.][8.] han utilizado el concepto de *Honeynet* basada en máquinas virtuales, como una herramienta de seguridad cuyo propósito es el estudio de las técnicas y motivaciones de los atacantes al romper los sistemas de seguridad. En este mismo contexto, El trabajo propuesto por Damiani [9.], describe un laboratorio virtual basado en tecnología de código abierto, utilizando la plataforma Xen, que tiene como objetivo la configuración de un firewall para proteger el servidor de ataques externos mediante Iptables. Todos estos trabajos han sido utilizados como insumos en esta investigación.

En relación a soluciones de recuperación de desastres mediante virtualización, el trabajo propuesto por [10.], demuestra que el uso de esta tecnología como una opción, debido a que minimizan el uso de servidores y liberan a los administradores del hecho de tener el mismo ambiente de hardware que los servidores en operación, representando una mayor flexibilidad y costos mucho menores de mantenimiento y administración.

En un contexto más cercano al nuestro, el trabajo propuesto por Ferrie [11.], utilizó código malicioso y ataques de denegación de servicio contra máquinas virtuales VMware, VirtualPC, Paralles e Hydra. Sin embargo en este estudio solo se recomiendan pero no se han desarrollado soluciones tangibles. Comparando este trabajo con el nuestro existen dos diferencias fundamentales, la primera hemos realizado la evaluación de diversos ataques de redes y hemos desarrollado e implementado un demonio que permita detectar, controlar y mitigar los ataques evaluados.

7 CONCLUSIONES

La presente investigación se enfocó en la evaluación de diversos ataques reales de redes IP utilizando plataformas de virtualización. Durante esta investigación, se diseñó e implementó varias topologías de experimentación basadas en entornos virtuales de red. Los tipos de ataques evaluados por ser tradicionales fueron escaneo de puertos, fuerza bruta, suplantación de identidad y denegación de servicios, tanto en una red de área local como en una extendida. Para cada topología, se utilizó diferente software libre tanto para producir el ataque como para obtener el flujo de tráfico, evaluándose las consecuencias del ataque. Para contrarrestar dichos ataques, se desarrolló un demonio en Shell script que detectó, controló y mitigó los ataques

mencionados de manera programable y constante. Los resultados redujeron las amenazas y vulnerabilidades de los ataques en redes en producción.

Como trabajo futuro se planea evaluar ataques distribuidos de denegación de servicio, utilizando otros mecanismos de mitigación como la encriptación, sistemas de detección de intrusos y VPNs en un entorno de red virtualizado.

En la actualidad existen muchos sistemas de seguridad que requieren herramientas sofisticadas, inversión en equipos de seguridad, entre otros. Pero asimismo existen personas que buscan vulnerabilidades sean éstos hackers, crackers o afines, que poseen una infinidad de herramientas, siendo prácticamente imposible proteger un sistema en su totalidad. Por lo tanto no existe sistema o mecanismo alguno que pueda considerarse cien por ciento seguro.

AGRADECIMIENTOS

Este trabajo ha sido financiado por el Vicerrectorado de Investigación y Vinculación con la colectividad de la Escuela Politécnica del Ejército de Ecuador, en el marco de los proyectos de iniciación científica. Los autores desean agradecer a los Departamentos de Ciencias de la Computación, Eléctrica y Electrónica por la asignación de laboratorios durante la investigación.

Referencias Bibliográficas

[1.] H. Tipton, M. Krause, "Information Security Management Handbook", Auerbach Publications. Fifth Edition. ISBN: 08493-1997-8

[2.] S. Garfinkel with Gene Spafford Web Security, Privacy & Comerse.. O'Really. Second Edition. ISBN 0-596000-456

[3.] W. Fuertes, J. E. Lopez de Vergara, F. Meneses, "Educational Platform using Virtualization Technologies: Teaching-Learning Applications and Research Uses Cases", In proceedings of II ACE Seminar: Knowledge Construction in Online Collaborative Communities, Albuquerque, NM - USA, October 2009.

[4.] K. Scarfone, M. Souppaya, P. Hoffman, "Guide to Security for Full Virtualization Technologies (Draft)", Special Publication 800-125 Recommendations of the National Institute of Standards and Technology, Gaithersburg, MD 20899-8930, July 2010.

[5.] J. Keller, R. Naues, "A Collaborative Virtual Computer Security Lab," e-science, In Proc. Second IEEE International Conference on e-Science and Grid Computing, pp. 126, CA, USA, Dec. 2006

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

[7.] F. Abbasi, R. Harris, "Experiences with a Generation III virtual Honeynet", In Proccedings of the Telecommunication Networks and Applications Conference (ATNAC), 2009 Australasian, Canberra, ACT , ISBN: 978-1-4244-7323-6. May 2009.

[8.] Fermín Galán, David Fernández, "Use of VNUML in Virtual Honeynets Deployment", IX Reunión Española sobre Criptología y Seguridad de la Información (RECSI), Barcelona (Spain), pp. 600-615, September 2006. ISBN: 84-9788-502-3.

[9.] 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.

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

[11.] P. Ferrie, Attacks on Virtual Machine Emulators, Symantec White Paper, 2008.

[12.] F. Galán, D. Fernández, W. Fuertes, M. Gómez and J. E. López de Vergara, "Scenario-based virtual network infrastructure management in research and educationnal testbeds with VNUML,", Annals of Telecommunications, vol. 64(5), pp. 305-323, May 2009.

[13.] Matthews, J., Hapuarachi, W., Deshane, Hu, M. T., Quantifying the Performance Isolation Properties of Virtualization Systems. In Proc. of Workshop on Experimental computer science ExpCS'07, 13–14 June, 2007, San Diego, CA.

[14.] 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 NVWS'09, Kassel-Germany, March 2009.

[15.] W. Fuertes and J. E. López de Vergara, "A quantitative comparison of virtual network environments based on performance measurements,", in Proceedings of the 14th HP Software University Association Workshop, Garching, Munich, Germany, 8-11 July 2007.
[16.] VMware home page, [Online:] http://www.vmware.com
[17.] VirtualBox home page [Online:] http://www.virtualbox.org
[18.] C. Lee, C. Roedel, E. Silenock, "Detection and Characterization of Port Scan Attacks", [Online:] "http://cseweb.ucsd.edu/users/clbailey/PortScans.pdf
[19.] Hacking: VII Ataques por Fuerza Bruta. [Online:]: http://jbercero.com/index.php?option=com_content&view=article&id=71:hacking-vii-ataques-por-fuerza-bruta&catid=40:hacking-tecnicas-y-contramedidas&Itemid=66
[20.] Laboratorios: Hacking, Técnicas y contramedidas, Ataques por fuerza bruta (Brute Force) III. [Online:] http://labs.dragonjar.org/laboratorios-hacking-tecnicas- fuerza-bruta-brute-force-iii
[21.] Jhon the Ripper 1.7.6., [Online:] www.openwall.com/jhon/
[22.] F. Callegati, W. Cerroni, and M. Ramilli, "Man-in-the-middle attack to the HTTPS protocol," IEEE Security and Privacy, vol. 7, no. 1, pp. 78–81, 2009
[23.] J. Li, N. Li, X. Wang, and T. Yu. Denial of Service Attacks and Defenses in Decentralized Trust Management. In ACM CCS, 2006.
[24.] Jacobson, V., Leres, C., and McCanne, S. Tcpdump. Available at anonymous@ftp.ee.lbl.gov
[25.] Wirewshark: http://www.wireshark.org/. Ultima comprobación, Jul. 2010.
[26.] Nmap, www.nmap.org. Ultima comprobación Octubre de 2010.
[27.] Nemesis, http://nemesis.sourceforge.net/. Ultima comprobación, 20 de octubre de 2010.
[28.] Ethercap, http://ettercap.sourceforge.net/. Ultima comprobación, 21 de octubre de 2010