ESCUELA POLITÉCNICA DEL EJÉRCITO

VICERRECTORADO DE INVESTIGACIÓN Y VINCULACIÓN CON LA COLECTIVIDAD

DEPARTAMENTO DE ELËCTRICA Y ELECTRÖNICA

PROGRAMA DE MAESTRÍA EN REDES DE INFORMACION Y CONECTIVIDAD

MRIC-I

TESIS DE GRADO

PLATAFORMA DE EXPERIMENTACIÓN DE ATAQUES REALES A REDES IP UTILIZANDO TECNOLOGÍAS DE VIRTUALIZACIÓN

LINA PATRICIA ZAPATA MOLINA

Sangolquí, 2012

AUTORIZACIÓN

Al presentar esta tesis como uno de los requisitos previos para la obtención del grado de magister de la Escuela Politécnica del Ejército, autorizo a la biblioteca de la ESPE para que haga de esta tesis un documento disponible para su lectura según las normas de la institución.

Estoy de acuerdo en que se realice cualquier copia de esta tesis dentro de las regulaciones internas de la ESPE, siempre y cuando esta reproducción no suponga una ganancia potencial.

Sin perjuicio de ejercer mi derecho de autor, autorizo a la ESPE la publicación de esta tesis, o de parte de ella, por una sola vez dentro de los treinta meses después de su aprobación.

Ing. Lina Patricia Zapata Molina

Sangolquí, julio del 2012

CERTIFICACION

Certifico que la elaboración de la presente tesis fue realizada en su totalidad por la ing. Lina Patricia Zapata Molina, como requisito previo a la obtención del título de MAGISTER EN REDES DE INFORMACION Y CONECTIVIDAD

_____

DIRECTOR

Ing. Walter Fuertes Díaz, PhD.

DIRECTOR DE LA UNIDAD DE GESTION DE POSTGRADO

AGRADECIMIENTO

En primer lugar quiero agradecer a Dios por proveer los medios necesarios para la consecución de mi tesis.

A mi esposo Francisco y mis hijas Diana e Iveth, quienes me apoyaron incondicionalmente en todo momento y han tenido mucha paciencia durante todos estos años. Gracias por su ayuda fundamental para conseguir esta gran meta.

A mis padres, y hermanos, que siempre me han animado a seguir adelante con mis estudios.

Finalmente quiero agradecer al Ing. Walter Fuertes PhD por su guía, compresión y ayuda generosa en todo momento.

Lina Patricia

DEDICATORIA

Este trabajo va dedicado a mi familia amorosa que es mi fortaleza, mi razón de ser y apoyo completo e incondicional en todo momento.

Además se lo dedico a todas aquellas personas que creyeron en mí, y apoyaron las dediciones tomadas por mi persona, durante la realización del mismo.

Lina Patricia

RESUMEN

El continuo aparecimiento de diversas amenazas, vulnerabilidades y tipos de ataques que implican hurto, modificación, espionaje, interrupción, falsificación, denegación de servicios etc., en las redes TCP/IP, perjudica directamente a los negocios que son altamente dependientes de sus sistemas y redes de información.

Para prevenir y contrarrestar una amplia gama de amenazas a las seguridades de las redes TCP/IP, es necesario conocer sus vulnerabilidades 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, a través de la utilización de 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.

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 dos escenarios de experimentación utilizado VMware Player y VirtualBox. Luego se aplicó diversos tipos de 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.

ÍNDICE

CAPÍTULO I

INTRODUCCIÓN

CAPÍTULO II

MARCO TEÓRICO

CAPÍTULO III

PLATAFORMA EXPERIMENTAL BASADA EN TECNOLOGÍAS DE VIRTUALIZACIÓN

CAPÍTULO IV

EMULACIÓN DE ATAQUES EN UN ENTORNO VIRTUAL DE RED

CAPÍTULO V

EVALUACION DE RESULTADOS

CAPÍTULO VI

CONCLUSIONES Y RECOMENDACIONES

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

GLOSARIO

Address Resolution Protocol (ARP): protocolo de la familia TCP/IP que asocia direcciones IP a direcciones MAC.

Denegación de servicio (DoS): ataque que hace una apropiación exclusiva de un recurso o servicio con la intención de evitar cualquier acceso a terceras partes. En inglés, deny of service.

Desbordamiento de buffer: posibilidad de corromper la pila de ejecución para modificar el valor de retorno de una llamada a función y provocar la ejecución de código arbitrario.

Dirección IP: Dirección definida por el Protocolo Internet.

DNS: ("Domain Name System", Sistema de Nombre de Dominio). Es una base de datos distribuida y jerárquica que almacena información asociada a nombres de dominio en redes como Internet, el DNS como base de datos es capaz de asociar diferentes tipos de información a cada nombre, los usos más comunes son la asignación de nombres de dominio a direcciones IP y la localización de los servidores de correo electrónico de cada dominio.

Enrutamiento: Es el mecanismo por el cual los paquetes de información viajan desde su origen hasta su destino, siguiendo un camino o ruta a través de la red.

Gateway: (Puerta de Enlace). Es un dispositivo que permite interconectar redes con protocolos y arquitecturas diferentes a todos los niveles de comunicación, tiene como propósito traducir información del protocolo utilizado en una red al protocolo usado en la red destino.

IPv6: Nomenclatura de la nueva generación de Internet, aporta más velocidad, mayor facilidad de uso y mejora la seguridad de acceso a la información.

ISO: ("International Oganization for Standardization", Organización Internacional para la normalización).

Enrutamiento: Es el mecanismo por el cual los paquetes de información viajan desde su origen hasta su destino, siguiendo un camino o ruta a través de la red.

Hacker: Persona que por diversos motivos irrumpe en un sistema de información.

Huella identificativa: información muy precisa que permite identificar un sistema o una red en concreto. En inglés, fingerprinting.

Escáner de vulnerabilidades: aplicación que permite comprobar si un sistema es vulnerable a un conjunto de deficiencias de seguridad.

Exploit: aplicación, generalmente escrita en C o ensamblador, que fuerza las condiciones necesarias para aprovecharse de un error de programación que permite vulnerar su seguridad.

Exploración de puertos: técnica utilizada para identificar los servicios que ofrece un sistema.

Explotación de un servicio: actividad realizada por un atacante para conseguir una escalada de privilegios, abusando de alguna deficiencia del servicio o del sistema.

Fingerprinting: ver Huella identificativa.

Firewall: ver Cortafuegos.

Fragmentación IP: proceso de división de un datagrama IP en fragmentos de menor longitud.

Internet Control Message Protocol (ICMP): protocolo encargado de realizar el control de flujo de los datagramas IP que circulan por la red.

Internet Protocol (IP): protocolo para la interconexión de redes.

IP flooding: ataque de denegación de servicio basado en una saturación de la red mediante la generación masiva de datagramas IP.

Maxim Transfer Unit (MTU): medida máxima de un datagrama IP dentro de una red.

Requests for Comments: conjunto de documentos técnicos y notas organizativas sobre internet.

Reensemblado IP: proceso de reconstrucción de un datagrama IP a partir de sus fragmentos.

Rootkit: recopilación de herramientas utilizadas en un ataque de intrusión para garantizar la ocultación de huellas, garantizar futuras conexiones, realizar otros ataques al sistema, etc.

Shellcode: código ensamblador inyectado en memoria que un exploit tratará de ejecutar.

Sniffer: aplicación que intercepta toda la información que pase por la interfaz de red a la que esté asociado.

SYN Flooding: ataque de denegación de servicio que se basa en no complementar intencionadamente el protocolo de intercambio de TCP.

Cortafuegos: elemento de prevención que realizará un control de acceso para proteger una red de los equipos del exterior (potencialmente hostiles).

Tablas de ruteo: Rutas o caminos a seguir por los paquetes que direccionan los Routers mediante algoritmos de trayectos.

Transmission Control Protocol (TCP): protocolo de transporte de la arquitectura de protocolos TCP/IP.

User Datagram Protocol (UDP): protocolo de transporte de la arquitectura de protocolos TCP/IP.

INTRODUCCIÓN

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, utilizando para ello 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.].

La documentación del presente trabajo ha sido organizado de la siguiente manera: El capítulo uno presenta el planteamiento del problema, una descripción de los objetivos planteados junto a las contribuciones alcanzadas con la realización del presente proyecto de tesis. En el capítulo dos se presenta el marco conceptual que fundamenta esta investigación. El capítulo tres describe el entorno en el que se desarrollaron los ataques, la configuración de la topología de pruebas y los diversos tipos de ataques evaluados. El capítulo cuatro describe la implementación del escenario de pruebas, la ejecución de diversos ataques y los resultados obtenidos, así como también la descripción de los mecanismos para contrarrestar dichos ataques. En el capítulo cinco se analiza y se evalúa los resultados obtenidos sobre los ataques ejecutados sin mecanismos de mitigación y con mecanismos de prevención implementados. Finalmente en el capítulo seis se establecen las conclusiones y recomendaciones obtenidas durante el desarrollo del presente trabajo.

CAPÍTULO I

INTRODUCCIÓN

1.1. Motivación

La dependencia cada vez mayor del mundo informático, obliga a que se tome mayor atención al tema de seguridad y confidencialidad en el uso de los servicios informáticos disponibles en Internet.

El problema de la seguridad radica en la existencia de vulnerabilidades en la gran variedad de servicios y utilidades que ofrece Internet, y es en donde los hackers sacan su mayor partido y aprovechan los más mínimos fallos de seguridad para cometer sus actos delictivos [1.]. Ante esto, la mayoría de las organizaciones trabajan día a día para proporcionar a sus usuarios datos e información con las mayores medidas de protección, a fin conservar y/o mejorar la imagen, credibilidad y en consecuencia ganar más clientes.

Evidentemente por desconocimiento de las normas básicas de seguridad, o por comodidad de ciertos administradores de sistemas no tienen sus equipos actualizados y protegidos, en el manejo de datos e información confidencial que puede llevar a diversas situaciones comprometedoras. Así se tiene el caso más sencillo y cotidiano, la simple infección de un virus informático del cual se puede convertir en una plataforma de expansión ocasionando daños irreversibles cuando se cae en cuenta de la infección [2.]. Un caso extremo es el ataque de un hackers que puede utilizar la PC víctima como escudo y plataforma para realizar conexiones ilegales a sitios web, direcciones electrónicas, servidores, entre otros, pudiendo comprometer al dueño de la PC en un serio problema legal.

El mundo de la seguridad informática no es sencillo, pues evoluciona muy rápido, en consideración a las dos posturas, la del experto en seguridades, y la del hacker o intruso, pues las dos partes intentarán conocer al máximo los sistemas para, el uno protegerlos, y el otro

atacarlos. Sobre quien alcance su objetivo dependerá que tan preparado e informado esté sobre el sistema en cuestión

1.2. Planteamiento del Problema

1.2.1. Descripción del problema

Los ataques a las redes de información se han venido incrementando en los últimos años, muchos de ellos difíciles de detectar. Algunos de estos ataques son los virus, gusanos, troyanos, ataques de denegación de servicio, modificación, intercepción, spaming, phishing, spyware, malware entre otros. Esto se debe al crecimiento exponencial que el Internet ha tenido en los últimos diez años así como a la utilización masiva de las redes de información.

El problema de la seguridad radica en la existencia de vulnerabilidades y amenazas informáticas que existen alrededor de las redes pero sobre todo de los servidores Web, que en ocasiones podrán terminar con la información o los equipos de una empresa, dando como resultado la pérdida de la imagen, credibilidad y en consecuencia pérdida de clientes.

Para realizar pruebas y emulaciones de ataques existe la alternativa de usar las plataformas de Virtualización que permiten detectar y contrarrestar los ataques en la red y los resultados obtenidos podrían son muy aproximados a los derivados en un escenario real, ya que se trata de una aplicación existente y no solamente un modelo simulado[3.].

Por lo expuesto anteriormente la realización de esta tesis, tiene como fin diseñar una plataforma de experimentación de ataques a la red IP utilizando tecnologías de Virtualización y definir un mecanismo para enfrentarlos.

1.2.2. Pregunta de la Investigación

¿Qué tipos de ataques a las redes IP son más frecuentes?

¿Cómo emular un ataque real utilizando tecnologías de Virtualización?

¿Qué mecanismo se puede utilizar para prevenirlo y neutralizarlo?

1.3. Justificación e Importancia

Las plataformas de Virtualización son tecnologías potenciales para reproducir redes reales en escenarios virtuales. Estas plataformas permiten ejecutar y probar múltiples ambientes de validación de software y brinda facilidades para realizar el dimensionado de prestación de servicios de redes. Las principales ventajas de la Virtualización son: Ahorro de costos de inversión, simplificación de la gestión dada la administración centralizada de todas las VMs, portabilidad entre servidores físicos, facilidad de técnicas de recuperación de desastres. Algunas desventajas son: falta de estandarización; dificultad en el acceso a la red de información del host desde las VMs y la introducción de una penalización u overhead debido al consumo de recursos reduciendo su rendimiento.

La investigación planteada en este proyecto cobra importancia debido a que permitirá analizar un problema actual y real, existente en las redes de información, mediante la utilización de la tecnología de Virtualización, que es una tecnología disruptiva para la presente década.

Este proyecto se desarrollará con plataformas de Virtualización de libre distribución, a fin de emular diversos ataques a redes IP.

Los principales beneficios que se pretende alcanzar son:

1. Desde el punto de vista investigativo, el diseño y configuración de plataformas virtuales para experimentación, con las mismas funcionalidades de una red real, tendrá el consiguiente ahorro en gastos en equipos y dispositivos de red

2. El análisis y evaluación de las vulnerabilidades en la red permitirá identificar algunos tipos de ataques a la red y pretende identificar medidas de seguridad preventivas para contrarrestarlos

3. Tecnológicamente el presente estudio de ambientes virtuales servirá como base para otras emulaciones o pruebas de temas relacionados con redes. Además se busca contribuir, de esta manera, para investigaciones futuras.

4. Con los resultados de esta investigación se pretende contribuir al conocimiento, por ende se aspira que sean susceptibles de publicación, lo que coadyuvará en incrementar la producción técnica-científica de la ESPE.

1.4 Objetivos

1.4.1. Objetivo General

Crear una plataforma informática de experimentación, en ambiente Linux, que permita emular ataques a redes IP mediante la aplicación de tecnologías de Virtualización con el fin de implementar mecanismos de seguridad para contrarrestarlo.

1.4.2 Objetivos específicos

Estudiar el Estado del Arte relacionado, en primer lugar, con las seguridades en las redes, particularizando el estudio sobre rastreo de máquinas, servicios, ataque a servidores, ataques a estaciones de trabajo, ataques de negación de servicio, código malicioso. En segundo lugar relacionado con las tecnologías de Virtualización:

- Diseñar e implementar varias topologías de prueba con el fin de emular ataques reales a redes en ambiente Linux, utilizando escenarios de redes virtuales.
- Investigar herramientas existentes para evaluar la vulnerabilidad de una red.
- Evaluar los daños causados por los ataques a la red virtual
- Diseñar un mecanismo de protección ante los ataques reales a redes Linux.

1.5 Contribuciones de la Tesis

El problema de la seguridad radica en la existencia de vulnerabilidades y amenazas informáticas que existen alrededor de las redes pero sobre todo de los servidores Web, que en ocasiones podrán terminar con la información o los equipos de una empresa, dando como resultado la pérdida de la imagen, credibilidad y en consecuencia pérdida de clientes.

El presente proyecto pretende crear una plataforma de experimentación de ataques reales a redes IP, configuración de topologías de red virtuales, programación de scripts y utilización de programas de libre distribución existentes en el mercado.

Con lo antes expuesto se dará cumplimiento a los objetivos planteados, generando las siguientes contribuciones:

- Desde el punto de vista investigativo, el diseño y configuración de plataformas virtuales para experimentación, con las mismas funcionalidades de una red real, tendrá el consiguiente ahorro en gastos en equipos y dispositivos de red

- El análisis y evaluación de las vulnerabilidades en la red permitirá identificar algunos tipos de ataques a ella y pretende identificar medidas de seguridad preventivas para contrarrestarlos

- Tecnológicamente el presente estudio de ambientes virtuales servirá como base para otras emulaciones o pruebas de temas relacionados con redes, contribuyendo de esta manera, para investigaciones futuras.

- Con los resultados de esta investigación se pretende contribuir al conocimiento, por ende aspira que sean susceptibles de publicación, lo que coadyuvará en incrementar la producción técnica-científica de la ESPE.

CAPÍTULO II

MARCO TEÓRICO

2.1. Seguridades en Redes de Información

2.1.1. Fundamentos de seguridades en redes IP

Las redes y en particular Internet, han introducido nuevas aplicaciones que también implican riesgos. Estos surgen a partir de las vulnerabilidades que poseen los sistemas. La existencia de vulnerabilidades implica amenazas, cuya concreción son los ataques. Estos ataques a las redes pueden dejar inoperativos los recursos y causar por tanto pérdidas económicas a las organizaciones, además de exponer a intrusos datos que pueden ser privados.

Internet está plagada de personas que se dedican a robar y destruir todo tipo de información. Estos pueden ser gente de la misma empresa o externa a la institución que tiene fácil acceso a la red y también existen personas que logran violar la seguridad de la red. Estas personas son muy peligrosas ya que pueden robar cualquier información que esté en las computadoras y los usuarios o administradores pueden darse cuenta del ataque después de varios días, semanas o meses.

A estas personas se les conoce como hackers. Estos son personas que, ya sea por dinero o placer, logran quebrantar la seguridad de los equipos, bajar información confidencial y robarla o destruirla.

Por lo expuesto anteriormente, es preciso contar con políticas de seguridad, normas que, directa o indirectamente, permiten discernir los eventos y acciones permitidos o prohibidos para un sistema, desde el punto de vista de su seguridad. En el presente proyecto

se hace referencia al nivel de seguridad informática lógica, más no a la seguridad física de equipos y dispositivos[1].

2.1.1.1. Ataque o intrusión a una red ip

Se denomina intruso o atacante a la persona que acceda o intente acceder sin autorización a un sistema informático ajeno, ya sea en forma premeditada o no.

En cualquier sistema informático existe tres elementos básicos a proteger: hardware, software[2] y los datos. Para cualquiera de estos elementos, existe una gran cantidad de ataques que se los puede clasificar en:

a) Ataques pasivos

"El atacante no altera la comunicación, sino que únicamente la escucha o monitoriza, para obtener información que está siendo transmitida. Sus objetivos son la intercepción de datos y el análisis de tráfico"[3]. Generalmente se emplean para:

- Obtención del origen y destinatario de la comunicación, a través de la lectura de las cabeceras de los paquetes monitorizados.
- Control del volumen de tráfico intercambiado entre las entidades monitorizadas, obteniendo así información acerca de actividad o inactividad inusuales.
- Control de las horas habituales de intercambio de datos entre las entidades de la comunicación.

Este tipo de ataque es difícil de detectar, ya que no provocan alteraciones en los datos. No obstante es posible evitarlo mediante el cifrado de la información entre otros mecanismos.

b) Ataques activos

Estos ataques implican algún tipo de modificación del flujo de datos transmitido o la creación de un falso flujo de datos. Generalmente son

[1] Carracedo, J. (2004). Seguridad en redes telemáticas. McGraw Hill.
[2] Carlos Barra, Software e Ingeniería de Software, http://www.revistamarina.cl/revismar/revistas/1998/1/barra.pdf
[3] Diseño y Evaluación de un Sistema de Seguridad para la RDSI-BA, Tesis Doctoral, Jordi Forné Muñoz, 1997, http://tdx.cat/handle/10803/7045

realizados por hackers, piratas informáticos o intrusos, y se los puede subdividir en las siguientes categorías:

- Interrupción: si hace que un objeto del sistema se pierda, quede inutilizable o no disponible.
- Intercepción: si un elemento no autorizado consigue el acceso a un determinado objeto del sistema.
- Modificación de mensajes: donde el intruso si además de conseguir el acceso consigue modificar el objeto.
- Fabricación: se consigue un objeto similar al original atacado de forma que es difícil distinguirlos entre sí.
- Destrucción: es una modificación que inutiliza el objeto.

2.1.2. Fases de una Intrusión a una red IP

Una intrusión es una secuencia de acciones realizadas por un usuario o proceso deshonesto, con el objetivo final de provocar un acceso no autorizado sobre un equipo o un sistema completo[4]. La penetración en un sistema o subred suele proceder de acuerdo con el siguiente esquema

1. Identificar las máquinas objetivo.

a) Recabar información sobre la red a atacar.

- Qué hosts y subredes existen en ella y qué nombres tienen
- Qué usuarios pueden existir
- Qué servicios de red ofertan las máquinas
- Qué potenciales vulnerabilidades presentan
- Qué sistemas operativos corren sobre las máquinas.

b) Formas de exploración

Existen abundantes métodos y software, para realizar esta tarea, siendo los siguientes:

[4] González, D. (2002), Mecanismos para la detección de ataques e intrusiones, http://ocw.uoc.edu/computer-science-technology-and-multimedia/advanced-aspects-of-network-security/advanced-aspects-of-network-security/P06_M2107_01773.pdf

- Ping y traceroute
- Exploradores de red
- Exploradores de puertos
- Finger printing para detección de sistema operativo
- Fragmentación anómala de paquetes para eludir el registro (logging)

2. Obtener acceso ilícito en alguno de los hosts, sin disponer de un login, a través de alguna de las siguientes formas:

a) Exploit de un bug en el software de servicios de red ofrecidos por el sistema. Los bugs en el software o hardware pueden ser explotados para realizar intrusiones. Ejemplos:

- Buffer overflows, ejemplo: finger, sendmail, wu-ftpd, etc.
- Fallos de autenticación o de protocolo
- Facilidad para esnifar una Ethernet local (Tcpdump, Dsniff, Radiusniff, etc)
- Duplicar las credenciales de un usuario autorizado (ejemplo, robo de contraseñas, sniffing, ingeniería social, impostura, entre otros).
- Hay que considerar que una vez que el intruso puede acceder a una cuenta no privilegiada, todo es muchísimo más fácil para él en su siguiente tarea que consiste en conseguir el control total como supe usuario.
- La mejor oportunidad de detener las intrusiones ocurre en el punto del acceso a los servicios. Más allá, el ataque es mucho más difícil de contener.

b) Secuestro de una conexión existente

Por algunas debilidades típicas

- Permisos incorrectos
- Directorios de ejecutables con permiso de escritura
- Cambios temporales realizados por administradores que olvidan restablecer la configuración por defecto

c) Obtener el control de alguno de los hosts (como super usuario)

Medidas sanitarias

- Control riguroso de su número e integridad (checksums)
- Verificadores de integridad y HIDS (Host Intrusion Detection Systems) (COPS, Farmer & Spafford).

3. Borrar las pruebas

Una vez alcanzado el acceso como root, se borran las pistas dejadas por la intrusión. Las víctimas habituales son logs en /var/log.

2.2. Tipos de Ataques a una Red IP

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:

2.2.1. Rastreo de Sistemas (Escaneo de Puertos)

Rastreo de Sistemas (Escaneo de Puertos).- 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 analizador 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)[5].

Cabe mencionar que las herramientas utilizadas para detectar sistemas emplean uno de los siguientes métodos de análisis para descubrir redes:

[5] Fyodor ,Guía de referencia de Nmap (Página de manual), http://nmap.org/man/es/index.html

- Análisis activo: El sistema que rastrea una red se dedica a enviar información a la red para que el resto de sistemas detecten dicha información y envíen algún tipo de respuesta.

- Análisis pasivo: El sistema que analiza la red no envía nada, simplemente está a la escucha, recogiendo y analizando datos de lo que está pasando.

Es evidente que será más fácil detectar un análisis activo que uno pasivo, diferencia que se radica en la forma de empleo de las técnicas para analizar una red.

2.2.2. Ataque de Fuerza Bruta

Un ataque de Fuerza Bruta "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"[6]. 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."[7]. 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 (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 John The Ripper y Medusa.

2.2.3. Ataque de Suplantación de Identidad (Spoofing)

"Es una técnica más o menos sofisticada de autenticar una máquina en otra máquina con una dirección fuente de confianza. En otras palabras, consiste en modificar en los mensajes que salen de una máquina la dirección fuente haciendo creer a los destinatarios de esos mensajes que es otra dirección quien le envía el mensaje y además esa es una dirección de confianza"[8].

---

[6] Hacking: VII Ataques por Fuerza Bruta, http://jBercero.com.
[7] Labs DragonJAR, Ataques por fuerza bruta (BruteForce) III., http://labs.dragonjar.org/laboratorios-hacking-tecnicas- fuerza-bruta-brute-force-iii
[8] ADLS, Tipos de Ataques en Internet, http://www.adslayuda.com/foro/adsl/vuestros-documentos/tipos-de-ataques-en-internet-t34936.html

La mecánica de un ataque Spoofing

Es conocido que el mecanismo de conexión, aparte de requerir una dirección IP correcta, requiere un dialogo entre las dos máquinas. Siendo necesario como primer paso que lleguen los paquetes intactos al host destino. Cuando se inicia la conexión TCP el primer paquete del cliente se identifica con un número de secuencia aleatorio, los siguientes paquetes están numerados con respecto a este número. El servidor responde a este mensaje con un número de secuencia propio que debe ser reconocido por el cliente.

El problema para quien quiere generar un ataque de tipo Spoofing está en que primero debe bloquear (dormir) la dirección fuente y segundo debe mantener un diálogo con la máquina objetivo (proceso que hace complejo el ataque). Esta complejidad se debe a que la máquina objetivo es quien genera el número de secuencia al cual debe responder el atacante con el reconocimiento pertinente. Además el atacante debe adivinar este número de secuencia ya que no recibe los paquetes que envía el objetivo puesto que la dirección IP fuente que el atacante ha enviado estaba trucada.

Existen diferentes tipos como el IPSpoofing, ARPSpoofing, DNS Spoofing, Web Spoofing o e-mail Spoofing. Para efecto del presente estudio se ha enfocado a IPSpoofing y ARPSpoofing.

IPSpoofing (enmascaramiento de la dirección IP)

Mediante IP Spoofing un atacante consigue modificar la cabecera de los paquetes enviados a un determinado host para simular que proceden de un equipo distinto al que verdaderamente los ha originado. Para ello, el atacante trataría de seleccionar una dirección IP correspondiente a la de un equipo confiable (legítimamente autorizado) para acceder al sistema que pretende ser engañado.

IPSpoofing solamente puede ser implementado contra un cierto tipo de máquinas corriendo ciertos servicios, tal como:

- Cualquier dispositivo ejecutando Sun RPC[9].
- Cualquier servicio de red que utiliza autenticación de las direcciones IP.

[9] Martín López Nores, Introducción a Sun RPC, Primera práctica, http://www-gris.det.uvigo.es/wiki/pub/Main/PracticasRO/msg-slides.pdf

ARPSpoofing

Un ataque ARP Spoofing es una técnica que altera la caché ARP. La caché ARP contiene una tabla en la que se relacionan direcciones máquina (MAC) con direcciones IP. Este sistema consiste en mantener tu dirección hardware en la tabla pero relacionada con la dirección IP de una máquina legítimamente autorizado. Esta información se le envía a la caché ARP de tu máquina y a la caché ARP de la máquina objetivo. A partir de ese momento los paquetes del objetivo se encaminan hacia la máquina atacante ya que la máquina objetivo cree que es una máquina de confianza.

La mayor limitación en este tipo de ataques, es el atravesar routers inteligentes, y es cuando fallará el invento. Además se tiene la expiración en muy poco tiempo de las cachés, por lo que se debería actualizar constantemente la caché mientras se genera el ataque.

2.2.4. Ataque de Denegación de Servicios (DoS)

La idea principal de este ataque es inundar un sistema con paquetes de datos que alteren, afecten o degraden seriamente la conexión a Internet, inmovilizando los servidores locales hasta el extremo de no poder atender las peticiones legítimas es decir causar la baja del sistema totalmente. Un ataque DoS puede ser perpetrado en varias formas, siendo el de tipo SYN Flood el seleccionado en esta investigación, ataque 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 (parte del proceso de establecimiento de conexión TCP de 3 vías[10]). 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.

[10] Explicación de enlace de tres vías a través de TCP/IP , http://support.microsoft.com/kb/172983/es

2.2.5. Ataques a la WEB

Dirección URL

La dirección URL (Uniform Resource Locatior) es la forma de indicarle a un servidor web cual es la página que se quiere visualizar desde el cliente.

Estructura URL

La dirección URL es el mecanismo que se crea entre el usuario y los servidores para poder acceder a recursos, páginas web, sitios FTP, etc. La estructura o sintaxis de una dirección URL es:

Protocol://server/path/resource?parameters

Protocol: Especifica el protocolo a utilizar de la capa de aplicación, como por ejemplo: https (http cifrado), ftp (transmisión de ficheros), ldap (acceso a un directorio), pop3 (acceso al buzón de correo), etc.

Server: Indica el nombre del servidor donde se quiere acceder ya sea por medio de DNS, el nombre de red NetBios o la dirección IP de la máquina.

Path/resource: Se especifica el directorio de la ruta y el nombre del recurso de la petición. El recurso puede ser un fichero estático o una aplicación que genera una salida dinámica.

Parameters: Este valor es opcional, lo que se indican son los parámetros que se le quieren pasar al recurso que se esté solicitando (depende del sitio web donde se realicen las peticiones). A este conjunto de valores se le suele denominar en inglés Query String.

Ejemplo:

http://192.168.1.36/cgi-bin/vervariables.cgi?hola+mundo

El resultado obtenido en el navegador web es:

Numero de argumentos: 2

Argumentos: hola mundo

GATEWAY_INTERFACE = CGI /1.1

SERVER_PROTOCOL = HTTP/1.1

REQUEST_METHOD = GET

SCRIPT_NAME = / cgt-bin/vervariables.cgi

QUERY_STRING = hola+mundo

El símbolo + del query string significa el carácter en blanco y por ello muestra que hay dos parámetros. Se puede ver en el listado anterior el valor de variables de entorno.

Formularios HTML

Los formularios (forms) html permiten una interacción entre el usuario y el servidor web. Es necesario conocer el funcionamiento de los forms, a fin de poder explotar todas las opciones de un sitio web. Una form puede ser utilizada, por ejemplo, en los siguientes sitios:

Buscadores de Internet.

Web mail.

Sitios de solicitud de información.

Sitios de consulta de datos.

Un formulario se encuentra definido o delimitado por las etiquetas:

< form>..... ..< / form >

Todo el código html que se encuentre dentro de estas etiquetas pertenece al formulario.

2.2.5.1. Tipos de intrusiones

Existen tres tipos de intrusiones que puede hacer un atacante sobre un sitio web de Internet:

El acceso a los servicios no autorizados que el sitio web ofrece, suplantando la identidad de un usuario registrado y así poder tener acceso a información restringida, pudiendo utilizar como técnica el ataque de fuerza bruta.

El acceso al servidor web directamente, ya sea a sus directorios y ficheros, base de datos, así como reconocer todo el entorno cercano de máquinas en la red. En este tipo de intrusión el daño que se puede cometer en el servidor puede ser de magnitudes considerables, ya que se podrán poner en práctica todas las técnicas existentes en incursión a un sistema web; siendo entre las más conocidas: modificación de ficheros y sustitución de la página principal del sitio web.

El último método, es la denegación de servicios (DoS) sobre el sitio web, cuyo objetivo es que la página web este caída o que el sitio deje de prestar sus servicios.

2.2.5.2. Ataque basado en la vulnerabilidad del sistema de validación de HTML

Ataque de Cross Site Scripting

Un ataque de Cross Site Scripting, cuya abreviatura es XSS, consiste en inyectar código en los intérpretes del navegador web.

El uso de lenguajes HTML, JavaScript, VBScript, ActiveX, Flash, entre otros, en aplicaciones Web, se puede realizar del lado del cliente robos de sesiones, cambiar configuraciones del usuario, mostrar anuncios falsos o incluso realizar ataques de denegación de servicio. Cualquier página web que publique entradas del usuario (blogs, foros de mensajes, libros de visita, etc.) es propensa a contener XSS.

Fundamento de XSS

XSS aprovecha la falta de mecanismos de filtrado y validación en campos de entrada, en cualquier página web, permitiendo así el envío de scripts X completos (como Visual Basic Scripts o Java Scripts) con comandos maliciosos que podrían impactar directamente en el sitio web o en el equipo de un usuario[11].

Esta limitación se debe a que el código HTML se interpreta en el navegador de un usuario y no en el servidor. Es por ello que si alguien inyecta código HTML en alguna aplicación web no podría hacer daño alguno al servidor, ya que éste nunca interpreta el código HTML, sólo los navegadores.

Tipos de Ataques XSS

- *Persistente*: se produce generalmente en sectores donde podemos comentar, de modo que el código que inyectemos quedará almacenado en el servidor y se ejecutará cada vez que abramos la página.

[11] Web Attacks, Mauren Alies, Sergio García, Fabián Molina, Juan Felipe Montoya, María Isabel Serrano,Marzo de 2002, www.acis.org.co/memorias/JornadasSeguridad/IIJNSI/weba.doc

- *No Persistente*: también conocido como reflejado, consiste en inyectar código pasándolo mediante la URL, no se almacena en el servidor, solo es explotable el XSS mediante la URL.

Detectar aplicación web vulnerable a ataques XSS

- Detectar la vulnerabilidad a través de un formulario. En este caso los datos proporcionados a la aplicación web por un usuario se almacenan en un servidor y luego se muestran a los demás usuarios sin ser codificados como entidades HTML. Ejemplo

  <script>alert();</script>

- Detectar la vulnerabilidad a través de URL. Como algunos browsers no exigen una formación correcta de la URL, se podría inyectar código JavaScript, escribiendo por ejemplo:

http://vulnerable.com/index.html?nombre=<script>alert();</script>

El ataque XSS es un tipo de inseguridad informática o agujero de seguridad basado en la explotación de vulnerabilidades del sistema de validación de HTML incrustado.[12]

Fases de un ataque XSS

1. Inserción de código HTML, a fin de identificar que código puede introducir
2. Determinación de lo que puede hacer con XSS? Robo de credenciales, alterar el sitio web, etc.
3. Llevar a cabo el daño, robo de credenciales, suplantación de identidad, envío de emails falsos, identificación de posibles víctimas, etc.

2.2.6. Ataques a Bases de Datos

En la actualidad, un gran número de sitios y aplicaciones web interactúan con bases de datos, debido a que se maneja información, sensible o no, que necesita ser

[12] Carlos Tori,2008 Rosario Argentina, Hacking Ético

almacenada, consultada o modificada. El almacenamiento de datos, en una base de datos, es forma ordenada, coherente e íntegra, y puede contener todo tipo de información, Por ejemplo:

- Catálogo de productos de consumo masivo.
- Registros relacionados con los datos personales del personal de una empresa.
- Órdenes de entrega-recepción de mercancía.
- Base de Datos sobre información económica y financiera de una empresa.

Actualmente existe un gran número de empresas que utilizan interfaces web para consultar sus bases de datos, las mismas que a pesar de utilizar los últimos parches de seguridad pueden convertirse en víctimas de intrusiones, que no sólo están basadas en vulnerabilidades en las aplicaciones, sino que aprovechan la inseguridad de las aplicaciones web.

La información, que se encuentra en una base de datos, es manejada a través de sentencias SQL, infiltradas desde el mismo código fuente de la página.

La interacción ocurre más o menos de este modo:

- El usuario introduce datos en un formulario o hace clic en un link.
- El sitio web podría estar programado en algún lenguaje de alto nivel y a su vez contendrá, en su código fuente, sentencias SQL.
- Estas sentencias SQL, junto con los datos suministrados por el visitante, irán directamente a la base de datos para lograr el resultado esperado por el usuario como: consulta, almacenamiento, modificación, borrado, ejecución, etcétera.
- El resultado puede mostrarse al usuario o no, o bien puede dar un error de sistema.

2.2.6.1. Ataque SQL Inyection

Consiste en la manipulación de los datos de tránsito, ingresado en los campos de datos (Textfield) en los que se escribe la información que va a procesar el servidor

(usuario, password u otro tipo de datos personales o relativos al sitio web seleccionado). La intensión de este tipo de ataque es conseguir información contenida en el servidor o tomar el control por completo de la máquina.

El intruso logra inyectar una sentencia SQL cuando en lugar de colocar el nombre de usuario, password y cuenta (verdaderos), logra insertar código como 'or 1=1-- y seguido a ella, el resto de lo que sería interpretado por el gestor de base de datos como código SQL de la siguiente manera:

Select * from usuarios where usuario='or 1=1-- and password ='or 1=1--

Con este tipo de sentencia se busca conseguir el acceso con el primer usuario existente en la tabla Usuarios, y así tener acceso a la aplicación con el mismo. Una vez que se pueda conseguir acceso con el primer usuario, el próximo paso cosiste en acceder con un usuario que se pensaría que existe en la aplicación. Para ello se debe incluir en el campo usuario y contraseña lo siguiente: usuario=admin'--, password='or 1=1 --; quedando la sentencia SQL de la siguiente manera:

select * from usuarios where usuario=admin' -- and password = 'or 1=1 --

De este modo, si el usuario es el admin y tiene un password, si 1 es igual a 1, éste será validado y tendrá acceso a la intranet.

Si un intruso logra inyectar código SQL, podrá llevar a cabo acciones intrusivas en el servidor, comprometer información, destruir, copiar o modificar. Incluso sobre la red interna si ésta se encuentra detrás.

En principio, hay que encontrar un campo o punto vulnerable a la inyección SQL. Ya sea un formulario de acceso user/pass, uno de búsqueda, de recuperación de password, de comprobación de cualquier dato, de contacto, link con variables, links ocultos al público ligados a la DB, scripts y archivos de testeo, CGIs por defecto, aplicaciones del tipo foro y otras de terceros públicas y licenciadas, ya sea vía método GET o POST.[13]

[13] Carlos Tori, 2008 Rosario Argentina, HACKING ÉTICO

2.2.7. Ataques a Correo Electrónico

Un servidor de correo es una aplicación que nos permite enviar mensajes (correos) de unos usuarios a otros, con independencia de la red que dichos usuarios estén utilizando.

Protocolos:

Simple Mail Transfer Protocol (SMTP)
Protocolo de red basado en texto utilizado para el intercambio de mensajes de correo electrónico entre computadoras o distintos dispositivos.

Post Office Protocol (POP).
Permite trabajar en modo "offline". El buzón de correo se consulta a intervalos regulares.

Internet Message Access Protocol (IMAP).

IMAP es un protocolo de acceso interactivo ("online") al buzón de correo. Se actúa directamente sobre el buzón en el servidor.

En la actualidad el correo electrónico es el medio de comunicación más utilizado en la red de redes, lo que lo hace el más deseado por los hackers, escritorios de virus, lammers.

SMTP (Simple Mail Transfer Protocol) (RFC: 821, 822, 1869)

Es el método definido por Internet para transferencia de correo electrónico. Emplea el puerto TCP 25. Trabaja por medio del empleo de colas o spooler donde va almacenando los mensajes recibidos en los servidores hasta que un usuario se conecte y transfiera su correspondencia, si esto no sucede en un determinado tiempo (Programable), los mensajes son descartados o devueltos a su origen.

2.2.7.1. Comandos y códigos

Todos los comandos, réplicas o datos intercambiados son líneas de texto. Todas las réplicas tienen un código numérico el comienzo de la línea. La secuencia de envío y recepción de mensajes es la siguiente:

1. El emisor SMTP establece una conexión TCP con el SMTP de destino y espera a que el servidor envíe un mensaje "220 Service ready" o "421 Service not available" cuando el destinatario es temporalmente incapaz de responder.
2. Se envía un HELO (abreviatura de "hello"), con el que el receptor se identificará devolviendo su nombre de dominio. El SMTP emisor puede usarlo para verificar si contactó con el SMTP de destino correcto.
3. El emisor inicia ahora una transacción enviando el comando MAIL al servidor. Este comando contiene la ruta de vuelta al emisor que se puede emplear para informar de errores.
4. El segundo paso del intercambio real de correo consiste en darle al servidor SMTP el destino del mensaje (puede haber más de un receptor). Esto se hace enviando uno o más comandos RCPT TO: <forward-path>. Cada uno de ellos recibirá una respuesta "250 OK" si el servidor conoce el destino, o un "550 No such user here" si no.
5. Cuando se envían todos los comandos rcpt, el emisor envía un comando DATA para notificar al receptor que a continuación se envían los contenidos del mensaje. El servidor replica con Seguridad por Niveles "354 Start mail input, end with <CRLF>.<CRLF>". Nótese que se trata de la secuencia de terminación que el emisor debería usar para terminar los datos del mensaje.
6. El cliente envía los datos línea a línea, acabando con la línea <CRLF>. <CRLF> que el servidor reconoce con "250 OK" o el mensaje de error apropiado si cualquier cosa fue mal.
7. Ahora hay varias acciones posibles:

   - El emisor no tiene más mensajes que enviar; cerrará la conexión con un comando QUIT, que será respondido con "221 Service closing transmission channel".
   - El emisor no tiene más mensajes que enviar, pero está preparado para recibir mensajes (si los hay) del otro extremo. Mandará el comando TURN.
   - Los dos SMTPs intercambian sus papeles y el emisor que era antes receptor puede enviar ahora mensajes empezando por el paso 3 de arriba.

2.2.7.2. Vulnerabilidades del correo electrónico

Las dos grandes vulnerabilidades que sufre el correo electrónico son referidas a su privacidad y su seguridad, dentro de ellas existen debilidades concretas que se tratan a continuación:

- La privacidad es fácilmente vulnerable pues el correo viaja como texto plano, es decir, que si no se emplea algún algoritmo criptográfico, cualquiera puede tener acceso al mismo. En este tema, la mejor analogía es la del correo postal, en el cual a nadie se le ocurre enviar una carta sin el sobre.

- La seguridad es atacada con dos técnicas puntuales: las bombas de correo (varias copias de un mismo mail a un solo destino) y el Spam (Correo no autorizado).

Las herramientas con que se cuenta para defenderse de estas vulnerabilidades son:

S/MIME: Desarrollado por RSA el cual es una especificación para obtener autenticación por medio de firmas digitales. Se lo considera uno de los más seguros

PGP: Pretty Good Privacy, el cual es un producto completo desarrollado por Phillip Zimmerman que ofrece que dentro de sus muchas funciones ofrece también autenticación, no repudio y criptografía siendo soportado por la mayoría de los clientes de correo.

PEM: Privacy Enhanced Mail, el cual es una norma para permitir la transferencia de correo seguro. Con cualidades similares a PGP, siendo el estándar más reciente de los tres.

2.3. Herramientas Utilizadas para la Ejecución de Ataques a una Red IP

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

2.2 describe el tipo de ataque y las diversas herramientas utilizadas para llevar a cabo dicho ataque sobre una red IP.

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 Nikto	
3	Suplantación de Identidad	Ubuntu	Hping3	Wireshark
		Windows	Nemesis	Wireshark
4	Denegación de Servicio	Ubuntu	Nemesis	Ettercap,
		Windows	Ettercap	Wireshark

Tabla 2.1. Resumen sobre las herramientas utilizadas para la ejecución de ataques a redes IP.

2.3.1. Herramienta Nmap

La herramienta Nmap tiene como misión permitir hacer un barrido a las redes informáticas y a ordenadores a fin de determinar que puertos tienen activos, y así solucionar posibles debilidades en su seguridad, para ello se emplearon algunas técnicas entre las que cabe destacar las siguientes[4.]:

Tipo de escaneo	Descripción
TCP connect()	Intentar establecer una conexión con cada uno de los puertos del host a escanear. Si la conexión se establece, el puerto está abierto; en caso de recibir un aviso de cierre de conexión (RST), el puerto estará cerrado; y en caso de no recibir respuesta, se deduce que el puerto está silencioso.
TCP SYN	Envía un paquete SYN que finge intentar establecer una conexión y se espera la respuesta. Si llega SYN/ACK significa que el puerto está abierto; si llega RST, el puerto está cerrado; y si no se recibe respuesta se asume que está silencioso.
TCP FIN	Envía un paquete FIN al puerto del host destino que se quiere escanear. Si se recibe RST por respuesta, el puerto está cerrado, y en caso de no recibir respuesta (se ignora el paquete FIN) el puerto puede encontrarse abierto o silencioso.
UDP scan	Manda un paquete UDP vacío al puerto que se desea escanear. Si el puerto está cerrado, el sistema responderá con un paquete ICMP de tipo 3 (destino inalcanzable). En caso de no responder, el puerto puede estar abierto o silencioso.

Tabla 2.2. Algunas Técnicas de Rastreo de Sistemas

2.3.2. Herramientas Medusa y John The Ripper

Medusa es una herramienta desarrollada para llevar ataques del tipo fuerza bruta, de forma rápida, paralela y modular [5.]:

Características

- Permite llevar a cabo múltiples ataques a diferentes sistemas con diferentes usuarios.
- Tiene como objetivo auditar el inicio de sesión de varios servicios entre ellos MS-SQL Server, VNC, HTTP, SMB, Telnet, el servicio SSH (Secure Shell), entre otros

Con el servicio SSH (SecureSHell), Medusa puede acceder a máquinas remotas a través de una red. Además permite copiar datos de forma segura (tanto ficheros sueltos como simular sesiones FTP cifradas), gestionar claves RSA para no escribir claves al conectar a las máquinas y pasar los datos de cualquier otra aplicación por un canal seguro tunelizado mediante SSH[14].

John The Ripper es un programa de criptografía utilizado para descifrar contraseñas. Es capaz de fragmentar varios algoritmos de cifrado o hash, como DES, SHA-1[15]y otros.

Es una herramienta que permite a los usuarios de sistemas comprobar la robustez de las contraseñas de sus sistemas.

John The Ripper es capaz de auto detectar el tipo de cifrado de entre muchos disponibles, y se puede personalizar su algoritmo de prueba de contraseñas [12.].

Características:

- Funciona en varias arquitecturas y sistemas operativos.
- Se utiliza para ataques de diccionario (fichero con un gran número de palabras) y por fuerza bruta (formación de palabras mediante combinación de caracteres).
- Permite definir el rango de letras que se usará para construir las palabras, y las longitudes.

[14]Red Hat Enterprise Linux 4: Manual de referencia, http://web.mit.edu/rhel-doc/4/RH-DOCS/rhel-rg-es-4/ch-ssh.html
[15]What is a sufficient encryption? http://penguinsecurity.net/wiki/index.php?title=What_is_a_sufficient_encryption

- Permite parar el proceso (con control-C), y continuarlo más adelante (con –restore).

- Se puede automatizar; por ejemplo, ponerlo en cron[16].

2.3.3. Herramienta Ettercap

Ettercap es una herramienta que hace uso del envenenamiento ARP en entornos GNU/Linux. Está diseñada con el fin de analizar, filtrar y escuchar determinado tráfico circulando por la red[7.]. Es capaz de realizar ataques MITM (Man in the Middle) entre diferentes hosts de la red, con el fin de interponerse en su comunicación y obtener información valiosa como el tipo de contraseñas POP, SSH, Telnet, FTP, Https, etc.

En un envenenamiento ARP[17], el host atacante busca conseguir modificar la tabla ARP en caché del host victima introduciendo una entrada falsa. En consecuencia, cuando el host victima quiera enviar información, a un destinatario, la encapsulará en una trama ethernet pero con la dirección MAC del host atacante, haciendo que cuando llegue la trama al switch, éste la encamine a la boca de salida que corresponde con la dirección MAC especificada, es decir, hacia el host atacante[18] .

Las características de Ettercap son:

- Capacidad para monitorear usuario y contraseña de conexiones ssh, en modo Fullduplex.
- Capacidad de esnifar datos cifrados con SSL. Un certificado falso es presentado al host víctima y la sesión es desencriptada.
- Inyección de caracteres en una conexión establecida.
- Filtrado de paquetes
- Capacidad para finalizar conexiones TCP.

[16] Como agregar tareas al Cron de Linux, http://www.guatewireless.org/os/linux/como-agregar-tareas-al-cron-de-linux/
[17] Envenenamiento ARP, Seguridad en redes conmutadas, Sergio Valín Cabrera. http://ownz.despai.es/trabajo_arp.pdf
[18] La importancia de los protocolos cifrados: Envenenamiento ARP mediante ettercap, http://www.sahw.com/wp/archivos/2010/05/31/la-importancia-de-los-protocolos-cifrados-envenenamiento-arp-mediante-ettercap/

- Capacidad de creación de tus propios plugins utilizando las API de Ettercap.
- Capturador de contraseñas: TELNET, FTP, POP, SSH1, MySQL, HTTP, IRC, RIP, BGP, LDAP, NFS, SNMP, MSN, YMSG.
- Posibilidad de escanear la red local en modo pasivo (sin enviar ningún paquete) y obtener información detallada sobre los hosts, como el sistema operativo que utilizan, servicios en ejecución, puertos abiertos, IP, direcciones MAC y vendedor del adaptador de red.

2.3.4. Herramienta Némesis

"Némesis es un inyector de paquetes, su funcionamiento consiste en enviar tramas de datos a una red utilizando el protocolo TCP/IP. Se utiliza para probar y depurar servicios de red específicos. Es ideal para auditar servicios de red pero también puede ser utilizada para realizar ataques a una red, como una inundación UDP"[19].

Una inanición UDP consiste en el consumo de los recursos del sistema, a la saturación de la CPU, memoria, etc., hasta que la máquina se cae. Este tipo de ataque, generalmente provoca un fallo general del sistema, o procesos que se cuelgan porque necesitan CPU que el sistema no le está proporcionando o se lo proporciona escasamente[8.].

2.3.5. Herramienta Hping

Hping es una herramienta muy versátil, que permite la manipulación de paquetes TCP/IP desde línea de comandos. Es factible modificar la información contenida en las cabeceras de los paquetes ya sean TCP, UDP e ICMP, en función de los parámetros con que se ejecute Hping[9.].

Con la herramienta Hping se puede hacer un ataque DoS. El ataque se basa en el envío de paquetes de ping al broadcast de una red grande haciendo un spoofing a la dirección IP que envía la petición. Así, todos los ordenadores de la red (el broadcast)

[19] Introducción al némesis, http://bad-robot.blogspot.com/2009/04/introduccion-al-nemesis.html

responderán a la vez a la máquina que se quiere atacar (la IP spoofeada), y si hay muchas peticiones esta seguramente se colapse.

Un ataque DoS (Denial of service / Denegación de servicio) tiene como objetivo derribar un host temporalmente, o definitivamente hasta que se reinicie[20].

2.3.6. Herramientas utilizadas en ataques WEB

Telnet

C: \>telnet -h

telnet t-al [-e escape char] [-f log file] [-l user] [-t term] [host [port]]

En donde cada uno de los parámetros significa lo siguiente:

-a: Se utiliza para conectarse automáticamente a un sistema. Para ello utiliza el nombre de usuario y contraseña con el que está conectado actualmente.

-e: Con este parámetro se puede especificar el carácter de escape que se va a utilizar en la sesión remota. Dicho comando se utiliza para salir temporalmente de la sesión al menú de comandos del comando telnet.

-f: Nombre del fichero donde se va a guardar la sesión remota.

-l: Parámetro para indicar el nombre de usuario para el sistema remoto, donde se quiere conectar.

-t: Asigna el tipo de terminal/emulación que se quiere utilizar en la sesión remota.

host: Nombre o dirección IP del equipo remoto.

port: En esta parte se especifica el puerto al que se quiere conectar, por defecto es el puerto número 23 corresponde al puerto telnet.

Nikto

Nikto es una herramienta orientada al rastreo de sistemas web. Está desarrollada en Perl, lo que permite que pueda ser ejecutado en diferentes sistemas operativos: Windows, Linux y Mac OS X.[21]

[20] HPING tutorial by Philippe Bogaerts, Version 1.5 24-08-2003 http://www.radarhack.com/dir/papers/hping2_v1.5.pdf

Características principales:

La licencia de la herramienta es Open Source, GPL.

Conoce hasta 900 versiones de servidores web distintas.

Actualización automática de los complementos de la herramienta.

Plantillas HTML para la generación de informes.

Soporte de proxy con autentificación.

Tratamiento de cookies.

Técnicas de evasión de IDS.

Parámetro a utilizar:

-host: permite especificar el nombre del sistema que se quiere rastrear. Ejemplo

$ perl- nikto.pl -host localhost

-port: Indica el puerto del servidor web.

-verbose: Ofrece un informe más detallado.

-ssl: Se activa el uso del protocolo SSL en los servidores web que utilizan el protocolo HTTPS.

-Format: Se define como será el formato del resultado de salida, dicho resultado puede ser en formato HTML, cSV o texto.

-output: Indica cual es el nombre del archivo donde se va aguardar el resultado.

-id: Indica las credenciales de autentificación básicas para el protocolo HTTP.

-cookies: Visualiza las cookies retornadas desde el servidor web.

-vhost: Utiliza un sistema virtual como objetivo en vez de la dirección IP.

-Cgidirs: Examina los directorios CGI que existan en el servidor web'

-evasion:Indica que Nikto utilice técnicas de evasión para los sistemasIDS.

-update: Actualización de los complementos del programa Nikto'

-userproxy: Indica que se va a utilizar el proxy definido en el archivo de configuración config.txt.

[21] Nikto v2.1.4 - The Manual,Chris Sullo and David Lodge, http://cirt.net/nikto2-docs/

-debug: Activa los mensajes de depuración.

-timeout N: Define el tiempo de rastreo en segundos en caso de no recibir datos. El valor predeterminado es 10 segundos.

-nolookup: No convierte las direcciones IP en nombres de sistemas según el servidor DNS.

2.3.7. Herramientas Utilizadas en Ataques a Base de Datos

Ataque manual de SQL Inyection

A continuación se describe como sería un posible ataque completo con sólo utilizar un navegador para realizar las pruebas

El ataque se inicia con la técnica más básica para saber si existe una posible vulnerabilidad de SQL Injection, en una página web la cual solicita un usuario y contraseña, para ello se podría proceder de la siguiente manera:

Revisar el código fuente (HTML) de la página web para ver de qué manera se envía el usuario y contraseña de dicha web.

Asignar el siguiente usuario y contraseña, para ver qué resultado se genera.

Usuario=is'ma

Password=pass

La consulta que se realizaría al servidor de base de datos es el siguiente:

Select * from Usuarios where usuario='is'm and password='pass'

Ejecutada la sentencia anterior el servidor devolverá el siguiente error:

Server: Msg 170, Level 10, State 1, Line 1

Line 1: Incorrect syntax near 'ma'

Lo que se comprueba que la aplicación es vulnerable a un posible ataque SQL Injection, debiendo ahora intentar entrar como cualquier usuario, si se conoce alguna persona dada de alta en la aplicación[22].

2.3.8. Herramientas Utilizadas en Ataques a Correos Electrónicos

Postfix.

El MTA (Mail Transportation Agent) Postfix no es un programa monolítico, sino una combinación de pequeños programas, cada uno de los cuales lleva a cabo una función especializada para la entrega y recepción de correos[23].

TLS.

Para solventar los problemas de seguridad en Internet, Netscape, Inc. introdujo el protocolo SSL (Secure Sockets Layer), que ha ido evolucionando en el protocolo estandarizado TLS (Transportation Layer Security). Ofrece tanto cifrado de la comunicación (frenando las escuchas) como autenticación fuerte (asegurando que ambas partes de una comunicación son correctamente identificadas y que la comunicación no puede ser alterada). Postfix/TLS no implementa el protocolo TLS por sí mismo, sino que usa el paquete OpenSSL para esta tarea.

SASL

SASL (Simple Authentication and Security Layer), es un método para añadir soporte para la autenticación a protocolos basados en la conexión. Se usa en servidores (en este caso Postfix) para manejar las peticiones de autenticación de los clientes[24].

SpamAssassin

SpamAssassin es un filtro de correo que trata de identificar el spam mediante el análisis del texto y el uso en tiempo real de algunas listas negras a través de Internet.

A partir de su base de datos de reglas, el correo puede ser opcionalmente marcado como spam o más tarde filtrado usando el cliente de correo del usuario.

---

[22] María Teresa Jimeno, Carlos Míguez, Mariano Matas y Justo Perez, 2010, La Biblia Hacker
[23] Postfix Basic Configuration, http://www.postfix.org/BASIC_CONFIGURATION_README.html
[24] Simple Authentication and Security Layer, http://asg.web.cmu.edu/sasl/

SpamAssassin normalmente identifica acertadamente entre un 95 y un 99% del spam, dependiendo del tipo de correo que se reciba.

Clam AntiVirus

ClamAV es una herramienta antivirus para UNIX. El propósito principal de este software es la integración con los servidores de correo (escaneo de datos adjuntos). Algo importante que resaltar es que la base de datos se mantiene actualizada constantemente. Otras características destacables son el soporte de firmas digitales en la actualización de la base de datos, el análisis durante el acceso bajo Linux, la detección de más de 20000 virus, gusanos y troyanos, el soporte integrado para archivos comprimidos con Rar, Zip, Gzip y Bzip2 y formatos de correo Mbox, Maildir.

SquirrelMail

SquirrelMail es un paquete de correo por web basado en estándares y escrito en PHP 4. Incorpora soporte PHP para los protocolos IMAP y SMTP. SquirrelMail tiene toda la funcionalidad que se espera de un cliente de correo electrónico, agendas de contactos y gestión de carpetas[25].

2.4. Tecnología de Virtualización

La virtualización es la forma de particionamiento lógico de un equipo físico en diversas máquinas virtuales (MVs), para compartir recursos de hardware, como CPU, memoria, disco duro y dispositivos de entrada y salida. 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[26].

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

[25] Squirrelmail, http://www.squirrelmail.org/documentation/
[26] Virtualización en GNU/Linux, Alberto Abián Belmonte,Madrid, julio 2007, http://www.whyfloss.com/pages/conference/static/editions/mad07/charla2.pdf

Virtualización, del Instituto Nacional de Estándares y Tecnología (NIST) la virtualización podría reducir el impacto de esta explotación [11.]. Bajo este precepto, el trabajo propuesto por Keller y Naues[12.], formula la implementación de un laboratorio colaborativo de seguridad utilizando máquinas virtuales. Li y Mohammed [13.], 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[14.][15.], 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 [16.][17.][18.], han utilizado las plataformas de virtualización para recuperación de desastres y mitigación de ataques reales a redes IP.

2.4.1. Tipos de Virtualización

La virtualización se puede hacer desde un sistema operativo Windows (XP, Vista u otra versión), siempre y cuando las otras versiones sean compatibles con el programa que se utilice[27]. Los tipos de Virtualización más comunes son:

a) Emulación

La emulación se basa en crear máquinas virtuales que emulan el hardware de una o varias plataformas hardware distintas. Este tipo de virtualización es la más costosa y la menos eficiente, ya que obliga a simular completamente el comportamiento de la plataforma hardware a emular e implica también que cada instrucción que se ejecute en estas plataformas sea traducida al hardware real.

Sin embargo la emulación tiene características interesantes, como poder ejecutar un sistema operativo diseñado para una plataforma concreta sobre otra plataforma, sin tener que modificarlo, o en el desarrollo de firmware para dispositivos hardware, donde se pueden comenzar estos desarrollos sin tener que esperar a tener disponible el hardware real.

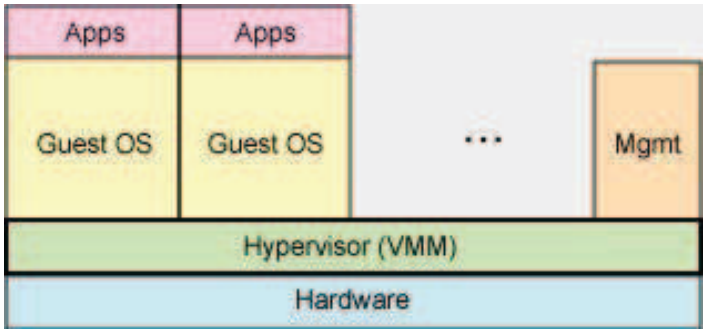
b) Virtualización completa

"Con este término se denominan aquellas soluciones que permiten ejecutar sistemas operativos huésped (Guest), sin tener que modificarlos, sobre un sistema

[27] Taller 3 de sistemas operativos, VIRTUALIZACIÓN y MULTIHILOS, Ing. Ms. Jairo E. Márquez D., http://es.scribd.com/doc/50398451/17/Tipos-de-virtualizacion

anfitrión (Host), utilizando en medio un Hypervisor[28] o Virtual Machine Monitor que permite compartir el hardware real. Esta capa intermedia es la encargada de monitorizar los sistemas huésped con el fin de capturar determinadas instrucciones protegidas de acceso al hardware, que no pueden realizar de forma nativa al no tener acceso directo a él"[29].

Su principal ventaja es que los sistemas operativos pueden ejecutarse sin ninguna modificación sobre la plataforma, aunque como inconveniente frente a la emulación, el sistema operativo debe estar soportado en la arquitectura virtualizada, ver figura 2.7.



URL: http://www.arcos.inf.uc3m.es/~folcina/pfc-html/node16.html

Figura 2.1. Virtualización completa

Se debe tener en cuenta también que la virtualización completa no se refiere a todo el conjunto de hardware disponible en un equipo, sino a sus componentes principales, básicamente el procesador y la memoria. De esta forma, otros periféricos como tarjetas gráficas, de red o de sonido, no se virtualizan. Las máquinas huésped no disponen de los mismos dispositivos que el anfitrión, sino de otros virtuales genéricos.

c) Paravirtualización

"La paravirtualización surgió como una forma de mejorar la eficiencia de las máquinas virtuales y acercarlo al rendimiento nativo. Para ello se basa en que los sistemas virtualizados (huésped) deben estar basados en sistemas operativos especialmente modificados para ejecutarse sobre un hypervisor. De esta forma no es necesario que éste monitorice todas las instrucciones, sino que los sistemas operativos huésped y anfitrión colaboran en la tarea"[30].

[28] Hypervisor-Based Redundant Execution on a Single Physical Host, http://scholar.google.com/scholar?start=20&q=Hypervisor&hl=es&as_sdt=0
[29] Máquinas virtuales, http://www.arcos.inf.uc3m.es/~folcina/pfc-html/node16.html
[30] VMware, http://www.vmware.com/es/virtualization/why-virtualize.html

2.4.2. Máquinas Virtuales

Una máquina virtual es un contenedor de software perfectamente aislado que puede ejecutar sus propios sistemas operativos y aplicaciones como si fuera un ordenador físico. La idea principal es la de permitir ejecutar varios sistemas operativos simultáneamente sobre el mismo hardware. Una máquina virtual se comporta exactamente igual que lo hace un ordenador físico y contiene sus propios CPU, RAM, disco duro y tarjetas de interfaz de red (NIC) virtuales (es decir, basados en software).

"El sistema operativo no puede establecer una diferencia entre una máquina virtual y una máquina física, ni tampoco lo pueden hacer las aplicaciones u otros ordenadores de una red. Incluso la propia máquina virtual considera que es un ordenador "real". Sin embargo, una máquina virtual se compone exclusivamente de software y no contiene ninguna clase de componente de hardware. El resultado es que las máquinas virtuales ofrecen una serie de ventajas con respecto al hardware físico"[31].

2.4.3. Herramientas de Virtualización

2.4.3.1. VMware.

VMware es un sistema de virtualización por software, en donde se puede crear un sistema de cómputo dividido en forma lógica que se ejecuta sobre una plataforma presente. El rendimiento del sistema virtual varía dependiendo de las características del sistema físico en el que se ejecute, y de los recursos virtuales (CPU, RAM, etc) asignados al sistema virtual. La mayor parte de las instrucciones en VMware se ejecutan directamente sobre el hardware físico.

Ventajas:

- Facilidad de uso.
- Posibilidad de ejecutar imágenes de máquinas virtuales creadas en Virtual PC.
- Buen rendimiento obtenido mediante la técnica de virtualización.

[31] VMWare, http://www.vmware.com/es/virtualization/virtual-machine.html

Desventajas:

Aunque se pueden utilizar los productos gratuitos de VMware para ejecutar máquinas virtuales y existen otros productos en el mercado para crearlas, si se quiere tener un rendimiento óptimo es necesaria una licencia para VMware ESX Server.

VMware cuenta con varios productos:

- **VMware Player**: Es un producto gratuito que permite correr máquinas virtuales creadas con otros productos de VMware, o creadas por él mismo.
- **VMware Server (antes GSX)**: En un principio era una versión de pago, luego fue liberada para ser descargada y utilizada de forma gratuita. Esta versión, a diferencia de la anterior, tiene un mejor manejo y administración de recursos.
- **VMware Workstation**: Esta versión es una aplicación que se instala dentro de un sistema operativo (host) como un programa estándar, de tal forma que las máquinas virtuales corren dentro de esta aplicación, existiendo un aprovechamiento restringido de recursos.
- **VMware ESX Server**: Esta versión es un sistema complejo de virtualización, pues corre como sistema operativo dedicado al manejo y administración de máquinas virtuales dado que no necesita un sistema operativo host sobre el cual sea necesario instalarlo. Pensado para la centralización y virtualización de servidores, esta versión no es compatible con una gran lista de hardware doméstico.

2.4.3.2 Oracle MV VirtualBox 3.2

VirtualBox es una herramienta de virtualización de código abierto con la que puedes ejecutar Linux bajo Windows y viceversa.

VirtualBox crea una unidad virtual en el disco duro donde se instala el sistema operativo virtualizado, al que podrás acceder como si lo estuvieras

ejecutando realmente. Dicha unidad puede tener un tamaño fijo (estática) o variable (dinámica)[32].

Las ventajas de la virtualización son varias: permite acceder y ejecutar una aplicación de un sistema operativo en otro; crear instantáneas del sistema operativo que pueden utilizarse para recuperar el sistema tras una caída inesperada; ahorrar tiempo y dinero en complicadas configuraciones de hardware [19.].

VirtualBox es un programa multiplataforma, pues está disponible para los principales sistemas operativos como: Windows, Linux, Macintosh y OpenSolaris acoge y apoyos a gran número de sistemas operativos invitados incluyendo pero no limitado a Windows (NT 4.0, 2000, XP, Server 2003, Vista), DOS / Windows 3.x, Linux (2.4 y 2.6), Solaris y OpenSolaris, y OpenBSD [33].

2.5. Direccionamiento y Enrutamiento IP

2.5.1. Direccionamiento IP

"Cada host TCP/IP está identificado por una dirección IP lógica. Esta dirección es única para cada host que se comunica mediante TCP/IP. Cada dirección IP de 32 bits identifica la ubicación de un sistema host en la red de la misma manera que una dirección identifica un domicilio en una ciudad"[34].

Una dirección IP consiste de dos niveles jerárquicos, los cuales son: un identificador de red y un identificador de host:

- El identificador de red (dirección de red) identifica un único segmento de red dentro de un conjunto de redes. Además se utiliza para identificar de forma exclusiva cada red en un conjunto de redes más grande.

- El identificador de host (dirección de host) identifica un nodo TCP/IP (estación de trabajo, servidor, enrutador u otro dispositivo TCP/IP) dentro de cada red.

[32] Oracle, VirtualBox, http://www.virtualbox.org/
[33] Documentación VirtualBox,http://creativx.net/forums/es/general-software/17094-virtualbox-3-2-8-a.html
[34] Microsoft, Direccionamiento y enrutamiento IP, http://technet.microsoft.com/es-es/library/cc776674(WS.10).aspx

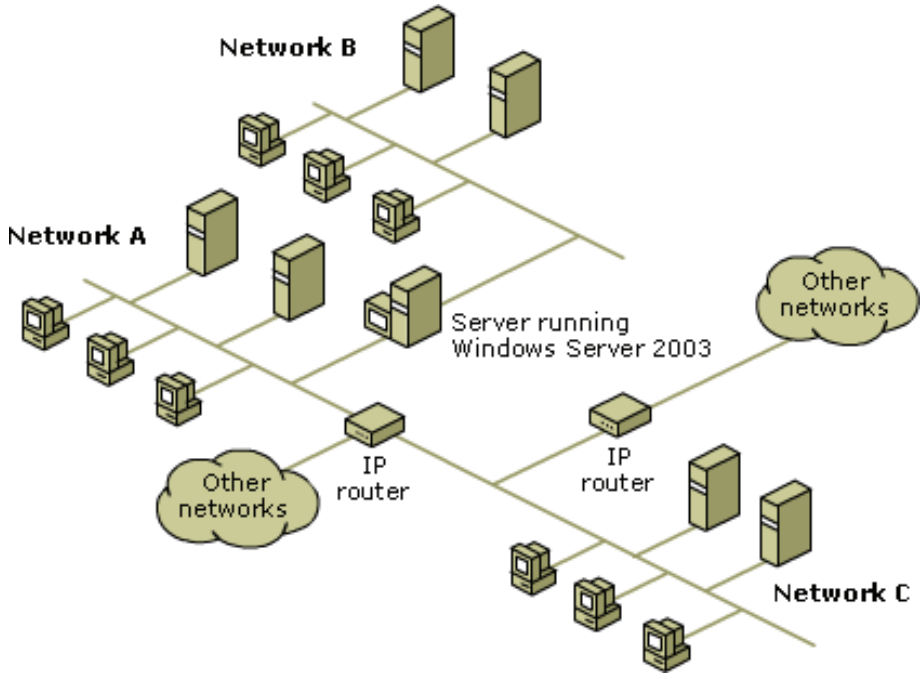
2.5.2. Enrutamiento IP

El proceso de lograr que cada máquina de una red se pueda comunicar con otra en la Internet se denomina **enrutamiento**. Sin éste, la máquina estaría limitada sólo a una red local, definida por el dominio de difusión (broadcast). El enrutamiento permite que el tráfico de una red busque el camino óptimo a un destino en cualquier lugar del mundo, pasando eventualmente a través de varias redes[35].

El enrutamiento forma parte del Protocolo Internet (IP) y se utiliza junto con otros servicios de protocolo de red para proporcionar capacidades de reenvío entre hosts que se encuentran en segmentos de red distintos dentro de una red basada en un TCP/IP más grande.

Enrutadores IP

Los segmentos de red TCP/IP están conectados entre sí mediante enrutadores IP, que son los dispositivos que transmiten los datagramas IP desde un segmento de red a otro. Este proceso se conoce como enrutamiento IP y se muestra en la siguiente figura:



URL: http://technet.microsoft.com/es-es/library/cc785246(WS.10).aspx

Figura 2.2. Enrutamiento IP

[35] MISCHA Schwartz, *Redes de telecomunicaciones Potrocolos, modelado y análisis, año* 1994, 62924

Los enrutadores IP proporcionan el medio principal para unir dos o más segmentos de red IP separados físicamente. Todos los enrutadores IP poseen dos características fundamentales:

- Los enrutadores IP son de hosts múltiples. Entendiéndose por hosts múltiples aquel equipo de host de red que utiliza dos o más interfaces de conexión de red para conectarse a cada segmento de red separado físicamente.

- Los enrutadores IP permiten el reenvío de paquetes a otros hosts TCP/IP.

Los enrutadores IP se pueden implementar mediante varios productos de hardware y software posibles. Comúnmente se utilizan enrutadores basados en hardware (dispositivos de hardware dedicados que ejecutan software especializado). Además, se pueden utilizar soluciones de enrutamiento basadas en software, como los servicios de enrutamiento y acceso remoto.

2.5.3. Tabla de Enrutamiento IP

Tanto los enrutadores como los hosts guardan una tabla de enrutamiento para mantener información acerca de otras redes IP y hosts IP. Las tablas de enrutamiento son importantes ya que proporcionan la información necesaria a cada host local respecto a cómo comunicarse con redes y hosts remotos. El daemon de enrutamiento de cada sistema actualiza la tabla con todas las rutas conocidas. El núcleo del sistema lee la tabla de enrutamiento antes de reenviar paquetes a la red local. La tabla de enrutamiento enumera las direcciones IP de las redes que conoce el sistema, incluida la red local predeterminada del sistema. La tabla también enumera la dirección IP de un sistema de portal para cada red conocida. El portal es un sistema que puede recibir paquetes de salida y reenviarlos un salto más allá de la red local. A continuación se incluye una tabla de enrutamiento simple en una red de IPv4:[36]

[36] Tablas y tipos de enrutamiento, http://download.oracle.com/docs/cd/E19957-01/820-2981/gdyen/index.html

```
Tabla de rutas IP del núcleo
Destino        Pasarela     Genmask      Indic  Metric    Ref      Uso      Interfaz
192.168.11.0   0.0.0.0      255.255.255.0  U    1         0        0        eth0
169.254.0.0    0.0.0.0      255.255.0.0    U    1         0        0        eth1
0.0.0.0        192.168.11.0 0.0.0.0        U    1         0        0        eth0
```

Tabla 2.3. Tabla de enrutamiento IP en Ubuntu

La tabla de enrutamiento se genera automáticamente y está basada en la configuración de TCP/IP actual del equipo. Cada ruta ocupa una sola línea en la tabla mostrada. El equipo busca en la tabla de enrutamiento la entrada que más se parezca a la dirección IP de destino.

El equipo utiliza la ruta predeterminada si no hay otra ruta de host o red que coincida con la dirección de destino incluida en un datagrama IP. Normalmente, la ruta predeterminada reenvía el datagrama IP (para el que no hay una ruta local coincidente o explícita) a la dirección de una puerta de enlace predeterminada de un enrutador en la subred local.[37]

2.5.3.1. Secciones de la tabla de enrutamiento IP

Destino de red

El destino de red se utiliza junto con la máscara de red para la coincidencia con la dirección IP de destino. El destino de red puede encontrarse entre 0.0.0.0, para la ruta predeterminada, y 255.255.255.255, para la difusión limitada, que es una dirección de difusión especial para todos los hosts del mismo segmento de red.

Máscara de red

La máscara de red es la máscara de subred que se aplica a la dirección IP de destino cuando se produce la coincidencia con el valor del destino de red. Cuando la máscara de red se escribe en binario, los "1" deben coincidir pero no es necesario que los "0" coincidan.

[37] TANENBAUM Andrew, *Redes de Ordenadores*, segunda edición, año 1991

Puerta de enlace

La dirección de la puerta de enlace es la dirección IP que utiliza el host local para reenviar datagramas IP a otras redes IP. Puede tratarse de la dirección IP de un adaptador de red local o de un enrutador IP (por ejemplo, el enrutador de una puerta de enlace predeterminada) del segmento de red local.

Interfaz

La interfaz es la dirección IP configurada en el equipo local para el adaptador de red local que se utiliza cuando se reenvía un datagrama IP en la red.

Métrica

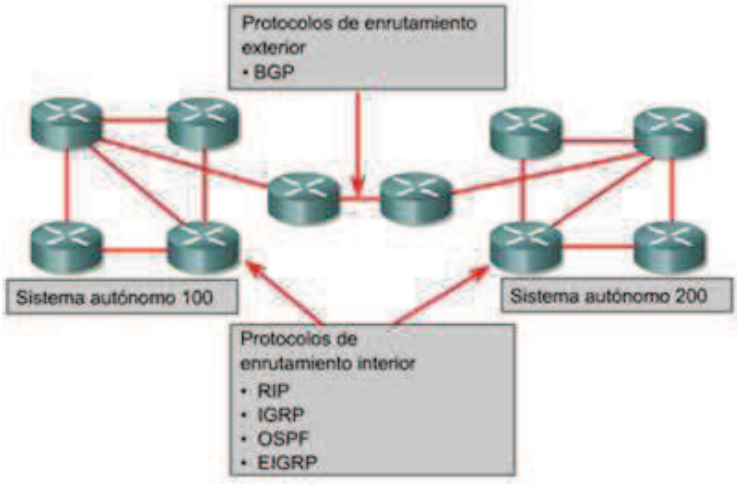
La métrica indica el costo del uso de una ruta, que suele ser el número de saltos al destino IP. Cualquier destino en la subred local está a un salto de distancia y cada enrutador que se atraviesa en la ruta es un salto adicional. Si existen varias rutas al mismo destino con diferentes métricas, se selecciona la ruta con menor métrica.

2.5.4. Protocolos de Enrutamiento

Los protocolos de enrutamiento son el conjunto de reglas utilizadas por un router cuando se comunica con otros routers con el fin de compartir información de enrutamiento. Dicha información se usa para construir y mantener las tablas de enrutamiento.

En muy importante tener en cuenta que a la hora de seleccionar un protocolo de ruteo para la red, se deben tener en cuenta las características de los protocolos y servicios de aplicaciones, los diseños de red que permiten un único protocolo de ruteo son los mejores para el rendimiento, mantenimiento y el diagnóstico de la red[38]. En la figura 2.9 se tiene el esquema general de los protocolos de enrutamiento.

[38] Revista Electrónica de Estudios Telemáticos, Volumen 4, Edición No 1, Año 2005, paginas 94,95



URL: http://fortalezadigital08.wordpress.com/2008/09/23/protocolos-de-enrutamiento-parte-1/
Figura 2.3. Esquema de Protocolos de Enrutamiento

2.5.5. Enrutamiento Dinámico

El ruteo dinámico utiliza diferentes protocolos cuyo fin es el de intercambiar rutas entre dispositivos intermedios con el objetivo de tener una red totalmente accesible. En este caso, los routers envían y reciben información de enrutamiento que utilizan para armar sus tablas de ruteo[39], ver figura 2.10.

Ventajas:

- El administrador tiene menos trabajo en el mantenimiento de la configuración cuando agrega o quita dispositivos en redes.
- Los protocolos reaccionan automáticamente a los cambios de topología.
- La configuración es menos propensa a errores.
- El crecimiento de la red normalmente no representa un problema debido a que es escalable.

Algunos protocolos de enrutamiento dinámicos son:

RIP: Protocolo de enrutamiento de Gateway Interior por vector distancia.

[39] Blog de WordPress.com, Protocolos de enrutamiento dinámico, http://vnanock.wordpress.com/2007/05/06/protocolos-de-enrutamiento-dinamicointroduccion/

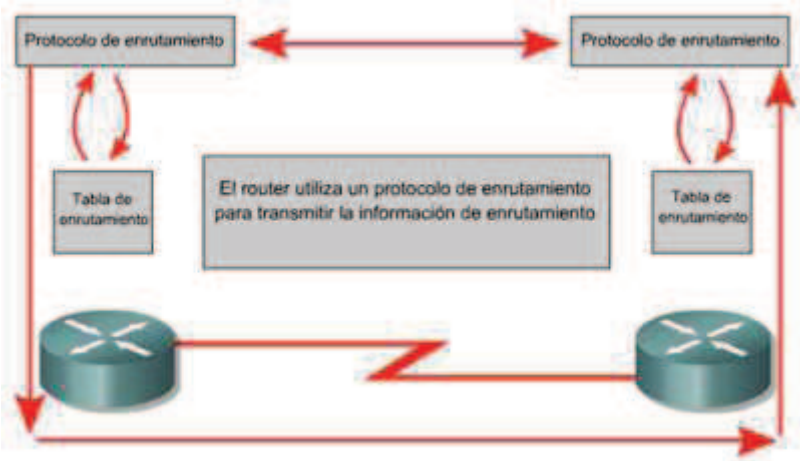
IGRP: Protocolo de enrutamiento de Gateway Interior por vector distancia, del cual es propietario CISCO.

EIGRP: Protocolo de enrutamiento de Gateway Interior por vector distancia, es una versión mejorada de IGRP.

OSPF: Protocolo de enrutamiento de Gateway Interior por estado de enlace.

BGP: Protocolo de enrutamiento de Gateway exterior por vector distancia.

Ejemplo de tablas de enrutamiento entre dos Routers, configurados con un protocolo dinámico.



URL: http://fortalezadigital08.wordpress.com/2008/09/23/protocolos-de-enrutamiento-parte-1/

Figura 2.4. Enrutamiento entre dos routers con un protocolo dinámico

2.5.6. Enrutador Basado en Software

Los enrutadores basados en software son esencialmente computadores personales con un sistema operativo estándar donde se encuentran instalados paquetes computacionales que permiten el manejo del enrutamiento. Estos paquetes computacionales pueden ser parte del sistema operativo estándar o pueden ser paquetes desarrollados para determinado sistema o sistemas operativos cuyo objetivo específico es el manejo de los protocolos de enrutamiento.

Los enrutadores basados en software no poseen elementos de hardware especializado para el propósito de encaminar la información, sino que más bien son computadores personales a los cuales se les ha dado la característica de poder enrutar

utilizando programas que ejecutan los diferentes algoritmos de enrutamiento y se adaptan a las características de hardware que poseen estos computadores.

2.5.6.1. Herramientas para enrutadores basadas en software (Quagga)

Entre los paquetes software de routing, de libre distribución, más usados cabe destacar Zebra y Quagga. Las mismas que soportan los protocolos de encaminamiento RIPv1, RIPv2, OSPFv2 y BGPv4. Además, estas herramientas también soportan protocolos de encaminamiento para IPv6 como RIPng, OSPFv3 y extensiones BGPv4.

La principal ventaja que proporcionan Zebra y Quagga es que son herramientas software de libre distribución (bajo la Licencia General Pública GNU) que implementan protocolos de encaminamiento no propietarios extensamente difundidos y utilizados. Otra ventaja es que este software permite un aprendizaje importante sobre la configuración de routers y puesta en práctica del encaminamiento dinámico, ya que los comandos utilizados para la configuración de los protocolos de encaminamiento son muy similares a que los que se utilizan para configurar equipos routers de proveedores como Cisco.

Sin embargo, para la configuración de este software se requiere mayor especialización, ya que no sólo es necesario prestar atención a las funciones de encaminamiento, sino que también es necesario gestionar el sistema operativo del PC router. Además, el hecho de disponer de pocas interfaces limita sus aplicaciones prácticas a redes con pocos enlaces.

En Zebra/Quagga, se tiene un demonio de routing para la gestión de cada protocolo de encaminamiento dinámico. De este modo, se tienen los demonios ripd, ospfd y bgpd que soportan los protocolos RIP o RIPv2, OSPFv2 y BGP-4, respectivamente. A su vez, todos se comunican con el demonio gerente (demonio zebra), el cual interacciona con el sistema operativo para la configuración general de las interfaces y para poner al día las tablas de encaminamiento del kernel.

A continuación se detalla en la Tabla 2.5 los demonios y los puertos utilizados por los mismos:

Demonio	Puerto	Transporte
zebra	2601	TCP
Ripd	2602	TCP
ripngd	2603	TCP
Ospfd	2604	TCP
Bgpd	2605	TCP

Tabla 2.4. Puertos utilizados por los demonios en zebra

La última versión del paquete de software Zebra fue lanzada el 27 de noviembre del 2003, por lo que actualmente se considera una versión antigua y con muchos errores, además con el desarrollo de las diferentes distribuciones, se necesitan muchas más consideraciones en este paquete; es por eso que nace el proyecto QUAGGA [40] que es una bifurcación del proyecto Zebra, siendo éste el encargado de ir desarrollando a Zebra a través de los diferentes errores y problemas que presentaba este paquete.

2.6. Firewall con Iptables

El término de Firewall proviene de la terminología, "paredes que resisten al fuego", la misma que es aplicada a la protección de una red de área local, en donde las máquinas externas que pretendan atacar al sistema se encuentran con un Firewall que limita el posible daño generado sí un intruso logra acceder a una máquina de la red interna que se intenta proteger y causarle daño, el Firewall protegerá a las otras máquinas de manera que el atacante no alcance niveles más altos dentro de la red [33.].

La idea principal de la construcción del firewall es la de cerrar todos los servicios e ir abriendo estos conforme requieran los usuarios.

La herramienta IPTables forma parte del sistema operativo LINUX y la función que desempeña es la de aplicar reglas para el filtrado de paquetes. Una vez que los paquetes comienzan a entrar o a salir, no importando de que máquina viene el paquete llega al kernel y el kernel decide qué hacer con este paquete; esto es si el paquete es para la propia máquina o para otra, para ello consulta las reglas del firewall y procede según las reglas del firewall[41].

[40] QUAGGA, http://www.quagga.net
[41] SEGURIDAD EN LA RED, http://bibing.us.es/proyectos/abreproy/11499/fichero/05+-+Seguridad+en+la+red+en+GNU-Linux.pdf

Iptables está basado en el uso de TABLAS dentro de las tablas, CADENAS, formadas por agrupación de REGLAS, parámetros que relativizan las reglas y finalmente una ACCION, que es la encargada de decir qué destino tiene el paquete.

2.6.1. Manejo de Cadenas dentro del Firewall

Las cadenas dentro del firewall tienen la función principal de organizar el flujo de los paquetes así como de definir el orden de ejecución. El orden de ejecución de estas cadenas tiene suma importancia, ya que de ello depende el que un paquete acceda o no al sistema esto es; si en un primer intento de pasar un paquete se accede a la red y después de esta regla se encuentra una que bloquee de alguna manera al paquete la jerarquía debería ser la opuesta[42].

Las cadenas que definen una acción son:

ACCEPT.- Paquete aceptado.

REJECT.- Paquete rechazado. Se envía notificación a través del protocolo ICMP.

DROP.- Paquete rechazado. Sin notificación

MASQUERADE.- Enmascaramiento de la dirección IP origen de forma dinámica. Esta acción es sólo válida en la tabla NAT en la cadena postrouting.

DNAT.- Enmascaramiento de la dirección destino, muy conveniente para re-enrutado de paquetes.

SNAT.- Enmascaramiento de la IP origen de forma similar a masquerade, pero con IP fija.

PREROUTING.- Lo que se hará antes de encaminar el paquete.

POSTROUTING.- Lo que se hará inmediatamente después de encaminar el paquete.

[42] IPTABLES, http://eisc.univalle.edu.co/materias/Administracion_De_Redes_Y_Servidores/material/IPTABLES_CORTO.pdf

2.6.2. Tipos de Filtrados.

Para realizar el filtrado de paquetes, se definen reglas basadas en:

- Direcciones
- Puertos (TCP, UDP, ICMP)
- Interfaz

Las reglas de filtrado incluyen estos aspectos junto con su estado de conexión que permiten saber si la conexión fue requerida por las máquinas que están detrás de la red o solo es un intento de conexión. De esta manera los filtrados se realizan:

- INPUT, es dirigido hacia el firewall.
- OUTPUT, es desde el firewall.
- FORWARD, es lo que pasa a través del Firewall.

La estructura es la siguiente:

#iptables –I | -A INPUT | OUTPUT | FORWARD

2.6.3. Elementos Básicos

En IPtables se utilizan script, que comienzan cargando los módulos necesarios (como el de ftp masquerading), establecen algún bit como por ejemplo el de forwarding, luego borra todas las reglas actuales, establece las políticas por defecto para la aceptación, reenvío y salida, y finalmente va aplicando todas las reglas de firewall, que varían dependiendo de las necesidades de cada red[43].

En IPtables se construyen reglas a partir de las siguientes órdenes básicas:

[43] PRÁCTICA 5: USO DE CORTAFUEGOS IPTABLES,http://gseguridad.unicauca.edu.co/talleres/practica_iptables.pdf

COMANDO	FUNCION
-A	Agregar nueva regla a la cadena especificada.
-I	Insertar nueva regla antes de la regla número_regla(rulenum) en la cadena especificada de acuerdo a los parámetros sometida.
-R	Reemplazar la regla (rulenum) en la cadena especificada.
-E	Modifica el nombre de la cadena. [nombre-anterior-cadena por nombre-nueva-cadena]
-L	Realiza el listado de reglas que se están aplicando. Si no se determina una cadena en particular, listará todas las cadenas existentes.
-N	Crear nueva cadena asociándola a un nombre.
-P	Modifica la acción por defecto de la cadena preseleccionada.
-D	Eliminar específicamente una de las reglas
-Z	Pone los contadores de paquetes y bytes a cero en la cadena seleccionada. De no poner seleccionar una cadena, pondrá a cero todos los contadores de todas las reglas en todas cadenas.

Tabla 2.5 Comandos de IPTables

Parámetros

Todas las reglas en iptables tienen definida su condición por los parámetros, que constituyen su parte primordial. Algunos de estos parámetros son:

PARAMETRO	FUNCION
-i	Interfaz de entrada (eth0,eth1,eth2...)
-o	Interfaz de salida (eth0, eth1, eth2...)
--sport	Puerto de origen
--dport	Puerto destino
-p	El protocolo del paquete a comprobar, tcp, udp, icmp ó all. Por defecto es all
-j	Esto especifica el objetivo de la cadena de reglas, o sea una acción
--line-numbers	Cuando se lista las reglas, agrega el número que ocupa cada regla dentro de la cadena

Tabla 2.6. Parámetros y sus funciones de IPTables

2.6.4. Correspondencia de Tablas, Cadenas y su función

TABLA	FUNCION	CADENA	FUNCION de la CADENA
FILTER	Filtrado de paquetes	INPUT	Filtrado de paquetes que llegan al firewall
		OUTPUT	Filtrado de los paquetes de salida
		FORWARD	Permite el paso de paquetes a otra dirección del firewall
NAT	Enrutamiento de direcciones de red	PREROUTING	Chequea la dirección de red antes de reenviarla. Facilita la modificación de la información para facilitar el enrutado
		POSTROUTING	Tratamiento de la dirección IP después del enrutamiento. Esto hace que no sea necesario la modificación del destino de la dirección IP del paquete como en pre-routing.
		OUTPUT	Interpretación de las direcciones de Red de los paquetes que salen del firewall. Escasamente usado.
MANGLE	Modificación de las cabeceras de TCP	PREROUTING POSTROUTING INPUT OUTPUT FORWARD	Permite la modificación del paquete como puede ser TOS (type of Service), marcado de los mismos para QOS o calidad de servicio

Tabla 2.7. Correspondencia entre tablas, función y cadena

2.7. Métodos Estadísticos

"La estadística es un método científico que se define como la recolección, presentación, análisis e interpretación de datos numéricos"[44].

La estadística tiene dos significados:

- Las estadísticas constituyen un conjunto de eventos comparables, referidos a un objeto
- La estadística es un conjunto de métodos para tratar o procesar series de eventos.

[44] EL PAPEL DE LA ESTADISTICA EN O Y M, http://html.rincondelvago.com/estadistica_23.html

2.7.1. Poblaciones y Muestras

La **población** representa un colectivo homogéneo que reúne características determinadas. En cambio, la **muestra** es el subconjunto de la población accesible y limitado sobre el que se realiza las mediciones o el experimento con la idea de obtener conclusiones.

Cuando se realiza un estudio de investigación, se pretende generalmente inferir o generalizar resultados de una muestra a una población. No obstante, para esta investigación, se estudia a un reducido número de ataques a redes IP, a los que se tiene acceso con la idea de poder generalizar los hallazgos a la población de la cual esa muestra procede.

En el presente trabajo se tomaron muestras. Entre las razones para estudiar muestras en lugar de poblaciones se pude señalar:

a) Ahorrar tiempo. Estudiar un determinado número de ataques a redes IP es evidente que lleva menos tiempo.

b) Ahorrar costos.

c) Aumentar la calidad del estudio. Las observaciones y mediciones realizadas a un reducido número de ataques pueden ser más exactas y plurales que si se realiza a toda una población.

d) La selección de muestras específicas permite reducir la heterogeneidad de una población al indicar los criterios de inclusión y/o exclusión.

Tipos de datos

Lo que se estudia en cada ataque a redes IP de la muestra son las variables: tiempo (duración de ataque), consumo de recursos de red, consumo de CPU y cantidad de paquetes transmitidos. Los datos son los valores que toma la variable en cada caso.

2.7.2. Tipo de variables

Variables cuantitativas. Son las variables que pueden medirse, cuantificarse o expresarse numéricamente.

Variables cualitativas. Este tipo de variables representan una cualidad o atributo que clasifica a cada caso en una de varias categorías. La situación más sencilla es aquella en la que se clasifica cada caso en uno de dos grupos (acierta/no acierta, descifra/no descifra).

2.7.3. Estadística Descriptiva

Una vez recogido los valores que toman las variables del presente estudio (datos), se procedió al análisis descriptivo de los mismos. Para variables categóricas, como el acierto en descifrar una clave, fue necesario conocer el número de casos en cada una de las categorías, reflejando habitualmente el porcentaje que representaron del total[45].

Para variables numéricas, en las que puede haber un gran número de valores observados distintos, se ha optado por un método de análisis distinto, respondiendo a la pregunta: ¿Alrededor de qué valor se agrupan los datos?

a) Medidas de tendencia central

La medida más evidente que se puede calcular para describir un conjunto de observaciones numéricas es su valor medio. La media no es más que la suma de todos los valores de una variable dividida entre el número total de datos de los que se dispone.

Más formalmente, si se denota por (X_1, X_2,...,X_n), los n datos que se tiene recogidos de la variable en cuestión, el valor medio vendrá dado por:

$$Media(X) = \frac{\sum_{j=1}^{n} X_j}{n}$$

[45] Estadística descriptiva de los datos, http://www.fisterra.com/mbe/investiga/10descriptiva/10descriptiva.asp

Otra medida de tendencia central que se utiliza habitualmente es la mediana. Es la observación equidistante de los extremos.

La mediana por lo general es el valor que deja a la mitad de los datos por encima de dicho valor y a la otra mitad por debajo.

Si la media y la mediana son iguales, la distribución de la variable es simétrica. La media es muy sensible a la variación de las puntuaciones. Sin embargo, la mediana es menos sensible a dichos cambios.

Finalmente, otra medida de tendencia central, no tan usual como las anteriores, es la moda, siendo éste el valor de la variable que presenta una mayor frecuencia.

b) Medidas de dispersión

Otro aspecto tomado en cuenta al describir datos continuos fue la dispersión de los mismos. Existen distintas formas de cuantificar esa variabilidad. De todas ellas, la varianza (S^2) de los datos fue la más utilizada. Es la media de los cuadrados de las diferencias entre cada valor de la variable y la media aritmética de la distribución.

$$S_X^2 = \frac{\sum_{j=1}^{n} \left(X_j - Media\,(X)\right)^2}{n}$$

Esta varianza de muestras se obtiene como la suma de las de las diferencias de cuadrados y por tanto tiene como unidades de medida el cuadrado de las unidades de medida en que se mide la variable estudiada.

CAPÍTULO III

PLATAFORMA EXPERIMENTAL BASADA EN TECNOLOGÍAS DE VIRTUALIZACIÓN

3.1. Escenario Virtual de Red

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[13.].

Para implementar los escenarios virtuales para el presente trabajo de investigación, se ha elegido VMware Player 3.0, que es una plataforma de libre distribución basada 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.

La plataforma VMware Player 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[46].

3.2. Diseño y Configuración de las Máquinas Virtuales

Para la realización del presente trabajo de investigación, se ha elegido las herramientas de VMWare Player 3.0.1 y VirtualBox 3.2, para la creación de la plataforma de virtualización, por las ventajas que ofrece esta herramienta.

La herramienta facilita enormemente la gestión de los escenarios de prueba, permitiendo su arranque de forma sencilla, su transporte entre máquinas, su

[46] VMware home page, [Online:] http://www.vmware.com

almacenamiento para usos posteriores, etc. Todo ello sin que sea posible diferenciar el comportamiento de los sistemas virtuales de lo que sería el mismo sistema ejecutado sobre una máquina real [20.]. Además permite crear escenarios de red más complejos de lo que permite el equipamiento existente, proporcionando al usuario escenarios de red más reales sobre los que trabajar y mejorar con ello sus pruebas[47].

3.2.1. Máquinas Virtuales en VMware

VMWare Player es una herramienta gratuita de virtualización que permite ejecutar simultáneamente varios sistemas operativos sobre el mismo hardware.

VMware Player simula un PC dentro de nuestro sistema operativo, permitiéndonos analizar otros sistemas operativos sin necesidad de instalarlos en nuestro disco duro. Se pueden crear máquinas virtuales para instalar muchos sistemas Linux, Windows, Nobel, Solaris y FreeBSD. Además permite ejecutar máquinas virtuales creadas con VMware Workstation o VMware Server, e incluso máquinas virtuales en formatos Microsoft.

Requisitos de Hardware

Los requisitos de hardware recomendados para instalar VMWare Player son:

- Un procesador de 2 Ghz
- 2 GB de RAM
- Espacio libre de 150 MB en disco. Hay que tener en cuenta el espacio libre en el disco a la hora de crear múltiples máquinas virtuales y el uso de la memoria RAM.

El funcionamiento básico de VMWare Player se puede resumir en el siguiente esquema:

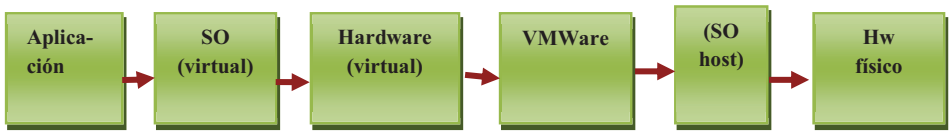


Figura 3.1 Funcionamiento básico de VMWare Player

[47] virtualizacion de servidores de telefonia ip en gnu/linux, http://www.adminso.es/images/6/6d/Eugenio_cap1.pdf

Una aplicación corriendo en el sistema operativo virtual ejecuta sus llamadas al sistema actuando sobre elementos de hardware también virtuales. Estas llamadas son capturadas por la aplicación VMWare que las traduce a instrucciones sobre elementos físicos reales y las devuelve de nuevo hacia el sistema operativo virtual. De esta manera, el SO virtual se ejecuta a una velocidad menor que en el caso de estar instalado directamente sobre la máquina, pero con un rendimiento bastante bueno.

Una vez que se accede a la ventana principal, se tiene las siguientes opciones: la creación de una nueva máquina virtual, abrir una máquina virtual y actualización de VMWare (ver figura 3.3).

Figura 3.2. Ventana de inicio de VMWare Player

Para crear una máquina virtual se elige la opción New Virtual Machine que hará que se ejecute el asistente de creación de máquinas virtuales.

3.2.2. Máquinas Virtuales en Oracle VM Virtualbox 3.2

Oracle VM VirtualBox es un programa de virtualización con el que es posible instalar sistemas operativos adicionales, conocidos como sistemas invitados, dentro de otro sistema operativo anfitrión, cada uno con su propia máquina virtual. En nuestro caso,

el sistema operativo anfitrión será Ubuntu y los invitados los sistemas operativos Windows y Ubuntu que se irán añadiendo a VirtualBox.

Virtualbox es un software que nos permite virtualizar sistemas operativos. Fue adquirido por Sun y después Sun fue adquirido por Oracle, es por esa razón que el nombre del producto es Oracle VM VirtualBox.

Para poder ejecutar VirtualBox en su máquina, se necesita:

Requisitos de Hardware

- Cualquier procesador Intel o AMD recientes deben hacer.
- Dependiendo de lo que los sistemas operativos invitados que desea ejecutar, necesitará al menos 512 MB de RAM (pero probablemente más, y mientras más, mejor).

Espacio en disco duro que puede alcanzar varios GB de tamaño.

La siguiente captura de pantalla muestra la ventana de inicio de VirtualBox con las siguientes opciones: Nueva, Configuración, iniciar y Descartar, ver figura 3.4.

Figura 3.3 Ventana de inicio de Oracle MV VirtualBox

Para crear una máquina virtual se elige la opción **Nueva**, seguidamente se ejecuta el asistente de creación de máquinas virtuales.

3.3. Diseño del Esquema Virtual de Red IP

Hoy en día la virtualización no sólo se circunscribe a la ejecución de sistemas operativos de propósito general como GNU/Linux o Windows, si no que permite también la emulación (limitada) de equipos de comunicaciones como routers CISCO mediante el software Quagga o Zebra.

Las posibilidades de la virtualización no sólo se limitan a la creación de máquinas aisladas; mediante la adecuada interconexión de dichas máquinas por medio de redes virtuales es posible crear escenarios de red completos. La herramienta se ocupa de arrancar y configurar automáticamente los escenarios definidos y de conectarlos con el exterior a través de los interfaces de red del equipo anfitrión.

VMware también permite definir de forma sencilla cómo los escenarios virtuales se conectan con máquinas externas, permitiendo escenarios complejos formados por equipos reales sobre los que trabajen directamente los usuarios y por otros virtuales que aumentan el realismo del escenario (por ejemplo, servicios típicos como DNS, mail, web, etc, máquinas de diagnóstico para poder ejecutar pings o traceroutes, routers que incrementen el tamaño de las tablas de encaminamiento, etc.)

La necesidad de crear una plataforma de experimentación con diferentes escenarios para llevar a cabo los ataques reales a redes IP descritos en el capítulo 2, se ha diseñado una topología de prueba con VMware Player, tomado como modelos aquellos escenarios de uso más común en pequeña y mediana organización. La Figura 3.5 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.

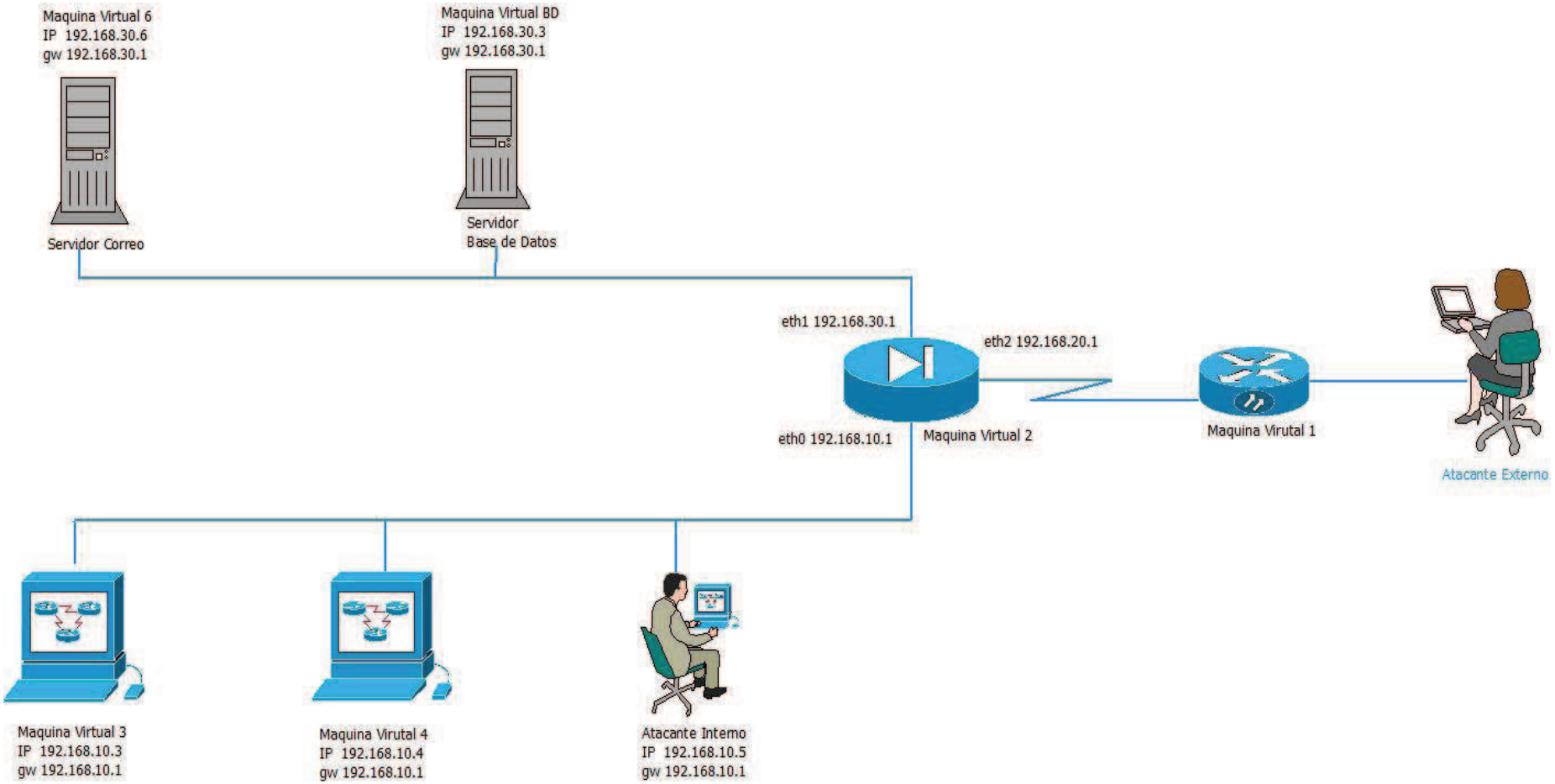


Figura 3.4 Diseño de la topología de prueba

3.4. Configuración del Escenario Virtual de Red.

El procedimiento utilizado para implementar el experimento consistió en los siguientes pasos:

3.4.1. Creación de la Máquina Host (Anfitriona)

Todas las pruebas se desarrollaron sobre Linux Ubuntu Server -i386, en un computador Pentium Intel Core Duo, RAM de 4GB y 512 GB en disco duro. En todas las VMs se instaló el mismo sistema de ficheros y el mismo kernel.

Este servidor puede verse como un particionado de un servidor físico de manera que pueda albergar distintos servidores dedicados (o privados) virtuales que ejecutan de manera independiente su propio sistema operativo y dentro de él los servicios que quieran ofrecer, haciendo un uso común de manera compartida y aislada sin ser conscientes del hardware subyacente.

3.4.2 Creación de la Plataforma de Virtualización.

Para la creación de la plataforma de virtualización se utilizó la herramienta VMWare Player 3.0.1. Una vez instalada y configurada se procedió a la creación de seis máquinas virtuales, las mismas que al realizar la abstracción de todo el hardware subyacente de la plataforma creada pueden ejecutarse de manera independiente, con la ilusión de que los recursos abstraídos les pertenecen en exclusiva. Esto es muy importante, ya que cada máquina virtual no ve a otra máquina virtual como tal, sino como otra máquina independiente de la que desconoce que comparte con ella ciertos recursos.

3.4.3 Direccionamiento IP sobre el Esquema Virtual de Red

Para identificar cada una de las VM's y considerando que el alcance de la red es local, se usaron los siguientes parámetros mostrados en la tabla 3.1.

Máquina Virtual	Dirección IP	Máscara de Subred	Puerta de enlace predeterminada
MaqVirtual01	Eth0 192.168.11.1 Eth1 192.168.20.2	255.255.255.0 255.255.255.0	0.0.0.0 0.0.0.0
MaqVirtual02	Eth0 192.168.10.1 Eth1 192.168.20.1 Eth2 192.168.30.1	255.255.255.0 255.255.255.0 255.255.255.0	0.0.0.0 0.0.0.0 0.0.0.0
MaqVirtual03	Eth0 192.168.10.3	255.255.255.0	192.168.10.1
MaqVirtual04	Eth0 192.168.10.4	255.255.255.0	192.168.10.1
MaqVirtual05	Eth0 192.168.10.5	255.255.255.0	192,168.10.1
MaqVirtualBD	Eth0 192,168.30.3	255.255.255.0	192.168.30.1
MaqVirtual06	Eth0 192,168.30.4	255.255.255.0	192.168.30.1

Tabla 3.1. Esquema de direccionamiento de red para VM's

3.4.4 Servidor de Correo Electrónico Seguro (Webmail) con Postfix – Sasl –Tls – Dovecot –Squirrelmail

La MV4 hace la función de Servidor Web, DNS y Correo electrónico seguro y se encuentra ubicada 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).

Un servidor de correo es una aplicación que nos permite enviar mensajes (correos) de unos usuarios a otros, con independencia de la red que dichos usuarios estén utilizando. La mejor manera de preservar la intimidad en los mensajes de correo electrónico es recurrir a la criptografía, añadida de otros servicios, como la integridad, la autenticación de usuarios, la certificación de conexiones, que autorizan el uso del servicio, de manera se puede estar seguro de que fue escrito por quien lo envió y no ha sido falsificado[48].

En el ANEXO A, se describe el manual de instalación y configuración de Postfix – Sasl – Tls – Dovecot, para conseguir un servidor de correo Seguro.

[48] Linux, Servidor OpenLDAP, http://www.humbug.in/docs/ubuntu-server-guide-es-10.04/openldap-server.html

3.4.5. *Enrutador Basado en Software de Código Abierto para Linux*

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).

3.4.5.1. Instalación y configuración de Quagga

Para la instalación de Quagga se debe ejecutar el siguiente comando:

```
#apt-get install quagga
```

El directorio de configuración de quagga, se encuentra en el directorio: */etc/quagga/*.

Los archivos que se tomarán de referencia para realizar la configuración están en: */usr/share/doc/quagga/examples/* y son:

- Daemons: Contiene los protocolos soportados y los cuales pueden activarse o desactivarse.
- Debian.conf: Contiene la configuración general para habilitar la consola de trabajo del paquete quagga.

La configuración completa de Quagga se describe en el ANEXO B,

3.4.6. Configuración del Firewall

La configuración del firewall tiene como fases la asignación de IP, la implementación de las interfaces de red según el modelo de la figura 3.5 y la generación de scripts que contengan las reglas para el filtrado de paquetes dentro de la red privada y los redireccionamientos para la interacción entre los servidores, la red privada y el exterior.

El estableciendo de la comunicación entre el firewall Linux y la Intranet se dará a través de la configuración de la máquina Firewall y la Subred, ésta se realiza con el paquete IPtables. Se debe contar con tres tarjetas de red en la máquina firewall; la eth0 es la tarjeta de red que conecta a la máquina firewall con Internet con una IP 192.168.20.1, la eth1 es la tarjeta de red que conecta la máquina firewall con la red local mediante la asignación de la dirección 192.168.10.1, la eth2 es la tarjeta de red que conecta a la máquina firewall con la zona desmilitarizada en la que se encuentran los servidores, cuya IP es la 192.168.30.1.

CAPÍTULO IV

EMULACIÓN DE ATAQUES EN UN ENTORNO VIRTUAL DE RED

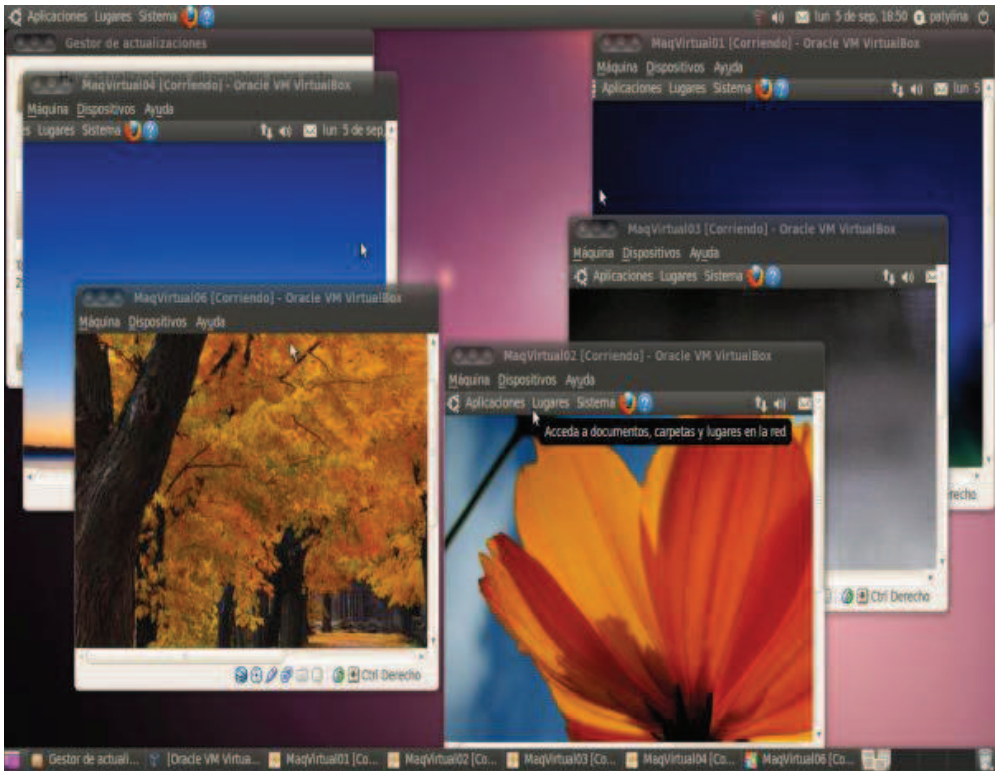
4.1. Implementación del Escenario de Red Virtual

El diseño de la topología de prueba de la figura 3.5 fue implementada tanto en VMWare (figura 4.1-a), como en VirtualBox (figura 4.2-b). Sobre éstas dos plataformas se puso en funcionamiento la red interna LAN y la zona desmilitarizada, permitiendo la interacción entre cada máquina virtual (estaciones) con el servidor web y de correo, además de la salida hacia Internet (exterior)[21.].

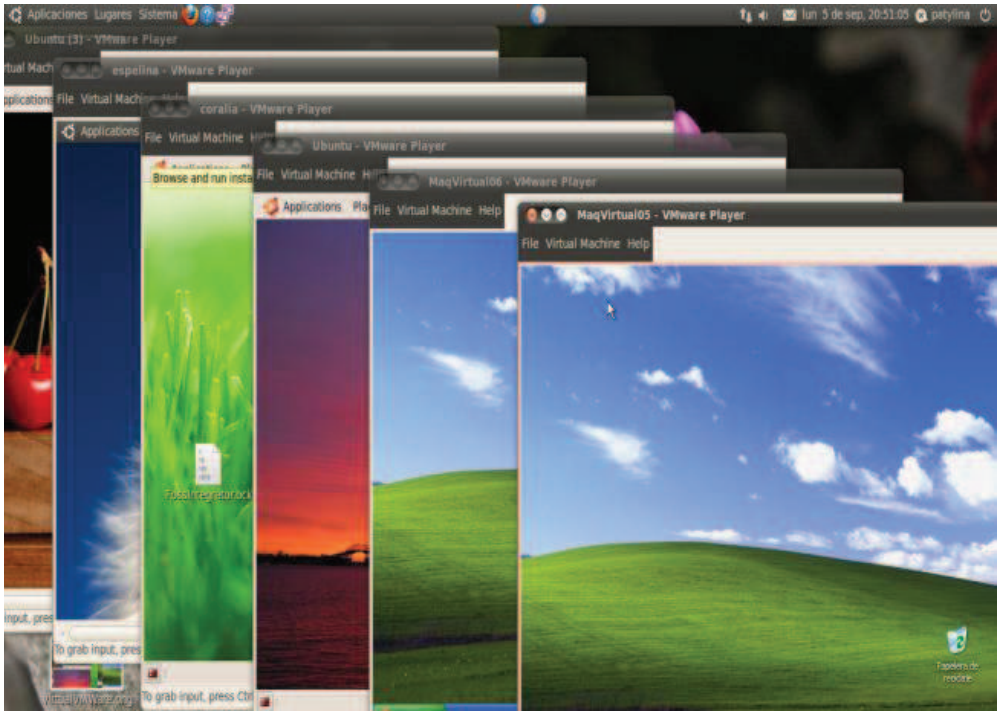
Para la comunicación entre los equipos virtuales fue necesario configurar los adaptadores de red respectivos. Cabe mencionar que en la configuración de las tarjetas se tiene tres posibilidades:

- Bridged: De esta forma se le asigna a la tarjeta de red de la máquina virtual una IP real visible desde toda la red real.
- NAT: La máquina real actuará como router NAT convirtiendo las direcciones internas en direcciones compatibles con el resto de la red real.
- Host-only : Se crea una red privada entre el ordenador real y la máquina virtual

Para la realización del presente trabajo se utilizó la conexión Bridged logrando tener comunicación entre los equipos virtuales de la red LAN interna y la salida de las mismas hacia el internet.



a) Escenario de pruebas con VMWare Player 3.01



b) Escenario de pruebas con Oracle VirtualBox 3.01

Figura 4.1 Escenarios de Pruebas con Tecnología de Virtualización

En las dos plataformas virtuales creadas con VMware y VirtualBox respectivamente, el servidor Virtual funciona sobre Ubuntu 10.04 y constituye la máquina anfitriona sobre la cual se crearon y configuraron seis MVs, cada una con su dirección IP respectivas, de las cuales cuatro tienen sistema Operativo Ubuntu y dos Windows XP[22.][23.].

4.2. Implementación Funcional del Ruteador a Través del Software Quagga en una Mv

Para establecer una máquina con Linux como enrutador, se configurará la máquina con dos o tres interfaces de red, de acuerdo a la topología descrita en la figura 3.5, cada ruteador está conectada a una red local de tal manera que las diferentes subredes formadas dentro del laboratorio puedan interactuar entre sí. Para ello se usará enrutamiento dinámico.

Para lograr el objetivo planteado, el de configurar una máquina como ruteador, es necesario realizar las configuraciones respectivas que se describen más adelante y utilizar las direcciones IP de las tarjetas indicadas en la Figura 4.2, que resume la topología de la figura 3.5 para una mayor comprensión.

4.2.1. *Configuración de Enrutamiento Dinámico Ipv4 utilizando el Demonio Ripd*

El demonio ripd es el demonio encargado de administrar el enrutamiento dinámico utilizando el protocolo RIP. Para que pueda funcionar el demonio ripd es necesario que el protocolo Zebra esté inicializado previamente, por ser éste el demonio gerente que administra a los demás demonios de enrutamiento; además es necesario configurar las interfaces en el demonio Zebra. El demonio Zebra funciona el puerto 2601 y el demonio ripd en el puerto 2602, ambos pueden ser configurados, por separado, a través de Telnet.

En la Figura 4.2 se observa el escenario donde se analizará la configuración del demonio ripd.

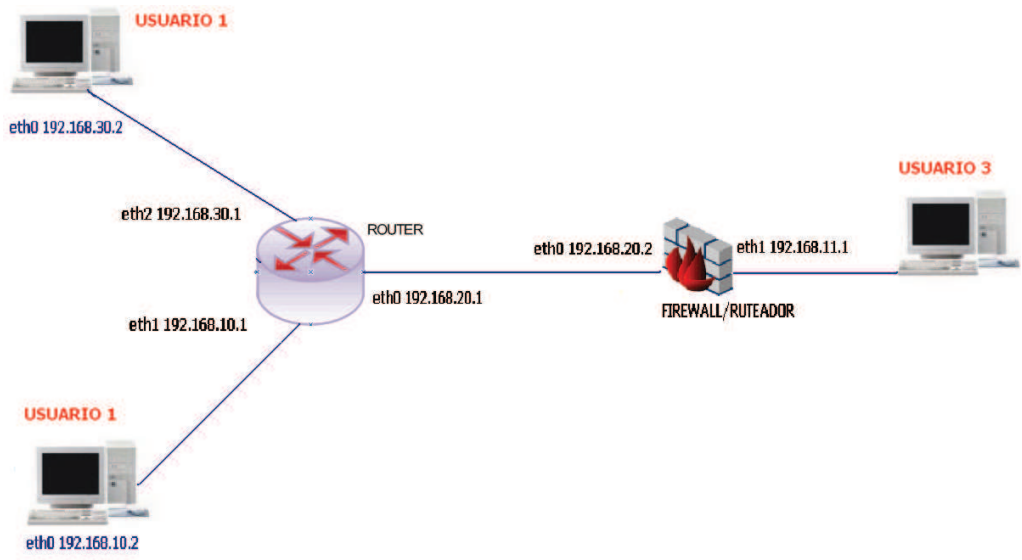


Figura 4.2. Escenario para enrutamiento dinámico IPv4 con RIP

Lo que se busca sobre este escenario es establecer la comunicación entre el Usuario 1, el Usuario 2 y Usuario3, utilizando RIP en los enrutadores A y B; para esto es necesario en primera instancia configurar el demonio Zebra, en donde se debe incluir la información de la interfaces, sus direcciones IP y máscaras de subred; como se describe a continuación:

Enrutador A	Enrutador B
Ingreso al modo de configuración de consola	**Ingresar al modo de configuración de consola**
# telnet localhost 2601	*# telnet localhost 2601*
Password:	*Password:*
ROUTER_A>	*ROUTER_B>*
Ingreso al modo de configuración global	**Ingreso al modo de configuración global**
ROUTER_A>	*ROUTER_B>*
ROUTER_A> enable	*ROUTER_B> enable*
Password:	*Password:*
ROUTER_A# configure terminal	*ROUTER_B# configure terminal*
Configuración de interfaces	**Configuración de interfaces**
ROUTER_A(config)# interface eth0	*ROUTER_B(config)# interface eth0*
ROUTER_A(config-if)#ip address	*ROUTER_B(config-if)#ip address*
192.168.20.1/24	*192.168.20.2/24*
ROUTER_A(config-if)# no shutdown	*ROUTER_B(config-if)# no shutdown*
ROUTER_A(config-if)# exit	*ROUTER_B(config-if)# exit*
ROUTER_A(config)# interface eth1	*ROUTER_B(config)# interface eth1*

Enrutador A	Enrutador B
ROUTER_A(config-if)#ip address	*ROUTER_B(config-if)#ip address*
192.168.10.1/24	*192.168.11.1/24*
ROUTER_A(config-if)# no shutdown	*ROUTER_B(config-if)# no shutdown*
ROUTER_A(config-if)# exit	*ROUTER_B(config-if)# exit*
Guardar configuración	**Guarda configuración**
ROUTER_A#write	*ROUTER_A#write*
ROUTER_A#exit	*ROUTER_A#exit*

Tabla 4.1. Configuración demonio Zebra**.**

El próximo paso es configurar RIP a través del demonio ripd; esta configuración se la puede realizar mediante consola, como se observa a continuación:

Enrutador A	Enrutador B
Ingreso al modo de configuración de consola	**Ingreso al modo de configuración de consola**
# telnet localhost 2602	*# telnet localhost 2602*
Password:	*Password:*
ROUTER_RIP_A>	*ROUTER_RIP_B>*
Ingresar en el modo de configuración global	**Ingresar en el modo de configuración global**
ROUTER_RIP_A >	*ROUTER_RIP_B >*
ROUTER_RIP_A > enable	*ROUTER_RIP_B > enable*
Password:	*Password:*
ROUTER_RIP_A # configure terminal	*ROUTER_RIP_B # configure terminal*
Configurar las redes para RIP	**Configurar las redes para RIP**
ROUTER_RIP_A(config)# router rip	*ROUTER_RIP_B(config)# router rip*
ROUTER_RIP_A(config-router)#network	*ROUTER_RIP_B(config-router)#network*
192.168.20.0/24	*192.168.20.0/24*
ROUTER_RIP_A(config-router)#network	*ROUTER_RIP_B(config-router)#network*
192.168.10.0/24	*192.168.11.0/24*
ROUTER_RIP_A(config-router)#exit	*ROUTER_RIP_B(config-router)#exit*
Guarda configuración	**Guarda configuración**
ROUTER_A#write	*ROUTER_A#write*
ROUTER_A#exit	*ROUTER_A#exit*

Tabla 4.2. Configuración demonio RIP.

4.2.2. *Verificación de las Rutas*

Una vez configurado los demonios Zebra y Rip de los equipos ruteadores, se procede a verificar que las rutas consten en las tablas de enrutamiento de cada ruteador, para esto se procede a ingresar al daemon zebra y con la opción *show ip route* se puede obtener el contenido de la tabla de enrutamiento correspondiente del ruteador. Claramente se puede observar sentencias como: *R>* 192.168.10.0/24 [120/2] via 192.168.20.1, eth0, 01:18:05, para el caso de Ruteador B, y R>* 192.168.2.0/24 [120/2] vía 192.168.20.2, eth0, 01:11:09*, en el Ruteador A, lo que significa que cada ruteador tiene habilitado su respecta vía de acceso por donde fluirá el tráfico desde o hacia el otro ruteador respectivamente, como se muestra a continuación:

Enrutador A	Enrutador B
Ingreso al modo de configuración de consola *# telnet localhost 2601* *Password:* *ROUTER_A>* *ROUTER_A> enable* *Password:* *ROUTER_A# show ip route* *Codes: K - kernel route, C - connected, S - static, R - RIP, O - OSPF,* *I - ISIS, B - BGP, > - selected route, * - FIB route* *C>* 127.0.0.0/8 is directly connected, lo* *K>* 169.254.0.0/16 is directly connected, lo* *C>* 192.168.10.0/24 is directly connected, eth1* *R>* 192.168.2.0/24 [120/2] via 192.168.20.2, eth0, 01:11:09* *C>* 192.168.20.0/24 is directly connected, eth0*	Ingreso al modo de configuración de consola # telnet localhost 2601 *Password:* *ROUTER_B>* *ROUTER_B> enable* *Password:* *ROUTER_B# show ip route* *Codes: K - kernel route, C - connected, S - static, R - RIP, O - OSPF,* *I - ISIS, B - BGP, > - selected route, * - FIB route* *C>* 127.0.0.0/8 is directly connected, lo* *K>* 169.254.0.0/16 is directly connected, lo* *C>* 192.168.10.0/24 is directly connected, eth1* *R>* 192.168.10.0/24 [120/2] via 192.168.20.1, eth0, 01:18:05* *C>* 192.168.20.0/24 is directly connected, eth0*

Tabla 4.3. Tabla de Enrutamiento de los Ruteadores A y B.

Para el correcto funcionamiento del ruteo se debe considerar que los usuarios deben tener la siguiente configuración en su respectiva interfaz de red:

Usuario 1	Usuario 2	Usuario 3
Dirección IP 192.168.10.2 Máscara de Subred 255.255.255.0 Puerta de enlace 192.168.10.1	Dirección IP 192.168.20.2 Máscara de Subred 255.255.255.0 Puerta de enlace 192.168.20.1	Dirección IP 192.168.2.2 Máscara de Subred 255.255.255.0 Puerta de enlace 192.168.2.1

Tabla 4.4. Configuración de la IP de los usuarios de la Red.

4.3. Implementación de los Ataques y Análisis de Resultados.

En esta sección se describe la ejecución de los cuatro ataques en estudio: Rastreo de Sistemas o Escaneo de Puertos, Fuerza Bruta, Spoofing y Denegación de Servicio, haciendo uso de herramientas o aplicaciones de libre distribución.

4.3.1. Técnicas de Rastreo de Sistemas con NMAP

Como se mencionó anteriormente Nmap es un buen rastreador de puertos, que usado con las directivas adecuadas se puede evitar hacer ruido y dejar huellas en las máquinas objetivo[24.]. Desde las máquinas de pruebas se puede realizar los siguientes barridos:

-sT: Se basa en la metodología de inicio de conexión que posee el TCP, conocida como el "three way handshake", el cual funciona como sigue:

a) La máquina destino está preparado para recibir una conexión.

b) La máquina cliente lanza una conexión activa llamada connect(). Con esto envía un mensaje con el segmento SYN activo para informarle al equipo destino el número inicial de secuencia para los datos que el cliente va a enviar en la conexión.

c) La máquina destino debe dar por conocido el SYN enviando un ACK-SYN con su número de secuencia, en un solo paquete TCP.

d) La máquina cliente debe dar por conocido el SYN enviando con un ACK.

En consecuencia se obtendrá la respuesta con datos de la máquina objetivo. Este método de escaneo posee dos ventajas:

1) Es rápido

2) No necesita privilegios especiales para realizarlo en la máquina que lanza el barrido.

-sS : Rastreo de segmento SYN medio abierto o Half Open. Esta es una técnica que envía un segmento SYN, si recibe como respuesta un ACK es porque ha detectado un puerto activo en la máquina objetivo, después de lo cual envía un Reset para cortar de forma abrupta la comunicación. Pero Si en vez de un ACK recibe un RST significa que el puerto de la máquina objetivo no se encuentra activo. De este modo el barrido posee la desventaja que se deben tener privilegios de root para ejecutarlo.

-sF, -sX, -sN : Escaneo utilizando segmentos FIN y que responden con un paquete RST. Los puertos activos en este caso ignoran dichos paquetes por lo que se debe observar los puertos que no contestan para tener un concepto claro de cuáles son los puertos abiertos bajo.

Estas son algunas de las técnicas más conocidas y utilizadas para realizar rastreos de puertos o escáner de estos.

Caso 1. Rastreos con banderas FIN

```
#Nmap  -sF 192.168.30.2 -p 21
```

En este ataque se realizaron dos pruebas:

a) Envío de un paquete con un Flag FIN al puerto 80 de Google, por lo que nos tiene que responder con un RST si el puerto está abierto, ver Figuras 4.3-a . Y el envío del mismo paquete pero al puerto 21.

b) Envío de un paquete con un Flag FIN al puerto 80 del Servidor Web de una LAN interna (192.168.30.2). Igualmente el envío del mismo paquete pero al puerto 21, ver Figura 4.3-b.

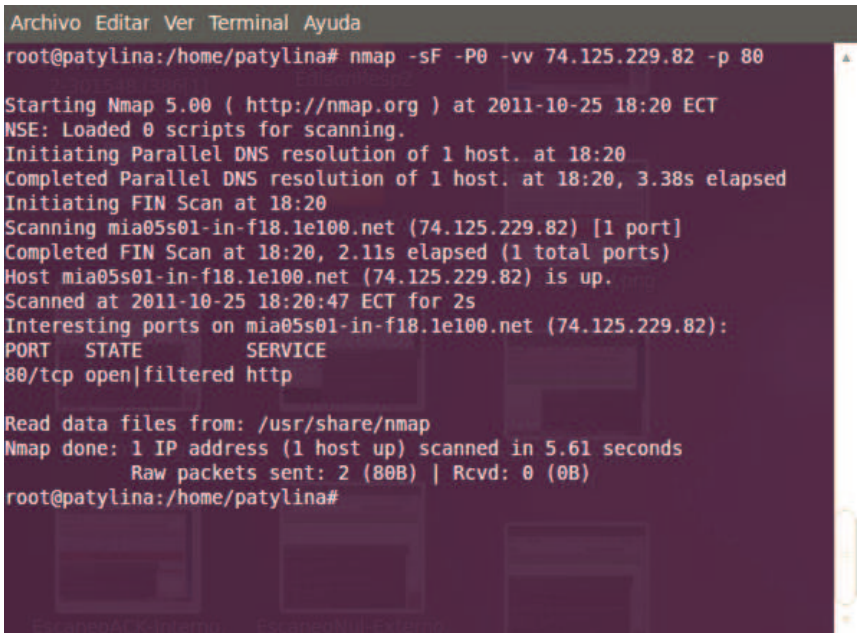


Figura 4.3-a. Escaneo de Puertos TCP FIN (Externo).

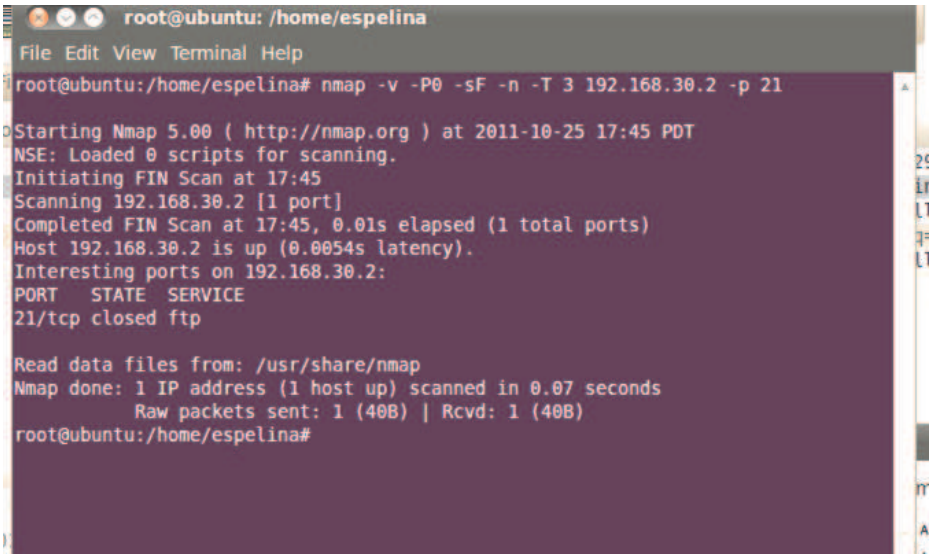


Figura 4.3-b. Escaneo de Puertos TCP FIN (Interno).

Resultados Obtenidos

A continuación se describe las pantallas de captura de paquetes del sniffer Wireshark[31.], ante un ataque de escaneo de puertos TCP FIN, figuras 4.4-a y 4.4-b.

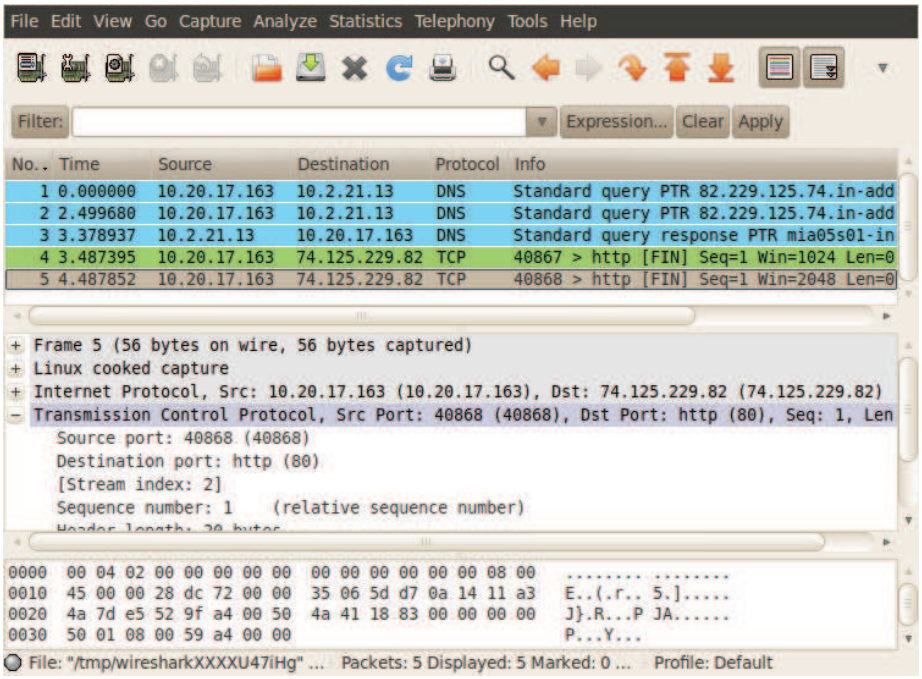


Figura 4.4-a. Captura de paquetes de un Escaneo TCP FIN (Externo).

En verde aparecen los paquetes enviados desde el equipo atacante.

La captura nro. 4 es el primer envío (sale del puerto 40867 hacia el 80, desde 10.20.17.163 a 74.125.229.82).

Y se puede observar que no se recibe respuesta por parte del servidor de Google, lo que significa que el puerto está en estado silencioso.

Cabe mencionar que este tipo de escaneo genera una enorme lista de puertos con estado abierto aunque realmente pueden estar cerrados o silenciosos.

En el envío de paquetes hacia el puerto 80, del mismo servidor Web, los resultados fueron los mismos que los obtenidos para el puerto 21.

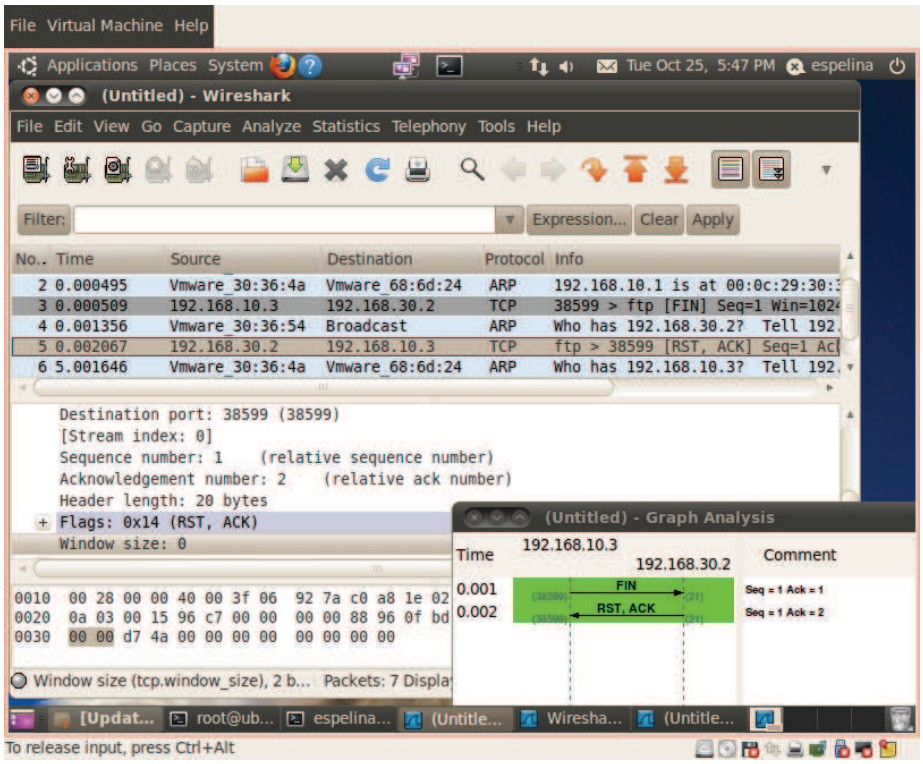


Figura 4.4-b. Captura de paquetes de un Escaneo TCP FIN (interno).

Caso 2. Rastreos con banderas SYN

```
#Nmap  -sS 192.168.30.2 -p 21
```

Para saber si un puerto X, de una host Destino Y, está disponible no se necesita establecer realmente una conexión, basta con enviar un único paquete SYN y esperar a ver que responde como se observa en la figura 4.5.

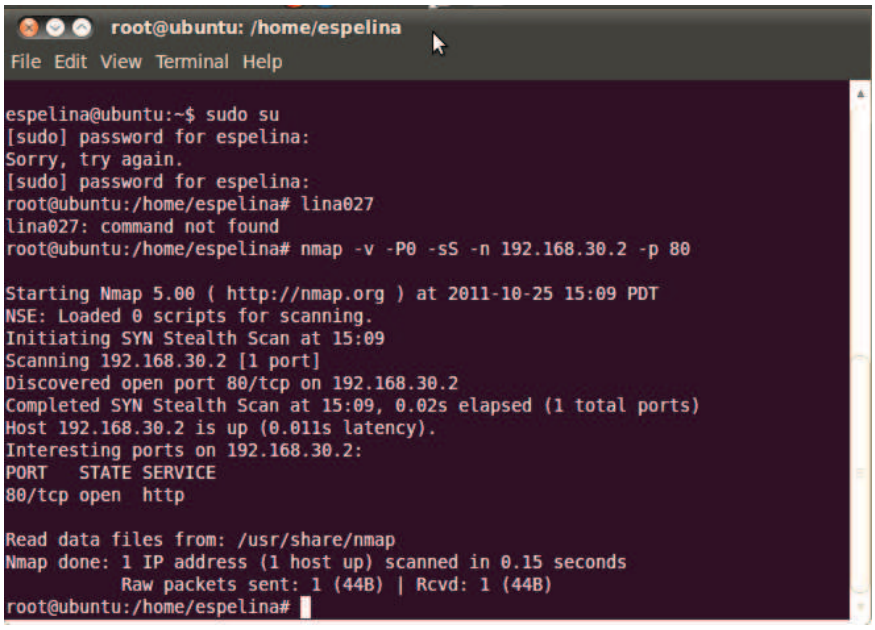


Figura 4.5 Escaneo de Puertos TCP FIN (Interno).

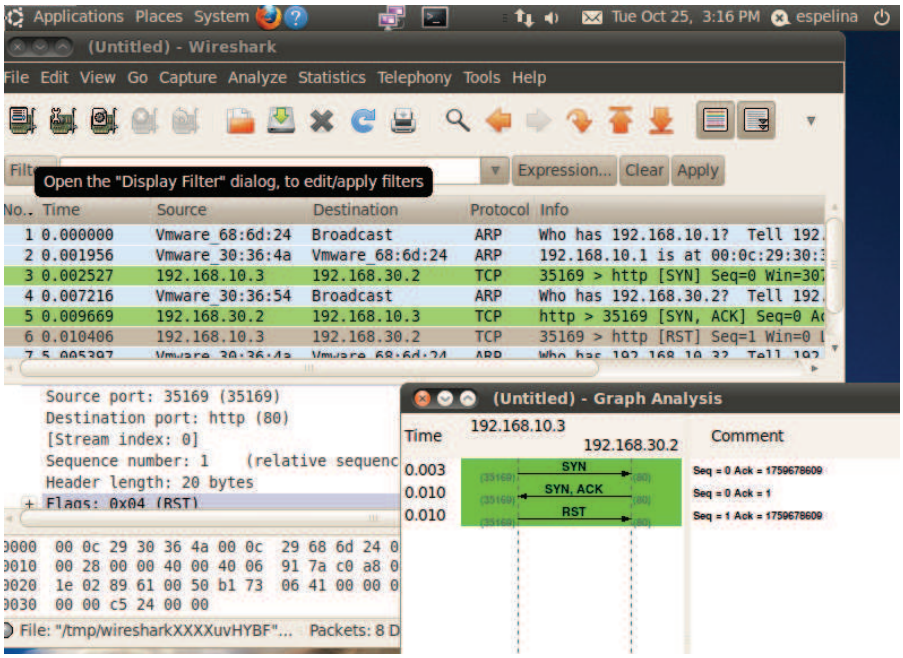


Figura 4.6. Captura de paquetes de un Escaneo TCP SYN.

Si se observa en los paquetes del sniffer (Ver figura 4.6), se tiene que el host atacante (192.168.10.3) envió un paquete SYN, el host destino (la IP 192.168.30.2) respondió con un SYN+ACK y por último el host emisor envía un paquete RST.

En las capturas Nro. 3, 5 y 6 se percibe que cuando los paquetes salen desde la PC atacante, el puerto origen es uno aleatorio, Nmap utilizan puertos dinámicos por encima de 35000 más o menos y cuando el host emisor recibe el SYN+ACK desde el host destino, lo hace desde ese mismo puerto dinámico que abrieron las aplicaciones[32.].

Adicionalmente se puede observar en el host destino no envía el paso 3 (ACK), del saludo en tres vías de TCP[49], sino que envía un RST/ACK para que no se establezca nunca una conexión completa.

Caso 3. Rastreos con banderas ACK

```
#Nmap  -sA 192.168.30.2 -p 21,22,80
```

Este sondeo es distinto a los otros que se han discutido en que no puede determinar puertos abiertos (o incluso abiertos/filtrados). Esta técnica es usada también para poder escanear hosts que estén detrás de un firewall que bloquee los intentos de conexión (paquetes SYN).

Con Nmap se puede invocar un escaneo ACK mediante el comando descrito en la figura 4.7 y 4.8:

[49] María Teresa Jimero, Carlos Migues, Abel Matas, Justo Pérez, , Edición 2009, La Biblia del Hacker

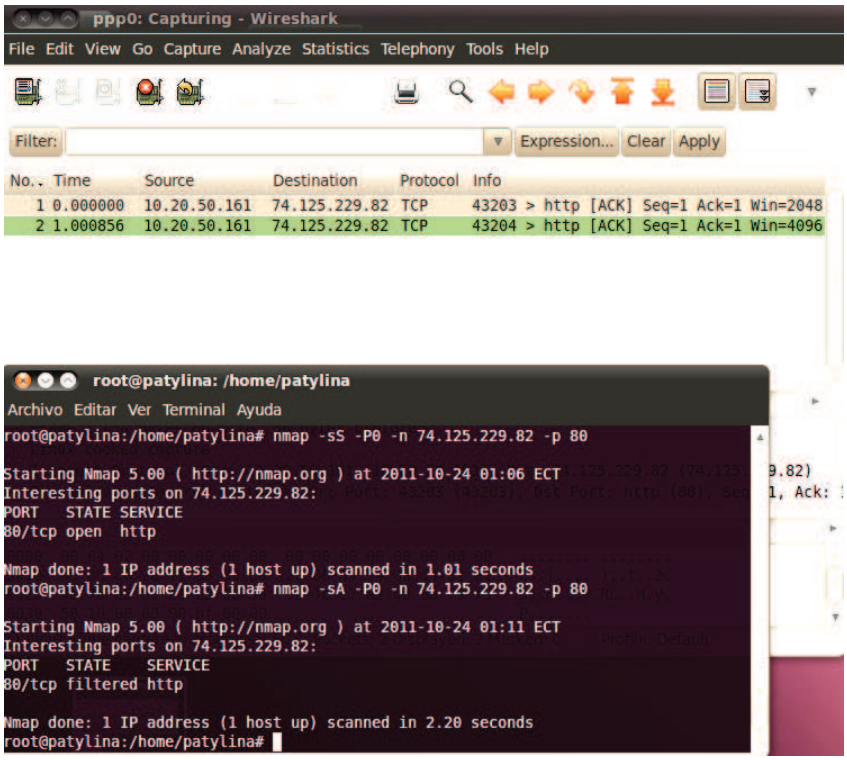


Figura 4.7 Escaneo de Puertos ACK SCAN (Externo).

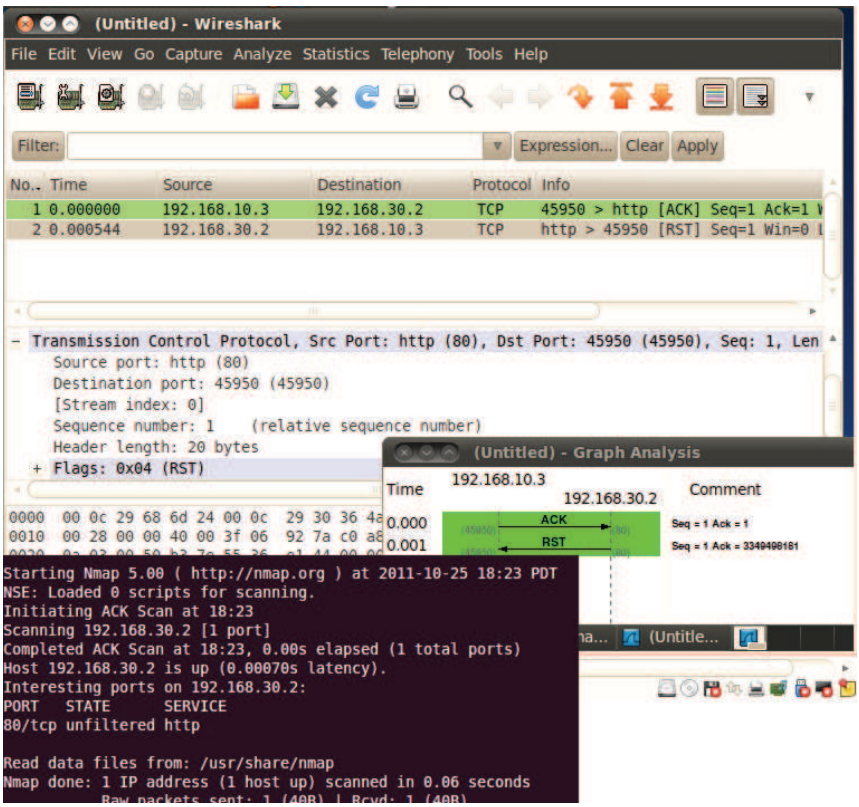


Figura 4.8 Escaneo de Puertos ACK SCAN (Interno).

Resultados Obtenidos

En la figura 4.7 se puede observar que Nmap marca el puerto como filtrado, lo que significa que es un puerto silencioso. Muy seguramente porque el servidor de Google se encuentra detrás de un firewall. En la misma figura se tiene la captura de paquetes, donde el escaneo ACK envía un número de secuencia y confirmación aleatorio. En las capturas nro. 1 y 2, se tiene el envío de paquetes hacia el host destino (192.168.30.2), pero no se obtiene respuesta alguna, identificando claramente el puerto 80 como filtrado (puerto silencioso).

Adicionalmente en la figura 4.8 se puede observar que Nmap marca el puerto como no filtrado, lo que significa que son alcanzables por el paquete ACK, pero no se puede determinar si están abiertos o cerrados. En la misma figura se tiene la captura de paquetes, desde Wireshak, en donde la captura nro. 1 es el envío del primer paquete ACK (desde 192.168.10.3 a 192.168.30.2). El nro. 2 se tiene la confirmación RST del host destino, lo que confirma el resultado obtenido por Nmap.

Caso 4. Rastreos con banderas TCP

```
#Nmap  -sT 192.168.30.2 -p 21,22,80
```

Nmap le pide al sistema operativo del host origen que establezcan una conexión con el sistema objetivo en el puerto indicado utilizando la llamada del sistema TCP connect(), cuyos resultados se observa en la figura 4.9.

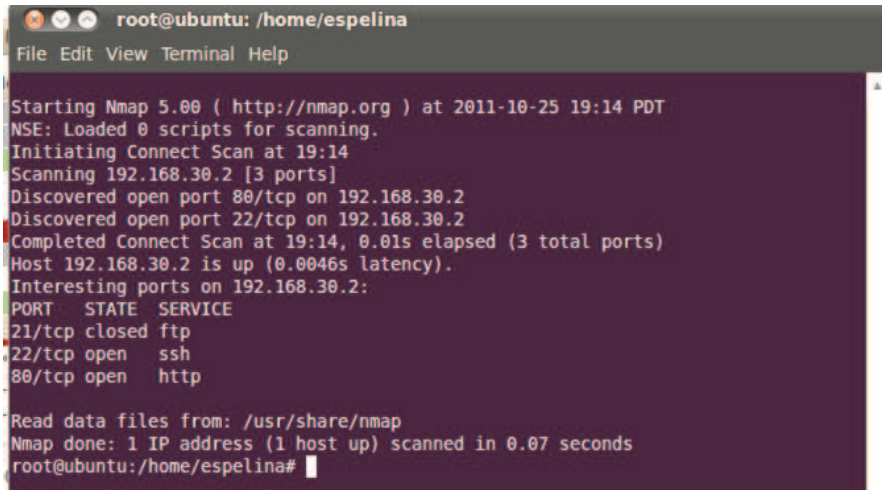


Figura 4.9 Escaneo de Puertos TCP connect().

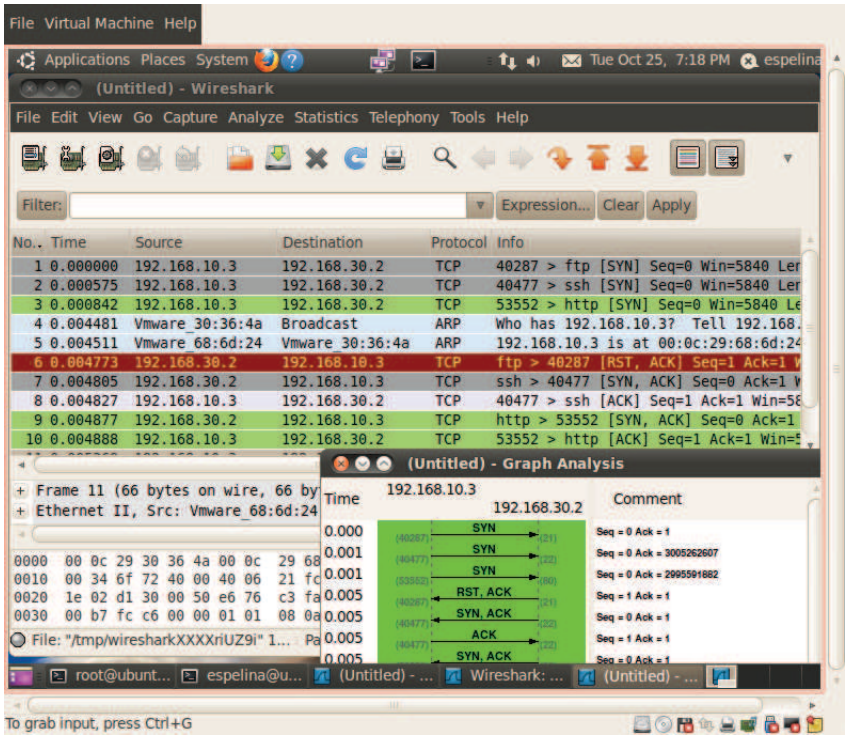


Figura 4.10. Captura de paquetes de un Escaneo TCP connect.

En la figura 4.10 se observa las llamadas connect de TCP para intentar establecer conexión con cada uno de los puertos del host a escanear, para este caso 21,22 y 80. La conexión se establece con los puertos 22 y 80, descrito en la captura Nro. 3 y 8, en donde se recibiendo un SYN, ACK por parte del host destino. En cambio la respuesta enviada por el puerto 80, un aviso de cierre de conexión RST, significa que el puerto está cerrado.

4.3.2. Ataque de Fuerza Bruta

En este tipo de ataques es importante disponer de lo siguiente:

- La máquina objetivo debe tener habilitado el servicio ssh y estar en ejecución
- Disponer del archivo de contraseñas o más conocido como diccionario de claves[25.].

4.3.2.1. Ejecución de ataque desde Ubuntu con Medusa

El ataque se inicia cuando la máquina atacante utilizando el programa Medusa y el diccionario de contraseñas, Ver figura 4.11, mediante la ejecución de siguiente comando:

```
#medusa -h 192.168.10.3 -u usuario -P passwords.txt  -M ssh
```

Dónde:

Medusa: Orden de ejecución de la herramienta[26.].

-h: Dirección IP objetivo (host)

-U: Usuario (en este caso root)

-P: Archivo contenedor de contraseñas.

-M: Módulo de ejecución de medusa, para este caso ssh (Secure Shell)

Resultados Obtenidos

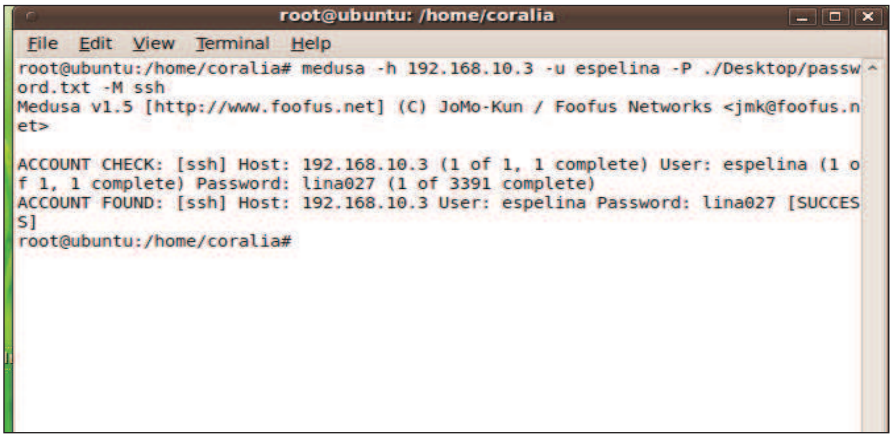


Figura 4.11 Ataque de Fuerza Bruta con Medusa.

En la figura 4.11 se puede ver el ataque al equipo 192.168.30.2, obviamente para la prueba se ha asegurado que el password se encuentra en el diccionario, el resultado fue asombroso tardó muy poco (8 segundos aproximadamente); hay que considerar que el tiempo depende de varios factores como el tamaño del diccionarios y la ubicación del password encontrado, de manera que tomará mucho menos si el password esta al inicio del diccionario de 3600 palabras.

Cabe mencionar que no es lo mismo hacer un ataque de fuerza bruta con diccionario de forma local que de forma remota, sin embargo al hacer la prueba de forma remota, los resultados fueron muy aceptables, se demoró 5 minutos aproximadamente en encontrar el password en un diccionario de 3600.

4.3.2.2. Ejecución del ataque desde Windows

Desde una ventada de comandos (DOS) se ingresa el siguiente comando: *john-386.exe -show pas2.txt* (Ver figura 4.12), para hacer uso de la herramienta PWDump, a fin de extraer los hashes de las contraseñas de Windows a auditar. Para este caso se auditará las contraseñas almacenas en un equipo local.

Una vez generado el archivo (pas2.txt) de contraseñas hasheadas se procede a auditar las contraseñas mediante un ataque de diccionario con el siguiente comando: *john-386.exe –wordlist -password.lst pas2.txt*

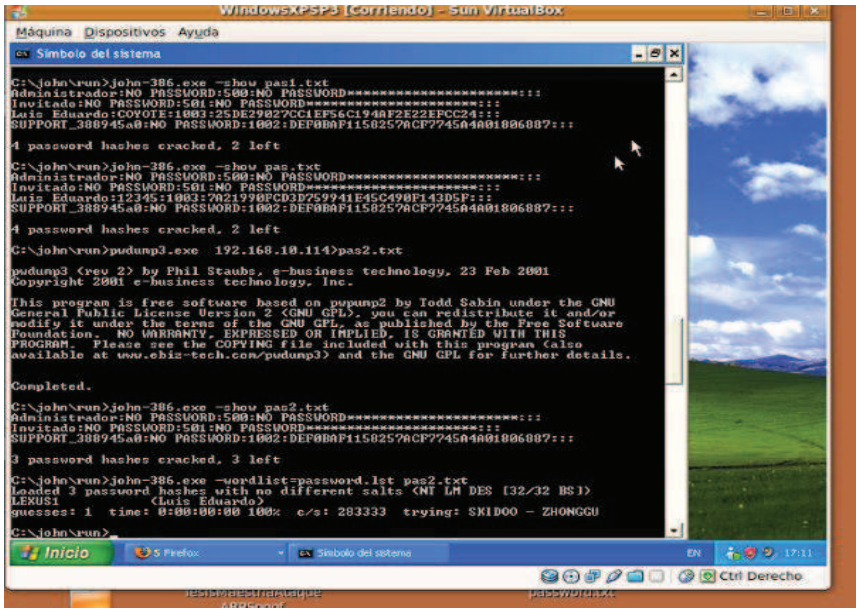


Figura 4.12 Ataque de Fuerza Bruta con John The Ripper (Windows).

Resultados Obtenidos

En la figura anterior también se observa los resultados obtenidos del ataque de fuerza bruta con John The Ripper, resultado que mucho depende de la personalización del archivo diccionario que se tenga. Para este caso las contraseñas desencriptadas, mostradas a pantalla, son el nombre de usuario, contraseña y el ID del usuario.

4.3.2.3. Ejecución del ataque desde Ubuntu con John The Ripper

Otra funcionalidad interesante de John The Ripper, es el poder crear sesiones de trabajo, entendiéndose por sesión el proceso de descifrar las contraseñas, de tal manera que sea posible iniciar, detener una sesión (a fin de continuarla en cualquier otro momento) y finalizar dicha sesión.

Antes de comenzar el ataque, se debe crear el archivo de contraseñas (pass.txt), es importante considerar que el archivo donde se encuentran las contraseñas del sistema es /etc/passwd, en el caso de utilizar shadow, estarán en el archivo /etc/shadow. El comando para la obtención del archivo de contraseñas es:

```
# unshadown /etc/password /etc/shadow> pass.txt
```

Con este comando el archivo es creado con las contraseñas cifradas del archivo /etc/shadow pero con la estructura del archivo /etc/password. Seguidamente se procede a descifrar las contraseñas utilizando el siguiente comando:

```
#./john pass.txt
```

Con este simple comando, el programa John The Ripper comienza a trabajar e irá mostrando automáticamente las contraseñas que va descifrando, como se observa en la figura 4.13. Según la calidad de las contraseñas cifradas en el archivo, John The Ripper puede llegar a tardas varios días, semanas o incluso meses encontrar las mismas. Ante esto, puede ser necesario tener que cortar la sesión antes, con la combinación de las teclas Ctr+c. Y para reanudar la búsqueda se ejecuta el siguiente comando:

#./John –restore

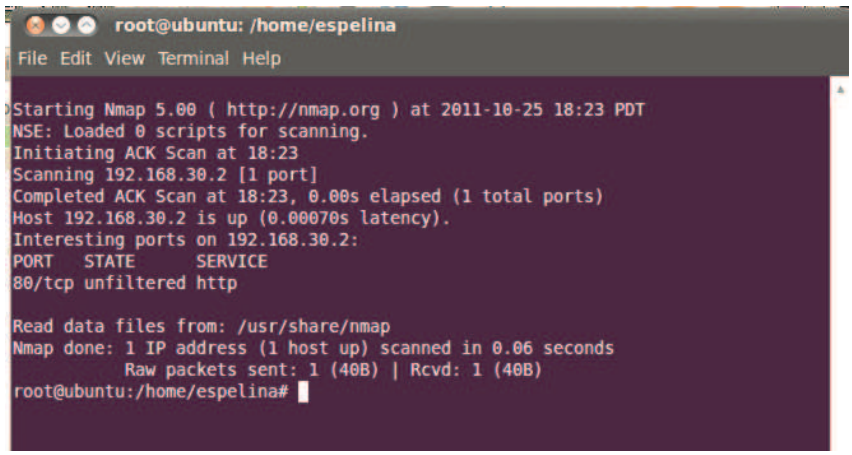


Figura 4.13 Ataque de Fuerza Bruta con John The Ripper (Ubuntu).

4.3.3. Suplantación de Identidad ARPSpoofing (Man In The Middle)

El Ataque *Man In The Middle* (MIM, Hombre en el Medio)[27.], consiste en interceptar o atravesar entre la conexión que establece la víctima con un router, servidor, etc. usurpando la identidad del router y haciendo que los paquetes que envíe la víctima al router, pasen primero por el host atacante[50].

Para entenderlo más fácilmente, se tiene las imágenes de la Figura 4.14-a. donde en una conexión normal entre un ordenador y su router G1, permite el tráfico entre H y T, como se muestra en la figura (líneas de celeste).

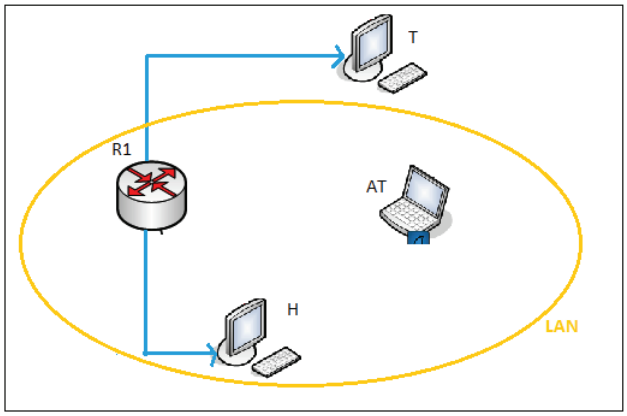


Figura 4.14-a. Conexión normal entre dos hosts

[50] Envenenamiento Arp, Seguridad en redes conmutadas, http://ownz.despai.es/trabajo_arp.pdf

En una conexión con un *Hombre en el Medio* sería así: AT (host atacante), envía un mensaje a H, para ello utiliza como IP origen del envío la IP de G1 (router). Logrando así que el tráfico pase por AT en lugar de seguir el camino inicial (línea roja de la izquierda), ver Figura 4.14-b.

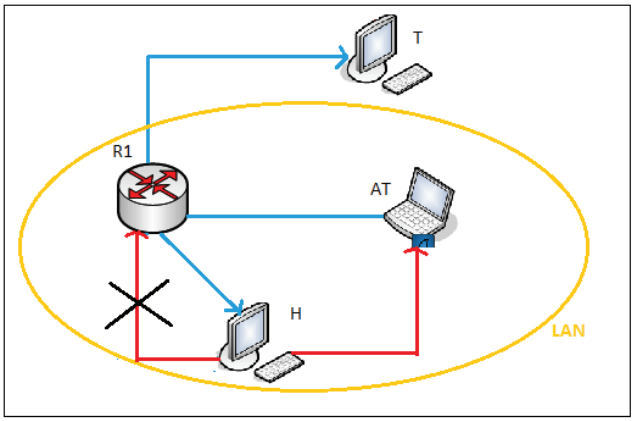


Figura 4.14-b Conexión con MITM (*Man In The Middle*) entre dos hosts

Ejecución del ataque

Para interponerse entre el host H(192.168.10.10) y el Gateway de nuestra LAN G1(192.168.10.1) bastará con ejecutar Ettercap de la siguiente forma: *root@maqvirtual03:~# ettercap –C.*

Seguidamente se indica a Ettercap que busque todos ordenadores conectados en este instante en la red, para ello se selecciona el menú *Hosts->Scan for host*, observando una pantalla temporal que tras unos segundos desaparece.

Una vez que ha terminado de buscar los ordenadores conectados a la red se procede a elegir entre que par de equipos se desea conectar, en este caso entre el equipo de la víctima (192.168.10.10) y el Router (192.168.10.1), para ello se selecciona la opción *Hosts->Host List* y se elige el primer equipo y se presiona la tecla 1, después se selecciona el segundo equipo (en este caso un Router) y se presiona la tecla 2 (ver Figura 4.15).

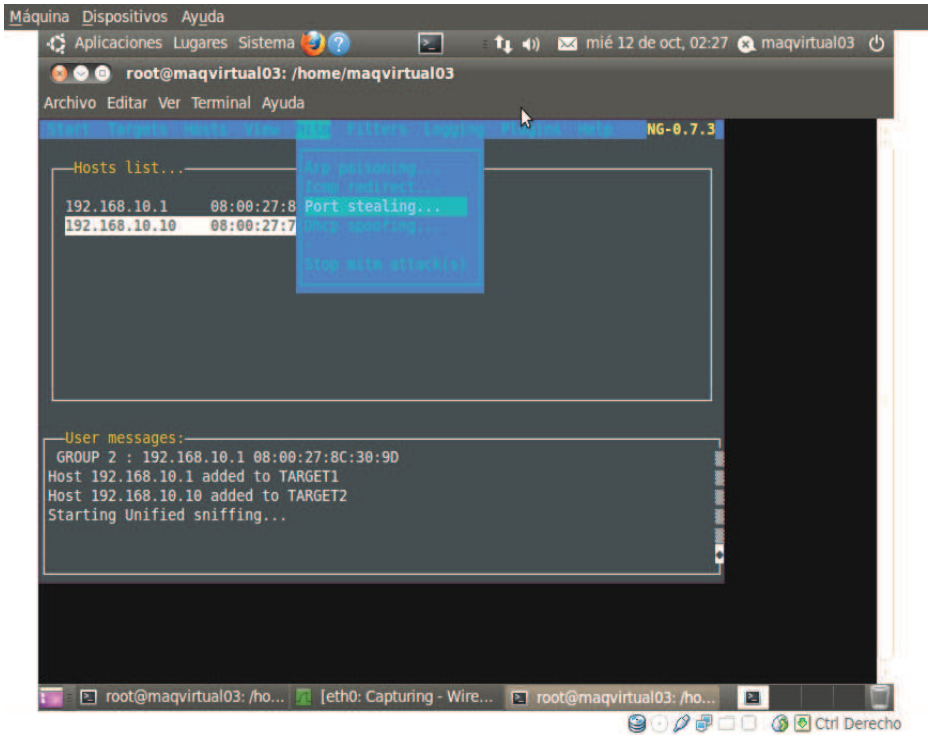


Figura 4.15 Activación de los dos equipos que se desea conectar, con Ettercap.

Finalmente se procede a decir a Ettercap que haga un ataque Man In The Middle por ARP Poisoning, para ello se selecciona el menú *MITM* opción *ARP Poisoning* y en parameters se debe escribir *remote* <*enter*>, desde ese momento el host atacante (192.168.10.20) se coloca entre el equipo víctima y el Router, procediendo al escaneo respectivo.

Resultados obtenidos con el sniffer Wireshark

En la Figura 4.16, se puede observar rápidamente la gran cantidad de tráfico ARP que se está recibiendo. Si se observa más detalladamente el comportamiento del protocolo, se puede dar cuenta de que el host de servicios (192.168.10.10) está siendo víctima de un ataque.

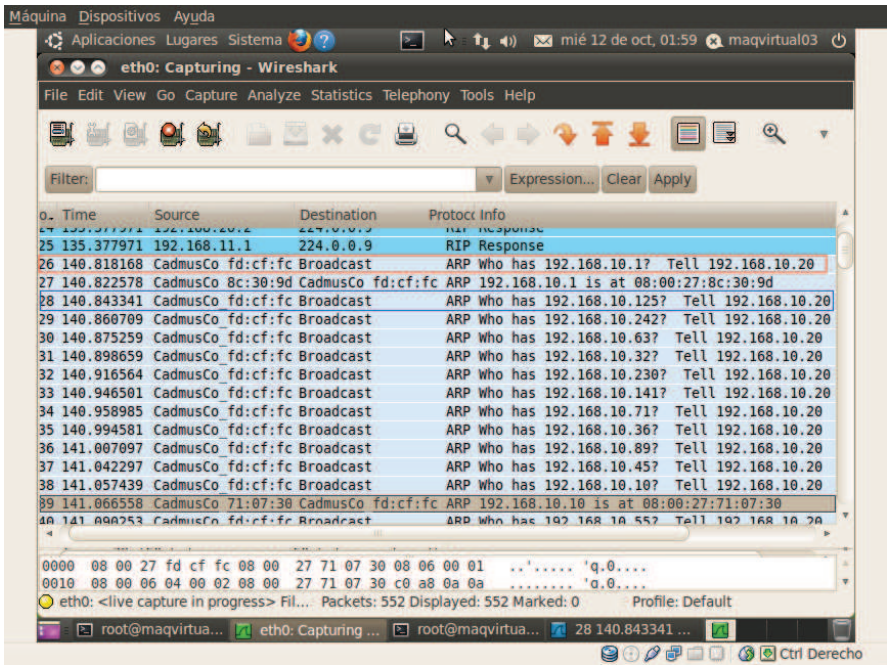


Figura 4.16 Tráfico generado ante un ataque MIM

En el paquete número 26 se observa cómo la máquina con IP 192.168.10.20, con una MAC ***CadmusCo_fd:cf:fc***, ha lanzado un ARP request a la dirección broadcast preguntando por la MAC de la IP 192.168.10.1 (el gateway de nuestra red). Seguidamente, el router contesta con un ARP reply indicando cuál es su dirección MAC. A continuación, la misma IP repite el proceso, con algunas IP sin lograr comunicación alguna, hasta que llega a pregunta por la MAC de la IP 192.168.10.10 mediante otra difusión broadcast. El host de servicios contesta con su dirección MAC ***(08:00:27:71:07:30)***. Se tiene como resultado una máquina de la red LAN (192.168.10.20), que ya tiene la MAC del servidor de servicios y la del router. El poder del ataque como tal, viene a partir del paquete 28, donde la máquina atacante envía reiteradamente al servidor de servicios y al router paquetes ARP reply falsos, asociando la IP de ambos con su propia MAC (08:00:27:fd:cf:fc). De esta forma, todo el tráfico que transite entre el router de la LAN y el host de servicios pasará a través de la máquina atacante.

A continuación, se muestra el detalle del formato en bruto de una respuesta ARP generada por el equipo atacante a un ARP-request (Figura 4.17). Se puede buscar estos paquetes con el filtro ***arp.opcode == 0x2*** dentro de Wireshark.

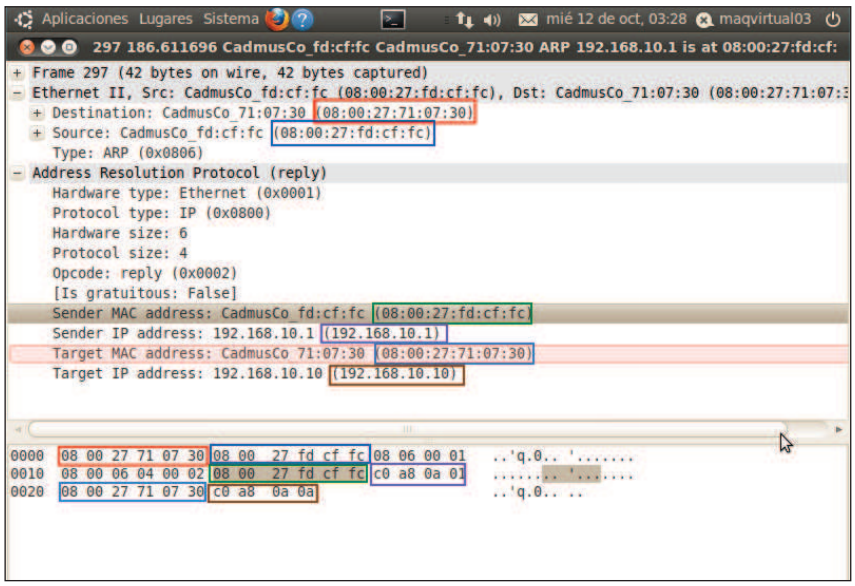


Figura 4.17 Detalle de las tramas generada en un ataque MIM

El texto hexadecimal mostrado en la zona inferior, de la figura anterior, se corresponde con la trama tal y como se trasmite por la red.

4.3.4. Denegación de servicios (DoS*)*

El Ataque que se presenta a continuación es un DoS en una Red Local (LAN), a través de un inyector de paquetes ARP automático. Para lograr envenenar las Tablas ARP del equipo víctima a través de ARP Spoofing (Némesis), en el paquete que se inyecta arbitrariamente, se cambia la dirección MAC del Router por una inexistente, por lo que la máquina víctima fracasará en el intento de encontrarla, con esto se consigue que el ordenador víctima quede sin Internet o sin acceso al servicio (Web, correo electrónico, etc.) por un lapso de tiempo controlable[28.].

Para lograr lo descrito anteriormente, ya sea desde un equipo atacante con sistema operativo Windows o Ubuntu, se debe realizar lo siguiente:

4.3.4.1. Ataque desde una máquina Windows

Desde una consola DOS, se procede a ingresar al directorio donde se encuentra Némesis[33.] (C:\cd Némesis), una vez ahí se comienza con el ataque con los siguientes comandos (figura 4.18):

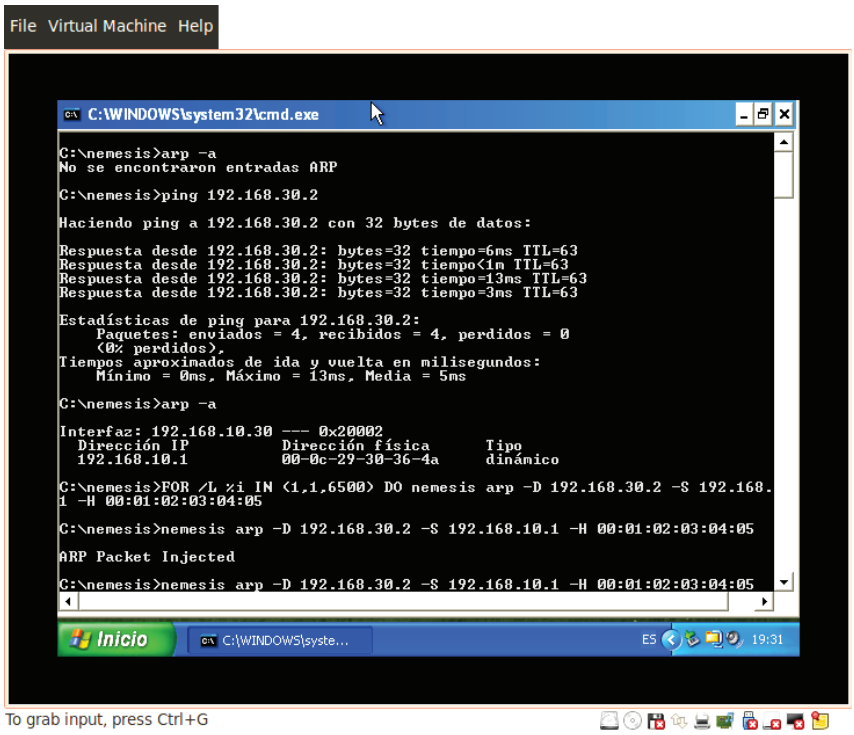


Figura 4.18 Ataque de DoS con Némesis desde Windows

En la figura anterior se observa primeramente el comando que permite conocer la dirección MAC del Router con el siguiente comando:

```
c:\>arp –a
```

Seguidamente se requiere saber la dirección MAC de la víctima, para ello se hace lo siguiente:

```
c:\>nemesis >ping 192.168.30.2
```

El ping permite que se actualice la tabla ARP del equipo atacante con la dirección IP y MAC de la víctima

Como ya se conoce las direcciones necesarias, se procede realizar el ataque escribiendo el siguiente comando desde el directorio del Inyector ARP, Némesis:

```
c:\Némesis\>FOR /L %i IN (1,1,6500) DO nemesis arp -D 192.168.30.2 -S
192.168.10.1 –H 00:01:02:03:04:05
```

Resultados Obtenidos

En la figura 4.19-a, se puede ver que en tan solo unos segundos, se ha inundado la red con casi cientos de mil paquetes transmitidos de forma ininterrumpida. Al realizar una inyección de ARP constante, en donde la dirección MAC del router es una inexistente, provoca que el acceso hacia el equipo víctima tarde en responder por las múltiples peticiones a ella, que se ejecutan con este proceso, lo que provoca que deniegue el servicio de correo electrónico para este caso (ver Figura 4.19-b).

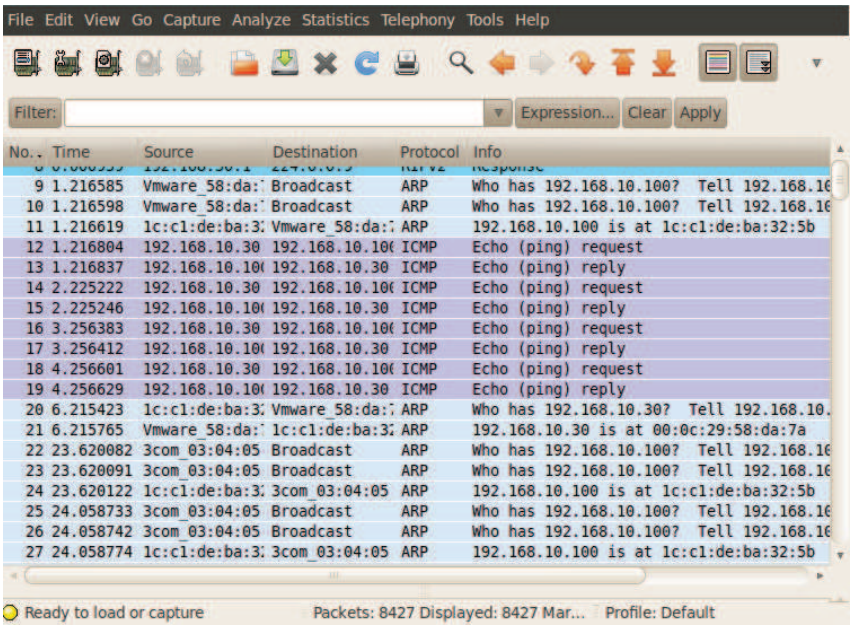


Figura 4.19-a Inyector de paquetes ARP automático **(** DoS) desde Windows

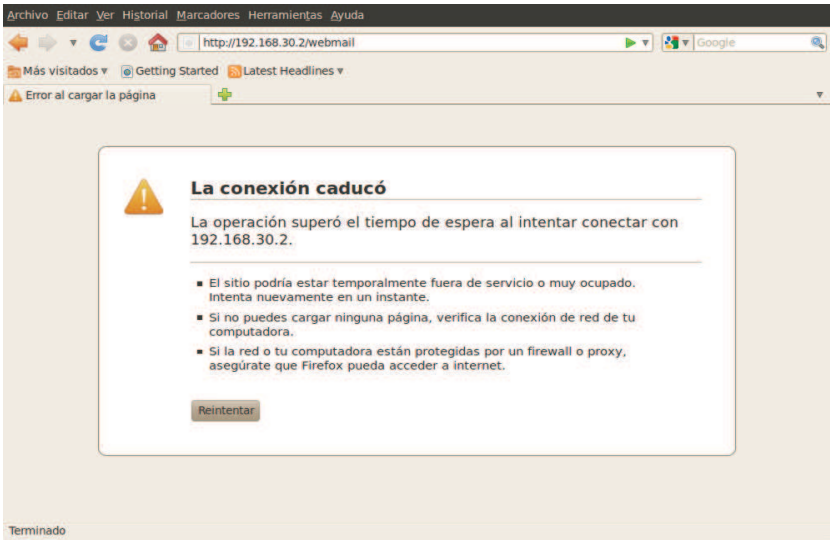


Figura 4.19-b Denegación del Servicio de Correo Electrónico ante un ataque **(** DoS)

4.3.4.2. Ataque desde una máquina Ubuntu

Un ejemplo de ataque de denegación de servicio (DoS), llevado a cabo por hping3, donde la víctima (con IP 192.168.30.2) es un servidor Apache y de Correo instalado, es el siguiente comando:

```
#hping3  -a –p80 192.168.30.2 192.168.30.200
```

Resultados Obtenidos

Claramente se puede observar, en la figura 4.20, gran cantidad de segmentos TCP con el flag SYN activados desde la misma IP, que no reciben respuesta alguna por parte del servidor web. El servidor trata de resolver la MAC de la máquina cliente en numerosas ocasiones, una de ellas se puede ver en el paquete 6, pero al no recibir respuesta alguna y al carecer de la dirección física del host, no puede enviar un ACK-SYN al mismo para continuar con el establecimiento de la conexión a tres pasos. Esto conlleva que el equipo atacante tenga que esperar por cada conexión un tiempo determinado, tiempo en el cual seguirán llegando más paquetes que irán creando nuevas conexiones. Conexiones, que al ser un número muy elevado, pueden acabar con los recursos de la máquina produciendo que el equipo deje de contestar más solicitudes de conexión.

Además se puede observar de forma gráfica, la secuencia de paquetes, a través de la herramienta Wireshark, activando la opción Statics -> Flow Graph. Como se ve en la imagen, el comportamiento de conexiones TCP se describe mediante flechas el origen y destino de cada paquete, resaltando los flag activos que intervienen en cada sentido de la conexión.

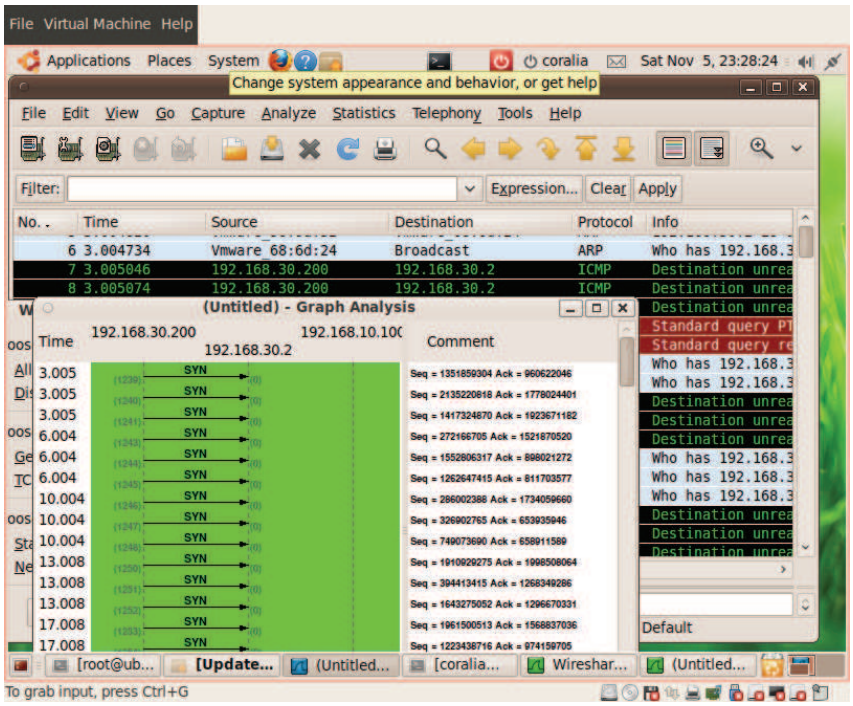


Figura 4.20 Ataque de denegación de Servicio **(** DoS) desde Ubuntu

4.3.5. Ataques a la WEB

Escaneo básico.

Para ejecutar un escáner básico con Nikto basta con ejecutar la sentencia:

./nikto.pl -h xxx.xxx.xxx.xxx o http://www.dominio.xx

La opción -h nos indica el host que queremos escanear pudiendo ser este una dirección IP o un nombre de dominio.

Si no se indica lo contrario, el puerto por defecto que utiliza Nikto es el 80, si se desea cambiarlo se debe indicaren la sentencia con la clave -p.

A continuación se muestra un ejemplo de rastreo de un máquina Ver figura 4.21, Nikto detecta que es un servidor web de tipo Apache/2.2.14 y a continuación muestra los posibles problemas de seguridad que contiene el sistema web escaneado.

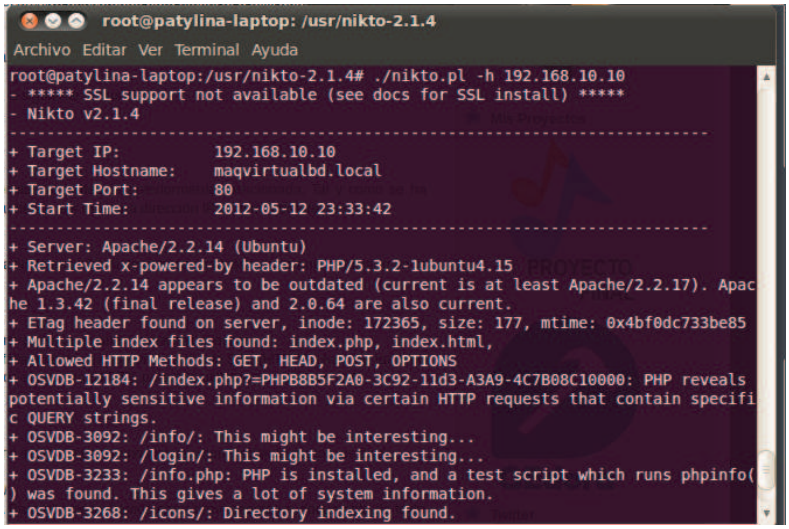


Figura 4.21. Escaneo básico de la WEB

Ataque de Fuerza Bruta en la web

Pasos a realizarse para este tipo de ataque:

Elegir una víctima, en este caso será: 192.168.10.10/verificar_usuario.php

Lo que se hará es crakear el password de esa web mediante http. Para ello se debe realizar lo siguiente:

- Primero introducir la dirección de la web en el target.
- Después seleccionamos el tipo de servidor.
- El puerto que corresponda, en conexiones ponemos “3” (se puede probar a poner más conexiones, ya que puede desconectarse)
- En *authentification options* marcamos la casilla de *Use username* y la casilla de *single username* y se pone el nombre del user y se selecciona un diccionario.
- Seguidamente se da *Start* y el programa empezara a *crakear*.

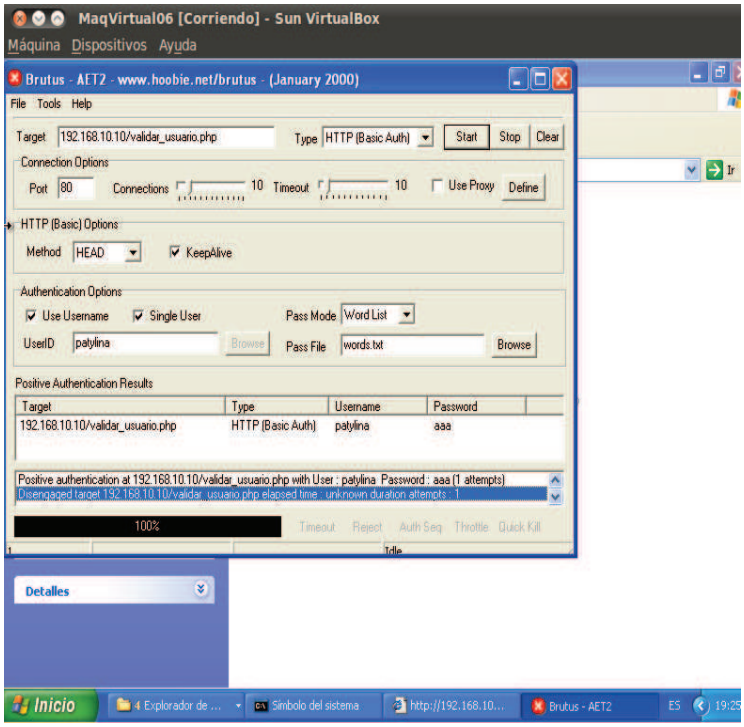


Figura 4.22. Ataque de Fuerza Bruta a la WEB

Una vez que el crakeo se termina y se ha obtenido como resultado "aaa", por lo general este proceso es bastante lento, debido a que el servidor es bastante lento, Figura 4.22. Seguidamente comprobamos el resultado obtenido y para ello se debe dirigir a la aplicación Web e introducir el usuario "patylina" y el password "aaa", logrando tener acceso al sitio Web.

Ataques XSS

Cabe mencionar que algunas de las demostraciones sobre este tipo de ataque fueron desarrolladas utilizando la aplicación DVWA (Damn Vulnerable Web App) de libre distribución.

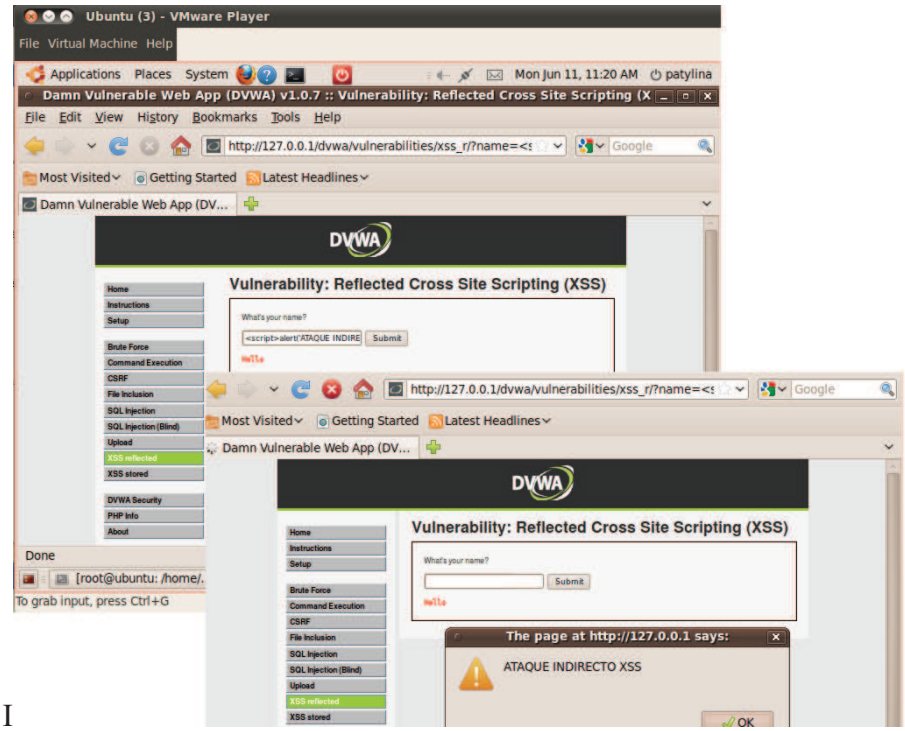
DVWA es una aplicación de entrenamiento en seguridad Web que se destaca por contener muchas aplicaciones Webs vulnerables a diferentes tipos de técnicas. Y está desarrollada con la finalidad de ofrecer a los profesionales, estudiantes e investigadores en seguridad informática una aplicación de entrenamiento, la cual permita poner a prueba sus conocimientos y herramientas enfocadas a la seguridad Web[51].

[51] DVWA – Damn Vulnerable Web App, http://www.dragonjar.org/dvwa-damn-vulnerable-web-app.xhtml

Algunas de las técnicas que el aplicativo DVWA (entrenamiento en Seguridad Web) permite llevar a cabo, se describe a continuación:

- ***XSS (Cross Site Scripting)***

  El ataque XSS, del inglés Cross-site scripting es un tipo de inseguridad informática o agujero de seguridad basado en la explotación de vulnerabilidades del sistema de validación de HTML incrustado.



I

Figura 4.23 Ataque XSS (Cross Site Scripting)

En la figura 4.23 se observa la digitación de una secuencia de comandos XSS escrito en lenguaje JavaScript. Este script genera una ventana de alerta con el único objetivo de causar molestia al usuario y comprobar así la vulnerabilidad de la página web.

- **Ataque CSRT**

El CSRF (Cross-site request forgery o falsificación de petición en sitios cruzados) es un tipo de exploit malicioso de un sitio web en el que comandos no autorizados

son transmitidos por un usuario en el cual el sitio web confía. Esta vulnerabilidad es conocida también por otros nombres como XSRF, enlace hostil y ataque automático.

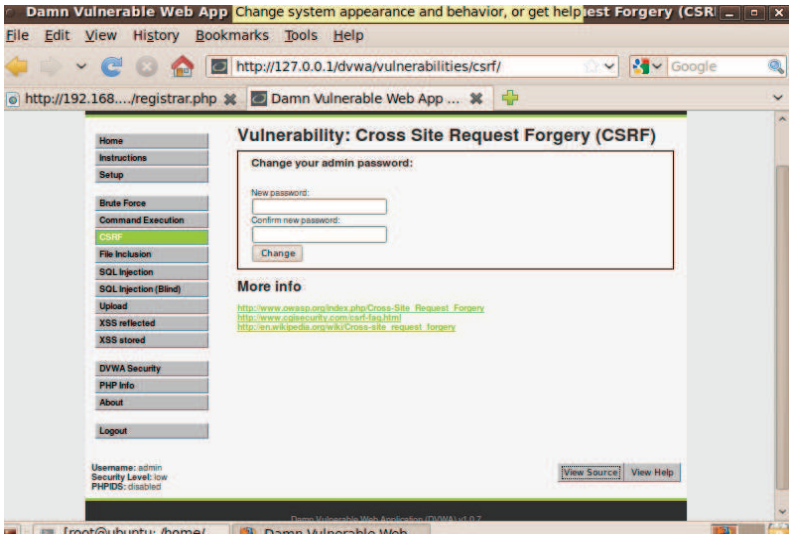


Figura 4.24. Sitio vulnerable que permite el cambio de clave

Figura 4.25 Formulario falsificado que roba la identidad de un usuario

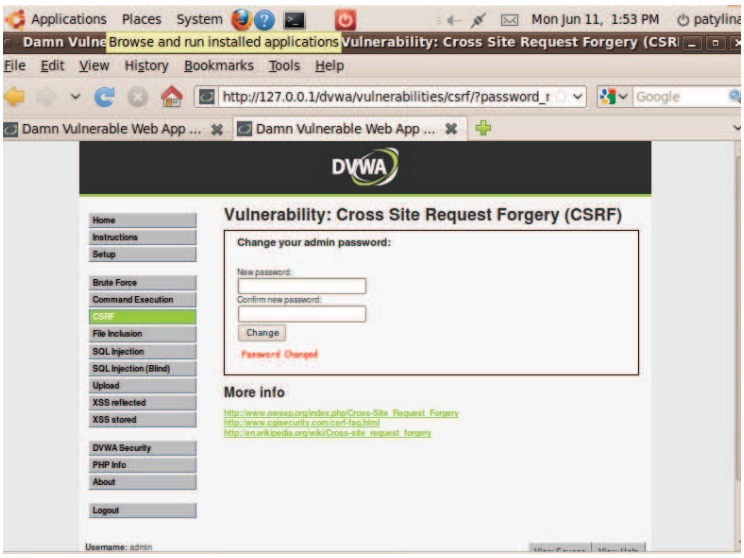


Figura 4.26 Formulario que confirma el cambio de clave al atacante

En las figuras 4.24, 4.25 y 4.26, se observa la ejecución de un ataque XSRT. El objetivo de este scripts es modificar la contraseña del usuario activo en dvwa, creando para ello un formulario falso, que permite al atacante robar la identidad del usuario, para este caso, para poder cambiar la contraseña, sin que este pueda llegar a sospecharlo. Este tipo de ataque funcionan de la siguiente manera: el usuario atacante a través del formulario falso, que contendrá las mismas variables de entrada del formulario válido y la dirección del sitio web a donde se enviará la información ingresada por el usuario relacionada con la nueva contraseña del usuario admin. El resultado obtenido se observa en la figura 4.26, donde se confirma el cambio de password.

4.3.6. Ataque a una Base de Datos

A continuación se tiene ejemplos de la técnica SQL Inyection, donde se tiene una página web la cual es aplicada a un login de autentificación o acceso para poder ingresar a la parte privada de ésta. Dentro del archivo jsp que permite el acceso a clientes que se verá a continuación, se encontrará la sentencia SQL o rutina de validación para el acceso a la web. Esta acción proveniente de la orden o sentencia verificará que usuario y password sean los correctos para dejar acceder al visitante en caso de que introduzca estos datos correctamente.

El código, con la transaccion SQL (a la database) se verá más o menos así:

SELECT usuario, passworsd FROM usuarios WHERE usuario = ´$usuario'
AND password= '$clave'

El proceso para que el intruso logre inyectar una sentencia SQL en lugar de colocar usuario, password, es muy simple. Si los campos de datos no están protegidos contra los caracteres especiales y asegurados, se podrá incluir una comilla simple ' y seguido a ella, el resto de lo que será interpretado por el gestor de base de datos como código SQL. La comilla simple es interpretada como terminador de carácter, cuya función al incluirle dentro de una sentencia SQL es hacer que servidor genere un error, ello dará como indicios de que se puede realizar posibles ataques SQLInyection.

Ataque 1

Introduciendo en los campos de usuario y passwod lo siguiente:

usuario=is'ma

passwod=pass

Como resultado se tiene la generación del siguiente error, Ver figura 4.27.

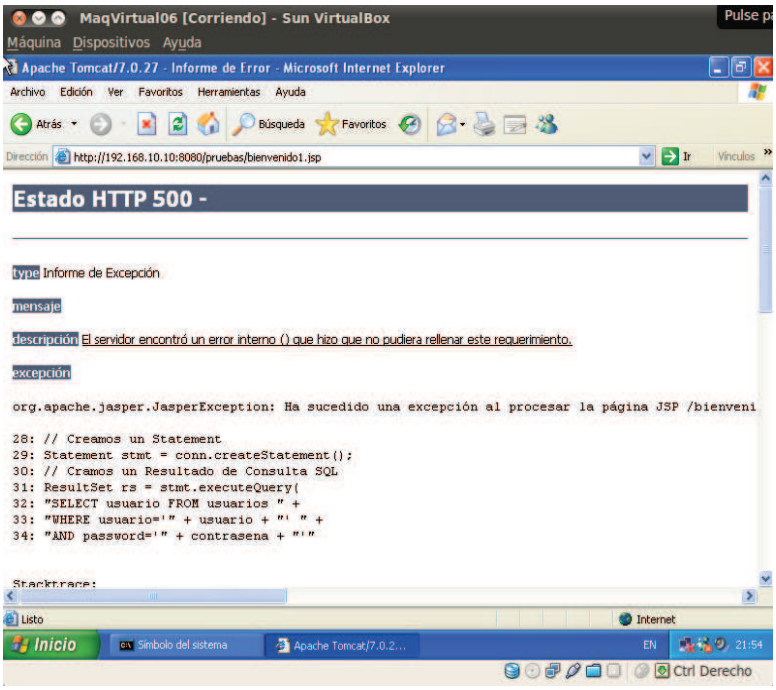


Figura 4.27 Error generado ante un ataque SQL Inyection

Lo que demuestra que la aplicación es vulnerable a un ataque SQL Inyection.

Además se ha podido averiguar el nombre de la tabla y los dos campos usuario y password.

Ataque 2

Pretender llegar a entrar con el primer usuario existente en la aplicación digitando en el campo usuario: ' or 1=1 --. Logrando ingresar a la página privada Ver figura 4.28.

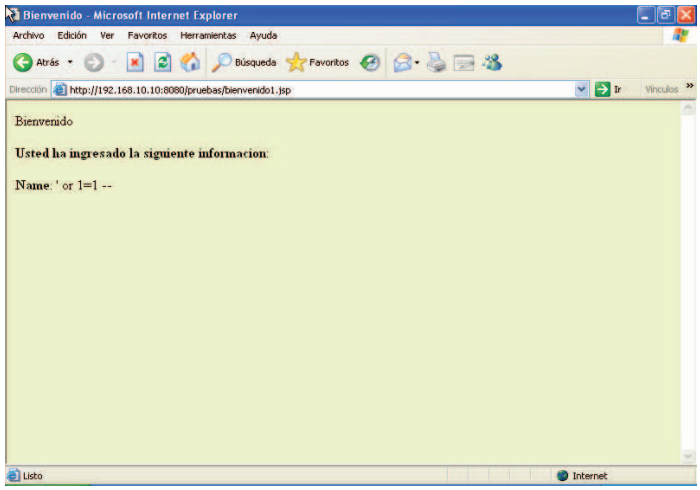


Figura 4.28 Ingreso a la página Web Privada con la sentencia *' or 1=1 –*

Ataque 3

Para el caso en que se conozca el nombre de algún usuario se podría intentar acceder colocando en el campo password, en lugar de la clave del usuario, el comando *'OR 1=1 --,* quedando la sentencia sql de la siguiente manera:

SELECT id FROM login WHERE usuario = 'francisco' AND clave = 'OR 1= 1 --

Con este tipo de sentencias se cambia el comportamiento de la aplicación. De este modo, si el usuario es francisco y tiene un password/condición, si 1 es igual a 1 (que por supuesto lo es), éste será validado y tendrá acceso al sitio web privado (como se ve en la siguiente figura), ver figura 4.29

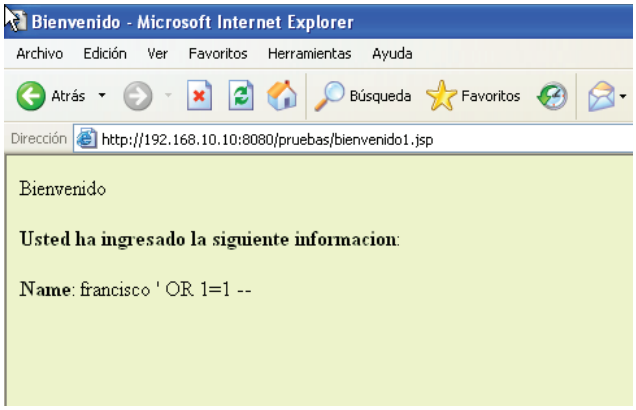


Figura 4.29 Ingreso a la página Privada conociendo nombre de usuario.

4.3.7. Ataques a Correos Electrónicos

a) Envío masivo de correo (Mail Bombing)

Para la ejecución de este tipo de ataque fue necesario programa un script, el mismo que permite que se envíe n veces un mensaje a un destinatario, Ver Figura 4.30.

```
#!/bin/bash
set nombre="emailmasivo"
if test -r $nombre
then $nombre  | nc -vv -w 3 192.168.30.4 25
else
touch $nombre
echo Nombre del remitente?
read "email"
echo Nombre deln destinanatario?
read "enviar"
echo Ingresa el asunto?
read "asunto"
echo cual es el mensaje?
read "mensaje"
echo Cuantas veces quieres enviar el email?
read "veces"
echo "Helo dastic.com.br"  >> $nombre
echo "MAIL FROM: $email"  >> $nombre
echo "RCPT TO: $enviar" >> $nombre
echo "DATA"  >> $nombre
echo "From: $email" >> $nombre
echo "To: $enviar" >> $nombre
echo "Subject: $asunto" >> $nombre
echo "$mensaje" >> $nombre
echo "." >> $nombre

echo "." >> $nombre
for cont in $(seq "$veces")
do
cat $nombre | nc -vv -w 3 192.168.30.4 25
done
fi
echo "######## mensaje enviado   ######"
```

Figura 4.30 Envío masivo de correos (Mail Bombing)

Este script solicita al usuario que ingrese los datos referentes al email (dirección origen, dirección destino, texto del email, mensaje), además debe

ingresar el número de veces que enviará el correo a su destino. Para poder logra todo esto fue necesario crear un archivo temporal con los datos necesarios ingresados sobre el correo a enviarse, y con el comando nc –vv –w 3 192.168.30.4.25 se logra el envío del de correo a la dirección de destino especificada a través del puerto 25.

Resultados obtenidos

En la figura 4.31 se observa en la bandeja de entrada del usuario destinatario, utilizado en la prueba del envio masivo de correos, el ingreso de correos seguidos del remitente usuario@gmail considerado en la prueba efectuada sobre el envio masivo de correos a través del script desarrolado.

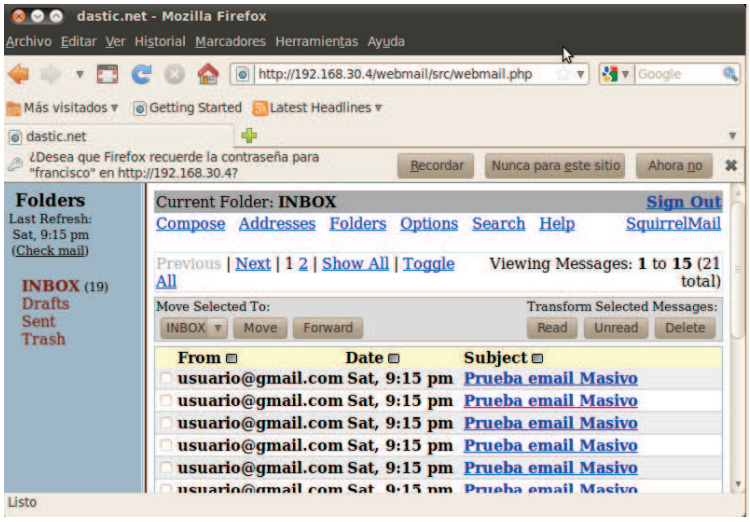


Figura 4.31 Verificación del mensaje masivo que ingresa al destinatario del correo masivo enviado.

Correo Anónimo

El envío de correo anónimo igualmente se lo hace a través del comando telnet, junto al puerto de conexión, que en vez del número 23 (puerto por defecto del puerto telnet) es el del servicio SMTP, ver figura 4.32.

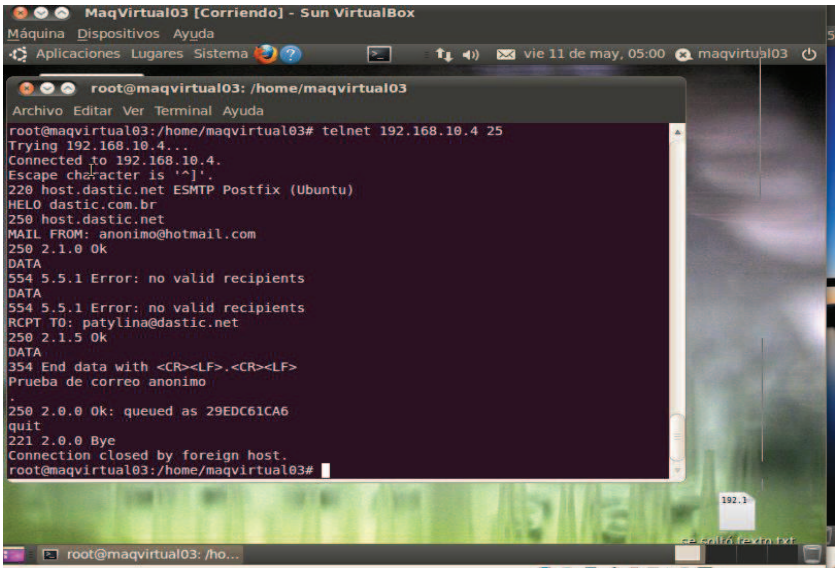


Figura 4.32 Envío manual de mensajes anónimos través de telnet.

En esta prueba se demuestra que la conexión por telnet al puerto 25 es posible. Una vez establecida la conexión telnet, todo lo que se digite se consideran órdenes para Postfix; en este ejemplo se le ha ordenado a la instrucción HELO dirigirse al dominio del servidor de correos, seguidamente se procede a indicar la fuente del correo y el destino del mismo, junto al DATA o texto del mensaje a enviarse, siendo para este caso una fuente anónima y un destino real, el resultado se puede observar en la figura 4.33.

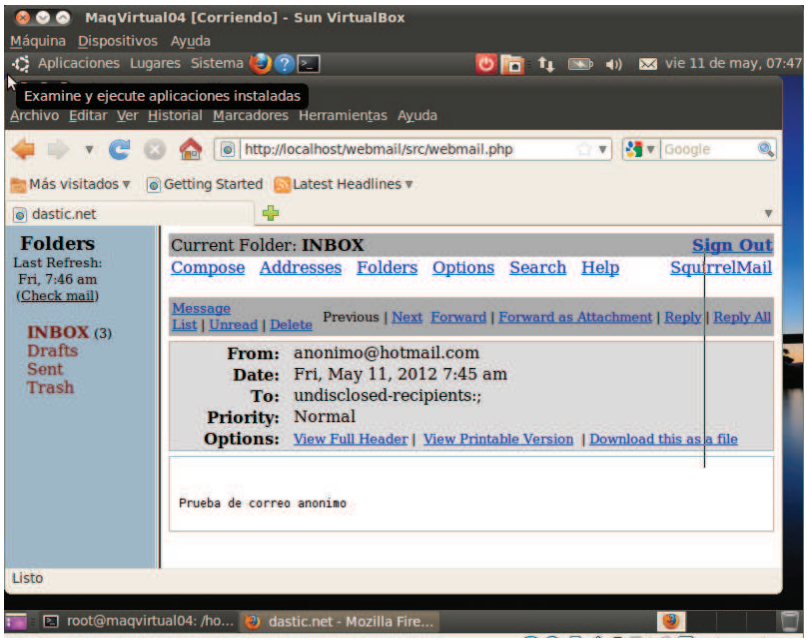


Figura 4.33 Verificación del mensaje **anónimo** enviado en forma manual a través de telnet.

Como se observa en la figura anterior el correo anónimo (anónimo@hotmail.com) llegó a su destinario se puede observar que el correo anónimo llego al destinatario patylina@dastic.net sin inconvenientes.

4.4. **Mecanismos para Contrarrestar Ataques a Redes IP**.

Una vez conocidas las vulnerabilidades y ataques a las que está expuesto un sistema es necesario conocer los recursos disponibles para protegerlo. Mientras algunas técnicas son evidentes (seguridad física por ejemplo) otras pautas no lo son tanto e incluso algunas pueden ocasionar una sensación de falsa seguridad.

Cabe mencionar que muchas de las vulnerabilidades estudiadas son el resultado de implementación incorrecta de tecnologías, otras son consecuencias de la falta de planeamiento de las mismas pero la mayoría de los agujeros de seguridad son ocasionados por los usuarios de dichos sistemas y es responsabilidad del administrador detectarlos y mitigarlos de la mejor manera.

Una vez revisadas los tipos de ataques, considerados para efecto de estudio y análisis en el presente trabajo, se ha determinado los mecanismos más apropiados para prevenir y mitigar dichos ataques, los que se describen a continuación.

4.4.1. Demonio o Administrador Regular de Procesos

Es un programa en segundo plano que ejecuta comandos programados en Shell Scripts[52] y tiene como objetivo programar cada cierto tiempo, a través de la configuración de crontab[53], la ejecución de los scripts que mitigue tanto un ataque por fuerza bruta como el de suplantación de identidad o de denegación de servicios.

Script para contrarrestar un ataque de fuerza bruta

El sistema de logs (registro) de Linux, es un mecanismo estándar que se encarga de recoger los mensajes generados por los programas, aplicaciones y demonios[54].

El Bash Script propuesto como contramedida ante intentos de conexión no solicitadas (ataque e fuerza bruta) monitoriza el fichero auth.log, cada cierto tiempo (dos segundos),

[52]Programación De Shell Scripts En Linux, http://juanin.bligoo.com/media/users/0/44513/files/3080/preliminares_shell.pdf
[53] Crontab, http://usemoslinux.blogspot.com/2010/11/cron-crontab-explicados.html
[54] Logs en Linux http://www.estrellateyarde.org/so/logs-en-linux

filtrando los intentos fallidos de conexión. Superado el número de intentos de conexión (tres), a través del script, se envía la dirección IP del atacante al fichero /etc/hosts.deny, a fin de denegar la conexión al equipo atacante desde el host víctima. Adicionalmente, este script genera un fichero con los IP que se van registrando en hosts.deny y envía un mail de notificación, al administrador de la red, sobre la IP que acaba de ser denegada la conexión y junto a la IP de su víctima.

En la figura 4.34 se describe el diagrama de secuencia sobre la lógica de funcionamiento del script bloque.sh, creado e implementado como contramedida ante intentos de conexión no solicitadas.

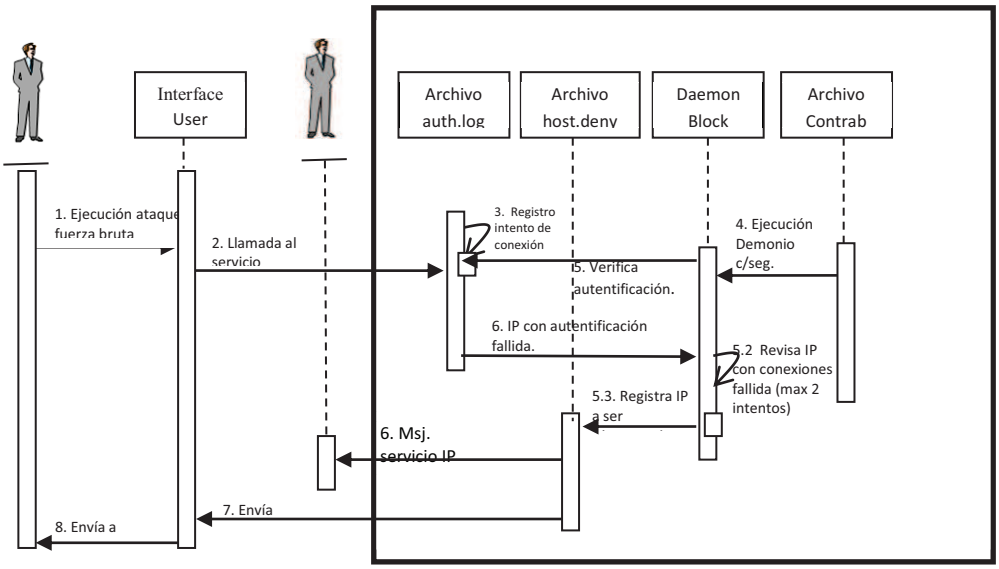


Figura 4.34 Diagrama de secuencias del proceso de mitigación a un ataque de fuerza bruta.

Cabe mencionar que para la ejecución automática del script bloqueo.sh se hizo uso del cron, el mismo que fue configurado para que revise la tabla de tareas crontab (/etc/crontab) en búsqueda de tareas que se deban cumplir, como es la ejecución del script bloqueo.sh cada segundo.

En el ANEXO C se observa un fragmento del script que permite mitigar un ataque de fuerza bruta.

4.4.2. Script que Modifica la Configuración del Firewall en Ubuntu

La configuración del firewall[55] 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[56], 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 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 permitidas 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 lo cambie por una dirección válida. Esto se lo hace con MASQUERADE.

En la Figura 4.35 está un resumen sobre los pasos a seguir, por el firewall, para contrarrestar ataques ARPSpoofing, IPSpoofing y Denegación de servicio DoS.

[55] http://es.tldp.org/Manuales-LuCAS/doc-iptables-firewall/doc-iptables-firewall.pdf
[56] http://www.pello.info/filez/firewall/iptables.html

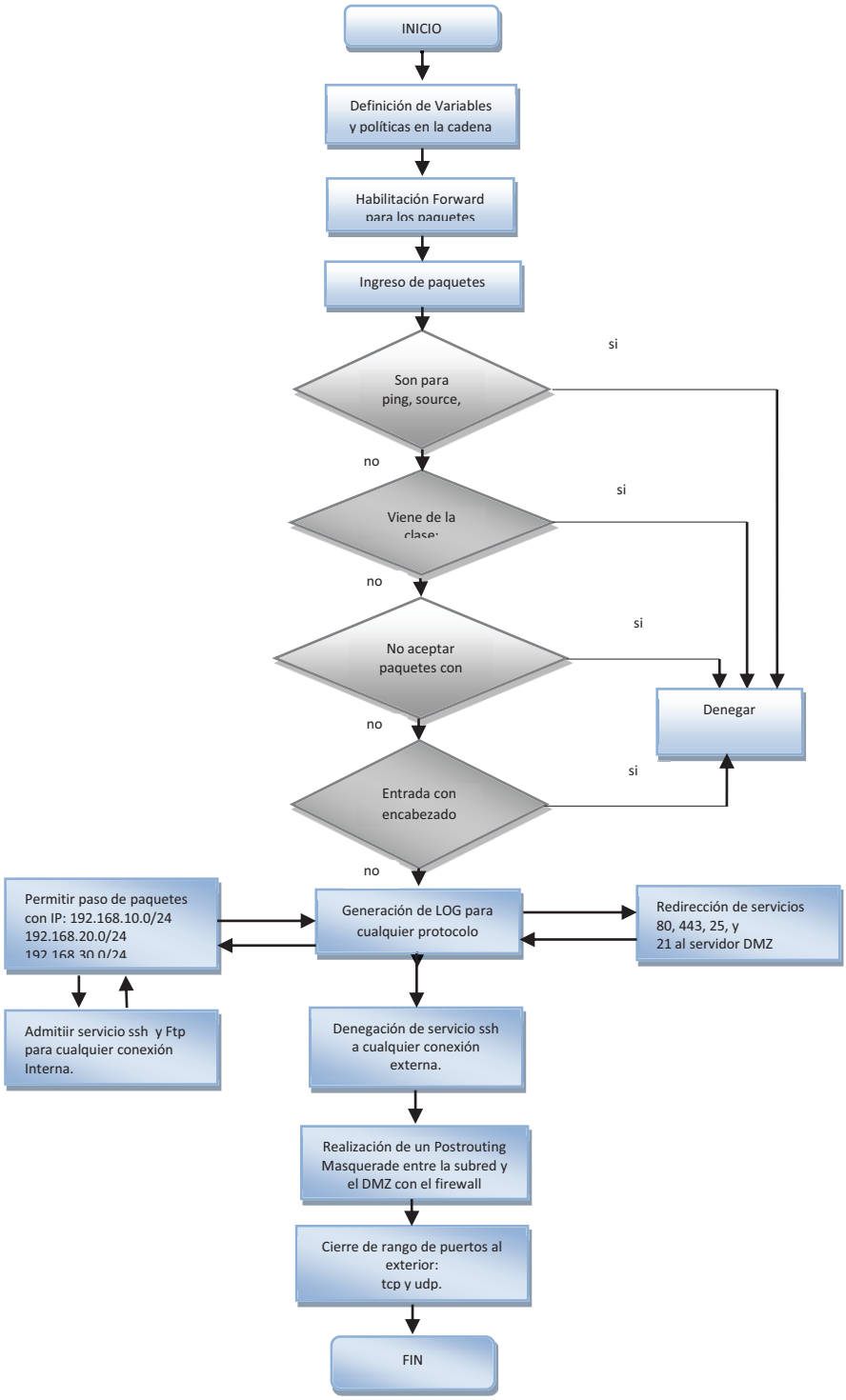


Figura 4.35 Diagrama de pasos del Firewall creado en Shell Script

En el ANEXO D se observa un fragmento del script para la creación de un Firewall que permite mitigar ataques de denegación de servicios (DoS) y suplantación de identidad (Spoofing).

4.4.3. Mecanismos de Seguridad para Protección Contra Ataques a Web, Base de Datos y de Correo Electrónico.

4.4.3.1.Mitigación ataque xss

Para proteger nuestra aplicación contra inyección XSS hay que centrarse en proteger la entrada de datos de los usuarios. Ya que esta es la manera que tiene el atacante para llevar a cabo la inyección. Toda información que el usuario pueda insertar mediante formularios, variables en el link, etcétera debe de ser analizada y filtrada para que su único fin sea el que el programador de la aplicación Web haya decidido. Para ello podemos utilizar las siguientes medidas:

- Limitación de tamaño de la entrada, la misma que debe de ir acorde con la función que vaya a desempeñar esa entrada de datos. Por ejemplo, para que un usuario introduzca su nombre y contraseña no debería disponer de un tamaño máximo mayor a 20 caracteres.

- Filtrado de caracteres especiales: Cómo hemos visto en el apartado anterior, el filtrado de caracteres especiales (“,’,<,>,/,…) que puede ser una medida simple para evitar ataques XSS a primera vista, pero hay que considerar que estos filtros pueden ser esquivados.

- Filtrado de etiquetas html: Una de las mejores medidas de prevención es filtrar las etiquetas html, para que un usuario externo sólo pueda introducir texto plano. Evitando así que inserte scripts, imágenes, o links.

4.4.3.2. Mitigación de ataques a base de datos

- Utilizar validaciones de campo para impedir la entrada de scripts en los campos de inserción de datos de la aplicación.
- Este tipo de validación hace que no se pueda incluir código malicioso O caracteres maliciosos en la inserción de datos.
- Limitar el tamaño de los campos de entrada de la aplicación, a fin de poder limitar los caracteres de entrada y no se puedan incluir código malicioso

4.4.3.3. Mitigación de Ataque a correos electrónicos

- Utilizar en la aplicación Secure Socket Layer (SSL). Cabe destacar que SSL es un complemento de protección. También se debería utilizar certificados de seguridad de una entidad certificadora reconocida.
- Cifrar las sesiones de usuario y las variables que se pasan entre páginas, fijar un tiempo prudente de caducidad de la sesión.

CAPÍTULO V

EVALUACIÓN DE RESULTADOS

5.1. Resultados Obtenidos por los Ataques Implementados

5.1.1. Ataque de Rastreo de Sistemas (Escaneo de puertos)

En la Tabla 5.1, se tiene 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, como se describe a continuación.

En la gráfica de la figura 5.1-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, no siendo así con los otros tipos que utilizan paquetes TCP, que maneja mensajes de error ICMP durante la conversación establecida entre el atacante y su víctima. En la Figura 5.1-b) se observa en un equipo con Ubuntu ocupa más recursos de red que un equipo con Windows. Adicionalmente en la figura 5.1-c. se observa 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.

Descripción Ataque		Software para el ataque	Muestra de Datos											
Escaneo de Puertos		Nmap , Zenmap												
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 (Mi B/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**	**10,0**	**3598,0**	**2,2**	**38,3**	**2002,0**	**1070,9**	**10,5**	**2026,0**

Tabla 5.1. Muestra de datos referente al tiempo, recurso de red y número de paquetes ocupados por un atacante de Rastreo de Sistemas.

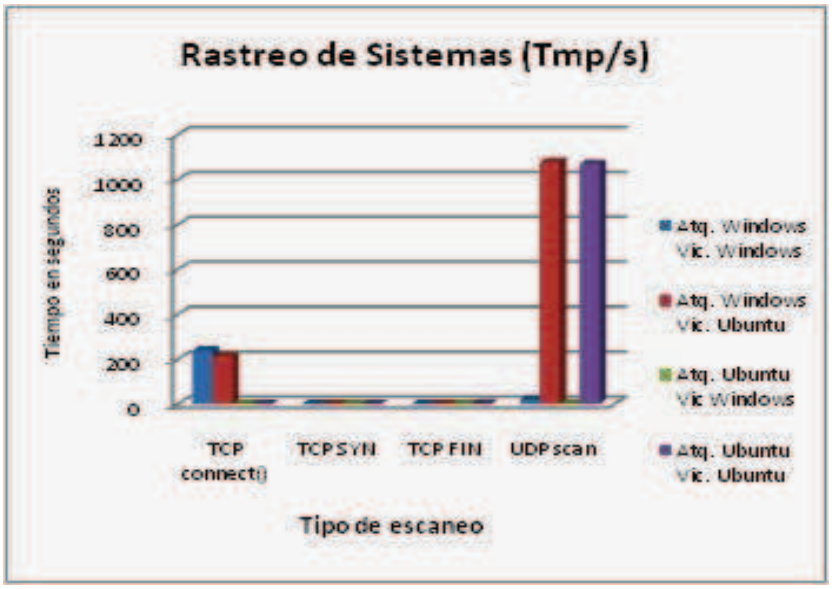


Figura 5.1-a) Tiempo en segundos.

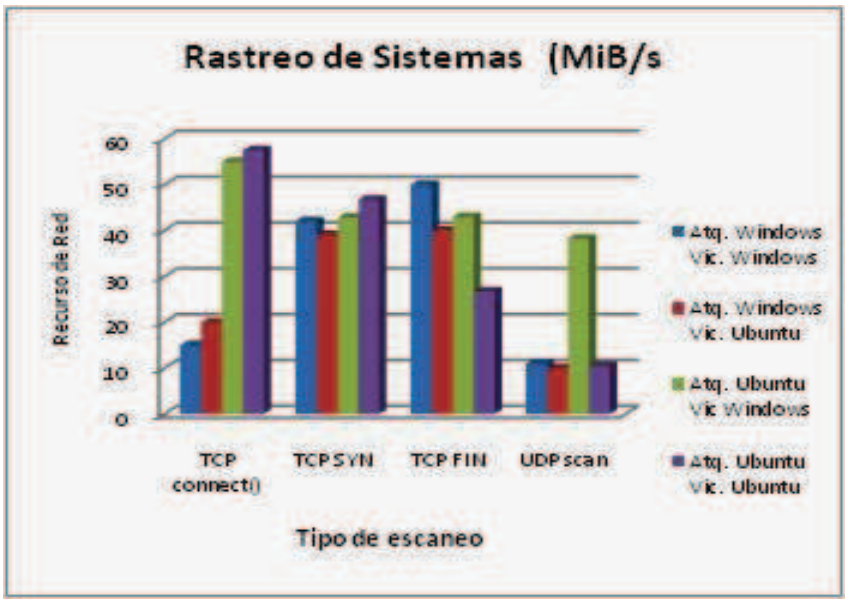


Figura 5.1-b) Recurso de red.

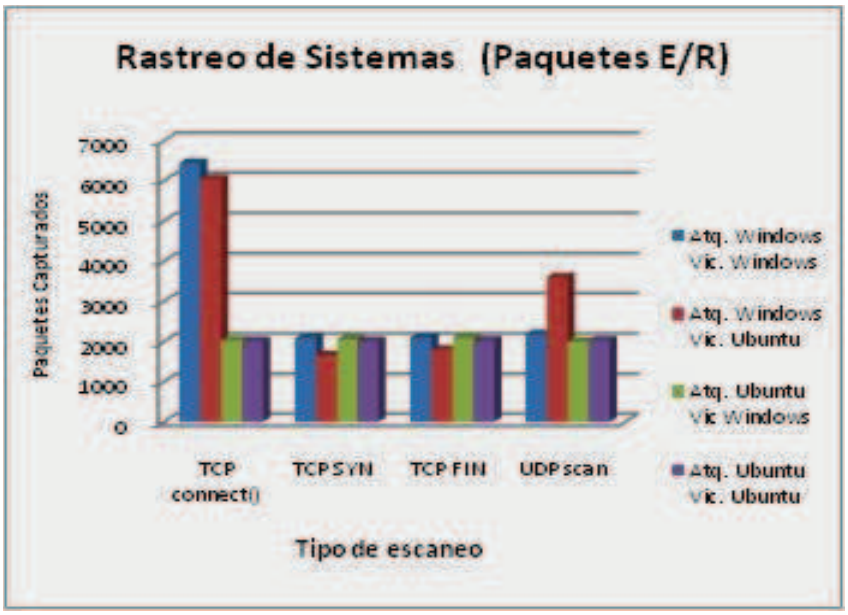


Figura 5.1-c) Número de paquetes capturados

Figura 5.1. Consumo de recursos por un Ataque de Rastreo de Sistemas.

5.1.1.1. Función de distribución de probabilidad acumulada en ataques de escaneo de puertos

En la gráfica de la figura 5.2-a, se observa el tiempo que se demora en hacer un escaneo de puertos el equipo atacante a un equipo víctima, la moda (valor que

más se repite) es de 1,75 seg, adicionalmente se ve que el tiempo promedio en realizar un escaneo de puertos es de 1,67 seg. Al calcular la desviación estándar 0,546 s. y la desviación de la media con la fórmula a), se obtiene una diferencia bastante pequeña lo que demuestra que las medidas tomadas son certeras.

$$\partial = \sqrt{\frac{\sum_i^n (Xi - X)^2}{n-1}} \qquad a)$$

En la gráfica de la figura 5.2-b. se observa las muestras tomadas referente a la cantidad de paquetes en KB/s que viaja por la red. El valor con mayor recurrencias es de 42 KB/s, valor relativamente bajo que no afectaría al rendimiento de la red.

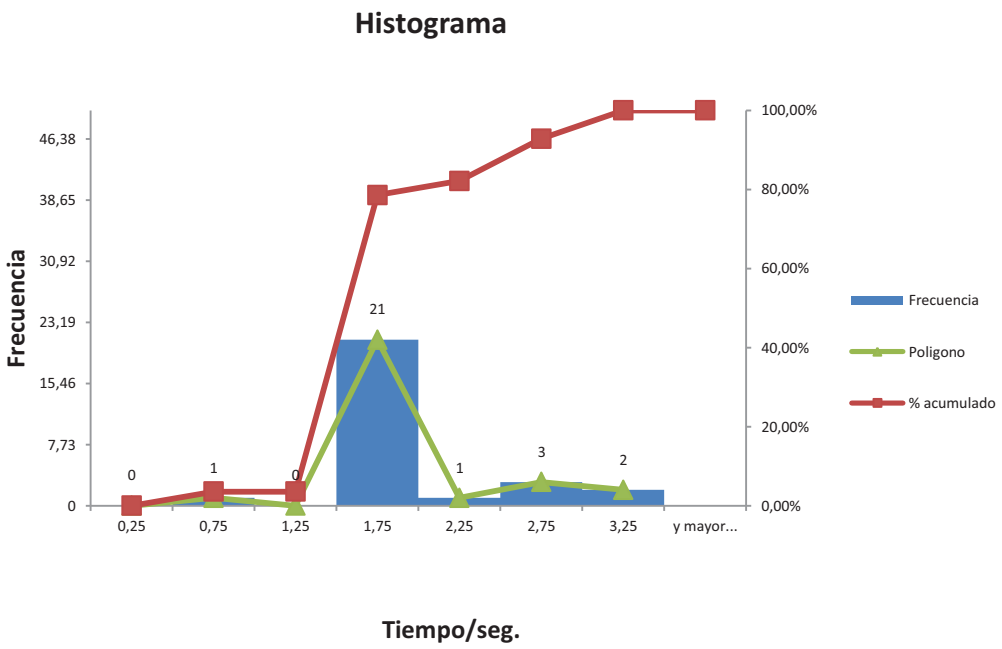


Figura 5.2-a) Tiempo en segundos que demora un ataque de rastreo de sistemas.

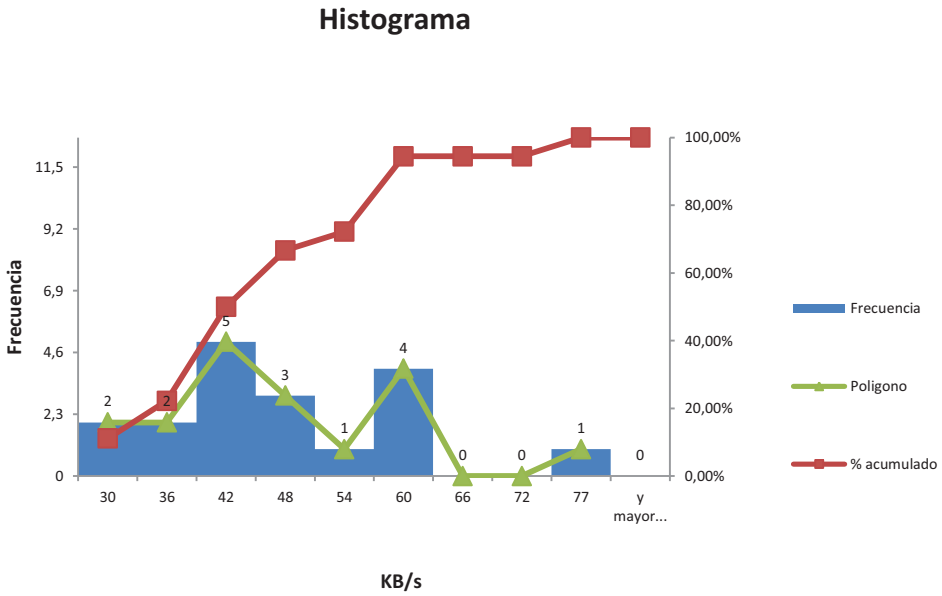


Figura 5.2-b) Recurso de red que ocupa un ataque de rastreo de sistemas

5.1.2. Ataque de Fuerza Bruta

En la Tabla 5.2 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, sobre Ubuntu, en función al tiempo que se demora en localizar la contraseña dentro de un fichero diccionario (inicio, medio y final).

En la gráfica de la figura 5.3-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 5.3-b, cuando mayor sea el archivo diccionario utilizado por el programa medusa, más posibilidades se tendrá de encontrar la contraseña pero en más tiempo.

Descripción Ataque		Software para el ataque	Muestra de Datos				*Sniffer*
Fuerza Bruta		**Medusa**					
		John the Ripper					wiresh ark
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 the 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	√

Tabla 5.2 Muestra de datos referente al tiempo, recurso de red y número de paquetes ocupados por un ataque de Fuerza Bruta.

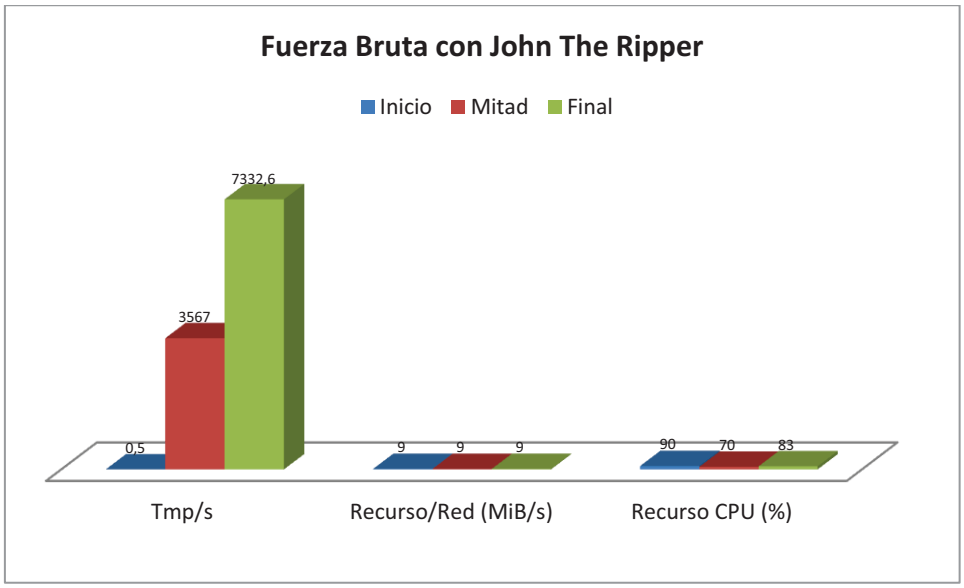


Figura 5.3-a) Recursos consumidos por un ataque utilizando Medusa

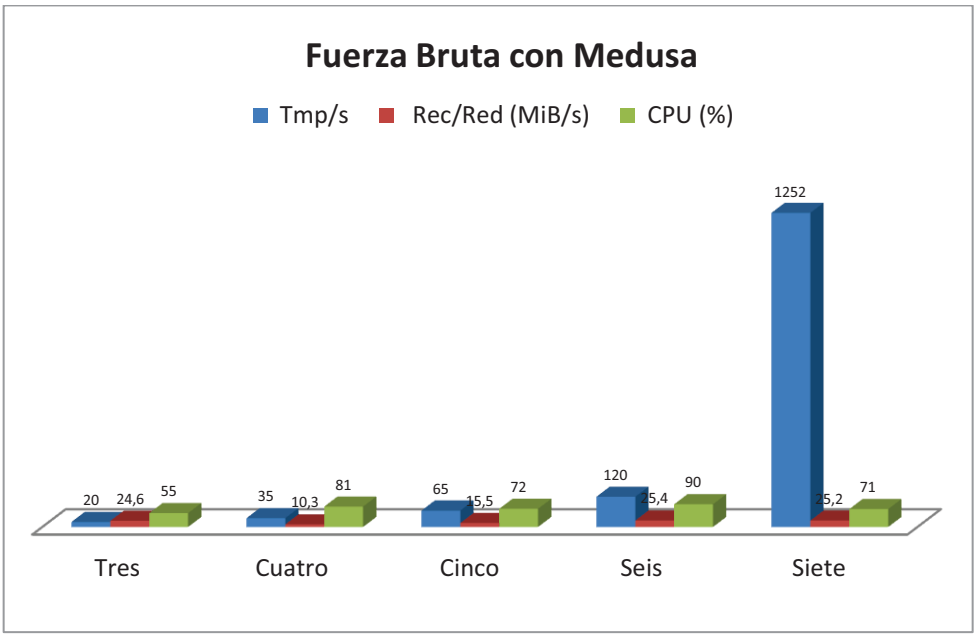


Figura 5.3-b) Recursos consumidos por un ataque con John The Ripper

Figura 5.3 Recursos de red, CPU y tiempo, consumido por un atacante de Fuerza Bruta

5.1.2.1. Función de distribución de probabilidad acumulad en ataques de Fuerza Bruta

En la gráfica de la figura 5.4 se observa el tiempo más frecuente en que se demora, un ataque de fuera bruta, en descifrar una clave. Cabe mencionar que las formas de combinación de caracteres (alfanuméricos) para las contraseñas tienen el siguiente tamaño: pequeño (2-3 caracteres), mediano (5-6 caracteres) y largo (más de 6 caracteres). Siendo la forma pequeña la más concurrente por su número de aciertos alcanzados Adicionalmente para el uso de la aplicación Medusa se empleó un archivo (diccionario) de contraseñas con 3600 palabras.

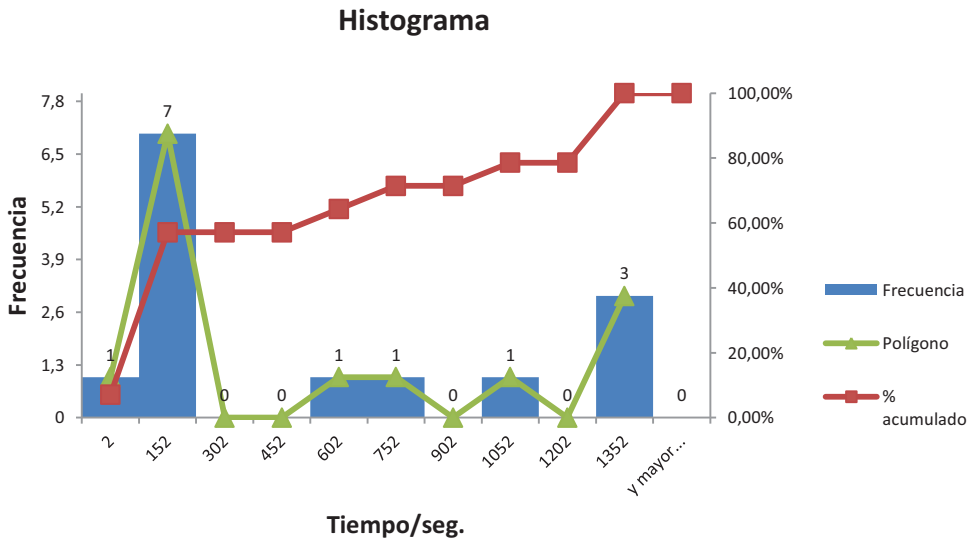


Figura 5.4 Tiempo consumido al realizar un ataque de Fuerza Bruta

5.1.3. Ataque de Suplantación de Identidad

La tabla 5.3, describe los recursos de red y de CPU que ocupa un ataque de suplantación de identidad, dada la rapidez con que se ejecutan el comando y el tiempo ilimitado que dura el ataque, se consideró un margen de duración del ataque, 60 segundos, a fin de poder tomas las muestras correspondientes.

En la Figura 5.5, se resume las mediciones obtenidas sobre los recursos de red y CPU ocupados durante un ataque IPSpoofing y ARPSpoofing. En la gráfica 5-a y 5-b se observa una similitud en datos obtenidos en los dos tipos de Spoofing realizados y que los recursos de red consumen muy poco en relación a los anteriores ataques.

Descripción Ataque		Software para el ataque	Muestra de Datos			*Sniffer*
Suplantación de Identidad		Hping2				
		Nemesis				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	√	√
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	√	√

Tabla 5.3. Recursos utilizados en un ataque de Suplantación de Identidad.

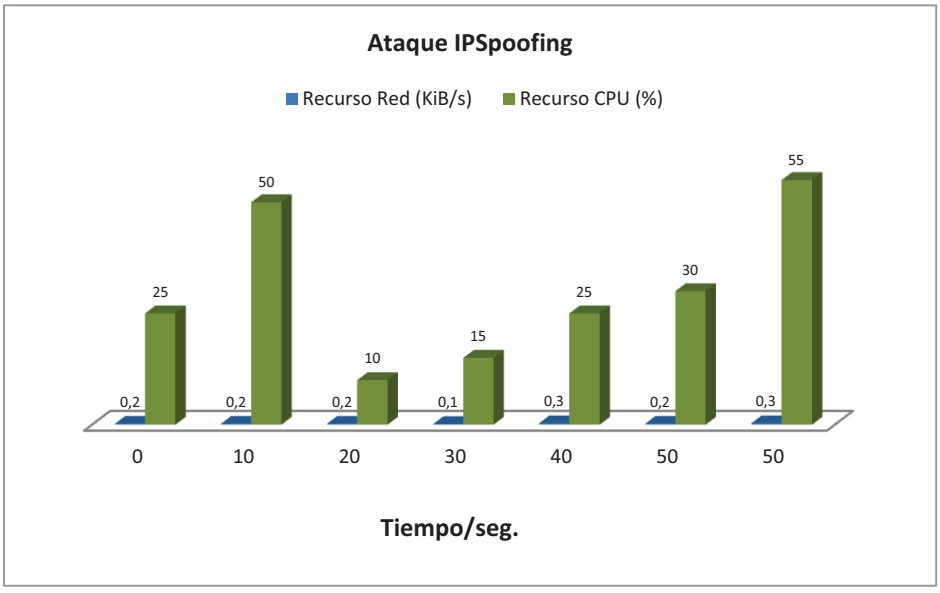


Figura 5.5-a) Recursos consumidos por un ataque IP Spoofing

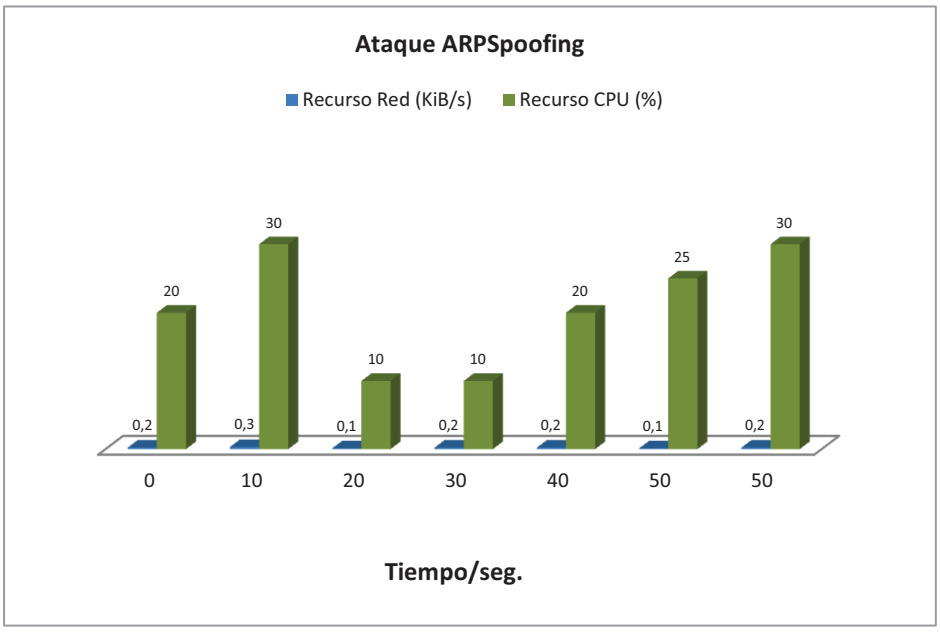


Figura 5.5-b) Recursos consumidos por un ataque ARP Spoofing

Figura 5.5. Recursos utilizados por un Atacante se Suplantación de Identidad.

La figura 5.6 demuestra la diferencia existente, en cuanto al número de paquetes transmitidos, entre el atacante A y su víctima B (y viceversa), al momento de producir un ataque de denegación de servicio. Se puede observar la diferencia bastante significativa al transmitir de A hacia B en relación de B hacia A, esto se debe a que A no recibe el mensaje de respuesta de B, ya que estos mensajes son enviados a la IP por la que A se hace pasar.

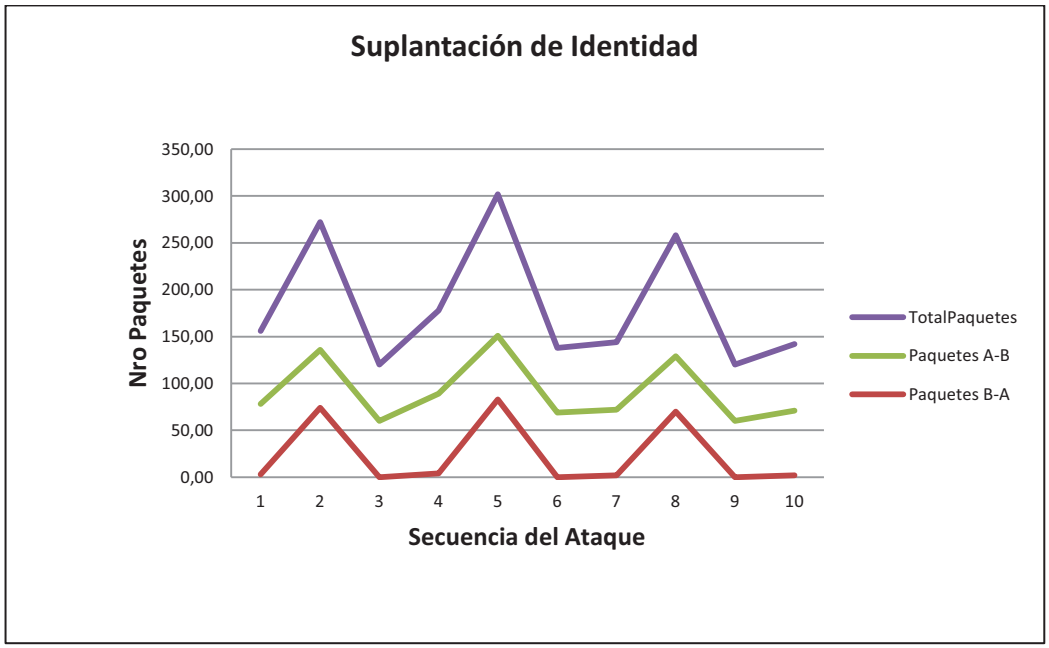


Figura 5.6. Número de paquetes transmitidos en un ataque de Suplantación de Identidad

5.1.4. Ataque de Denegación de servicios

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 resumidas en la tabla 5.4. La figura 5.7, muestra los resultados obtenidos y se puede apreciar en la gráfica que los equipos víctimas ocupan un ancho de banda de 963 kbps a los 60 segundos, alcanzado su máximo nivel, esto se debe a que mientras más tiempo dure el ataque la víctima será bombardeada con peticiones de conexión llegando a saturarse dicho equipo.

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%

Tabla 5.4 Recursos de ancho de banda consumido en un ataque DoS.

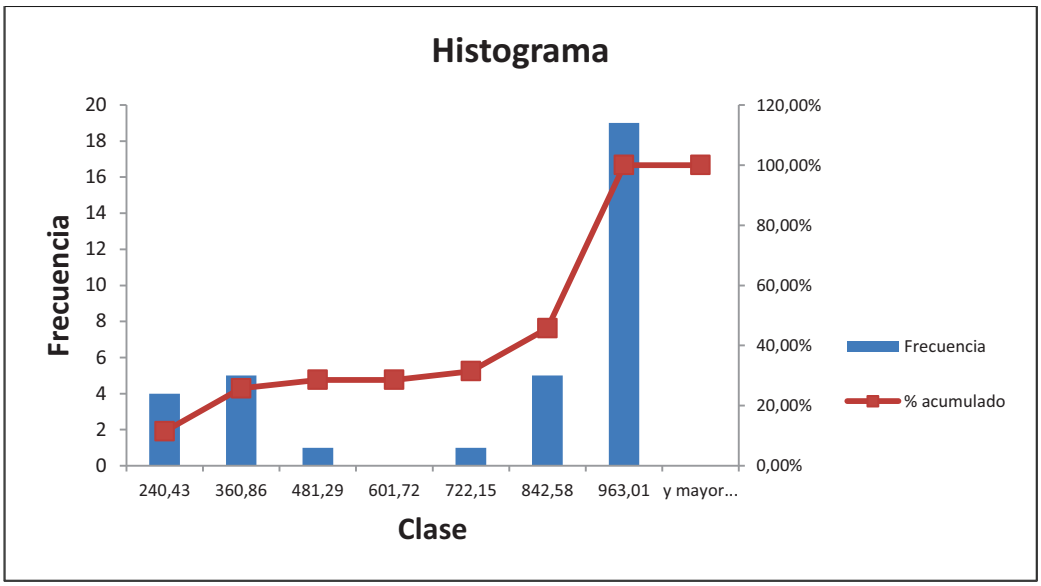


Figura 5.7. Porcentaje acumulado del consumo de ancho de banda en un ataque DoS.

En la figura 5.8 se pude observar la cantidad de bytes que se transmite entre el atacante A y su víctima B, siendo el equipo B el que transmite pocos bytes, en relación a la gran cantidad de bytes que trasmite su atacante A, esto se debe a que B queda sin servicio por el ataque generado y no puede trasmitir dado que no encuentra al ruteador.

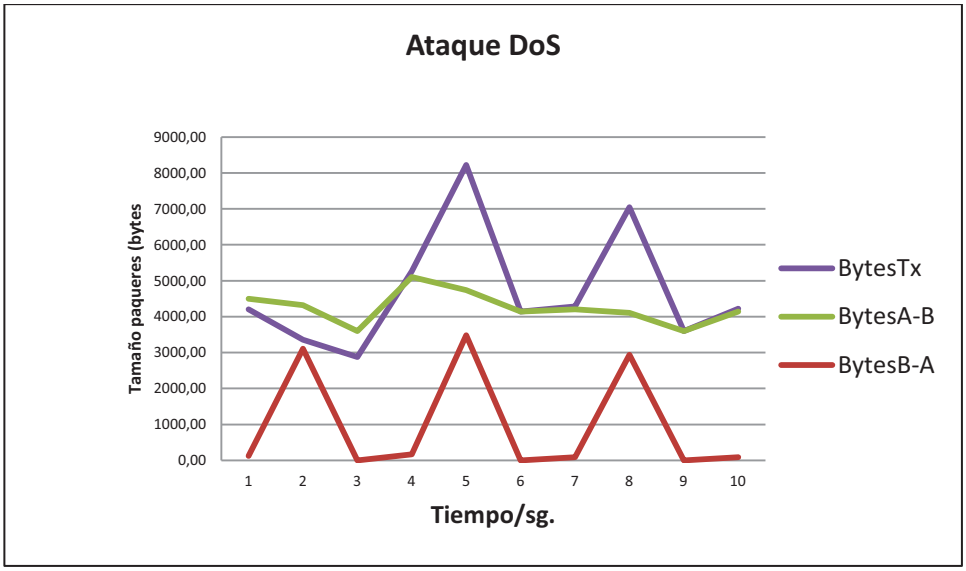


Figura 5.8 Cantidad de Bytes transmitidos en un ataque DoS.

5.2. Resultados de los Ataques Implementado los Mecanismos de Mitigación.

5.2.1. Rastreo de Sistemas o Escaneo de Puertos

Un paquete TCP con la bandera ACK activa nos indicará la presencia de un equipo al otro lado de la conexión.

El problema que brinda este tipo de ataque es el ruido que provoca. Este hecho provoca que cualquier firewall bien configurado deseche este tipo de paquetes que pretendan abrir una conexión.

Por ello es importante mencionar que en firewall además de cerrar los intentos SYN, también se obstruyen los intentos de conexión ACK.

Utilizando el programa <nmap> que se debe instalar en una máquina (atacante) que trata de acceder a la red que protege el firewall se realizaron las siguientes pruebas:

Caso 1 Rastreos con banderas TCP

```
# Nmap  -sT 192.168.30.1 -p 21,22,80
```

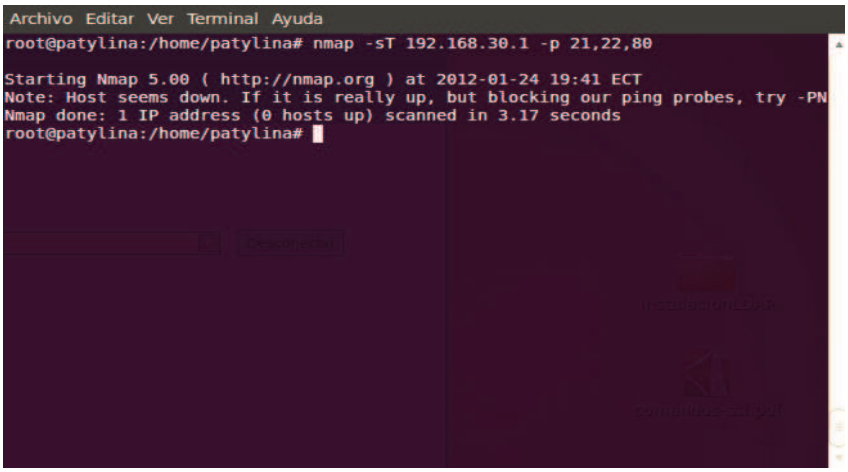


Figura 5.9 Resultados del efecto de un escaneo de puertos tipo TCP

Figura 5.9 Muestra los resultados obtenidos al efectuar una solicitud con NMAP, para establecer conexión con el firewall, y que este tipo de paquetes son

filtrados por el firewall y no proporciona respuesta alguna. Lo que significa que el firewall no deja pasar éste tipo de paquetes.

Caso 2 Rastreos con banderas SYN

Este tipo de intentos de conexión bajo equipos LINUX son respondidos con un RST, a continuación se observa la respuesta del servidor.

```
#Nmap  -sS –n –T 192.168.30.2
```

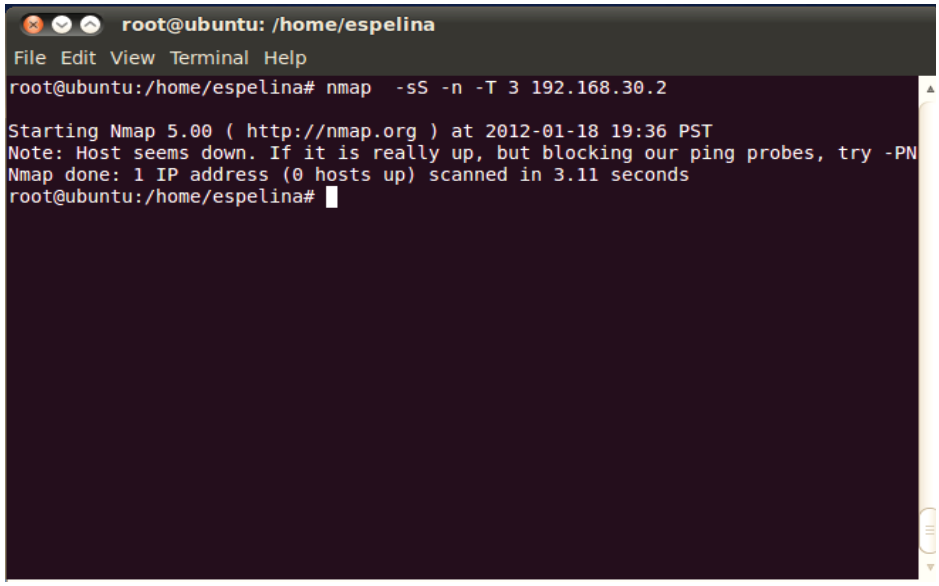


Figura 5.10 Resultados del efecto de un escaneo de puertos tipo SYN

En la Figura 5.10 se muestra la respuesta por parte del firewall, que fue nula, con las banderas ACK y SYN activas. Por otro lado, se realizó nuevamente la prueba para otro equipo y la respuesta se muestra fue exitosa, es decir que se estableció conexión con el equipo víctima.

Caso 3 Rastreos con banderas ACK

```
#Nmap  -sA 192.168.30.1 -p 21
```

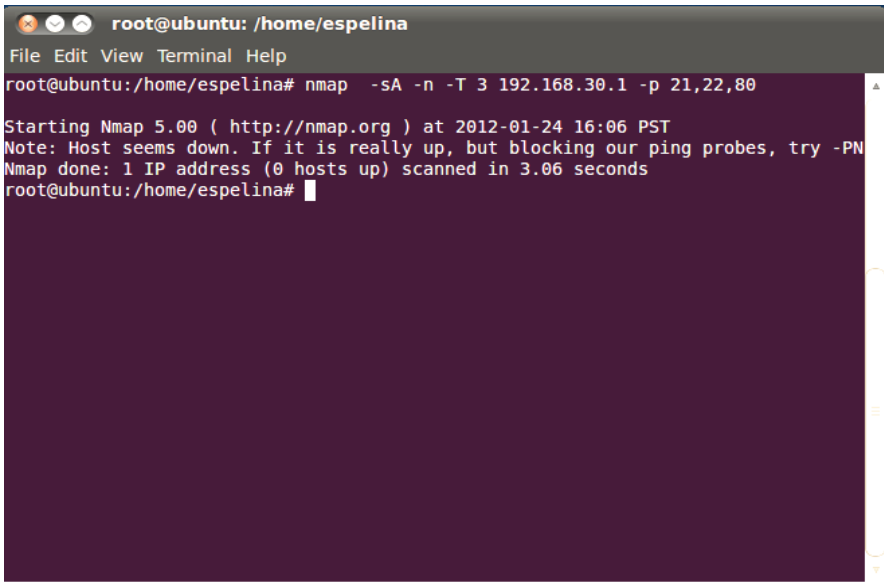


Figura 5.11 Resultados del efecto de un escaneo de puertos tipo ACK

Claramente en la figura 5.11 se observa que la respuesta es nula ante la petición de intento de conexión. Se puede interpretar esta información como que el firewall bloquea éste tipo de paquetes o que no se observa respuesta alguna.

Caso 4 Rastreos con banderas FIN

Un paquete con la bandera FIN activa deberá ser respondido de la siguiente forma:

- RST -ACK ante un intento a un puerto cerrado
- Ninguna respuesta ante un intento en un puerto abierto

La ausencia de respuesta ante un paquete FIN se debe a cualquiera de los siguientes casos:

- Paquetes FIN bloqueados por un firewall, routers o ACL´s.
- Se encuentra abierto el puerto elegido.
- El equipo remoto esta desconectado o apagado.

Ejecución del ataque:

```
#Nmap  -sF 192.168.30.2 -p 21
```

Los resultados se muestran a continuación, ver figura 5.12.

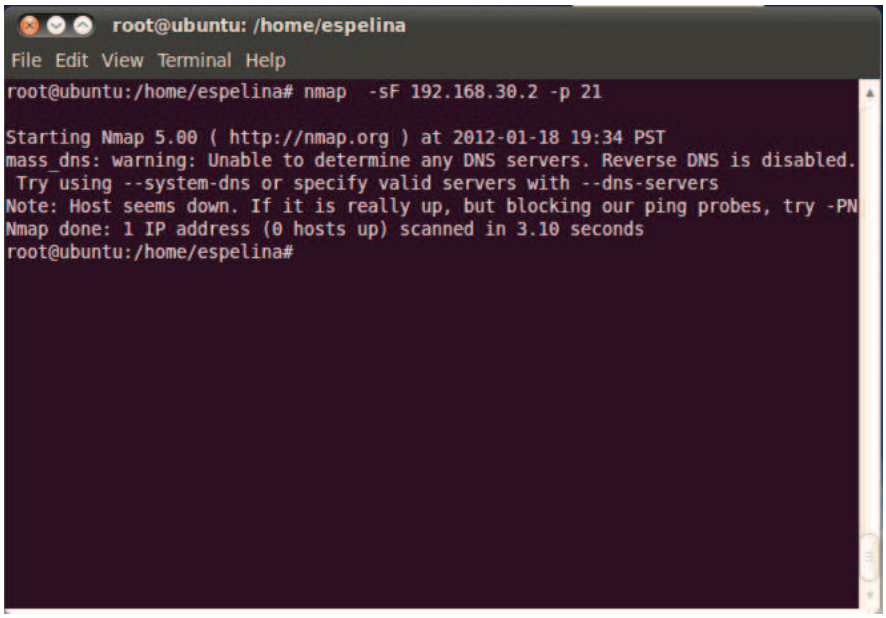


Figura 5.12 Resultados del efecto de un escaneo de puertos tipo FYN

Como se observa en la figura 5.12, para el 192.168.30.2 la respuesta es nula a éste tipo de intento de conexión. Se puede interpretar esta información como que el firewall bloquea éste tipo de paquetes o que no se observa respuesta a ellos como se observó al principio, en la utilización de FYN.

5.2.2. Ataque de Fuerza Bruta

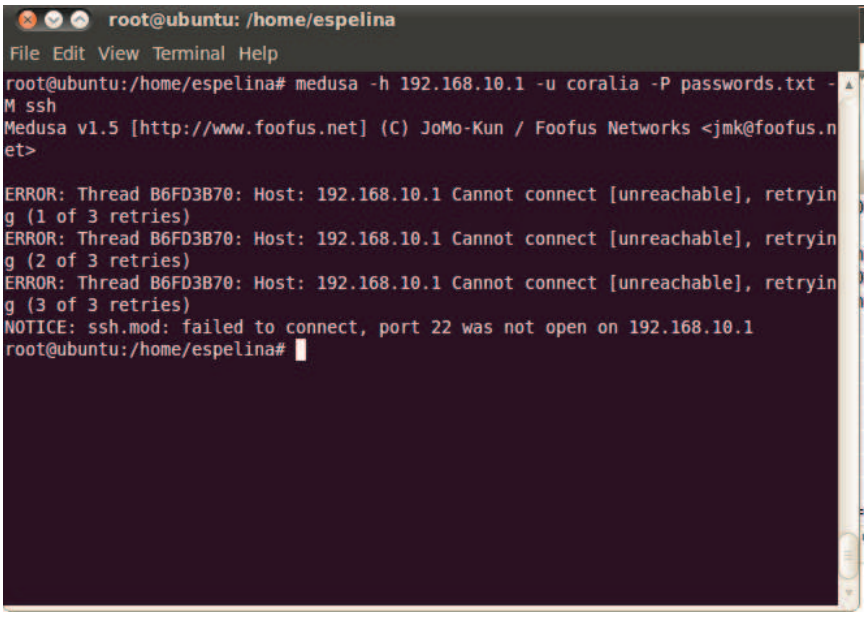


Figura 5.13 Muestra de Resultados del efecto de un ataque de fuerza Bruta con Medusa

Como se observa en la figura 5.13, el intento de conexión a la IP 192.168.10.1 la respuesta es nula. Se puede interpretar esta información como que el script BloqueoF.sh bloquea el acceso al equipo atacante, que tiene como fin descifrar la contraseña del equipo víctima, utilizando el puerto 22.

5.2.3. Ataque de Suplantación de Identidad (Spoofing)

Spoofing o suplantación de una dirección origen es donde el remitente utiliza otra dirección, en lugar de la propia, en el campo origen, a fin de fingir ser otra persona durante el ataque. Existen seis clases de direcciones que se deben negar en la interfaz externa y son:

1. La propia dirección IP.- Los paquetes legales nunca proceden de la máquina propia.

2. Direcciones IP privadas de clase A, B, C.- Son reservadas para su uso en la LAN (red de área local) privadas y no se usan en Internet,

3. Direcciones IP de multidifusión de clase D.- Estas se reservan para uso exclusivo como direcciones destino cuando se participa en una difusión de sonido o vídeo en una red de multidifusión.

4. Direcciones reservadas clase E.- Este intervalo es reservado para usos futuros y de ámbitos científicos o experimentales

5. Direcciones de Interfaz ciclo invertido.- Es una interfaz de red privada que utiliza UNIX para servicios locales de red

6. Direcciones de difusión mal formadas.- Estas son direcciones especiales que se aplican a todas las máquinas de una red. Por ejemplo la dirección 0.0.0.0 es de difusión especial.

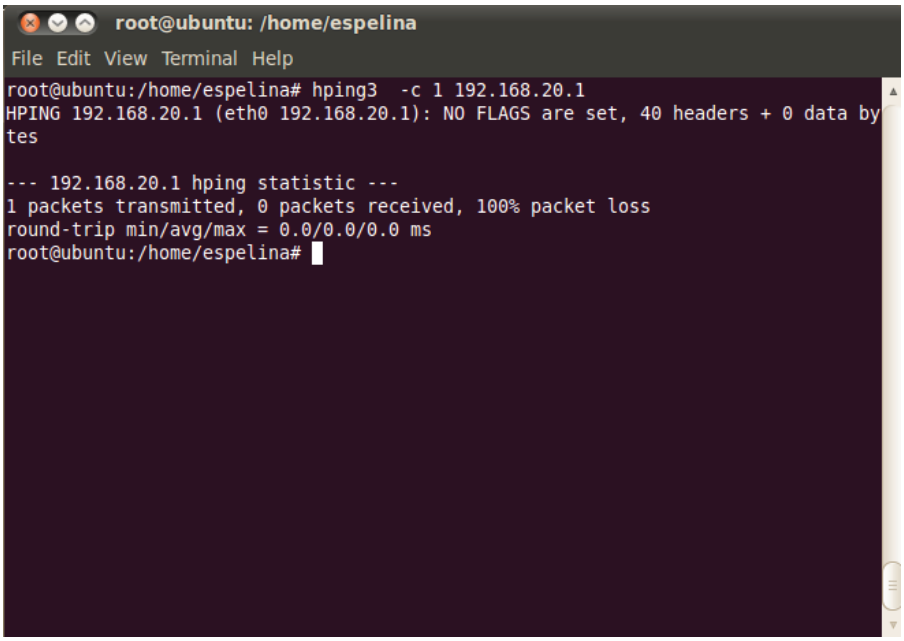


Figura 5.14 Muestra de Resultados del efecto de un ataque Spoofing

Como se observa en la figura 5.13, el intento de conexión a la IP 192.168.20.1 la respuesta es nula. Se puede interpretar esta información como que el firewall bloquea el acceso al equipo atacante, que tiene como fin establecer conexión con su víctima suplantando su identidad (dirección IP).

5.2.4. Ataque de Denegación de Servicios (DoS)

Un ataque de Dos sucede cuando un atacante intenta ocupar la mayoría de los recursos disponibles de una red IP e impide que los usuarios legítimos de ésta los utilicen.

Ejecución del Ataque

```
#Hping3 -a 192.168.10.1 192.168.100.10
```

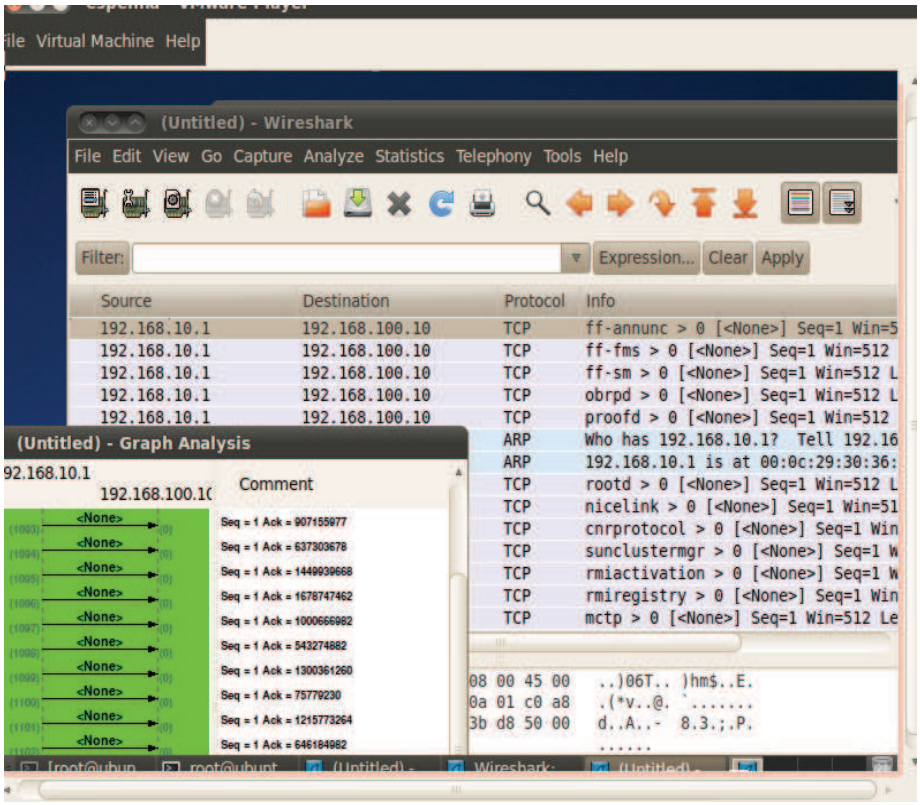


Figura 5.15 Muestra de Resultados del efecto de un ataque DoS

Como se observa en la figura 5.15, el intento de conexión a la IP 192.168.10.1 la respuesta es nula. Se puede interpretar esta información como que el firewall bloquea el acceso al equipo atacante. Adicionalmente se puede ver el análisis gráfico que proporciona la herramienta Wireshark, ante este tipo de ataque, que la transmisión de paquetes TCP es nula.

5.2.5. Ataques desde una Máquina Windows

Como se ha mencionado a lo largo de este documento una de las formas de atacar una PC es conociendo el sistema operativo, otra es mediante los servicios que la máquina ofrezca, o a través del sitio web, etc.

De esta manera el próximo paso es probar el firewall, en un intento de intrusión remoto, pero ahora utilizando una máquina Windows (atacante) y a través del programa llamado Putty, que lo que hace es generar una terminal shell para conectarse a un servidor LINUX. En la figura 5.16 se muestra la interfaz de este programa así como el puerto que utilizará, puerto 22 de servicio ssh, seguidamente se inserta la IP

de la máquina a la que se quiere entrar (victima) que en su momento al realizar la fase de obtención de cuentas de usuario no funcionó, ver figura 5.16

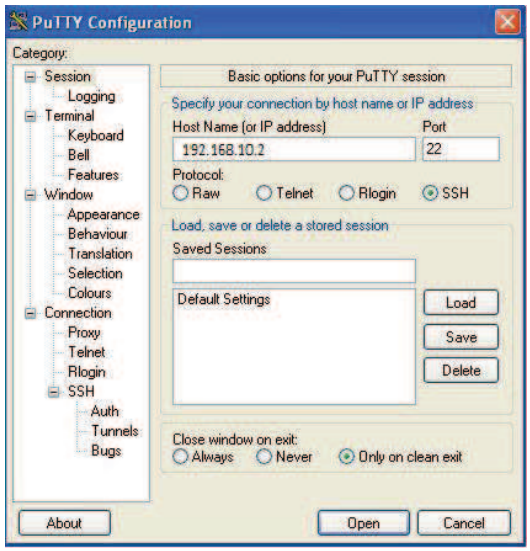


Figura 5.16 Pantalla de ingreso de datos para obtener una shell de la máquina objetivo.

Como se observa en la figura 5.17 el resultado obtenido es que no se estableció la conexión con 192.168.10.1, el firewall rechazó la conexión, por lo que se ha verificado una vez más que el firewall no ha podido ser infiltrado.

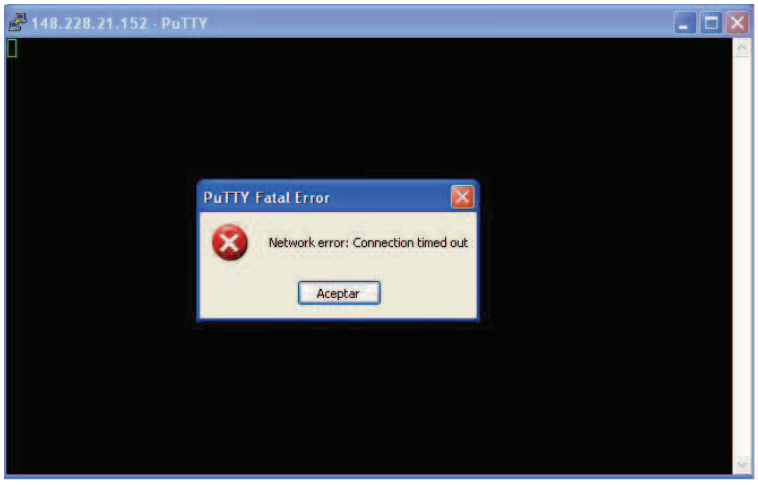


Figura 5.17 Resultados obtenidos de conexión, esta fue denegada

5.2.6. Ataques a la Web, Implementado Mecanismos de Protección

Ataque XSS

a) Limitación del tamaño de la entrada

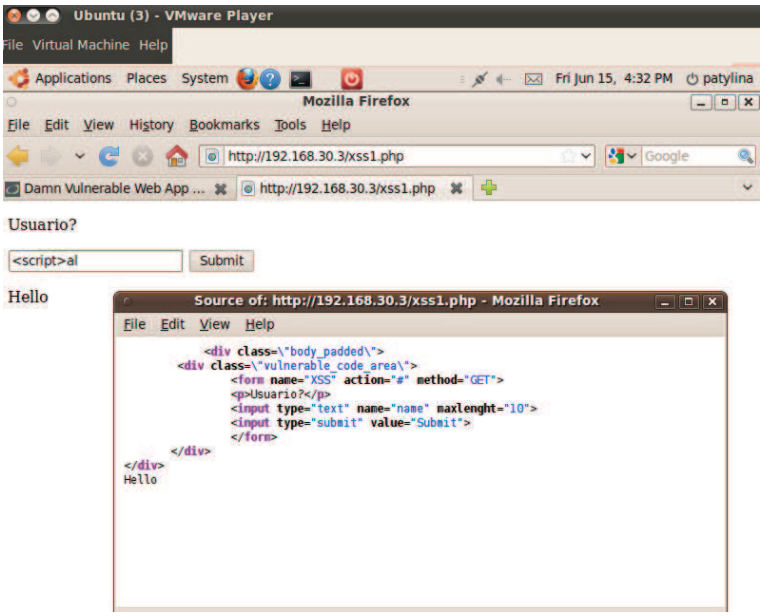


Figura 5.18 Resultados obtenidos, con limitación del tamaño del campo

En la figura 5.18 se observa la limitación a 20 caracteres el campo de entrada de datos de una página programada en php. Con ello se logra que no se pueda digitar más de 20 caracteres en dicho campo.

b) Filtrado de caracteres especiales html

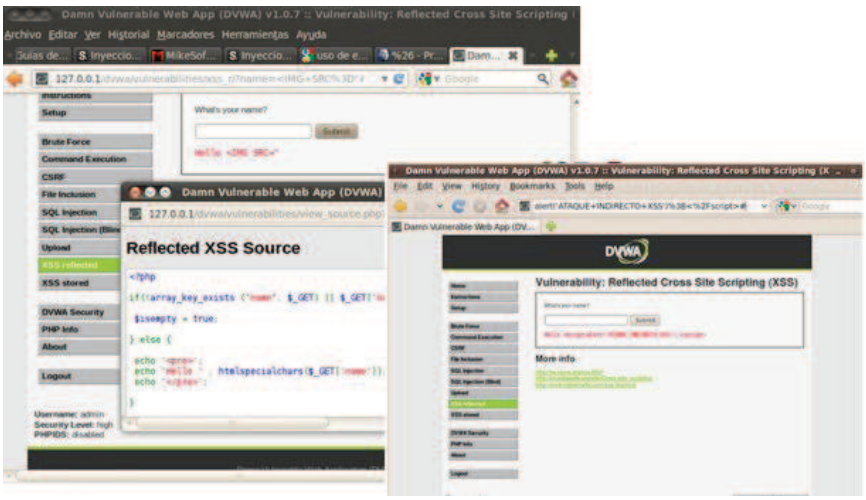


Figura 5.19 Resultados obtenidos, con el filtrado de códigos especiales en html

En la figura 5.19 se observa que al hacer uso de la función htmlspecialchars en php, se hace un filtrado de caracteres especiales en html de tal manera que son manejados como texto simple y no como sentencias html.

5.2.7. Ataques a un Servidor de Correo Electrónico Seguro con SSL

Usuarios Autorizados

Una vez que se ha Configurar SASL ("Simple Authentication and Security Layer") a fin de poder añadir soporte para autenticación a los protocolos como SMTP, y poder restringir el acceso de retransmisión, a través de los certificados digitales que sirven para identificar y firmar de forma digital las acciones que se realiza en internet. Para el caso en que se necesite tener unos cuantos certificados para realizar pruebas de programación, como este caso, lo mejor que se pudo hacer es crear una propia entidad certificadora, utilizando OpenSSL.

Adicionalmente con el cifrado del canal de comunicación usando TLS, se logra dar mayor seguridad en el manejo de correos electrónicos. A continuación se procede a comprobar el funcionamiento del correo electrónico seguro implementado.

Primera prueba:

Configuración de Outlook Express sobre Windows XP, a fin de crear una cuenta de correo, con conexión al servidor con IP 192.168.10.100. Una vez configurado Outlook se procede al envío de mensajes como se observa en la figura 5.18

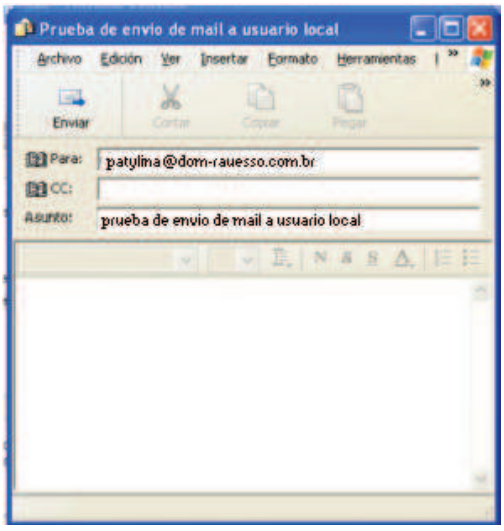


Figura 5.20 Envío de mensajes desde Outlook Express.

En esta primera prueba, no se activó la opción “mi servidor requiere autentificación”, de forma que los usuarios se intentarán conectar de forma anónima. En la figura 5.19 se observa la creación un mail para enviarlo a un usuario local:

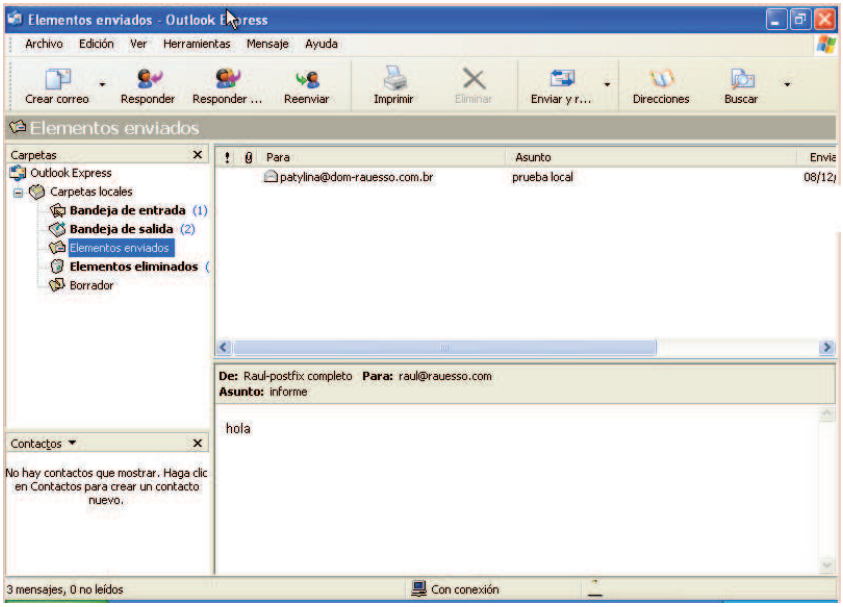


Figura 5.21 Envío de un mail a un usuario local

Como se puede apreciar en la figura anterior, el mail ha sido enviado correctamente, por lo que se puede afirmar, que sin autentificación, los usuarios pueden enviar mensajes en local.

Segunda prueba:

Esta prueba consiste en volver a confeccionar un mail para un usuario externo al servidor, el resultado que se obtiene se ve en la figura 5.20.

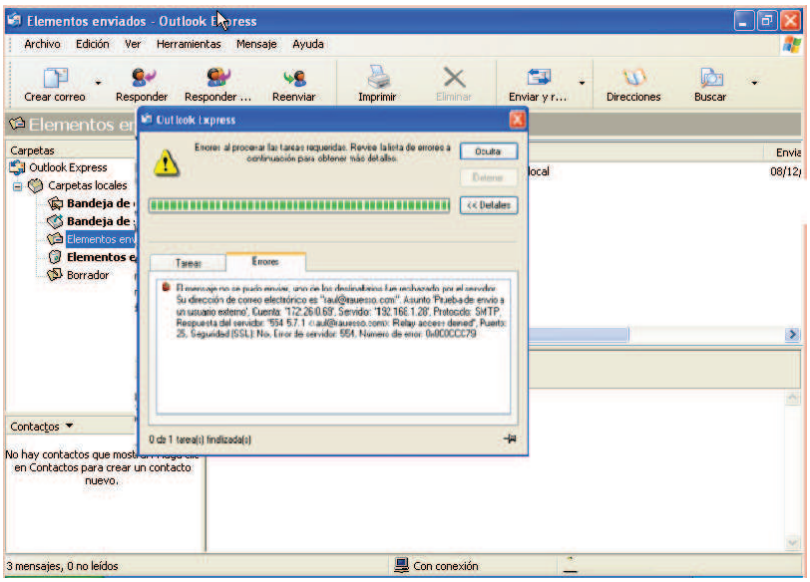


Figura 5.22 Envío de un mail a un usuario local no autentificado

El mail no puede ser enviado. Como se aprecia, el sistema nos devuelve un mensaje de error en el que se nos informa de "Relay Access denied", es decir, que el servidor nos deniega el reenvío del mail, al no haberse autentificado.

Tercera prueba:

Verificación del funcionamiento del cifrado, para ello se procede a decirle a Outlook Express que use el cifrado SSL para el servidor de correo saliente SMTP, sobre el puerto 25:

Una vez realizado esto, se compone un mail para envío, y se observa la siguiente reacción del sistema, ver figura 5.21.

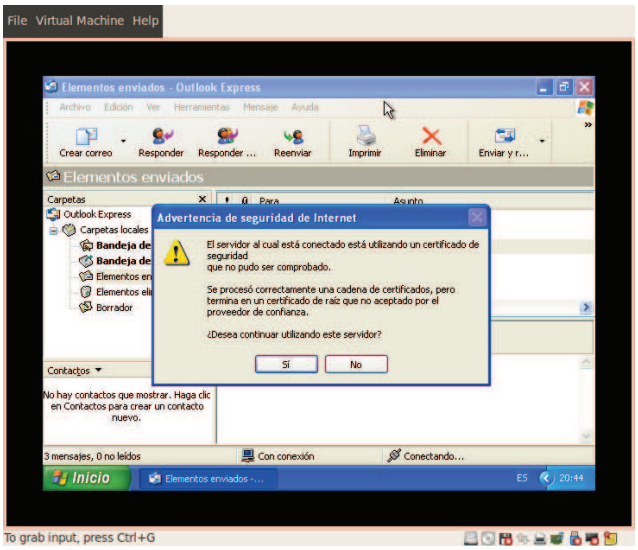


Figura 5.23 Envío de un mail con cifrado SSL

En esta segunda captura, se puede ver que el sistema informa de que el servidor hace uso de un canal cifrado (certificado) que no puede ser comprobado. Esto es así, ya que el certificado se ha creado manualmente a modo de prueba, y no ha sido certificado por una autoridad certificadora real. Seguidamente se le diría que SI, que se quiere continuar utilizado el servidor. Y el mail es enviado.

Control de SPAM

Una vez instalada y configurada las herramientas Spamassassin de filtrado, destinadas al control de SPAM y virus desde Postfix, dentro de los parámetros de configuración, se ha hecho que el proceso spam sobre escriba el asunto de un mail considerado como spam a [***** SPAM _SCORE_ *****].

Para realizar el chequeo del spam, Spamassassin realiza unos tests basándose en los grupos de reglas que hay en los ficheros del directorio /usr/share/Spamassassin. Así mismo, los scores de los tests son almacenados en un único fichero. La estructura de los ficheros consiste de un nombre de test, una descripción y la acción que se puede aplicar en la cabecera, el cuerpo del mensaje y el score, estos son conocidos como listas blancas o listas negras.

Listas blancas y listas negras

Se le puede indicar a Spamassassin que algunas direcciones no sean nunca marcadas como spam. Se le indica con la directiva whitelist_form en el fichero local.cf, ejemplo:

whitelist_form nagore@pfc-server.com

whitelist_from pfc-server.com, *pfc-server.com

whitelist_form *@pfc-server.com

Otra posibilidad muy útil es justamente la opuesta, esto es, la de las listas negras. En ella se listan los spammers conocidos mediante la directiva blacklist_from, ejemplo:

blacklist_from 0-sexshop.com 001bastconsumer.com

En la figura 5.24 se puede observar el fucionamiento del control de spam mediante el registro de e-mail en el archivo de listas negras. Los correos que son considerados peligros viene marcados en asunto con el mensaje [***** SPAM _SCORE_ *****].

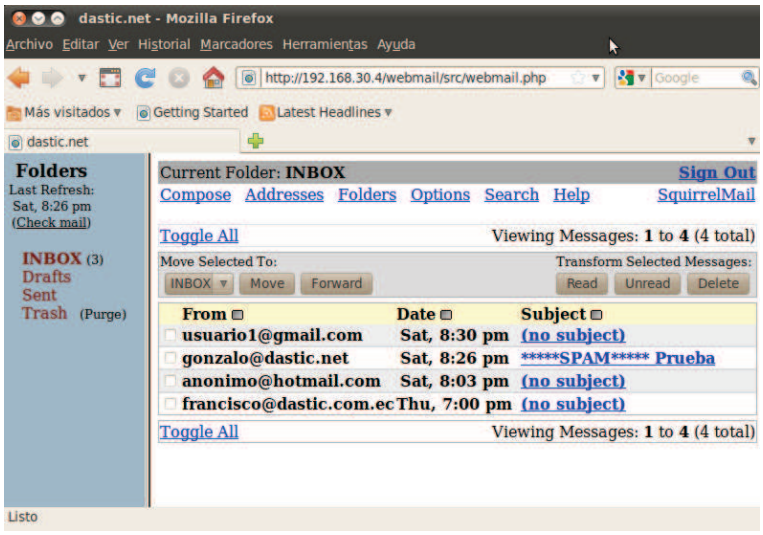


Figura 5.24 Detección de e- mail tipo SPAM.

5.2.8. Ataques a Base de Datos con Páginas Web Protegidas

Una vez visto en qué consiste la Inyección SQL y cuáles pueden ser sus graves consecuencias, el siguiente punto es explicar algunos métodos para evitar dicho

peligro. Cabe mencionar que se dará soluciones en concreto para PHP, pero son aplicables a otro tipo de lenguajes como ASP o JSP. Aunque hay varias formas de protegerse contra la Inyección SQL en PHP, a continuación se recomienda las siguientes reglas generales:

- Utiliza siempre las comillas simples (') para delimitar las variables que vayas a usar en una consulta a la base de datos. De esa manera la posibilidad de Inyección SQL se ve reducida de forma considerable:

Ejemplo

```
SELECT usuario FROM indentificacion WHERE usuario= ' $usuario ';
```

- Realiza las validaciones pertinentes para verificar el tipo de dato que debe tener una variable, ya sea en el lado del cliente media Javascript o bien por parte del servidor utilizando PHP.

- Evita el uso del operador asterisco (*), a pesar de que una tabla X no contenga más que 2 campos de debe tratar de evitar el uso del asterisco (*). Esto evitará que datos o campos que no son utilizado ser obtenidos mediante una Inyección SQL.

- También es recomendable tener configurado el servidor para que cambie las comillas de las variables pasadas, es decir que la comilla simple(') sea cambiada por (\').

  Ejemplo

```
<?
// Cómo evitar la inyección SQL
 // Modificamos las variables pasadas por URL
foreach($_GET as $variable=>$valor){
  $_GET[$variable] = str_replace("'","\'",$_GET[$variable]);
}
// Modificamos las variables de formularios
```

```
foreach($_POST as $variable=>$valor){
  $_POST[$variable] = str_replace("","\'",$_POST[$variable]);
}
?>
```

Luego se debe incluir este código en la página que se intente realizar querys contra la base de datos, Ver figura Figura 5.25:

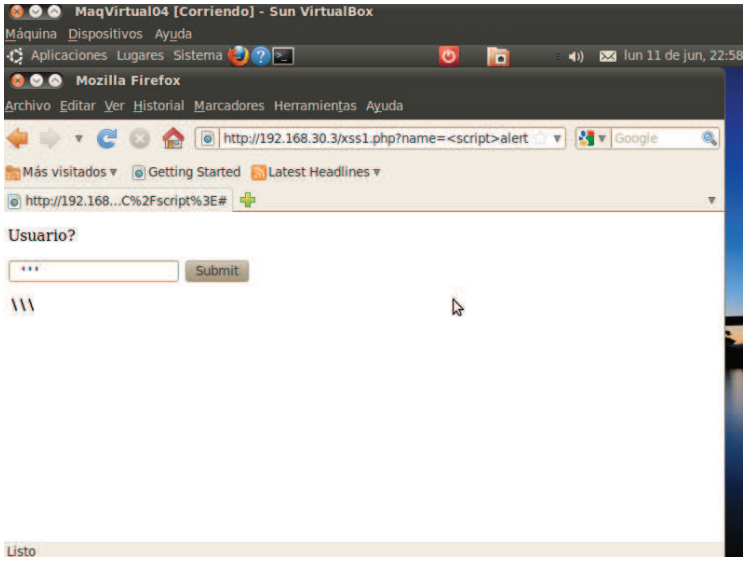


Figura 5.25. Resultados obtenidos, con el filtrado de "´" por "\"

5.2.9. Resumen General de Resultados

En la Tabla 5.5, se observa el resultado obtenido sobre la ejecución de los ataques de fuerza bruta, escaneo de puertos, suplantación y denegación de servicio, dentro del esquema descrito en la figura 3.5. Una vez implementado los mecanismos de prevención y mitigación de ataques tales como: la configuración de un firewall/ruteador interno de la red LAN, un demonio para detección de ataque de fuerza bruta y un segundo firewall externo (entre la red LAN e Internet), 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 5.5 Resumen sobre los ataques realizado en un escenario con mecanismos de prevención y mitigación de ataques.

CAPITULO IV

Conclusiones

- La utilización de máquinas virtuales para la realización de este proyecto fue un punto clave, ya que se pudo trabajar con varios equipos virtuales para establecer los diferentes escenarios de pruebas de ataques y mitigación a los mismos, sobre un mismo computador anfitrión; también se han podido establecer dos escenarios de pruebas utilizando distintos computadores anfitriones y máquinas virtuales invitadas. Estas opciones permitieron que pruebas de un laboratorio de computación sea más sencilla.

- Las plataformas de virtualización empleadas en el presente trabajo, VirtualBox y VMWare, permitieron implementar por completo la topología de pruebas propuesta en la figura 3.5. Sin embargo la herramienta de VMWare proporciona un entorno más amigable desde su instalación, configuración, creación y administración de las MVs.

- En la creación de los escenarios propuestos, algunas MV creadas tienen sistema operativo Windows XP y otras Linux (Ubuntu), a fin de dar una apariencia real a la topología propuesta, y que a pesar de haber encontrado herramientas de uso gratuito para Windows que permitieron ejecutar ataques a la red LAN, Linux no deja de ser uno de los sistemas operativos más destacados debido a su gran versatilidad y funcionalidad frente a otras soluciones. Adicionalmente otro punto importante que se puede destacar respecto a Linux es la gran cantidad de soporte que se encuentra disponible actualmente, la abundante información que se puede encontrar ya sea en libros o en el Internet, y la fácil ubicación y acceso a las comunidades existentes a nivel mundial que dan soporte para corregir y solucionar problemas referente al sistema Linux.

- Entre las herramientas de código abierto para enrutamiento basado en software más destacado se encuentra el paquete Quagga, debido a que su configuración en consola es muy parecida al realizado en los enrutadores comerciales como Cisco. Este ambiente de configuración permite que personas no familiarizadas con este tipo de equipos puedan hacerlo fácilmente, incluso si no se tiene conocimientos muy avanzados de Linux, por ser una aplicación fácil de instalar y configurar. Además por ser una herramienta de código abierto y de libre distribución constituye una alternativa de bajo costo y de buen funcionamiento ante escenarios de enrutamiento para el aprendizaje.

- La herramienta de enrutamiento Quagga puede complementarse con herramientas de igual manera de código abierto, como Iptables, para lograr mayor robustez en los sistemas contra ataques concretos a redes TCP/IP como denegación de servicios y suplantación de identidad.

- En las prácticas desarrolladas en el capítulo 4 se pudo analizar algunas de las actividades previas realizadas por los atacantes de redes TCP/IP para conseguir sus objetivos, como por ejemplo obtención de información de un sistema, descubrimiento de usuarios y exploración de puertos. Además se aprendió sobre cómo funcionan las técnicas de sniffing en redes TCP/IP para comprender el peligro que pueden acechar en la seguridad de una red local. Finalmente se realizó un estudio detallado de algunos ataques concretos contra redes TCP/IP como pueden ser los ataques de escaneo de puertos, fuerza bruta, suplantación de identidad y denegación de servicios, tanto en una red de área local como en una extendida.

- Para contrarrestar los ataques concretos a redes TCP/IP estudiados en el presente proyecto, se desarrolló un demonio en Shell script que detectó, controló y mitigó los ataques mencionados de manera automática y constante. Los resultados redujeron considerablemente las amenazas y vulnerabilidades de los ataques en redes en producción.

- En la evaluación de resultados, descritos en el capítulo 5, obtenidos durante las prácticas realizadas sobre los ataques a redes TCP/IP, en donde se consideró los mismos parámetros de medición y evaluación para las dos plataformas de

experimentación virtualizadas, no existiendo diferencias significativas en los resultados obtenidos.

- Al inicio del desarrollo del presente proyecto se utilizaron algunas herramientas para análisis de tráfico en una red TCP/IP, como es Abel y Cain, TcpDump y Ettercap. Sin embargo Wireshark, viene provista de innumerables funcionalidades gracias a las cuales se puede identificar y analizar múltiples problemas de red, no solo aquellos causados por malas configuraciones sino también por una gran variedad de ataques, externos e internos, por lo que es considerada una de las herramientas más utilizada para evaluar el flujo de datos en el esquema de redes TCP/IP propuestos, durante la evaluación de los ataques en el presente proyecto.

- El firewall construido con IPTables es una poderosa herramienta para filtrado de paquetes, denegando o aceptando ciertos tipos de paquetes, para el direccionamiento de éstos. Todo esto es factible mediante la correcta configuración de filtrado de paquetes a nivel de kernel, siendo para ello necesario y muy importante conocer la estructura de éstos y la manera en la que son transmitidos.

Recomendaciones

- En vista que el ataque escaneo de puertos del tipo TCP FIN, devuelve puertos que figuran como abiertos, aunque realmente estén cerrados o silenciosos. Se recomienda tener cuidado a la hora de usar esta técnica.

- 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 se recomienda fomentar trabajos de investigaciones referentes a seguridades en redes TCP/IP a fin de brindar mayor seguridad en las redes de datos.

Referencias Bibliográficas

Bibliografía Principal

[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.] Nmap, www.nmap.org. Ultima comprobación Octubre de 2010.

[5.] Comprobando contraseñas online: THC Hydra y Medusa, http://infow.wordpress.com/2010/01/02/comprobando-contrasenas-online-thc-hydra-y-medusa/

[6.] Jhon the Ripper 1.7.6., [Online:] www.openwall.com/jhon/

[7.] Ethercap, http://ettercap.sourceforge.net/. Ultima comprobación, 21 de octubre de 2010

[8.] Nemesis, http://nemesis.sourceforge.net/. Ultima comprobación, 20 de octubre de 2010.

[9.] Hping, http://www.hping.org/

[10.] Wireshark: http://www.wireshark.org/. Ultima comprobación, Jul. 2010.

[11.] 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.

[12.] 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

[13.] 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.

[14.] 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.

[15.] 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.

[16.] 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.

[17.] 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.

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

[19.] 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.
[20.] Hacker Prof, The ultimate Guide Network Security, by Lors Klander, Jamsa Press, Las Vegas 1997
[21.] 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.
[22.] 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.
[23.] 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.
[24.] C. Lee, C. Roedel, E. Silenock, "Detection and Characterization of Port Scan Attacks", [Online:] "http://cseweb.ucsd.edu/users/clbailey/PortScans.pdf
[25.] 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
[26.] 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
[27.] 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
[28.] J. Li, N. Li, X. Wang, and T. Yu. Denial of Service Attacks and Defenses in Decentralized Trust Management. In ACM CCS, 2006.
[29.] Jacobson, V., Leres, C., and McCanne, S. Tcpdump. Available at anonymous@ftp.ee.lbl.gov
[30.] Raúl Espinosa Soriano,10-11-2011, Instalación y configuración dePOSTFIX – SASL –TLS – DOVECOT –SQUIRRELMAIL -MAILMAN
[31.] Wireshark: http://www.wireshark.org/. Ultima comprobación, Jul. 2010.
[32.] Nmap, www.nmap.org. Ultima comprobación Octubre de 2010.
[33.] Nemesis, http://nemesis.sourceforge.net/. Ultima comprobación, 20 de octubre de 2010.

Bibliografía Secundaria

1. Redes informáticas: conceptos fundamentales: normas, arquitectura, modelo OSI, TCP/IP, Ethernet, Wi-FI...
2. MISCHA, schwartz, Redes de Telecomunicaciones Protocolos, modelado y análisis, 1994, 397
3. Implementación De Servidores Con GNU/Linux, Edición Julio 2008, Joel Barrios Dueñas, http://arpaneting.es
4. Pasarelas y Encaminadores, : Hurra Communications Espana S.L.U., http://www.docmirror.net/es/freebsd/books/handbook/network-routing.html
5. El Modelo TCP/IP (Parte II: El Protocolo IP), http://bitsnocturnos.wordpress.com/2010/01/15/el-modelo-tcpip-parte-ii-el-protocolo-ip/
6. Protocolo UDP, ttp://neo.lcc.uma.es/evirtual/cdd/tutorial/transporte/udp.html
7. Redes y Seguridad, Protocolo ARP,http://www.redesyseguridad.es/el-protocolo-arp/

8. El protocolo ICMP, http://neo.lcc.uma.es/evirtual/cdd/tutorial/red/icmp.html
9. Carracedo, J. (2004). Seguridad en redes telemáticas. McGraw Hill.
10. SOFTWARE E INGENIERIA DE SOFTWARE, Carlos Barra P.,http://www.revistamarina.cl/revismar/revistas/1998/1/barra.pdf
11. Diseño y Evaluación de un Sistema de Seguridad para la RDSI-BA, Tesis Doctoral, Jordi Forné Muñoz, 1997, http://tdx.cat/handle/10803/7045
12. Mecanismos para la detección de ataques e intrusiones, http://ocw.uoc.edu/computer-science-technology-and-multimedia/advanced-aspects-of-network-security/advanced-aspects-of-network-security/P06_M2107_01773.pdf
13. Fyodor ,Guía de referencia de Nmap (Página de manual), http://nmap.org/man/es/index.html
14. Hacking: VII Ataques por Fuerza Bruta, http://jBercero.com.
15. Ataques por fuerza bruta (BruteForce) III. , http://labs.dragonjar.org/laboratorios-hacking-tecnicas- fuerza-bruta-brute-force-iii
16. Tipos de Ataques eEn Internet, http://www.adslayuda.com/foro/adsl/vuestros-documentos/tipos-de-ataques-en-internet-t34936.html
17. Introducción a Sun RPC, Primera práctica, Martín López Nores,http://www-gris.det.uvigo.es/wiki/pub/Main/PracticasRO/msg-slides.pdf
18. Explicación de enlace de tres vías a través de TCP/IP , http://support.microsoft.com/kb/172983/es
19. J. Li, N. Li, X. Wang, and T. Yu. Denial of Service Attacks and Defenses in Decentralized Trust Management.In ACM CCS, 2006
20. Red Hat Enterprise Linux 4: Manual de referencia, http://web.mit.edu/rhel-doc/4/RH-DOCS/rhel-rg-es-4/ch-ssh.html
21. What is a sufficient encryption? http://penguinsecurity.net/wiki/index.php?title=What_is_a_sufficient_encryption
22. Como agregar tareas al Cron de Linux, http://www.guatewireless.org/os/linux/como-agregar-tareas-al-cron-de-linux/
23. ENVENENAMIENTO ARP, Seguridad en redes conmutadas, Sergio Valín Cabrera. http://ownz.despai.es/trabajo_arp.pdf
24. La importancia de los protocolos cifrados: Envenenamiento ARP mediante ettercap, http://www.sahw.com/wp/archivos/2010/05/31/la-importancia-de-los-protocolos-cifrados-envenenamiento-arp-mediante-ettercap/
25. Introducción al némesis, http://bad-robot.blogspot.com/2009/04/introduccion-al-nemesis.html
26. HPING tutorial by Philippe Bogaerts, Version 1.5 24-08-2003 http://www.radarhack.com/dir/papers/hping2_v1.5.pdf
27. Virtualización en GNU/Linux, Alberto Abián Belmonte,Madrid, julio 2007, http://www.whyfloss.com/pages/conference/static/editions/mad07/charla2.pdf
28. Taller 3 de sistemas operativos, VIRTUALIZACIÓN y MULTIHILOS, Ing. Ms. Jairo E. Márquez D.,http://es.scribd.com/doc/50398451/17/Tipos-de-virtualizacion
29. Hypervisor-Based Redundant Execution on a Single Physical Host, http://scholar.google.com/scholar?start=20&q=Hypervisor&hl=es&as_sdt=0
30. Máquinas virtuales, http://www.arcos.inf.uc3m.es/~folcina/pfc-html/node16.html
31. VMware, http://www.vmware.com/es/virtualization/why-virtualize.html
32. VMWare, http://www.vmware.com/es/virtualization/virtual-machine.html
33. MISCHA Schwartz, Redes de telecomunicaciones Potrocolos, modelado y análisis, año 1994, 62924
34. TANENBAUM Andrew, Redes de Ordenadores, segunda edición, año 1991

35. Revista Electrónica de Estudios Telemáticos, Volumen 4, Edición No 1, Año 2005, paginas 94,95
36. SEGURIDAD EN LA RED, http://bibing.us.es/proyectos/abreproy/11499/fichero/05+-+Seguridad+en+la+red+en+GNU-Linux.pdf
37. http://eisc.univalle.edu.co/materias/Administracion_De_Redes_Y_Servidores/material/IPTABLES_CORTO.pdf
38. VMware home page, [Online:] http://www.vmware.com virtualizacion de servidores de telefonia ip en gnu/linux, http://www.adminso.es/images/6/6d/Eugenio_cap1.pdf
39. La Biblia del Hacker, Edición 2009, María Teresa Jimero, Carlos Migues, Abel Matas, Justo Pérez.
40. Envenenamiento Arp, Seguridad en redes conmutadas, http://ownz.despai.es/trabajo_arp.pdf
41. Programación De Shell Scripts En Linux, http://juanin.bligoo.com/media/users/0/44513/files/3080/preliminares_shell.pdf
42. Crontab, http://usemoslinux.blogspot.com/2010/11/cron-crontab-explicados.html
43. Web Attacks, Mauren Alies, Sergio García, Fabián Molina, Juan Felipe Montoya, María Isabel Serrano,Marzo de 2002, www.acis.org.co/memorias/JornadasSeguridad/IIJNSI/weba.doc

Publicación de Artículos Técnicos

- W. Fuertes, P. Zapata, L. Ayala y M. Mejía, "Evaluación y Mitigación de Ataques Reales a Redes IP utilizando Tecnologías de Virtualización de Libre Distribución", Memorias del Tercer Congreso de Software Libre CONASOL-2010, realizado en la ciudad de Talara, Perú, entre el 1 y 3 de diciembre de 2010.

- W. Fuertes, P. Zapata, L. Ayala y M. Mejía, "Plataforma de Experimentación de Ataques Reales a Redes IP utilizando Tecnologías de Virtualización", Revista Técnica del Departamento de Ciencias de la Computación, ESPE, Decc-Report Tendencias en Computación, Vol. 1, Nro. 2-2010, Pg. 33-42.