UNIVERSIDAD NACIONAL ABIERTA Y A DISTANCIA. UNAD

ESCUELA DE CIENCIAS BÁSICAS, TECNOLOGÍA E INGENIERA
PROGRAMA: INGENIERÍA DE SISTEMAS
TESIS DE GRADO

EXTRACCIÓN DE DATOS PARA CLASIFICACIÓN Y FILTRADO DE IPS FALSAS EN ATAQUES DDOS

Yudy Angélica Ramírez Walteros
cód. 23561882

Director: Ing. (Msc) Carlos Alberto Amaya Tarazona

Duitama, Agosto de 2013

TABLA DE CONTENIDO

LISTA DE ANEXOS:

LISTA DE FIGURAS

LISTA DE TABLAS

1. INTRODUCCIÓN

El trabajo dado a continuación, se enmarca dentro del tema de seguridad y tratamiento de la información. Específicamente en uno de los ataque más comunes hoy en día que se perpetran tanto en redes locales como en la nube. Los ataques de denegación de servicio. Para el presente proyecto se ha tipificado un ataque distribuido tipo (DDoS) que a diferencia de los ataques DoS, llegan a tener mayor efectividad y son más comunes.

Dada la diferentes presentaciones de esos ataques, se ha escogido solo una variante del mismo: en el que se afecte el tratamiento de la pilas TCP, el llamado (SYN Flooding).

La estrategia usada para propender por los objetivos del proyecto, inicia con un simple escenario de pruebas (en el que con solo dos máquinas) un atacante y otra como víctima, se pueden conseguir los datos de la inundación masiva por IPs falseadas.

Posteriormente se hace uso de librerías que permitan la captura del tráfico para su posterior análisis.

El proyecto pretende en un escenario de pruebas real, aplicar en primera instancia una arquitectura en aplicaciones que detecte y monitoree estos ataques y posteriormente mediante ataques inducidos, filtrar, exportar, identificar y clasificar la procedencia de los mismos.

Las herramientas que se aplicaron en este proyecto, son típicas y comunes en la gestión y administración de redes. Parte del trabajo de ingeniería, es poderlas combinar, darles un uso proactivo, dinámico y crear un conjunto de trabajo acorde a un análisis previo al tipo de vulnerabilidad que se presentan en el nivel de transporte. Se genera un aporte tanto temático, procedimental, de implementación y de gestión cuando se trata la Administración de Redes y Telecomunicaciones; temática muy amplia y enriquecedora para ahondar en trabajos futuros que aporten a la temática.

2. TEMA

La temática del proyecto cubre los aspectos de telecomunicaciones, redes y transmisión de datos. Con temáticas en seguridad de la información, sistemas operativos, y software de aplicaciones específicas para monitoreo y gestión de datos. En el campo de la ingeniería de sistemas en aspectos básicos de programación, se abarcan temáticas básicas en códigos scripts y ejecución de comandos desde consola para plataformas de red.

3. JUSTIFICACIÓN

Dada las constantes amenazas y diferentes vulnerabilidades presentes en los diversos sistemas de comunicación e infraestructuras tecnológica, en las que cualquier dispositivo ya sea móvil, PC o estación de trabajo, que implemente un protocolo de comunicación como TCP, puede presentar estas amenazas y dada la naturaleza vulnerable y comprobada de la comunicación que se presenta en las conversaciones del protocolo, se ha recreado un ataque de denegación de servicio.

Además de poder ver el comportamiento del ataque en una estructura de red básica, se pretende ver en gran medida y evidenciar como las inundaciones de servicios son causadas por gran cantidad de IPs falsas y que no son filtradas o asociadas a un sistema de defensa típico como los que adopta los IDS (sistemas de detección de intrusos). Estos sistemas de detección de ataques DDoS establecen técnicas y herramientas de identificación del ataque y de bloqueo o prevención. Independiente del tipo de ataque o denegación.

El alcance del proyecto lleva a identificar, tipificar y caracterizar el ataque para efectos de detección y extracción de IPs masivas falsas. Dada situaciones generalizadas en que estos IDS son efectivos en las barreras y firewalls que aplican a los ataques pero sin clasificar o dar servicios a peticiones reales de estaciones de trabajo que son víctimas de la inundación.

Como trabajo y como se evidencia en el estado del arte, son muchos los autores que tratan la temática, y son muchas las soluciones que a nivel de hardware aplican identifican el origen de la vulnerabilidad, los host afectados y los servicios comprometidos deteniendo el servicio o cerrando puertos o sesiones de usuarios en general pero sin clasificar o permitir el servicio a IPS que verdaderamente están replicando o solicitando el servicio.

Una vez identificadas estas peticiones masivas y sus orígenes, se pretende hacer uso y manipular las primitivas y modificadores que las librerías de libpcap puedan ofrecer y poderlas llevar a un formato de datos compatible.

No hace parte de la investigación actual la fase o proyección de desarrollo de aplicaciones o de aplicación de sniffers con datos tipo pcap filtrados, que pueda hacer uso de estos archivos filtrados y generar una aplicación o sistema que permita redireccionar el servicio afectado.

3.1. DEFINICIÓN DEL PROBLEMA

Toda la problemática está definida en la capa de transporte del nivel 3 del modelo OSI. En esta capa podemos encontrar problemas de autenticación, de integridad y de confidencialidad. Para el presente proyecto se caracteriza un ataque de denegación de servicio, debido a las deficiencias que presentan los encabezados y las tramas que forman los paquetes TCP y UDP.

La solución al problema está globalizada y no específica para este tipo de ataques de denegación. Los IDS carecen de procedimientos específicos caros que permitan clasificar y bloquear inundaciones falsas. Los firewalls suele detener ataques de forma global y las configuraciones de filtrado no suelen ser tan efectivas. No se trata de crear un IDS o de modificar arquitecturas completas de seguridad en redes corporativas y locales. El problema estaría definido para poder dar un aporte y el primer paso al tratamiento de la información capturada con miras a establecer parámetros de inclusión en los mecanismos de defensa y que con opciones de código abierto, a futuro se puedan incluir en los núcleos de las aplicaciones para que se filtren peticiones falsas.

Cambiar un protocolo de comunicación como TCP desde la pila, lleva a otras instancias mayores el problema, dada la fragilidad en la autenticación de los equipos involucrados en una sección.

En el desarrollo del proyecto y para efectos de pruebas y de recrearlas vulnerabilidad del protocolo, no se han aplicado IDS ni Firewalls específicos. Se han establecido las comunicaciones a instancias nativas que permitan detectar todos los parámetros vulnerables del proceso. Los paquetes que viajan por la red, son interceptados para poder leerlos y ver su estructura.

La justificación del problema también está soportada por el hecho de tratar a fondo los ataques de denegación de servicio tipo TCP / SYNC Flooding con fines de establecer alarmas que minimicen su acción y prevengan su ejecución. Los estudios

que se han abordado en este tema son amplios pero han sido enfocados en cada caso a variables dependientes como el tipo de Topología de red, tipos de datos involucrados, selección de herramientas específicas y combinación de técnicas de escaneo y detección.

4. OBJETIVOS

4.1 OBJETIVO GENERAL

- Extraer y filtrar datos en flujos de tráfico afectado por ataques DDoS para identificar IPS falsas en peticiones masivas.

4.2. OBJETIVOS ESPECÍFICOS

- Recrear en un escenario real las denegaciones de servicio identificando la vulnerabilidad.
- Realizar un estudio de las librerías y primitivas necesarias que permitan extraer la información del tráfico de red cuando se presentan denegaciones de servicio.
- Partir del esquema Three Way Handshake para exportar los datos a un formato de archivo pcap filtrado y ordenado en los campos necesarios para la identificación de IPs falsas y el tipo de servicio afectado.
- Formular una propuesta de aplicación para el uso de los formatos pcap en la implementación o desarrollo de aplicaciones y sistemas de detección de ataques DDoS que detecte IPs falseadas

5. ESTADO DEL ARTE

Específicamente el estudio de este tipo de vulnerabilidades tiene soluciones para filtrado y detección con un software llamado JFFNMS desarrollado por Javier Szyszlican, en el año 2002, para monitoreo de dispositivos, que integra varias utilidades que interrogan y capturan los datos de los dispositivos.

Al igual que esta herramienta, han basado sus técnicas en el uso de nmap para verificar el dialogo correcto. También hacen uso de tcpdump y las herramientas que indican para verificar y comprender el dialogo completo.

Otros trabajos han llevado a identificar el origen del ataque, la IP que lanza la primera inundación y que sirve de puente para que otros actúen en nombre de él, generando IPs al azar en la que el cliente estaba mandando el paquete ACK en el tercer paso de la negociación en 3 pasos.

Posteriormente otras técnicas encontradas fueron el manejo de memoria, socket en el kernel Linux, que permite bloquear cierto número de peticiones cuando excedan un tipo definido en el kernel. Y siguiendo con las técnicas que se suelen usar, está hacer uso del framework NetFilter en el núcleo de Linux que es el encargado de manejar los paquetes que envían y llegan a nuestro sistema. La herramientas más conocida para administrarlo es IpTables, y que actúa a amanera de firewall, permitiendo crear reglas de filtrado de tráfico en nuestro sistema.

Y para ambientes Windows, es muy común encontrar aplicaciones que pretender dar solución a este tipo de problemas como “Arno's Firewall”[1]

Otras técnicas y estudios sobre todo en el campo de los IDS son el uso de “Snort + BASE” como un sniffer que implementa un motor de detección de ataques que funciona mediante reglas que nosotros mismos podemos definir, y alerta ante cualquier anomalía detectada por estas reglas. Esta fue la base de la que se partió para el desarrollo del proyecto, dado la flexibilidad, potencia y practicidad para aplicar y crear reglas de filtrado. Complemento a ello, muchas soluciones suelen usar BASE

[1] Disponible en Internet desde .<http://rocky.eld.leidenuniv.nl/joomla/>

(Basic Analysis and Security Engine), es una aplicación Web para la monitorización de Snort. Se necesita instalar un servidor apache con soporte PHP y SQL.

Dentro de la temática investigada, se encontró que existen aplicaciones que pueden perpetrar ataques con fines no malignos, para estudios y técnicas de detección (a amanera de laboratorio). Una de ellas es Slowloris que permite lanzar ataques de manera efectiva rápida.

Y amanera de protección se encontró dentro de los tanto IDS que hay un "HIDS (Host IDS)" que protege contra un único Servidor, PC o host. Monitorizan gran cantidad de eventos, analizando actividades con una gran precisión, determinando de esta manera qué procesos y usuarios se involucran en una determinada acción. Recaudan información del sistema como ficheros, logs, recursos, etc, para su posterior análisis en busca de posibles incidencias. Todo ello en modo local, dentro del propio sistema. Fueron los primeros IDS en desarrollarlos la industria de la seguridad informática.

Finalmente, cada resultado de una investigación, encadena otro que propone un complemento o mejora para resolver el problema, y cada autor le da camino a su investigación de acuerdo a la necesidad urgente o escenario propio de cada situación. Pero es en esta variedad de escenarios no definen claramente un esquema o sistema de filtrado de direcciones IP falsas por peticiones masivas o fraudulentas.

Puntualmente se encuentran los avances de CISCO en la fabricación de dispositivos activos de red capa 2 y 3 del modelo OSI. Una de las estrategias es la referencia en un papel *"Previniendo Ataques DDoS con Redes Cisco que Se Autodefienden"*[2], en la que defiende una estrategia de detección por capas, separando protocolos, administrando puertos y servicios, pero no ponen en manifiesto o no descubren la técnica usada en código o metodología a nivel de ingeniería de los procesos usados. Además las filtraciones son para todas las IPs involucradas sin hacer selección de las mismas.

Otras estrategias enmarcan soluciones como las que documenta GIL, Gutiérrez R "*Evasión de Sistemas de Detección de Intrusos y Cómo atacar sin ser detectado*

[2] *Todo el contenido tiene Derechos de Autor © 1992–2005 Cisco Systems, Inc. Todos los derechos reservados. Notificaciones*

Tácticas de defensa contra ataques DoS"[3] en las que trata los "Fws y routers" que implementan técnicas como:

Egress filtering: Filtrar entrada de paquetes con direcciones no enrutables y salida de paquetes con direcciones no de la organización (evita la acción de troyanos al no dejar salir sus paquetes) o no enrutables.

Protección contra ataques SYN flood en los fw's: Algunos protegen guardando el estado de las conexiones y existen parches especiales para DoS.

Bloqueo de broadcasts de IP en los routers de filtrado para evitar que la red se utilice como amplificador. Denegar todos los accesos a servicios no autorizados por políticas de seguridad (bloquear los puertos).

Otros temas como los que postula ARGENTE, Jorge en su libro "*Sistema Híbrido para la detección de código malicioso"* [4] formula como los IDS usan técnicas para prevenir intrusiones en las que utilizan:

- ✓ Detectar cambios en ficheros del sistema Tripwire
- ✓ Encontrar huellas. Comandos last, netstat, lastcomm
- ✓ Detectar sniffers. Antisniff. Detecta nics en modo promiscuo mediante arp.
- ✓ Ver usuarios activos y procesos
- ✓ Monitor de rendimiento para detectar ataques en tiempo real.
- ✓ Revisar a menudo las listas del CERT y el SANS.
- ✓ Filtrado de ciertos paquetes como ICMP en un momento dado
- ✓ Utilización de logs.

Todas estas técnicas son ágiles y efectivas en la detección y bloqueo. Se ha referenciado estos estudios para poder comprender como se filtrarían las direcciones IP falsas.

---

[3] GIL, Gutierrez. Evasión de Sistemas de Detección de Intrusos .p20.
4 ARGENTE, Jorge. Sistema Híbrido para la detección de código malicioso. P 262.

6. MARCO TEÓRICO

6.1. VULNERABILIDAD EN LA CAPA DE TRANSPORTE

La temática nos lleva a identificar vulnerabilidades en la capa de transporte.

Vulnerabilidades de la capa de transporte**:**[5]"La capa de transporte transmite información TCP o UDP sobre datagramas IP. En esta capa podamos encontrar problemas de autenticación, de integridad y de confidencialidad. Algunos de los ataques más conocidos en esta capa son las **denegaciones de servicio** debidas a protocolos de transporte.

En cuanto a los mecanismos de seguridad incorporados en el diseño del protocolo de TCP (como las negociaciones involucradas en el establecimiento de una sesión TCP), existe una serie de ataques que aprovechan ciertas deficiencias en su diseño. Una de las vulnerabilidades más graves contra estos mecanismos de control puede admitir la posibilidad de interceptación de sesiones TCP establecidas, con el objetivo de secuestrarlas y dirigirlas a otros equipos con fines deshonestos.

Estos ataques de secuestro se aprovechan de la poca exigencia en el protocolo de intercambio de TCP respecto a la autenticación de los equipos involucrados en una sesión. Así, sin un usuario hostil puede observar los intercambios de información utilizados durante el inicio de la sesión y es capaz de interceptar con éxito una conexión en marcha con todos los parámetros de autenticación configurados adecuadamente, podrá secuestrar la sesión."

6.2. SISTEMAS DE DETECCIÓN DE INTRUSOS

Un IDS es una herramienta de seguridad que intenta detectar o monitorizar los eventos ocurridos en un determinado sistema informático en busca de intentos de comprometer la seguridad de dicho sistema. También suelen aspirar detectar o

[5] *JOANCOMARTI, Jordi Herrera. Aspectos Avanzados de seguridad en redes. Seg edición. Barcelona España 2005. 62p*

monitorizar los eventos ocurridos en un determinado sistema informático buscando comprometer la seguridad de dicho sistema.

6.2.1. IDS y Snort: Los IDS buscan patrones previamente definidos que impliquen cualquier tipo de actividad sospechosa o maliciosa sobre nuestra red o host.

Dentro de sus funciones aplican.

- ✓ Los IDS aportan a nuestra seguridad una capacidad de prevención y de alerta anticipada ante cualquier actividad sospechosa. No están diseñados para detener un ataque, aunque sí pueden generar ciertos tipos de respuesta ante éstos.

- ✓ Los IDS: aumentan la seguridad de nuestro sistema, vigilan el tráfico de nuestra red, examinan los paquetes analizándolos en busca de datos sospechosos y detectan las primeras fases de cualquier ataque como pueden ser el análisis de nuestra red, barrido de puertos.

6.3. TIPOS DE IDS

6.3.1. Hids (Host Ids): Protege contra un único Servidor, PC o host. Monitorizan gran cantidad de eventos, analizando actividades con una gran precisión, determinando de esta manera qué procesos y usuarios se involucran en una determinada acción. Obtienen información del sistema como ficheros, logs, recursos, etc, para su posterior análisis en busca de posibles incidencias.

6.3.2. Nids (Net Ids): Protege un sistema basado en red. Actúan sobre una red capturando y analizando paquetes de red, es decir, son sniffers del tráfico de red. Luego analizan los paquetes capturados, buscando patrones que supongan algún tipo de ataque. Bien ubicados, pueden analizar grandes redes y su impacto en el tráfico suele ser pequeño.

Actúan mediante la utilización de un dispositivo de red configurado en modo promiscuo (analizan, “ven” todos los paquetes que circulan por un segmento de red aunque estos nos vayan dirigidos a un determinado equipo). Analizan el tráfico de

red, normalmente, en tiempo real. No sólo trabajan a nivel TCP/IP, también lo pueden hacer a nivel de aplicación.

6.4. ARQUITECTURA DE UN IDS

Normalmente formada por:

- ✓ La fuente de recogida de datos. Estas fuentes pueden ser un log, dispositivo de red, o como en el caso de los IDS basados en host, el propio sistema.

- ✓ Reglas que contienen los datos y patrones para detectar anomalías de seguridad en el sistema.

- ✓ Filtros que comparan los datos sniffers de la red o de logs con los patrones almacenados en las reglas.

- ✓ Detectores de eventos anormales en el tráfico de red.

- ✓ Dispositivo generador de informes y alarmas. En algunos casos con la sofisticación suficiente como para enviar alertas vía mail, o SMS.

6.4.1. Snort:

Es un IDS o Sistema de detección de intrusiones basado en red (NIDS). Implementa un motor de detección de ataques y barrido de puertos que permite registrar, alertar y responder ante cualquier anomalía previamente definida como patrones que intenten aprovecharse de alguna vulnerabilidad; analizan protocolos, conocidos, etc. Todo esto en tiempo real.

6.5. ARP EN LA DETECCIÓN DE REDES

En una red Ethernet cuando se desea enviar un paquete IP entre dos hosts conectados las únicas direcciones válidas son las MAC y lo que circula son tramas Ethernet. Es importante definir la función de ARP:

“En comunicaciones, ARP (del inglés Address Resolution Protocol o, en español, Protocolo de resolución de direcciones) es un protocolo de la capa de enlace de datos responsable de encontrar la dirección hardware (Ethernet MAC) que

corresponde a una determinada dirección IP. Para ello se envía un paquete (ARP request) a la dirección de difusión de la red (broadcast (MAC = FF FF FF FF FF FF)) que contiene la dirección IP por la que se pregunta, y se espera a que esa máquina (u otra) responda (ARP reply) con la dirección Ethernet que le corresponde. Cada máquina mantiene una caché con las direcciones traducidas para reducir el retardo y la carga. ARP permite a la dirección de Internet ser independiente de la dirección Ethernet, pero esto sólo funciona si todas las máquinas lo soportan."[6]

6.6. LIBRERIAS TCPDUMP Y WINDUMP

Son herramientas de administración de seguridad de redes de datos y de seguridad de ellas mismas, usan librerías comunes para analizar tanto tráfico entrante como saliente a través de los servicios que se proveen o instalen en la red. Detectan problemas y tráfico no esperado, en las puertas traseras (puertos) y sobre todo para escaneo de intrusos o cualquier otra intrusión.

Existen muchas herramientas que pueden ser muy útiles dependiendo del S.O. y tipo de red por ejemplo. Una de estas herramientas es un sniffer de red, basada en la librería de captura de paquetes (pcap) y que además funciona en plataformas tanto windows como GNU/Linux-UNIX es TCPDump (GNU/Linux) / Windump (Windows), ésta última hace uso de la librería Winpcap. Estas dos librerías son usadas por otras herramientas como Ethereal o Snort, e incluyen un lenguaje de filtros común para todos. Quizás Windump/TCPDump no sea la herramienta perfecta atendiendo a la interpretación fácil de los datos reportados, pero sí que es de las mejores en cuanto a su potencia y cantidad de datos de que nos provee.

6.7. PCAP

"El pcap es un interfaz de programación (actúa como un API) para captura de paquetes. La implementación del pcap para sistemas basados en Unix se conoce como libpcap; el port para Windows del libpcap recibe el nombre de WinPcap. **Libpcap** proporciona funciones para la captura de paquetes a nivel de usuario, utilizada en la monitorización de redes de bajo nivel."[7]

---

[6] Tomado desde internet: Con acceso <http://es.wikipedia.org/wiki/Address_Resolution_Protocol>

7 Sistema de detección de alarmas para ataques DDoS, UOC Catalunya, Amaya, C. 2, 127 p.

El programa utilizado para capturar y descargar los paquetes para futuros análisis es **tcpdump**[8]que se porta como un sniffer útil para los objetivos de este proyecto. La estrategia a usar con esta herramienta está basada en el análisis y descripción de los contenidos de los paquetes en una interfaz de red que coincidan con aluna expresión lógica determinada por la lectura de las cabeceras TCP.

6.7.1. Descripción De Logs Pcap: Los logs pcap son archivos que contienen una serie de registros del tráfico de red capturado, estos logs son usados como datos de entrada y guardan información tales como la fecha en la que se envió un paquete, direcciones MAC, direcciones IP, puertos, protocolos, data del paquete, entre otros.

Son muchos los programas que manejan logs pcap. Entre ellos son típicos los analizadores de protocolos como Wireshark que hacen uso de librerías diseñadas para cada entorno de programación; hay librerías para Java, C++, Visual Basic, entre otros. La ventaja de estos archivos es que permiten ser convertidos a diferentes formatos como el binario y texto plano para su posterior lectura y análisis.

6.8. XPLICO

Es una herramienta forense que nos permite revisar un extracto de los paquetes capturados del tráfico de red que están a un nivel muy bajo y nos muestra el resultado en un nivel más alto, capaz de armar y acomodar los paquetes capturados ya sean imágenes, páginas web, sonidos, vídeos incluso voz, dependiendo del protocolo. Esta herramienta es Open Source y al igual que la mayoría de proyectos de este tipo, tiene una comunidad grande que apoya y actualiza el proyecto constantemente.

El uso dado de la librería en el proyecto está dado al para interpretar las capturas pcap realizadas en Tshark, Wireshark, Windump, TCPDump, Con Xplico es posible dar apertura a un archivo pcap o realizar una captura in-situ (live), al momento y después “destripar” y analizar los datos. También es usado para análisis forense y en algunas técnicas de detección y corrección de errores.

8 Disponible en internet <http://www.tcpdump.org/tcpdump_man.html> Con acceso. November 19, 2009

Los protocolos y servicios que acepta esta herramienta son: HTTP, SIP, IMAP, POP, SMTP, TCP, UDP, IPv6, Facebook, MSN, RTP, IRC, Paltalk, entre otros permitiendo el acceso simultáneo a la aplicación por varios usuarios.

Está escrita en C, Python, PHP, JavaScript. La disponibilidad del paquete está en el sitio web del proyecto.[9]

[9] Disponible desde internet: http://www.xplico.org/status

7. METODOLOGÍA

7.1. DENEGACIÓN DE SERVICIO (DOS)

Es importante diferenciar dos tipos de ataques de denegación del servicio si se tiene en cuenta que solo en uno se va a centrar este proyecto, puede actuar teniendo un único origen o varios dependiendo la técnica usada para ejecutarlo:

- Ataques de denegación de servicio simples (Deny Of Service). Este tipo de ataques se caracterizan por tener un único origen desde el cual se realiza la denegación del servicio.

- Ataques de denegación de servicio distribuido (Distributed DOS o DDOS). En este tipo de ataques se utilizan varias fuentes coordinadas que pueden realizar un ataque progresivo, rotatorio o total.

Se define la denegación de servicio (Deny Of Service, DOS) como la imposibilidad de acceso a un recurso o servicio por parte de un usuario legítimo. De esta forma se puede definir un ataque de denegación de servicio (DOS Attack) como: [10]"La apropiación exclusiva de un recurso o servicio con la intención de evitar cualquier acceso de terceros. También se incluyen en esta definición los ataques destinados a colapsar un recurso o sistema con la intención de destruir el servicio o recurso".

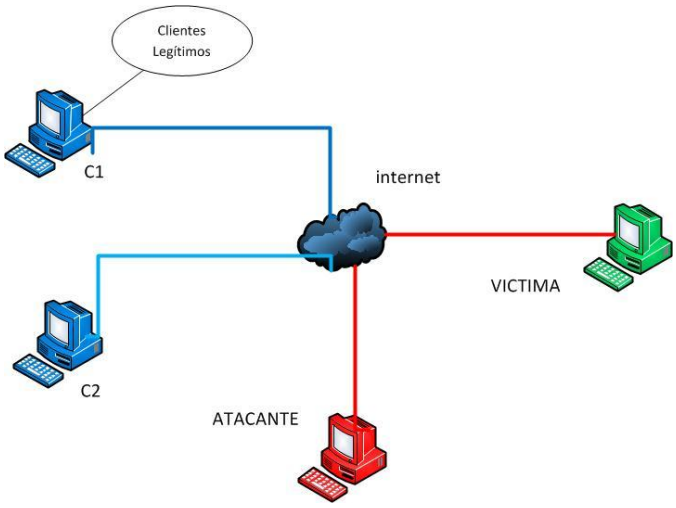
Siendo más concreto y enfocando los ataques a redes IP como el que se pretende llevar a cabo en el desarrollo del proyecto, se encuentra una definición con argumentos explicativos concretos que definen los ataques de denegación del servicio como: [11]"la consecución total o parcial (temporal o totalmente) del cese en la prestación de servicio de un ordenador conectado a Internet."

---

10 Search Software Quality. [Web en línea]. Disponible desde Internet en: <http://searchsoftwarequality.techtarget.com/sDefinition/0,,sid92_gci213591,00.html>

11 CERT® Advisory CA-1999-17 Denial-of-Service Tools. [Web en línea]. Disponible desde Internet en: <http://www.cert.org/advisories/CA-1999-17.html>

Figura 1. Escenario de un Ataque de Denegación de servicio (DoS)



Fuente: <El Autor> Adaptado < MIRKOVIC.J. D-WARD Source-End Defense Against Distributed Denial-of-Service Attacks.2003. p 38.>

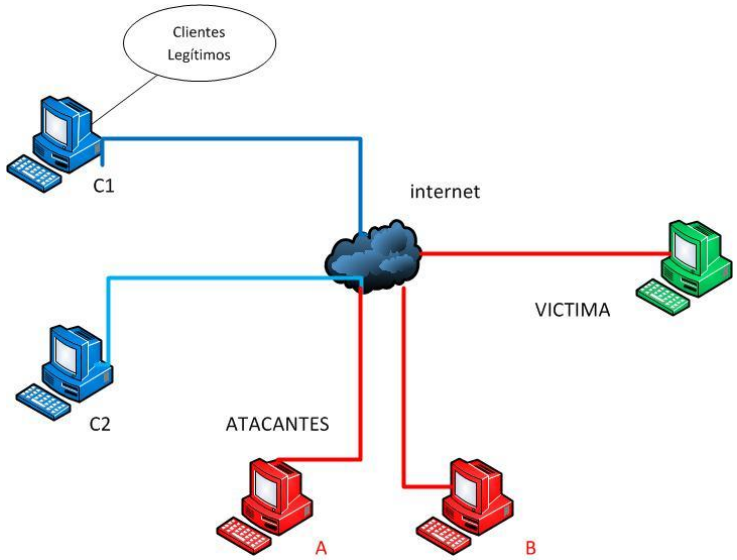
En la *Figura 1*, se explica una denegación del servicio típica. Que se produce cuando la víctima (V) recibe un malicioso flujo de paquetes obligándola a evitar recibir algún recurso o servicio de los clientes legítimos C1, y C2. Una característica de estos ataques es que los atacantes rara vez utilizan sus propias máquinas para llevar a cabo los ataques por lo que la máquina A se convierte en un agente o participante involuntario.

7.2. DENEGACIÓN DE SERVICIO DISTRIBUIDO: DDOS

Se han definido los ataques de denegación de servicio distribuido (DDOS) como [12]"un ataque de denegación de servicio (DOS) dónde existen múltiples focos distribuidos y sincronizados que focalizan su ataque en un mismo destino".

[12] [Web en línea].Disponible desde Internet en: <http://searchsecurity.techtarget.com/sDefinition/0,,sid14_gci557336,00.html>

Figura 2. Escenario de un Ataque de Denegación de servicio Distribuido (DDoS)



Fuente: <El Autor> Adaptado < MIRKOVIC.J. D-WARD Source-End Defense Against Distributed Denial-of-Service Attacks.2003. p 40.>

La *Figura 2*, muestra una negociación distribuida simple en el que varias máquinas (A,B) atacantes detienen un servicio al enviar paquetes de datos con flujos maliciosos a la víctima (V) negando el servicio a los clientes legítimos C1 y C2.

7.3. FUENTES DE ORIGEN DE LOS ATAQUES DOS / DDOS

El siguiente paso es identificar el origen de los ataques, que son muchos y que incluso existe herramientas y aplicaciones para perpetrarlos.

Los ataques de denegación de servicio suelen tener varios orígenes, lo que complica la posibilidad de mantener un control efectivo sobre todos los recursos o servicios ofrecidos. Una detección temprana con un sistema de alertas como el que se ha propuesto en este proyecto podrían minimizar el riesgo. No obstante, los distintos orígenes o fuentes pueden provenir principalmente de dos fuentes:

[13]"**Usuarios legítimos o internos:** Este grupo se subdivide en aquellos usuarios poco cuidadosos que colapsan el sistema o servicio inconscientemente (por ejemplo la persona que llena el disco duro del sistema bajando archivos de música), usuarios malintencionados (aquellos que aprovechan su acceso para causar problemas de forma premeditada) y usuarios ladrones que utilizan el acceso de un usuario legítimo (Ya sea robándolo del usuario legítimo o aprovechándose de la confianza de este).

Agentes externos: Este grupo hace referencia a los no usuarios del sistema. De esta forma se consigue acceso al recurso o servicio sin necesidad de ser un usuario legítimo (un sistema que no controle la autenticación de usuarios, un sistema que presente un "bug" conocido...). En este grupo usualmente se falsea la dirección de origen (faked/spoofed IP) con el propósito de evitar el origen real del ataque."

Antes de diferenciar algunos ataques y principalmente el que se va a tratar en esta tesis, es conveniente identificar como actúan las fases previas a un ataque.

7.4. PLATAFORMAS AFECTADAS

El otro paso es el de detectar que plataformas afecta:

El enfoque que abarca esta investigación comprende escenarios en los que se presenten ataques de servicio en redes IP. La trascendencia de estos ataques está dada por su independencia de plataformas en las que pueden perpetrarse y hacer efectivos estos ataques.

7.5. CARACTERIZACIÓN DE LOS ATAQUES DOS

7.5.1. Uso de IP Source Spoofing: En la que los atacantes utilizan la suplantación o falseo de una dirección IP durante el ataque emitiendo información falsa desde el mismo origen del ataque ocultando información de la cabecera de los paquetes IP. Una de las características de este ataque es que las máquinas "Agente" o las que envían el ataque, son difíciles de rastrear, incluso con la información

[13] VERDEJO. G. Seguridad en Redes IP. Universidad Autónoma de Barcelona. Barcelona Sep. 2003. p, 36.

que capturan y la dirección oculta de donde se realiza el ataque, se pueden realizar ataques futuros.

La otra característica que tiene esta falsificación de direcciones IP, es la que le permite a los atacantes hacer una “reflexión del ataque” que consiste que las solicitudes a un determinado servicio que realiza el atacante en nombre de la víctima, genera muchas respuestas de un tamaño pequeño. A medida que realice más solicitudes, estas obtendrán más respuestas que son recibidas por la víctima. Puede extender su ataque enviando solicitudes a servidores públicos que otorgan mayor número de respuestas.

7.5.2. Un Gran Número De Máquinas Agente: Acá se complica el ataque, porque incluso si un sistema de detección de intrusos (IDS) detecta las peticiones y el origen de las mismas, por el gran volumen de atacantes y la diversidad de paquetes enviados, pueden evitar que las repuestas lleguen al sistema de detección y tomar mucho tiempo en procesarse dando oportunidad a usar la “reflexión” explicada anteriormente para retomar el ataque.

7.5.3. Similitud De Tráfico Legítimo: Cualquier tipo de tráfico puede ser utilizado para ejecutar un ataque de denegación de servicio. Algunos tipos de tráfico requieren un mayor volumen de paquetes para que tengan mayor éxito que otros. El tráfico legítimo fácilmente puede ser confundido con el tráfico malicioso. Un sistema de defensa estaría basado en la obtención de un volumen de datos estadísticos para su análisis de la semántica o estructura de las transacciones habituales y compararlas con las que se tengan duda. Es un principio básico de muchos sistemas de detección de intrusos (IDS). Sin ser el objetivo de este proyecto, crear un (IDS), se abre una temática en cuanto a los aspectos de diseño de la red Ethernet e internet para prevenir los ataques DoS. Aspectos a tener en cuenta y que marcan la pauta para proyectar estrategias y mecanismos de defensa. Algunas de ellas son analizadas así.

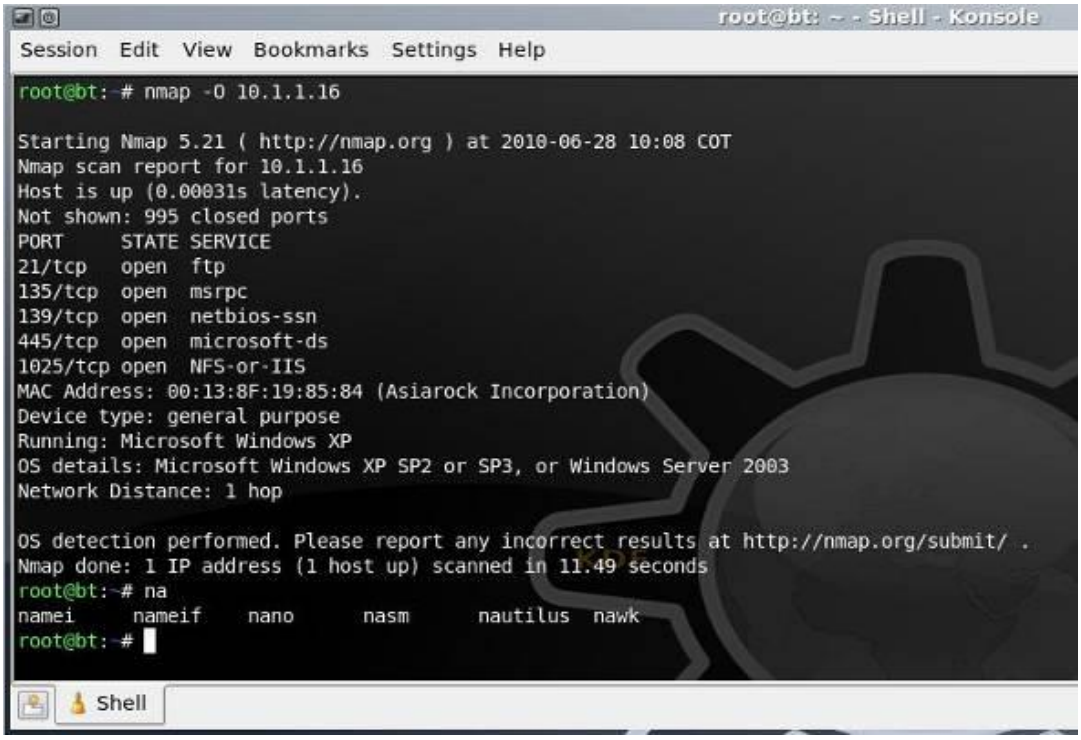
7.6. AFECTACIÓN DE LA COMUNICACIÓN EN TRES VÍAS

Para efectos de poder identificar en qué momento la comunicación en tres vías es afectada por las inundaciones masivas y peticiones simultáneas y evidenciar las direcciones falseadas, se analiza el estado del protocolo en su vulnerabilidad.

Se trata entonces de obtener información de un sistema concreto y de alguna vulnerabilidad específica. Esto se hace obteniendo su huella identificativa respecto de la pila TCP/IP de los equipos atacados. Esta técnica es la que se conoce como Fingerprinting y la información que puede brindar esta técnica dentro de las muchas opciones que tiene de descubrir datos de la víctima son:

- Permitir descubrir de forma muy fiable el sistema operativo que se ejecuta en la maquina analizada.
- Identificar el tipo de servidor y la versión en la que se soporta el servicio
- El tipo de servicio que se ejecuta y la versión.

Figura 6. Fingerprinting



Fuente: <El Autor>

De esta forma, un ataque DoS puede identificar qué tipo de servicio es vulnerable para poder ser explotado y atacado. La mayor parte de las técnicas para obtener huellas identificativas se basan en la información de la pila TCP/IP que puede obtenerse a partir de los mecanismos de control del intercambio de tres pasos del protocolo TCP/IP.

Un análisis previo como el que sigue de esta Pila TCP/IP permite ver como es de vulnerable esta implementación:

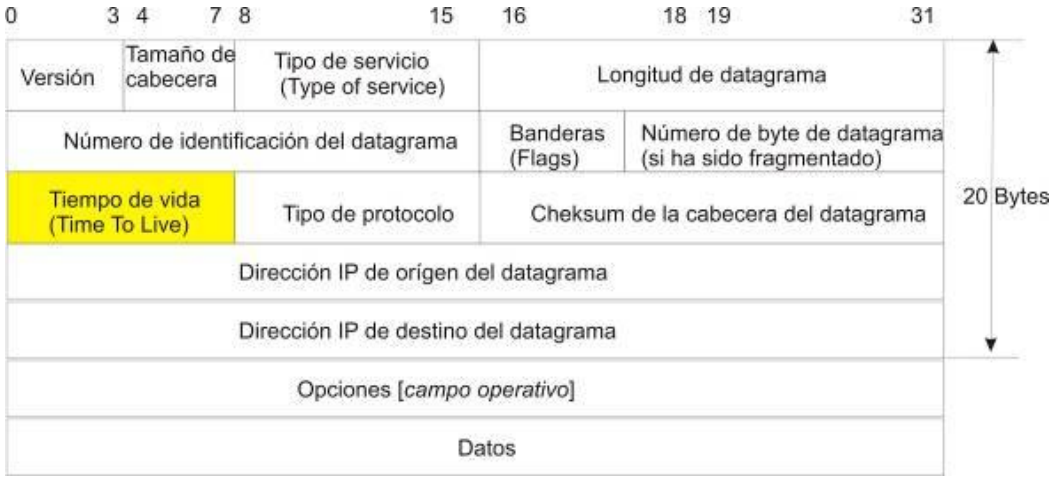
"En los ataques de Denegación de Servicio (DoS), que usan Inundación de paquetes tipo SYN, como el que se trabaja en este proyecto (SYN-Flood), las técnicas de Fingerprinting se hacen presentes. Determinados sistemas presenta cierta resistencia al envío masivo de este tipo de paquetes TCP, de tal modo que es posible determinar la versión del sistema remoto. En determinadas ocasiones algunos sistemas sólo aceptan **ocho (8) conexiones** de la misma fuente. Una prueba para verificar este comportamiento sería enviando 8 paquetes de estas características y verificar si se puede establecer una conexión al servicio atacado."

Una de las muchas técnicas que existen para levantar este tipo de información con respecto al levantamiento de una huella identificativa, es el uso de **"Escuchas de Red"** o escáneres de puertos como la herramienta nmap, con miras a establecer un posible ataque. En la *Figura 6* se muestra un escaneo a un host de una red local en busca de su huella identificativa. Se ha identificado información con el número de puertos cerrados que tiene la víctima, en este caso con la dirección IP 10.1.1.16, los puertos en servicio abiertos, la identificación del sistema Operativo y la dirección física de la interfaz de red.

7.6.1. Transmisión En El Protocolo De Control: Cuando TCP se encuentra entre IP y la capa de aplicación, el Protocolo de Control de Transmisión (TCP) garantiza una comunicación fiable orientada a conexión para las estaciones y para los servicios que estas implementan. TCP garantiza que los datagramas se entregan en orden, sin que haya errores, y sin duplicación. Los servicios se proporcionan utilizando mecanismos de control de flujo tales como el de "Ventana deslizante" (Three-way Handshake) y la adaptación de Técnicas de retransmisión.

7.6.2. Three-way Handshake: Para entender el proceso de establecimiento de conexión en tres vías se ha tomado una topología de una red genérica como la que se muestra en la *Figura 8*. Similar al escenario que el proyecto pretende recrear en la detección de ataques de denegación de servicio. Antes de poder transmitir datos entre un host de origen S y un host de destino D, TCP tiene que establecer una conexión entre S y D como se ve en la *Figura 7*. El establecimiento de esta conexión recibe el nombre de "conexión en tres vías". Se hace el análisis del comportamiento de los datos en las conversaciones establecidas usando el protocolo TCP/IP y los servicios implementados por el protocolo IP y su unidad de datos básica que es el **datagrama,** *Figura 3* cuyo tamaño máximo es de 65535 bytes (64K).

Figura 3. Estructura de un datagrama IP V 4.0



Fuente: <Guerrero, Cesar. Modelo de Generación de alertas tempranas en ataques DoS. p 32.>

Se parte de un análisis básico de la estructura de un datagrama IP, que está dividida en bloques de 32 bits (4 bytes). El datagrama IP se transmite enviando primero el bit 0, luego el bit 1, 2, 3... Y así sucesivamente hasta finalizar el datagrama. Este orden se denomina network byte order. Para el estudio que se lleva a cabo en la detección de ataques DoS, es muy importante conocer este orden de transmisión de la información, puesto que los diferentes ordenadores tienen diferentes sistemas de almacenamiento de bits en memoria.

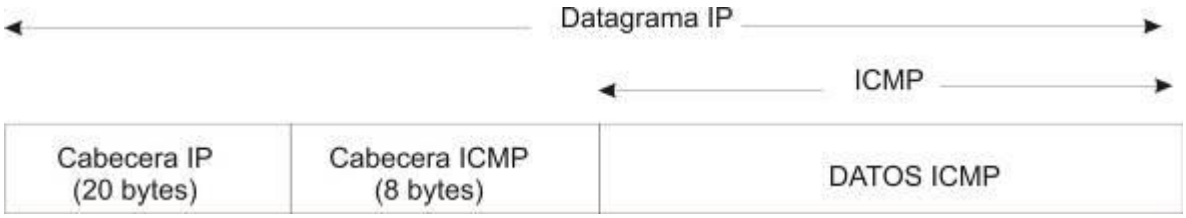
Para poder capturar esta información referente a la topología de red o distribución física identificando el tipo de sistema entre el atacante y las victimas, los ataques DoS manipulan el campo TTL de la cabecera IP de un paquete para determinar la forma que los saltos uno a uno se dan y por los que un determinado paquete avanza por la red TCP/IP. El campo TTL actúa como un contador de saltos, viéndose reducido en una unidad al ser reenviado por cada dispositivo de encaminamiento. *Este campo es observado en la figura (3).*

Específicamente este **campo TTL "tiempo de vida" (Time To Live)**, es un campo de 8 bits que indica el tiempo máximo que el datagrama será válido y podrá ser transmitido por la red. Esto permite un mecanismo de control para evitar datagramas que circulen eternamente por la red (por ejemplo en el caso de bucles). Este campo que se inicializa en el ordenador de origen a un valor (máximo 2^8 = 256) y se va decrementando en una unidad cada vez que atraviesa un router. De esta forma, si se produce un bucle y/o no alcanza su destino en un máximo de 255 "saltos", es descartado. En este caso se envía un datagrama ICMP de error al ordenador de origen para avisar de su pérdida.

7.6.3. Función de ICMP en los ataques DoS: Un análisis inicial muestra la importancia de este protocolo y su papel cuando se trata de extraer información de una conversación TCP con miras a ser atacada o a denegar algún servicio.

El protocolo ICMP (Internet Control Message Protocol) es un protocolo simple que va encapsulado en datagramas IP, *Figura 4* y que tiene por función el control del flujo de la comunicación así como la comunicación de errores [RFC792].

Figura 4. Estructura de un mensaje ICMP



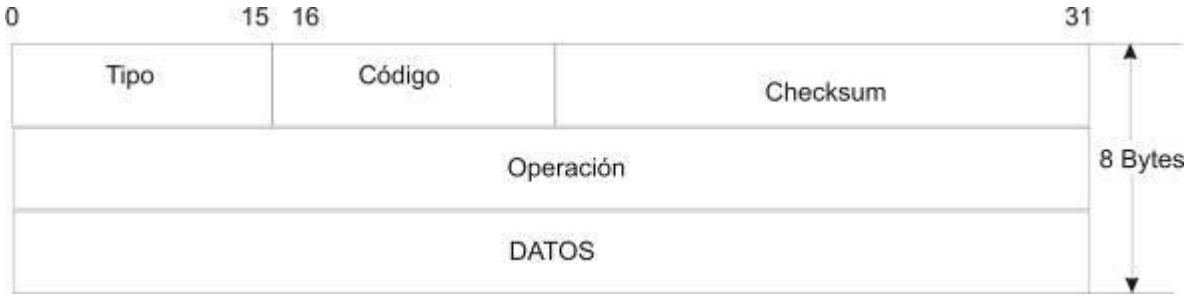
Fuente: < Guerrero, Cesar. Modelo de Generación de alertas tempranas en ataques DoS >

ICMP también permite obtener información como la franja horaria del sistema destino o la máscara de red. La simple ejecución del comando ping contra la dirección IP asociada a un nombre de dominio ofrece al atacante información de gran utilidad. Para empezar, esta información le permitirá determinar la existencia de uno o más equipos conectados a la red de este dominio.

Una explicación sencilla de cómo se identifican equipos conectados le da un servicio de entrega basado en el mejor intento (best effort) que adopta IP. Significa que cuando se detecta un error en la transmisión o funcionamiento irregular ya sea por tiempo de entrega de paquetes, tamaños, tipos de servicios, direcciones origen o destino, se contempla un sistema muy simple de tratamiento de errores. Este mecanismo de control de errores viene regulado por el protocolo ICMP (Internet Control Message Protocol).

Generalmente la víctima (equipo, host o cliente), descarta los datagramas y enviaría un mensaje de error ICMP al ordenador de origen sin encargarse de la retransmisión del datagrama, lo que no implica fiabilidad. La cabecera del protocolo ICMP tiene un tamaño de 8 bytes que contiene varios campos que permiten la identificación del mensaje. *Figura 5*.

Figura 5. Cabecera de un datagrama ICMP



Fuente: < Guerrero, Cesar. Modelo de Generación de alertas tempranas en ataques DoS >

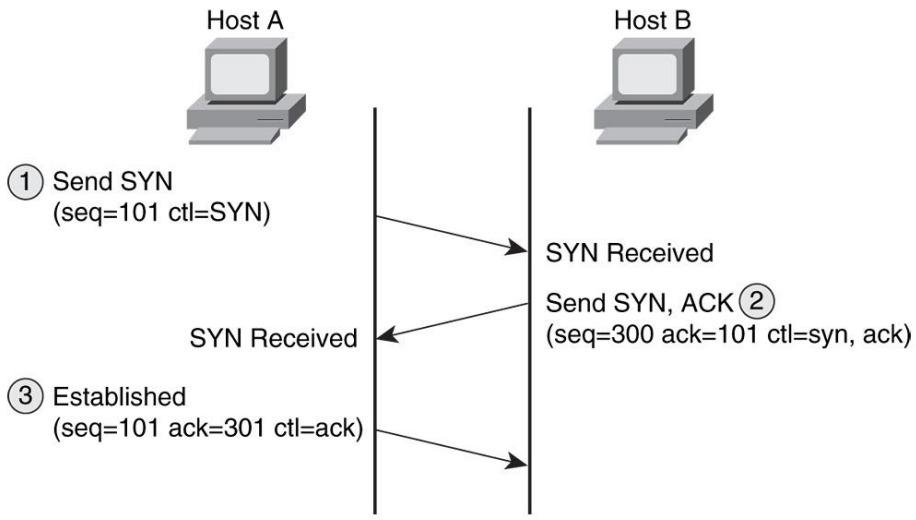
El Tipo (Type) indica el carácter del mensaje enviado, ya que el protocolo permite especificar una gran variedad de errores o mensajes de control de flujo de las comunicaciones.

Poder identificar los errores en el campo de Código (Code) que indica el código de error dentro del tipo de error indicado en el campo “tipo” agrupando los mensajes en

tipos y dentro de cada tipo especificando el código concreto al que se refiere, ayudan a determinar alertas sobre los fallos en la red o posibles ataques.

El Checksum simplemente permite verificar la integridad del mensaje enviado, lo que permite detectar posibles errores en el envío, transporte o recepción del mensaje de control ICMP.

Figura 7. Mecanismo de tres vías



Fuente: < Analysis of a Denial of Service Attack on TCP, Christoph L. Schuba.>

El mecanismo se describe de la siguiente forma:

[14]"0. El host receptor, que en el caso de más común será un servidor, espera pasivamente una conexión ejecutando las primitivas LISTEN y ACCEPT.

1. En primer lugar, el host que desea iniciar la conexión ejecuta una primitiva CONNECT especificando la dirección IP y el puerto con el que se desea conectar, el tamaño máximo del segmento que está dispuesto a aceptar y opcionalmente otros datos, como alguna contraseña de usuario. Entonces la primitiva CONNCET hace una apertura activa, enviando al otro host un paquete que tiene el bit SYN (ver formato de un segmento TCP más abajo) activado, indicándole también el número de secuencia inicial "x" que usará para enviar sus mensajes.

---

14 Protocolo de acuerdo a tres vías. Disponible desde Internet en: <http://neo.lcc.uma.es/evirtual/cdd/tutorial/transporte/twhandshake.html>

2. El segundo paso continúa con el host receptor que recibe el segmento revisa si hay algún proceso activo que haya ejecutado un LISTEN en el puerto solicitado, es decir, preparado para recibir datos por ese puerto. Si lo hay, el proceso a la escucha recibe el segmento TCP entrante, registra el número de secuencia "x" y, si desea abrir la conexión, responde con un acuse de recibo "x + 1" con el bit SYN activado e incluye su propio número de secuencia inicial "y", dejando entonces abierta la conexión por su extremo. El número de acuse de recibo "x + 1" significa que el host ha recibido todos los octetos hasta e incluyendo "x", y espera "x + 1" a continuación. Si no desea establecer la conexión, envía una contestación con el bit RST activado, para que el host en el otro extremo lo sepa.

3. El primer host recibe el segmento y envía su confirmación, momento a partir del cual puede enviar datos al otro extremo, abriendo entonces la conexión por su extremo.

4. La máquina receptora recibe la confirmación y entiende que el otro extremo ha abierto ya su conexión, por lo que a partir de ese momento también puede ella enviar datos. Con esto, la conexión ha quedado abierta en ambos sentidos.”

7.6.4. Exploración de puertos: El escaneo o exploración de puertos permite determinar las características de una red o sistemas remotos, con el objetivo de identificar los equipos disponibles y alcanzables desde Internet, así como los servicios que ofrece cada uno. Se puede llegar a conocer los sistemas existentes, los servicios ofrecidos por ellos, cómo están organizados los equipos, que sistemas operativos ejecutan y cuál es el propósito de cada uno.

Hay diferentes formas de explotar las vulnerabilidades de TCP/IP en los que los ataques de Denegación de Servicio hacen uso de diferentes técnicas para realizar esta exploración de puertos TCP. Entre las más conocidas, se destacan las siguientes:

- **TCP connect scan.** Mediante el establecimiento de una conexión TCP completa (completando los tres pasos del establecimiento de la conexión) la exploración puede ir analizando todos los puertos posibles. Si la conexión se realiza correctamente, se anotara el puerto como abierto (realizando una suposición de su servicio asociado según el número de puerto).

- **TCP SYN scan.** Enviando únicamente paquetes de inicio de conexión (SYN) por cada uno de los puertos que se quieren analizar se puede determinar si estos están abiertos o no. Recibir como respuesta un paquete RST-ACK significa que no existe ningún servicio que escuche por este puerto. Por el contrario, si se recibe un paquete SYN-ACK, podemos afirmar la existencia de un servicio asociado a dicho puerto TCP. En este caso, se enviara un paquete RST-ACK para no establecer conexión y no ser registrados por el sistema objetivo, a diferencia del caso anterior (TCP connect scan).

Existen otras “vulnerabilidades” que se pueden explotar pero que son dependientes de la implementación de la pila TCP/IP. Por Ejemplo:

- **TCP FIN scan.** Al enviar un paquete FIN a un puerto, deberíamos recibir un paquete de reset (RST) sí dicho puerto está cerrado. Esta técnica se aplica principalmente sobre implementaciones de pilas TCP/IP de sistemas Unix.

- **TCP Xmas Tree scan.** Esta técnica es muy similar a la anterior, y también se obtiene como resultado un paquete de reset si el puerto está cerrado. En este caso se envían paquetes FIN, URG y PUSH.

- **TCP Null scan.** En el caso de poner a cero todos los indicadores de la cabecera TCP, la exploración debería recibir como resultado un paquete de reset en los puertos no activos.

7.6.5. Exploración De Puertos Udp: Para poder detectar si existe un sistema de filtrado o cortafuegos, los sistemas de Detección de intrusos (IDS) utilizan el protocolo UDP para mandar paquetes UDP con 0 bytes en el campo datos, si el puerto está cerrado se debe recibir un ICMP Port Unreachable. Si el puerto está abierto, no se recibe ninguna respuesta. Si se detectan muchas puertas abiertas, puede haber un dispositivo de filtrado (firewall) en el medio. Para verificar esto, se envía un paquete UDP al puerto cero, si no se recibe una respuesta ICMP Port Unreachable, entonces hay un dispositivo de filtrado de tráfico.

Para el desarrollo de un sistema de alarmas que detecte ataques de denegación de servicio (DoS) en el caso de usar paquetes UDP para la exploración de puertos, hay

que tener en cuenta que a diferencia de las exploraciones TCP, se trata de un proceso más lento puesto que la recepción de los paquetes enviados se consigue mediante el vencimiento de temporizadores (timeouts).

7.6.6. Escaneo Basado En El Protocolo Icmp: La identificación de respuestas ICMP permite obtener huellas identificativas cuando se le da un mal uso a este protocolo. Muchas de las características que tienen las respuestas ICMP no son propias de los sistemas operativos. Algunos si responderán y otros no. Incluso y así la experiencia en la implementación de este protocolo dice que se pude deshabilitar este servicio de detección de errores en los sistemas de comunicación a través del mismo sistema operativo.

Por lo anterior, el análisis de respuestas ofrecidas mediante el tráfico ICMP como las del ICMP echo, ICMP timestap, ICMP information; se dará en la medida del desarrollo del proyecto y el análisis del ataque.

Básicamente la interpretación de estas respuestas ofrecidas es:

ICMP echo: Comúnmente llamado “Ping”. Se le utiliza para saber cuáles son los dispositivos que se encuentran activos en una subred. Esto se hace de forma masiva para detectar todos los host. Los IDS también detectan este tipo de diálogos.

ICMP broadcast: Se hace un único ping a la dirección de broadcast y se logra que todos los equipos contesten. Depende del sistema operativo la reacción a este tipo de cosas. Los sistemas Windows actuales no responden a estas solicitudes.

ICMP Timestamp: Se envía un paquete ICMP Timestamp y si el sistema objetivo está activo, responde, nuevamente depende la implementación del sistema operativo si se responde o no a estos paquetes.

Otros métodos tienen que ver con el uso indirecto de ICMP haciendo que un equipo conteste con paquetes de este tipo cuando se introducen errores en los paquetes IP:

- **IP bad header field:** Se introducen errores en los campos de la cabecera IP,

esto hace que si un equipo descarta el paquete debido a estos errores entonces notifica con un ICMP parameter problem. No todas las implementaciones responden a los mismos errores. Esto sirve, por ejemplo, para identificar los fabricantes de los routers.

Los firewall y los IDS deberían tener en cuenta este tipo de tráfico, y manejarlo con cuidado, ya que si un firewall simplemente no deja volver paquete ICMP parameter problem, entonces un atacante puede detectar la existencia del mismo.

- **IP non-valid field values:** Se ingresan valores incorrectos en los campos.

7.7. TIPOS DE ATAQUES

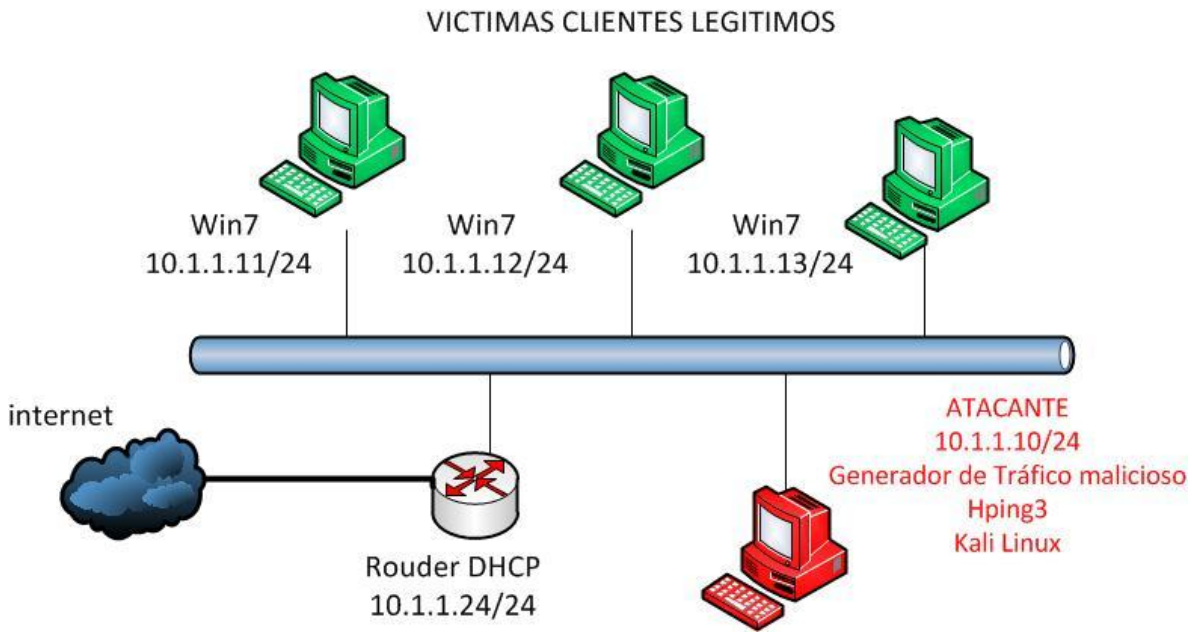
Para efectos de recrear el sistema para la detección de IPs, se delimita a las acciones que pueda ofrecer un solo ataque de denegación de servicio que pueda generar inundaciones masivas con IPs falsa. Para ello se ha escogido los ataques TCP SYN Flooding.

7.7.1. Ataque TCP/SYN Flooding:

Aprovechándose de la debilidad del protocolo TCP en las primeras implementaciones de las pilas TCP, los ataques a trabajar se basan en no poder completar intencionalmente el protocolo de intercambio TCP. Se ha seleccionado este tipo de ataque para su análisis y detección de alarmas ya que es uno de los más implementados con diferentes formas de actuar. No tiene límite de envío de paquetes por lo que el rendimiento en la red se baja de inmediato y considerablemente hasta llegar a denegar el servicio.

El ataque TCP/SYN Flooding se basa en una inundación masiva de la red mediante datagramas IP y por consiguiente lleva a no completar intencionalmente el protocolo de intercambio TCP para inundar la cola de espera. La víctima se queda esperando por establecer un servicio pues el atacante no responde con ACK los SYN/ACK de la víctima, Esto ocurre hasta saturar los recursos de memoria y así consigue la denegación de servicios de la víctima. En la *Figura 8* se observa en rojo la ruta del ataque.

Figura 8. Esquema del ataque IP Flooding



Fuente: <El Autor. Adaptado de Guerrero, Cesar. Modelo de Generación de alertas tempranas en ataques DoS >

La denegación de servicios se da por que el sistema está a la espera de que baje el umbral que generalmente es **1 minuto** para aceptar más conexiones, cada conexión generada por un SYN, tienen un temporizador de 1 minuto, cuando se excede el límite de tiempo o umbral, se libera la memoria que mantiene el estado de la conexión y la cuenta de la cola de servicios se disminuye en 1. El atacante debe usar un IP falso para que no le hagan seguimiento a las conexiones.

Tipo de datagramas: La inundación de datagramas IP puede ser de diferentes tipos:

UDP: Generar peticiones sin conexión a cualquiera de los 65535 puertos disponibles. En muchos sistemas operativos, las peticiones masivas a puertos específicos UDP (ECHO, WINS) causan el colapso de los servicios que lo soportan.

ICMP: Generación de mensajes de error o control de flujo malicioso. En este caso el objetivo es doble, degradar el servicio de red con la inundación de peticiones y/o conseguir que los sistemas receptores quede inutilizados por no poder procesar todas las peticiones que les llegan.

TCP: Genera peticiones de conexión con el objetivo de saturar los recursos de red de la máquina atacada. Este protocolo es orientado a conexión, y como tal consume recursos de memoria y CPU por cada conexión. El objetivo es el de saturar los recursos de red disponibles de los ordenadores que reciben las peticiones de conexión y degradar la calidad del servicio.

8. PROPUESTA DE LA SOLUCIÓN

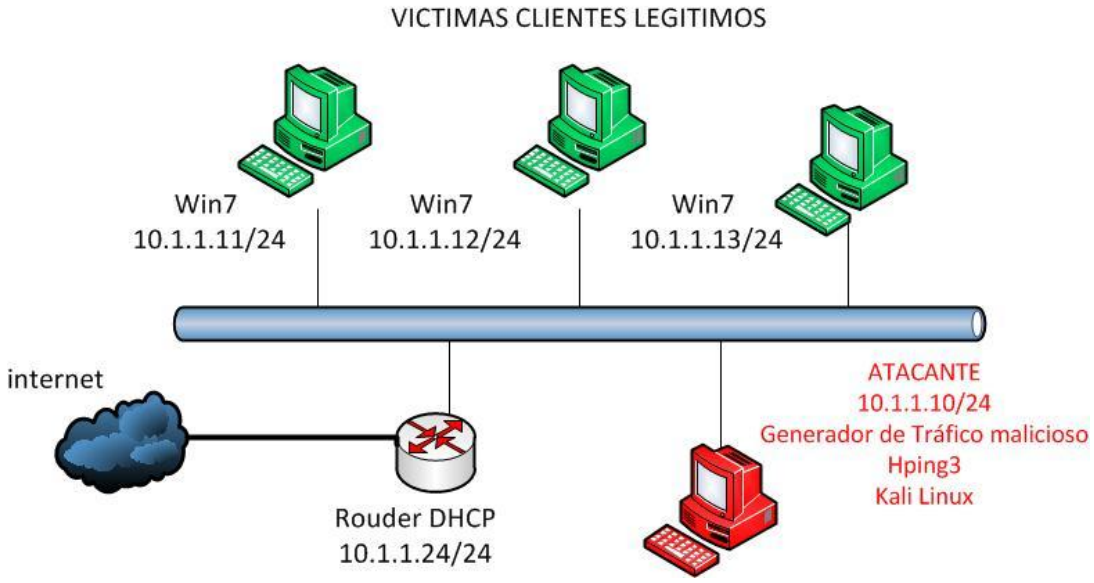
8.1. SELECCIÓN DEL SISTEMA OPERATIVO

1. Un sistema operativo Tipo Linux para perpetrar ataque: Como se identifica mediante un comando de identificación de sistema en ambientes Unix:
root@bt:~# uname -a
Linux kali 3.7-trunk-am64 #1 SMP Debian i686 GNU/Linux

8.2. ESCENARIO DE PRUEBAS

Se estableció un escenario red de are local LAN Ethernet 802.3 para las pruebas. *Figura 9*.

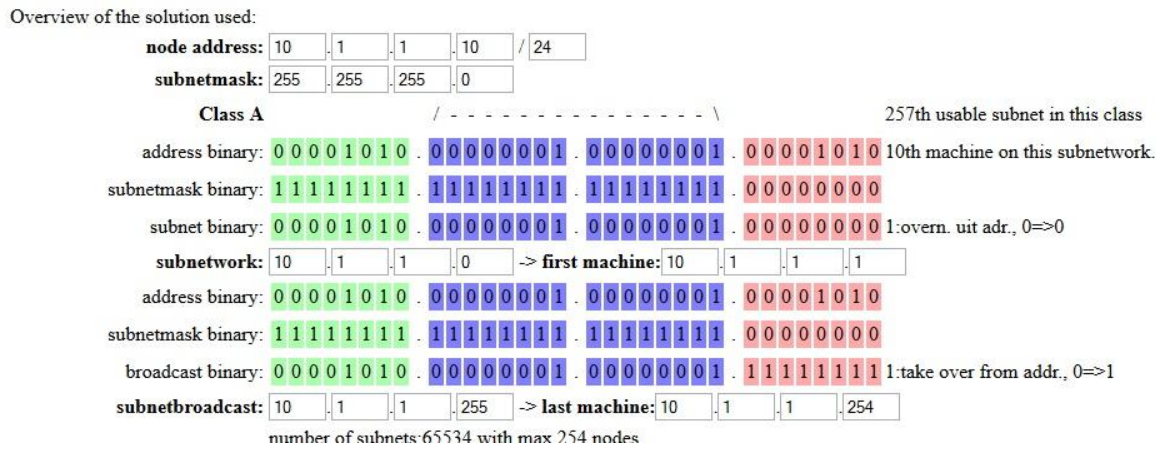
Figura 9. Escenario de Trabajo Red Ethernet



Fuente: <El Autor.>

Se establece un esquema Direcciones IP: 10.1.1.0/24 que se muestra en la figura. Usando calculadora IP. Esto para efectos de poder diferenciar las IPs falseadas en el análisis que se haga con “analizador de protocolos”.

Figura 10. Esquema IP a usar para escenario de pruebas.



Fuente: <El Autor. Software aplicado: Calcip.>

El escaneo de Equipos como primera fase de pruebas mediante el uso de nmap como herramienta de exploración habiendo escogido un esquema de direccionamiento IP V 4.0 tipo A. Se detectan otros equipos adicionales en la red.

linux:/home/yudy# nmap -sP 10.1.1.0/24

Starting Nmap 4.62 (http://nmap.org) at 2010-06-02 08:06 COT
Host 10.1.1.1 appears to be up.
MAC Address: 00:1C:F0:73:D6:97 (D-Link)
Host 10.1.1.9 appears to be up.
MAC Address: 00:1A:EF:05:97:74 (Loopcomm Technology)
Host LINUX (10.1.1.10) appears to be up.
Host 10.1.1.11 appears to be up.
MAC Address: 00:10:5A:A2:0C:2D (3com)
Host 10.1.1.13 appears to be up.
Nmap done: 256 IP addresses **(3 hosts up)** scanned in 12.563 seconds

8.3. CONSOLIDACIÓN DEL ATAQUE

La idea es manipular paquetes e inundarlos, aplicando el ataque TCP SYNC/Flooding). Para el escaneo se hizo uso de "**hping2**" como herramienta de red capaz de generar paquetes IP, paquetes ICMP/UDP/TCP hechos a medida, mostrando las respuestas del host destino de la misma manera en la que lo hace la herramienta "ping" normalmente con el propósito de verificar cortafuegos y sistemas de detección de intrusiones.

“Al usar hping2, se puede: evaluar el desempeño de la red utilizando diferentes protocolos, tamaños de paquetes, TOS (type of service, o sea, tipo de servicio), y fragmentación; realizar descubrimiento de camino utilizando el campo MTU (onda traceroute); transferir archivos (incluso ante reglas de firewall muy fascistas); realizar funciones al estilo `traceroute' pero bajo diferentes protocolos; detección remota de OS (`remote OS fingerprinting'); auditar una implementación de TCP/IP (`TCP/IP stack') en particular; etc. hping2 es una buena herramienta para aprender acerca de TCP/IP.” [15]

Es factible modificar la información contenida en las cabeceras de los paquetes ya sean TCP, UDP e ICMP, en función de los parámetros con que se ejecute hping2.

8.4. PARÁMETROS BÁSICOS DE FUNCIONAMIENTO

Entre los parámetros básicos de funcionamiento se incluyen los siguientes:

-c --count esta opción permite especificar el número de paquetes que queremos enviar o recibir.
-i --interval especifica el número de segundos, -iX donde X son los segundos, o micro segundos -iuX, donde X son los segundos, que tienen que transcurrir entre el envió de cada paquete.
--fast alias de tiempo para indicar -i u10000. Para enviar 10 paquetes por segundo.
--faster alias de tiempo para indicar -i u1.
-**n --numeric** trata la salida de datos únicamente como números, no hace resolución inversa de host.
-**I --interface** permite especificar el interfaz de red por donde enrutar el tráfico generado.
-D --debug la información que nos aporta el debug nos permitirá identificar errores de parámetros.
-z --bind asocia la combinación de teclas Ctrl+z para incrementar o reducir el TTL de los paquetes.
-**Z --unbind** desasocia la anterior combinación de teclas.

[15]<RODES, Michael.Pruebas de seguridad con Hping. “El Baile”. Magazine N 38>

8.4.1. Selección De Protocolos: Por defecto Hping2 utiliza el protocolo TCP en las conexiones incluyendo en las cabeceras TCP un tamaño de ventana de 64 y con puerto destino 0, sin especificar ningún otro parámetro.

Modo RAW (-0 --rawiP) Usando este modo se enviarán las cabeceras IP con datos añadidos mediante el uso de los parámetros --signature y/o --file.

Modo ICMP (-1 --icmp) Con este modo se enviarán paquetes ICMP echo-request, es factible emplear otros códigos o tipos de icmp mediante las opciones --icmptype e –icmpcode.

Modo UDP (-2 --udp) Con este modo el tráfico que se generará será UDP, los parámetros útiles para este modo son --baseport --destport y –keep.

Modo SCAN (-8 --scan) Permite realizar análisis de puertos, se pueden especificar distintas opciones para los puertos, rangos, excluir determinados puertos, los puertos incluidos en /etc/services, etc.

Modo LISTEN (-9 –listen) Este modo ejecutado con --listen información, realiza un volcado de la información contenida en los paquetes en busca de la palabra "información" y la muestra.

8.4.2. Parámetros De Configuración Adicionales: Los parámetros de configuración también nos permiten especificar las siguientes opciones:

-s --baseport puerto origen base (por defecto es aleatorio)
-p --destport puerto destino, por defecto es 0
-k --keep mantener el puerto origen
-w --win tamaño de ventana (por defecto 64)
-O --tcpoff enviar tamaños de datos modificados (en lugar de el tamaño de la cabecera entre 4, tcphdrlen / 4)
-Q --seqnum muestra únicamente los números de secuencia tcp
-b --badcksum (intentará) enviar paquetes con checksum inválido

-M --setseq se establece un número de secuencia TCP
-L --setack se envía un paquete TCP con el flag de inicio de conexión (ack)

Existen muchos otros parámetros en Hping que permiten la creación de paquetes TCP, UDP e ICMP con todas sus combinaciones.

Cuando se generan paquetes TCP se pueden especificar los siguientes flags:
-F --fin flag FIN
-S --syn flag SYN
-R --rst flag RST
-P --push flag PUSH
-A --ack flag ACK
-U --urg flag URG
-X --xmas flag X
-Y --ymas flag Y
Desde el equipo 10.1.1.10 **kalilinux** se lanza el ataque:
hping2 10.1.1.13 –-rand-source –S –-destport 80 –-faster --debug –w 2048

La explicación de esta instrucción está dada en la Tabla 1.

Tabla 1. Instrucciones para el ataque DoS TCP SYN Flooding

ORIGEN	DESTINO
IP=1.2.3.4 → SYN	*IP=10.1.1.13*
IP=1.2.3.4 ← SYN/ACK	*IP=10.1.1.13*
Nunca responde con ACK	*El IP=10.1.1.13 guarda en la cola la petición de conexión por 1 minuto*
Se repite la secuencia de requerimiento	*El IP=10.1.1.13 se satura por tanto requerimiento encolado y ocurre el DoS*
Cualquier IP cliente pide servicio al servidor	*El IP=10.1.1.13 no puede atender requerimientos pues está en medio de un ataque DoS. Solamente cuando cese el ataque automáticamente se atienden los requerimientos de los clientes*

Descripción de los parámetros:

PARÁMETRO	*DESCRIPCIÓN*

10.1.1.13	*IP de la Victima*
--rand-source	*IP ficticio o spoofed, se genera aleatorio, la idea es que no exista en la red, al no existir este no responde y así el atacante pasa inadvertido*
--debug	*Muestra cada intento*
-S	*Indica el flag "S" o SYN para solicitar un servicio*
--destport	*Indica el servicio requerido, es clave que este servicio este habilitado en la victima*
--faster	*Hace el intento de envió de SYN lo mas rápido que se pueda*
.w 2048	*La ventana de envío máximo será de 2048*

Fuente: <El Autor. Revisado y aplicado desde Guerrero, Cesar. Modelo de Generación de alertas tempranas en ataques DoS >

Las opciones de envío o inundación de peticiones que hacen de este ataque uno de los más perpetrados en redes Ethernet son comprobadas mediante el incremento del envío de paquetes por cada segundo que pasa. Debe inundarse la red con muchos paquetes por segundo para que el ataque sea efectivo. Estas opciones se ejecutan con los siguientes comandos:

hping2 10.1.1.13 –-rand-source –S –-destport 80 –-faster --debug –w 2048
-iu1000000, para 1 paquete por segundo
hping2 10.1.1.13 –-rand-source –S –-destport 80 –-faster --debug –w 2048
-iu500000, para 2 paquetes por segundo
hping2 1 10.1.1.13 –-rand-source –S –-destport 80 –-faster --debug –w 2048
-iu333333, para 3 paquetes por segundo (*)
hping2 10.1.1.13 –-rand-source –S –-destport 80 –-faster --debug –w 2048
-iu250000, para 4 paquetes por segundo (*)

8.4.3. Análisis Con Wireshark: [16]*Este análisis aplica para cualquier denegación genérica de servicios y la interpretación de las conexiones establecidas o interrumpidas es la misma para efectos de evaluar el Trhee Way Handshake.*

"Cuando un host A le manda datos a un host B, le está mandando un "paquete TCP". Dicho paquete, contiene los siguientes datos:

- puerto origen
- puerto destino
- IP origen
- IP destino

Además, dicho paquete contiene "flags", que determinan el tipo de paquete que se está mandando. Los distintos flags son los siguientes:

- SYN – Este flag indica que se está pidiendo una conexión

- ACK – Con este flag se responde cuando un paquete llega bien

- PSH – Este flag se usa para pasar datos a la aplicación. Por ejemplo, este flag tiene que estar activado en conexiones TELNET o RSH

- RST – Corta la conexión de manera brusca

- URG – Determina que un paquete es urgente

- FIN – Pide el cierre de la conexión

Es importante detectar que, el paquete contiene números de secuencia que identifican a cada paquete mandado, pero eso no es importante ahora.

Para este ejemplo se asigna el nombre de A para el cliente y B es el servidor:

1.	A	--SYN-->	B
2.	A	<--SYN/ACK--	B

[16] Disponible en internet. Con acceso desde: http://www.wiresharktraining.com/book.html

3. A --ACK--> B

1. A le pide a B la petición de la conexión mediante la activación del "flag" SYN.

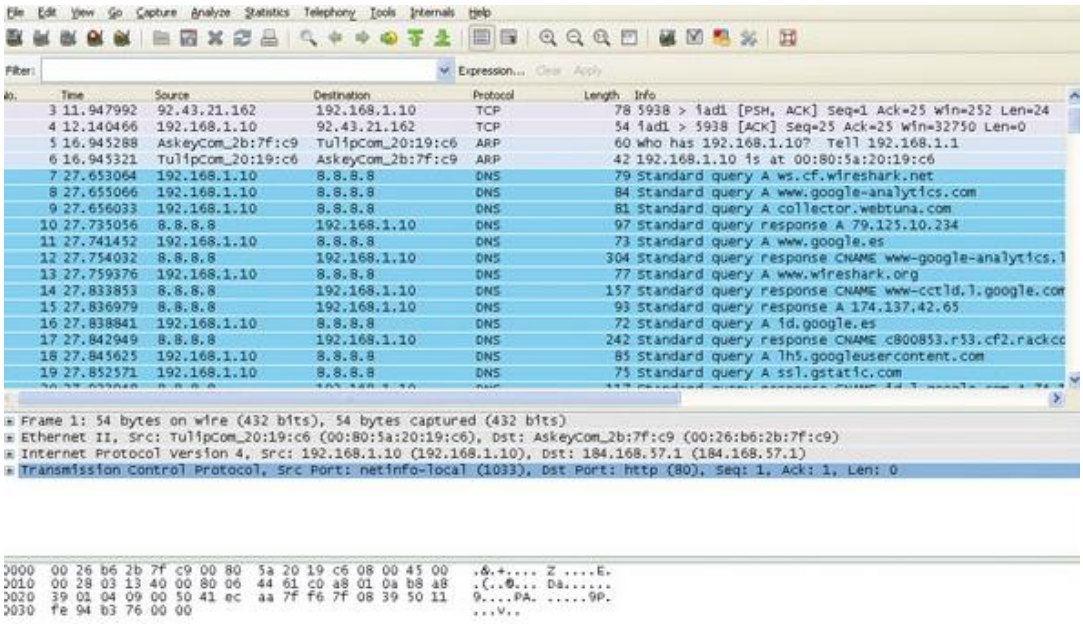
2. B acepta la petición de A.

3. A le responde a B diciéndole que su respuesta llegó bien.

Hay que mandarle a un host destino, una serie de paquetes TCP con el bit SYN activado, (es decir, una serie de peticiones de conexión) desde una dirección IP suplantada, la cual tiene que ser inexistente."

En la *Figura 11* se observa la evidencia el ataque en el analizador de protocolos. Las IPS aleatorias se inician con una dirección falsa 8.8.8.8

Figura 11. Ataque TCP Syn Flooding Visto en el Analizador Wireshark



Fuente: <El Autor.>

La víctima se queda esperando por establecer un servicio pues el atacante no responde con ACK los SYN/ACK de la víctima, Esto ocurre hasta saturar los recursos de memoria y así consigue la denegación de servicios de la víctima.

La denegación de servicios se da por que el sistema está a la espera de que baje el umbral que generalmente es 1 minuto para aceptar más conexiones, cada conexión generada por un SYN, tienen un temporizador de 1 minuto, cuando se excede el límite de tiempo o umbral, se libera la memoria que mantiene el estado de la conexión y la cuenta de la cola de servicios se disminuye en 1.

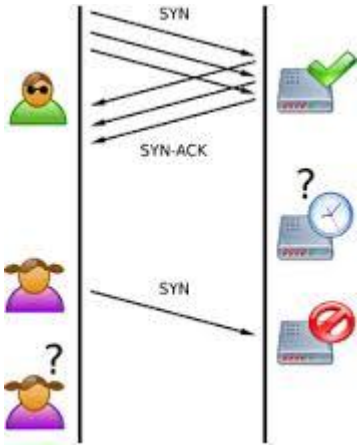
En el momento de presentar el ataque, se observa la petición por parte del atacante quién pregunta por 10.1.1.13. Se empieza a observar la inundación de paquetes SYN con la intención de complementar el protocolo.

El esquema del ataque es el siguiente:

```
A(C) –-SYN--> B
A(C) –-SYN--> B
A(C) –-SYN--> B
...
C <--SYN/ACK—- B
C <--SYN/ACK—- B
C <--SYN/ACK—- B
...
C <--RST-- B
```

8.4.4. Esquema Del Ataque: La conversación TCP que finalmente no se completa, se observa en la *Figura 12*.

Figura 12. Esquema del ataque TCP SYN Flooding



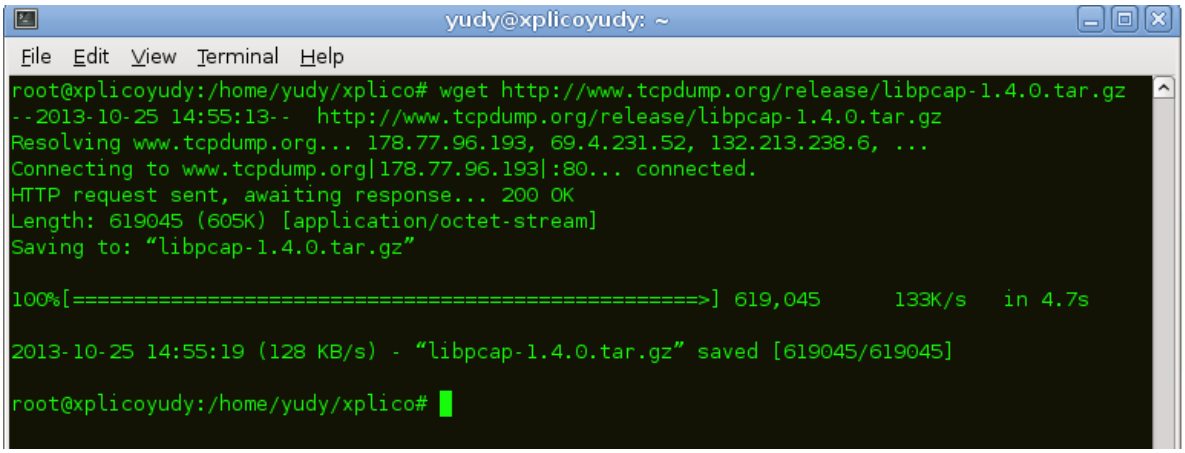
Fuente: < http://ja.wikipedia.org/wiki/SYN_flood>

8.4.5. Tratamiento De Los Datos: El procedimiento para aplicar el filtrado o selección de IPs falseadas y sobre todo masivas, se aplicó de la siguiente forma, se documenta el proceso desde pcap hasta Xplico.

DOCUMENTACION DE INSTALACION Y CAPTURA DE LIBRERÍA PCAP Y ANALISIS CON XPLICO

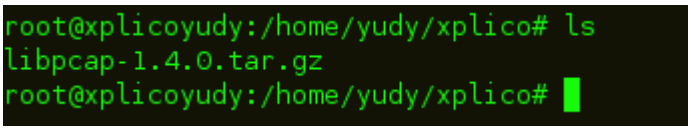
PARTE 1: INSTALACION DE LA LIBRERÍA PCAP: Instalación de la librería libpcap 1.4.0 (aplica a sistemas Unix). Para el caso del proyecto se instaló tanto en una máquina debían 7.0 como en Kali Linux.

Figura 13. Instalación De La Librería Pcap



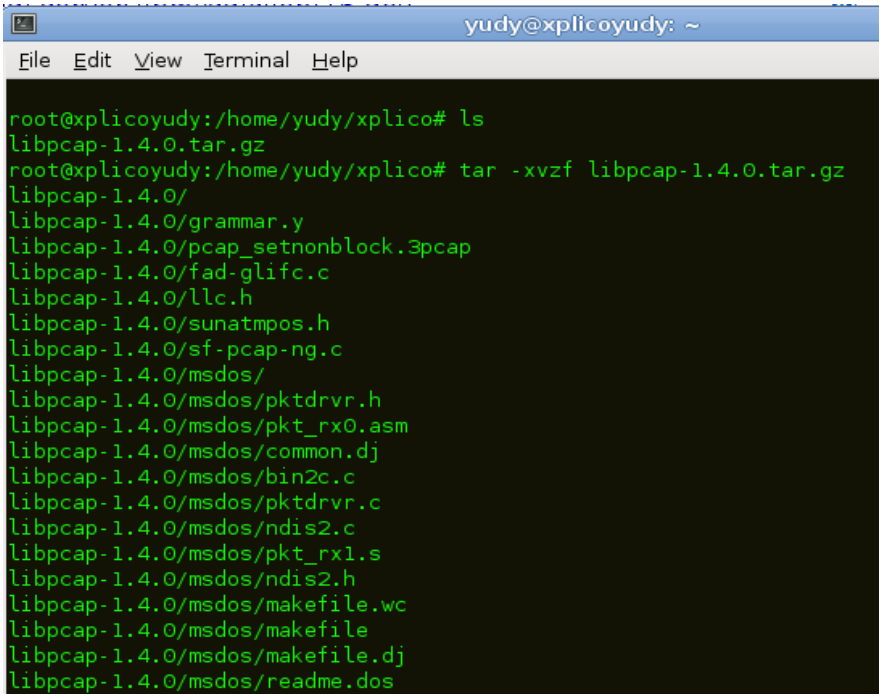
Fuente: <El Autor>

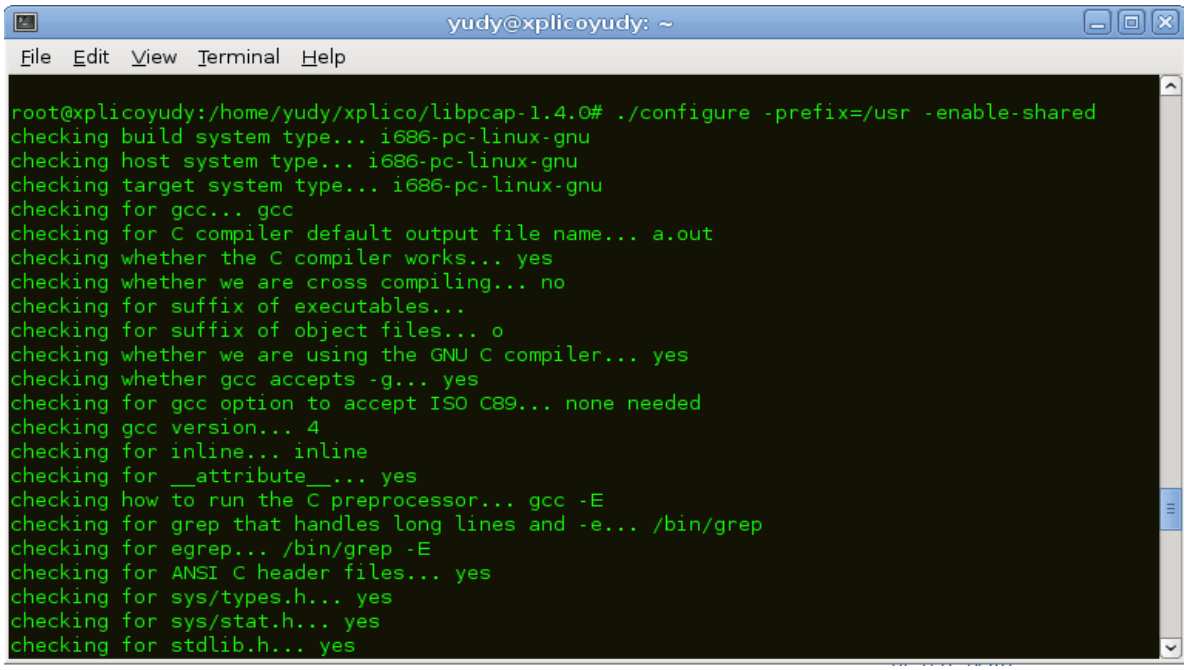
*Figura 14. El paquete **tar** descargado es ubicado en una carpeta del sistema.*



Fuente: <El Autor>

*Figura 15. Se desempaqueta los archivos de la librería para instalación y con **make** se compila (instalen el sistema).*

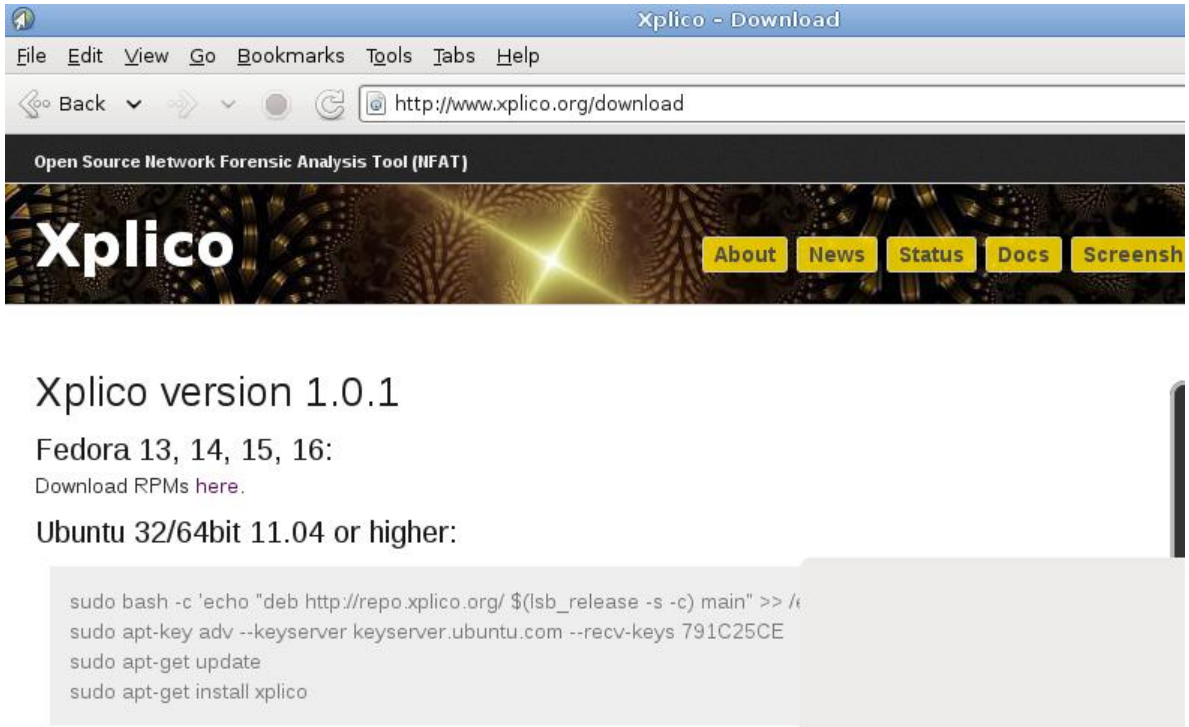




Fuente: <El Autor>

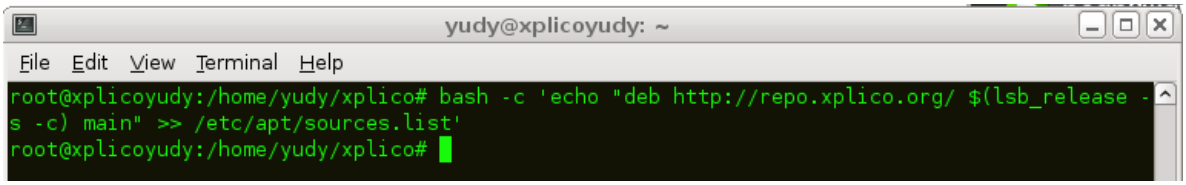
PARTE 2: INSTALACION DE XPLICO: Desde el sitio web del proyecto, se procede a ubicar las líneas que se deben cambiar en el archivo /etc/apt/ para la descarga desde los repositorios:

Figura 16. Instalación De Xplico.



Fuente: <El Autor>

Figura 17. Se adicionan las líneas al "source" origen de los repositorios

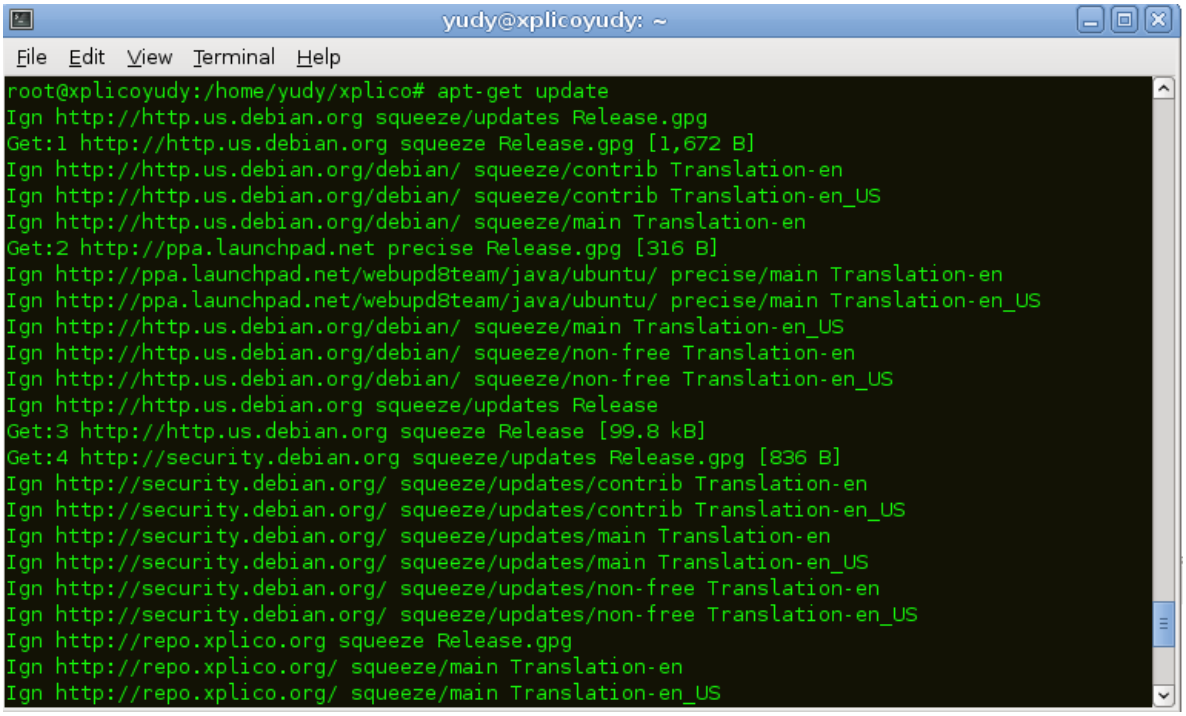


Fuente: <El Autor>

*Figura 18. Se agrega el **key** de autorización (archivo RSA de autenticación).*

Fuente: <El Autor>

Figura 19. Se actualizan repositorios del sistema.



Fuente: <El Autor>

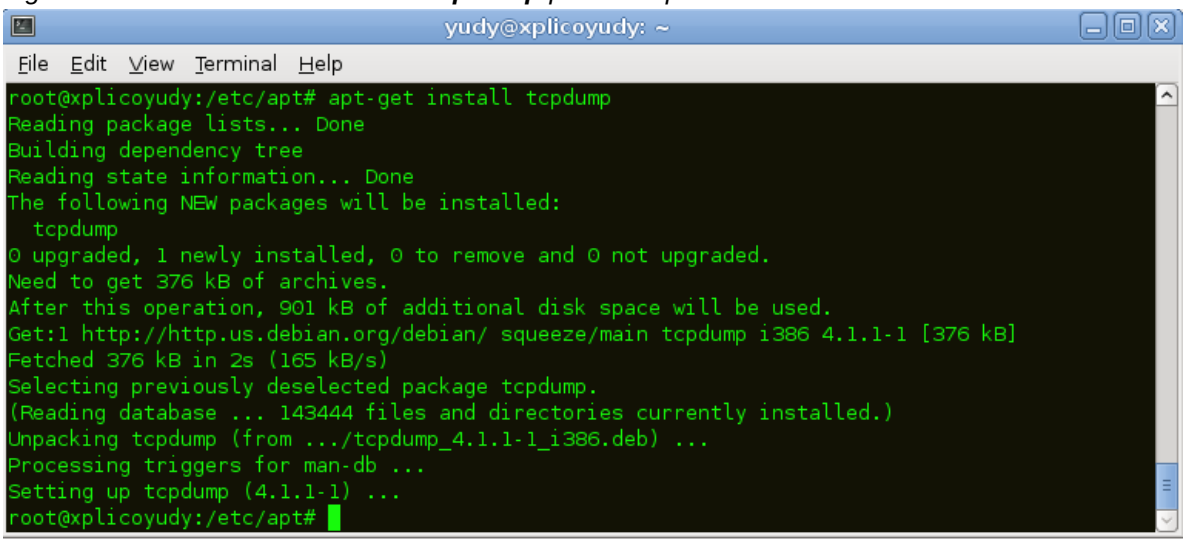
Figura 20. Para el filtro de paquetes, es necesario instalar la librería libssl para conexiones y sitios que tengan un certificado seguro ssl

```
root@xplicoyudy:/etc/apt# apt-get install libssl-dev
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  zlib1g-dev
The following NEW packages will be installed:
  libssl-dev zlib1g-dev
0 upgraded, 2 newly installed, 0 to remove and 128 not upgraded.
Need to get 2,349 kB of archives.
After this operation, 6,443 kB of additional disk space will be used.
Do you want to continue [Y/n]?
```

Fuente: <El Autor>

PARTE 3: LECTURA DEL TRÁFICO QUE CIRCULA: Para observar las IPs aleatorias. Se hizo uso de la librería **pcap** con la herramienta **tcpdump** para ejecutarla: el comando a usar es: **tcpdump –n**

*Figura 21. PARTE 3: Instalación de **tcpdump** para la captura del tráfico.*



Fuente: <El Autor>

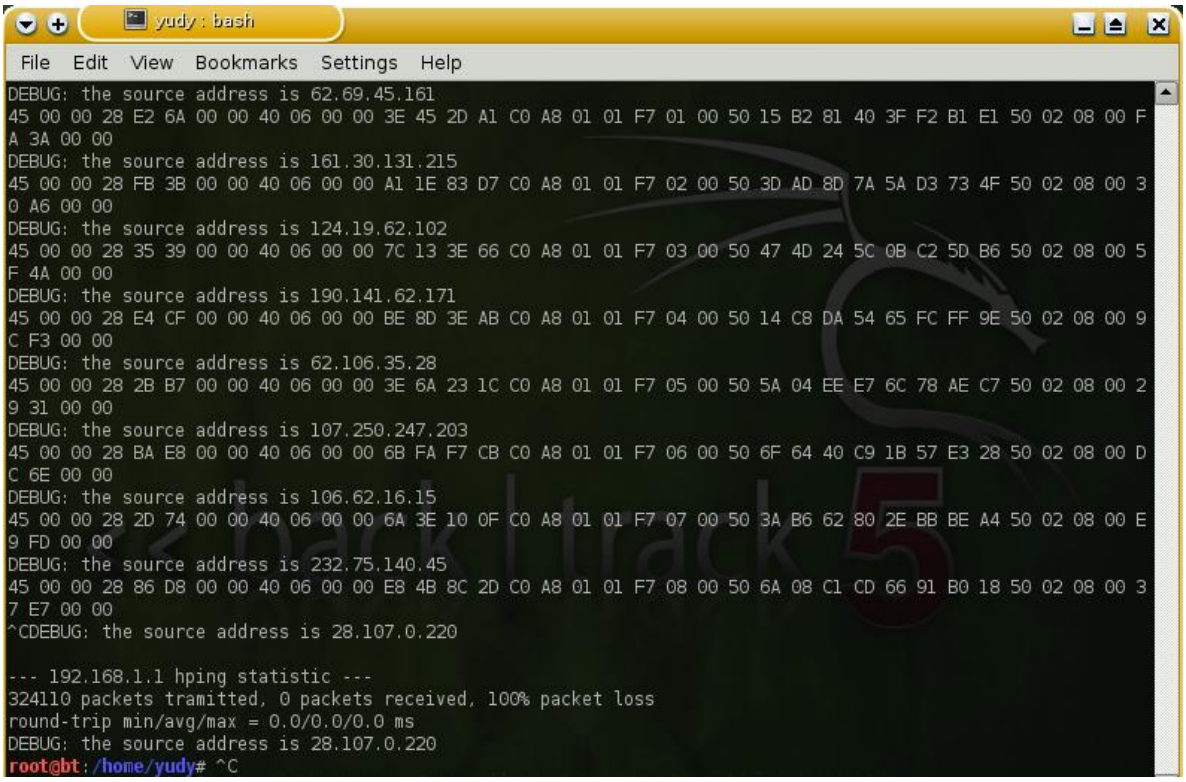
PARTE 4: Se debe mantener el ataque DDoS TCP SYn activo (se lanza)

Figura 22. PARTE 4: Se debe mantener el ataque DDoS TCP SYn activo (se lanza).

Fuente: <El Autor>

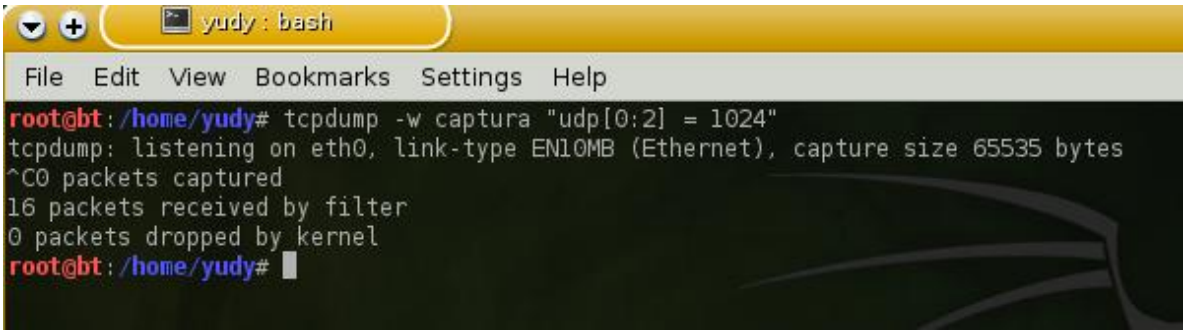
PARTE 5: Inundación como tal del servicio: Para ello se puede hacer uso de algún IDS que me lea el archivo pcap, en este caso la librería xplico puede aplicar a la solución del problema de este tipo de lecturas específicas.

Figura 23. Se evidencia la inundación de paquetes.



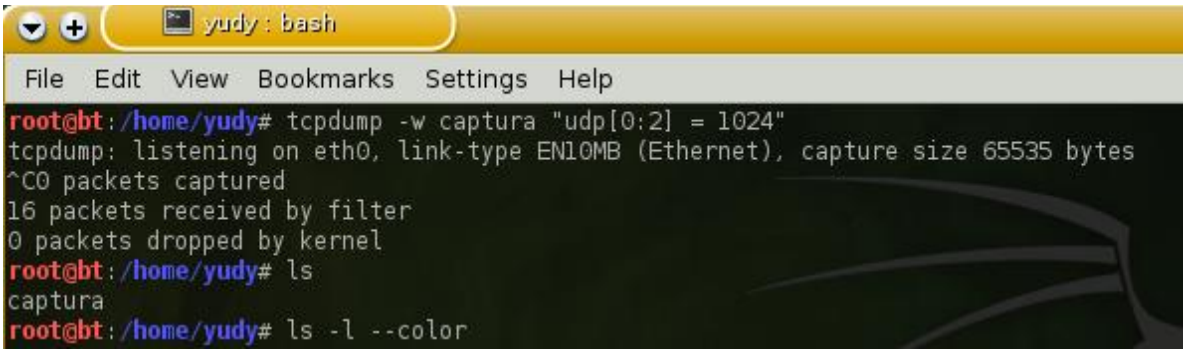
Fuente: <El Autor>

Figura 24. Se capturan los datos (con la opción w y con el nombre de archivo, en este caso "captura".



Fuente: <El Autor>

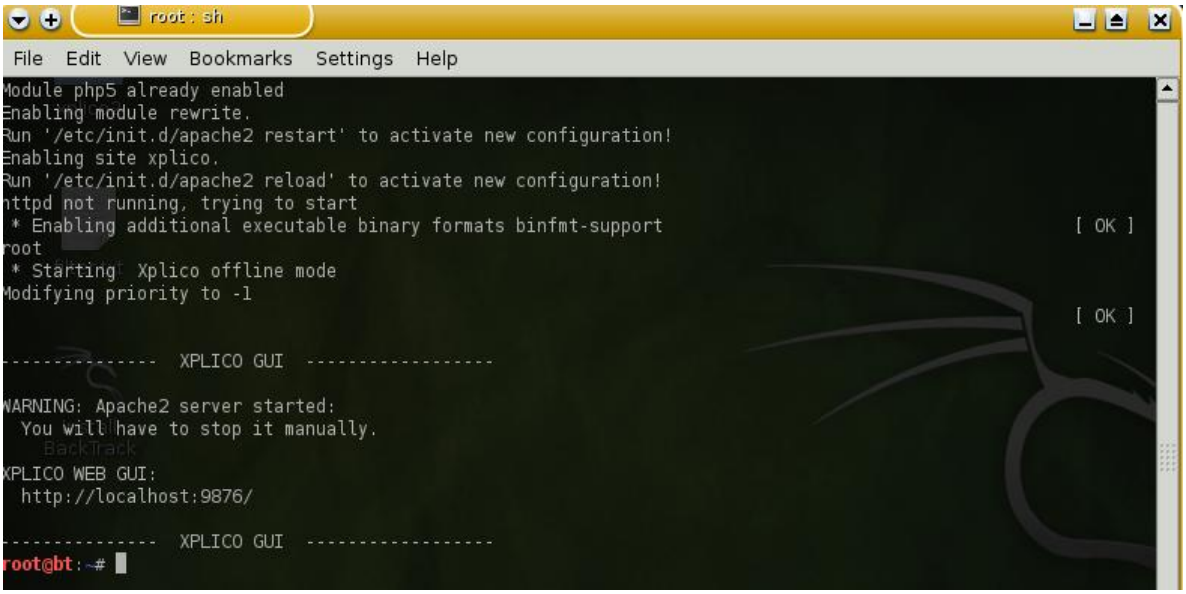
Figura 25. Se evidencia el archivo guardado en el sistema.



Fuente: <El Autor>

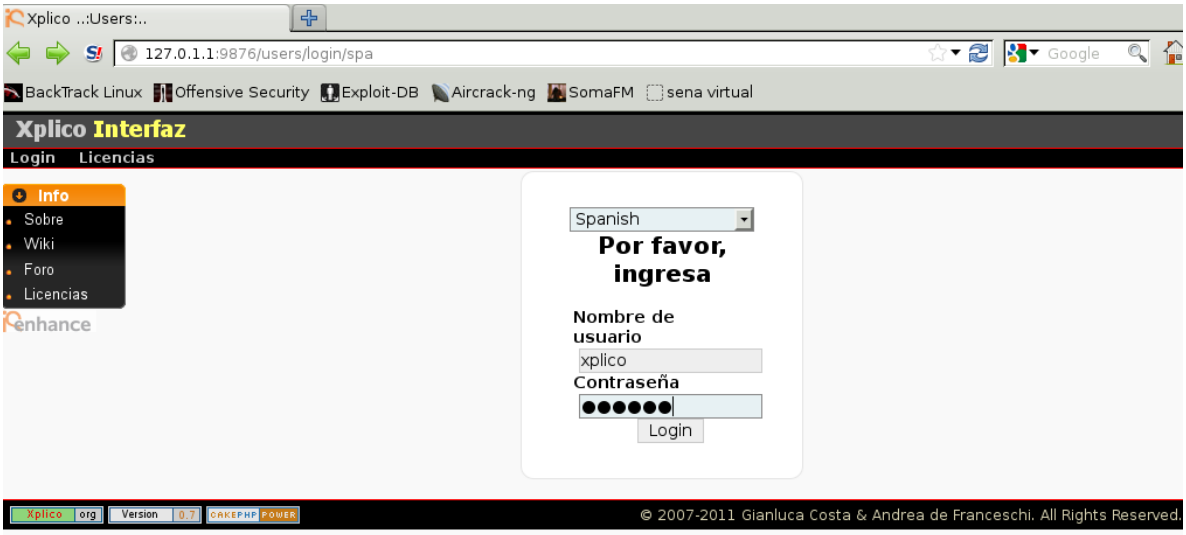
PARTE 6: Se lanza la aplicación xplico (en este caso la versión más reciente del proyecto la 1.0.1). Para el ejercicio se lanzó para ejecución con interfaz gráfica. Se deben tener los servicios activos de Apache, y MySql.

Figura 26. Aplicación Xplico.



Fuente: <El Autor>

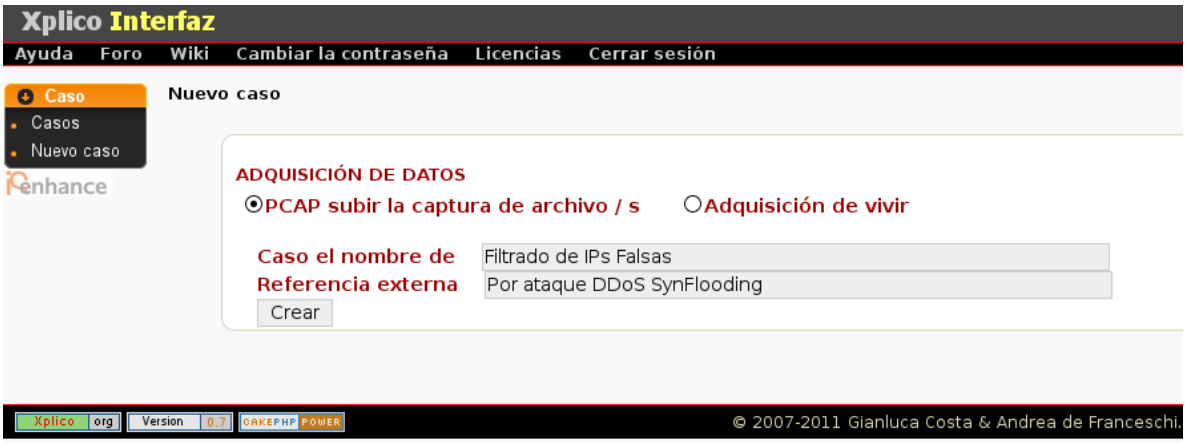
Figura 27. Desde un navegador se carga la interfaz gráfica



Fuente: <El Autor>

Se crea un nuevo caso con la opción que la lectura sea desde un archivo pcap previamente capturado.

Figura 28. Lectura De Archivo Pcap



Fuente: <El Autor>

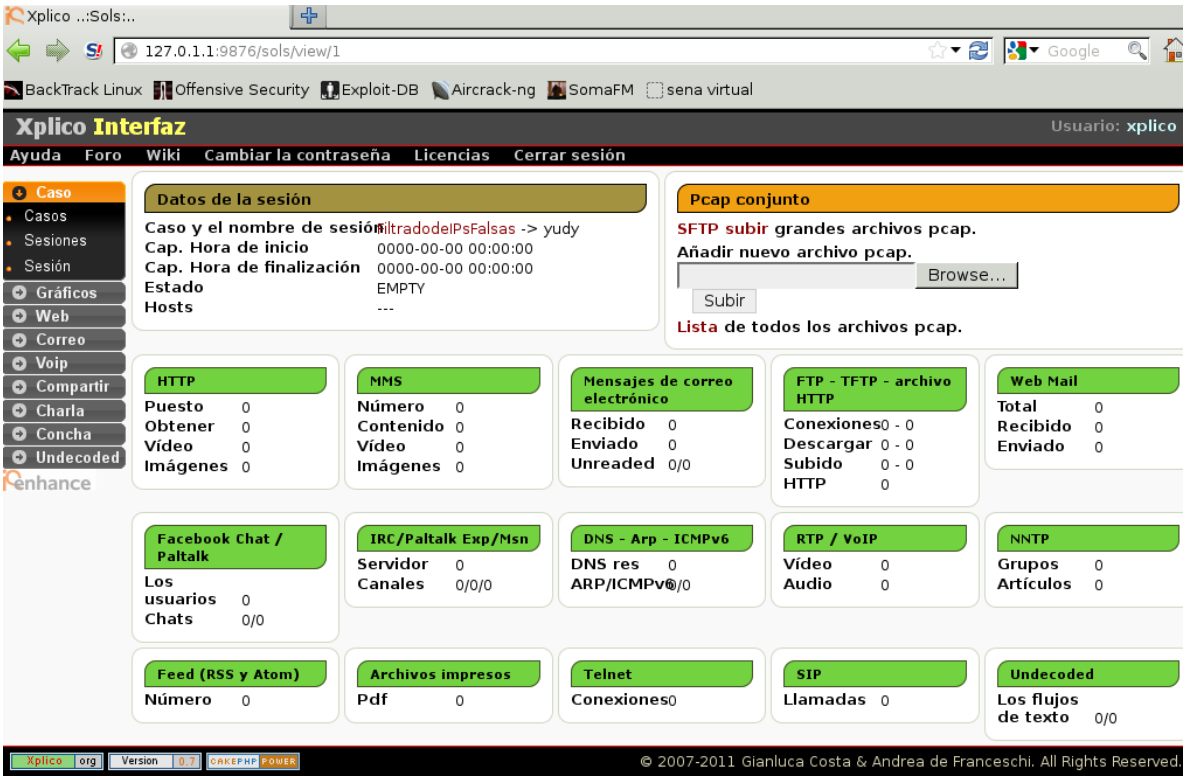
Se le adiciona una nueva sesión (sin límite de tiempo para el análisis del archivo en caso de resultar demasiado grande).

Figura 29. Al nuevo caso.

Fuente: <El Autor>

Posteriormente la aplicación pide abrir o subir un archivo pcap para el análisis. El archivo previamente capturado, tiene ya filtrado de paquetes del servicio del puerto 80 que haya sido denegado.

Figura 30. Interfaz Xplico.

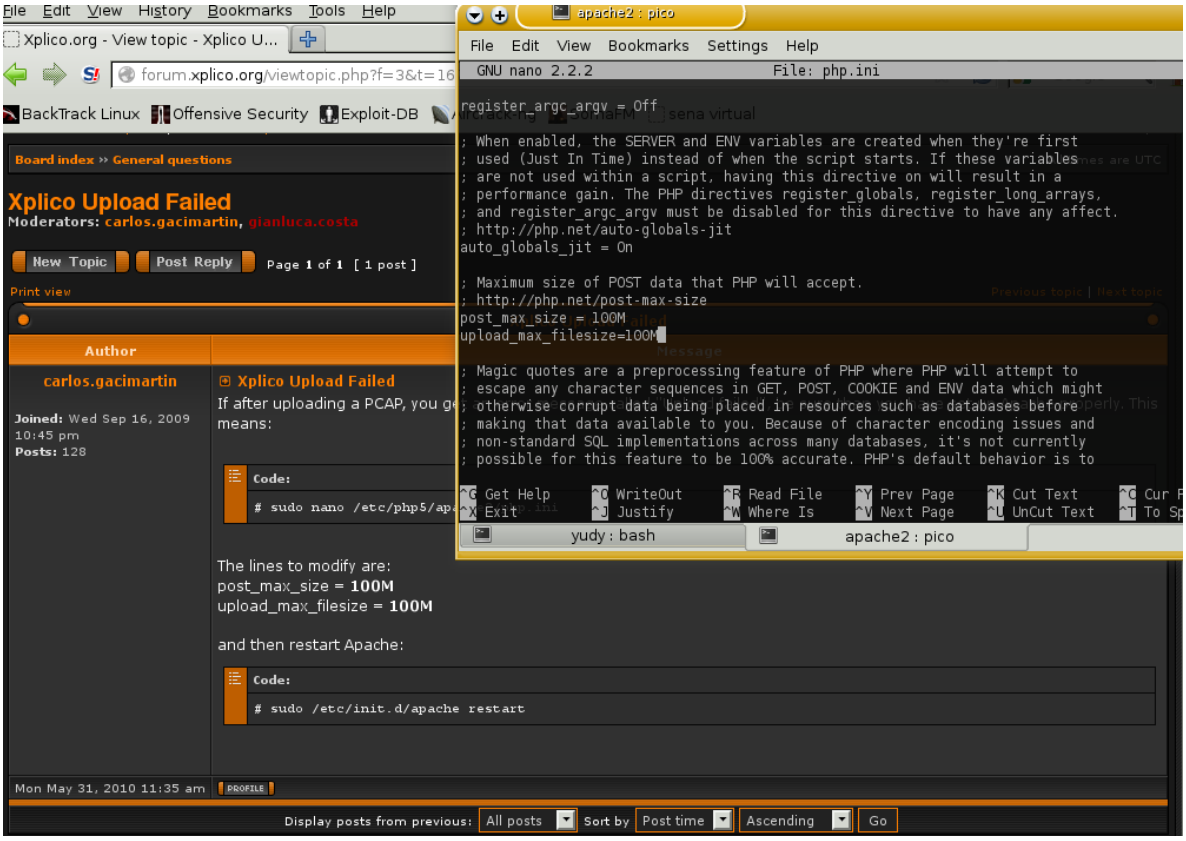


Fuente: <El Autor>

Hay que modificar el php.ini e indicarle el tamaño de upload o carga de archivos ya que los archivos que capturan tráfico como pcap son grandes en tamaño

dependiendo el tiempo de captura. En el ejercicio, al salir error, fue necesario modificar ese parámetro.

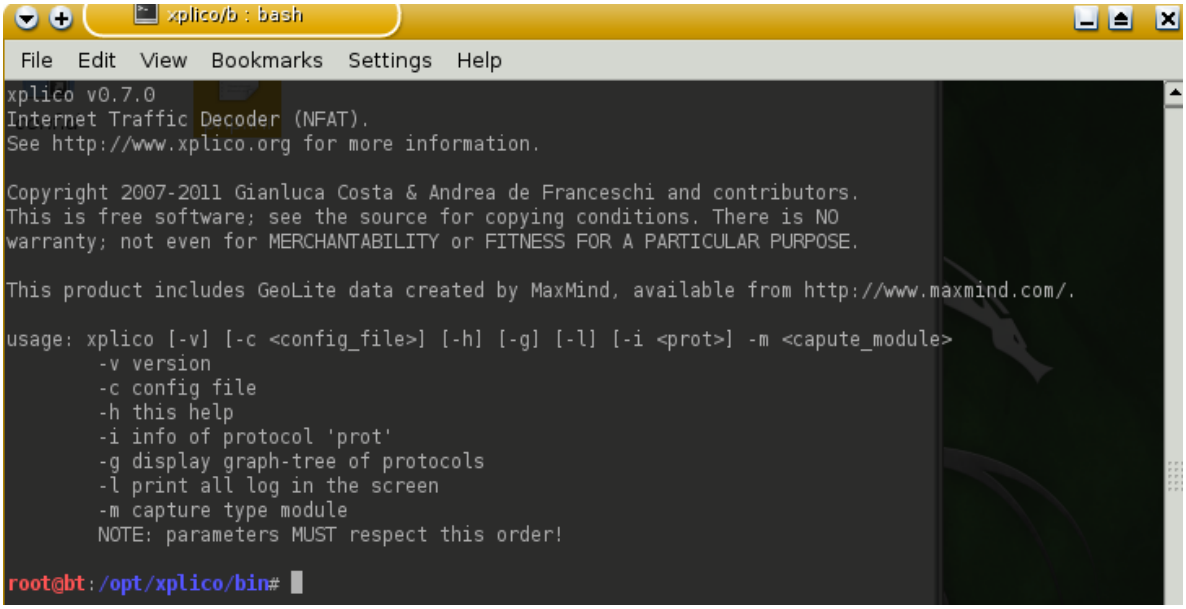
Figura 31. Modificación Php.ini.



Fuente: <El Autor>

PARTE 7: XPLICO DESDE UNA CONSOLA*:* Xplico también se lanza desde consola, para el caso en el que solo se requiera la visualización de información por archivos planos o de texto.

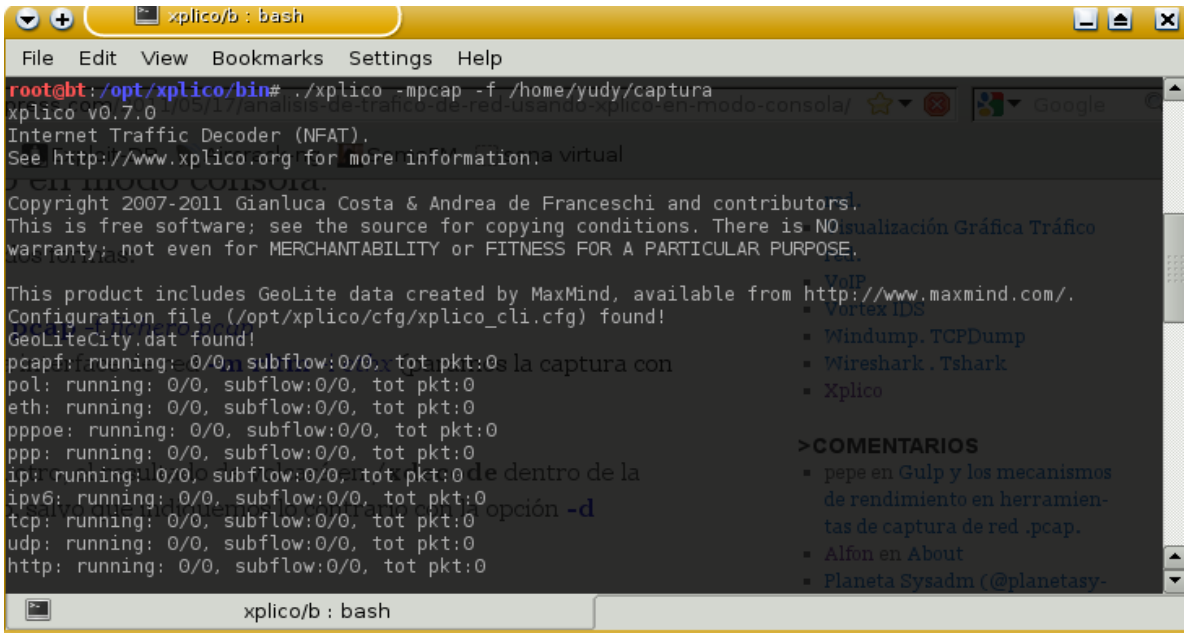
Figura 32. Xplico Desde Una Consola



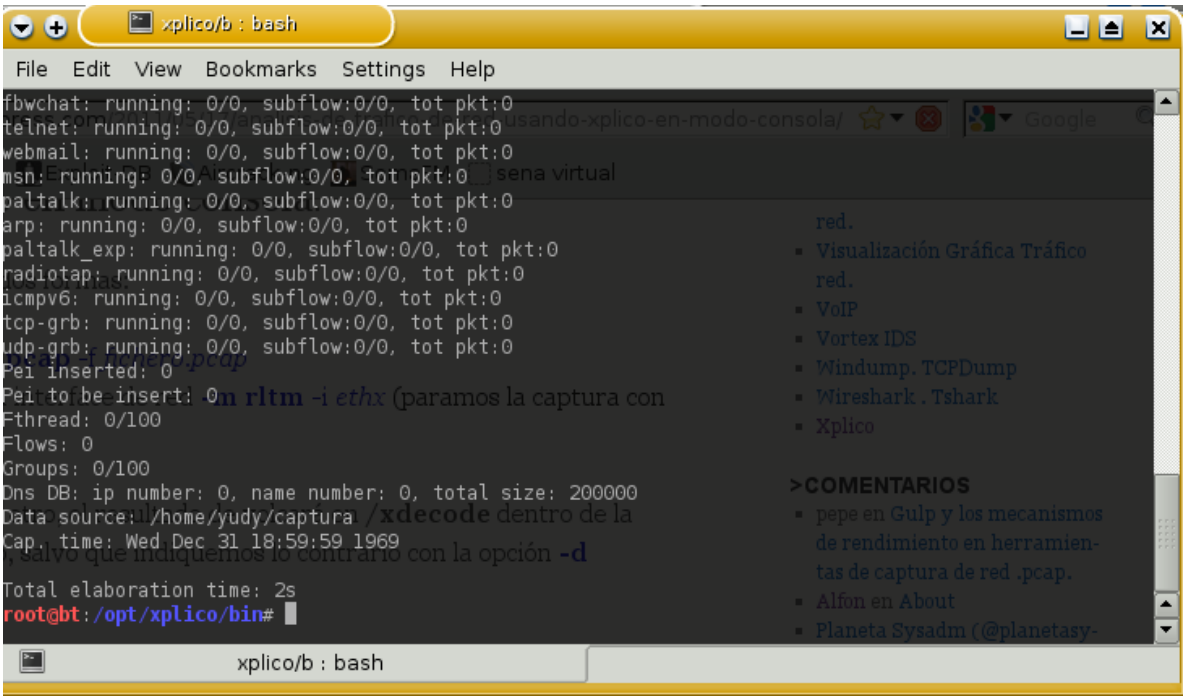
Fuente: <El Autor>

Desde la consola se carga también el archivo capturado en **pcap** que contiene todo el tráfico que ha pasado por la interfaz de red.

*Figura 33. Capturado En **Pcap***

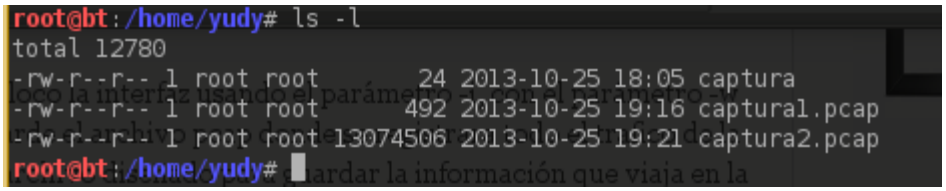


Fuente: <El Autor>
*Figura 34. Ya la librería ha sido ejecutada y el archivo **pcap** leído.*



Fuente: <El Autor>

Figura 35. Ese genera otros dos archivos con los datos filtrados que pueden ser leídos en formato texto



Fuente: <El Autor>

Finalmente se clasifican y cuentan las PI's aleatorias generadas por el ataque DDoS y que inundan el servicio, (en este caso el servicio dado por el puerto 80), En el archivo "filtradoip" se identifica en rojo la Ip aleatoria y el tipo de flag "SYN" no completado en la comunicación de tres vías. El tipo de protocolo, el servicio afectado, y el tamaño del intento de envío d etrama.

Este archivo debe ser leído con un editor de formato de texto enriquecido.

Figura 36. Se calsifican y cuentan las IP's aleatorias generadas por el ataque (DDoS).

```
No.     Time        Source                Destination
Protocol Length Info
      1 0.000000    25.42.238.24          192.168.1.1
TCP      54     57705 > http [SYN] Seq=0 Win=2048 Len=0

Frame 1: 54 bytes on wire (432 bits), 54 bytes captured (432
bits)
Ethernet II, Src: Hewlett-_9e:6a:49 (00:15:60:9e:6a:49), Dst:
Zte_fa:f5:d1 (00:15:eb:fa:f5:d1)
Internet Protocol Version 4, Src: 25.42.238.24 (25.42.238.24),
Dst: 192.168.1.1 (192.168.1.1)
Transmission Control Protocol, Src Port: 57705 (57705), Dst Port:
http (80), Seq: 0, Len: 0

No.     Time        Source                Destination
Protocol Length Info
      2 0.000075    55.240.92.25          192.168.1.1
TCP      54     57706 > http [SYN] Seq=0 Win=2048 Len=0

Frame 2: 54 bytes on wire (432 bits), 54 bytes captured (432
bits)
Ethernet II, Src: Hewlett-_9e:6a:49 (00:15:60:9e:6a:49), Dst:
Zte_fa:f5:d1 (00:15:eb:fa:f5:d1)
Internet Protocol Version 4, Src: 55.240.92.25 (55.240.92.25),
Dst: 192.168.1.1 (192.168.1.1)
Transmission Control Protocol, Src Port: 57706 (57706), Dst Port:
http (80), Seq: 0, Len: 0
```

Fuente: <El Autor>

9. MARCO CONCEPTUAL

ADDRESS RESOLUTION PROTOCOL (ARP): Protocolo de la familia TCP/IP que asocia direcciones IP a direcciones MAC.

ARP: Address Resolution Protocol.

APACHE: Es un software libre servidor HTTP de código abierto para plataforma Unix (BSD, GNU/Linux, etc.), Windows, Macintosh y otras, que implementa el protocolo HTTP/1.1, y la noción del sitio virtual.

API: A*pplication programming interface* Es una interfaz de programación de aplicaciones

ASN (Abstract Syntax Notation): Lenguaje utilizado para definir tipos de datos.

AUTENTICACION: El proceso de determinar que una entidad es quien o lo que dice ser.

BROADCAST: Ttransmisión de un paquete que será recibido por todos los dispositivos en una red

CSMA/CD (Carrier Sentido acceso múltiple / Detección de Colisión) es el protocolo usado en Ethernet en red para garantizar que sólo un nodo de red se transmite en el cable de red en cualquier momento.

CONFIDENCIALIDAD: Que la información solo sea vista por los lentes involucrados en la comunicación y que un tercer no pueda ingresar.

CONTROL DE ACCESO Y AUTORIZACIÓN: El proceso de determinar los recursos y servicios que puede usar una entidad.

CORTAFUEGOS: Elemento de prevención que realizara un control de acceso para proteger una red de los equipos del exterior (potencialmente hostiles).

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

DESBORDAMIENTO DE BUFFER: Posibilidad de corromper la pila de ejecución para modificar el valor de retorno de una llamada a función y provocar la ejecución de código arbitrario.

DISPONIBILIDAD: Los servicios y la información deben, en todo momento, estar disponibles.

DNS: Servicio de Nombres de Dominio.

DoS: (De las siglas en inglés Denial of Service), Ataque de Denegación de servicio

DDoS: (De las siglas en inglés Distribuited Denial of Service), Ataque de Denegación de servicio Distribuido

ESCÁNER DE VULNERABILIDADES: Aplicación que permite comprobar si un sistema es vulnerable a un conjunto de deficiencias de seguridad.

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

EXPLORACION DE PUERTOS: Técnica utilizada para identificar los servicios que ofrece un sistema.

EXPLOTACIÓN DE UN SERVICIO: Actividad realizada por un atacante para conseguir una escalada de privilegios, abusando de alguna deficiencia del servicio o del sistema.

FINGERPRINTING: Huella identificativa.

FIREWALL: Un equipo que impone un conjunto de directivas de seguridad que restringen de forma severa el acceso al sistema y a los recursos.

FRAGMENTACIÓN IP: Proceso de división de un datagrama IP en fragmentos de menor longitud.

HERRAMIENTA DE MONITOREO JFFNMS: Es un sistema de gestión y monitorización de red designando para monitorizar una red IP, puede ser utilizado para monitorizar cualquier dispositivo SNMP, router, switch, servidor, puerto TCP o cualquier elemento siempre que se programe una extensión adecuada a dicho elemento JFFNSM.

HIDS: Sistemas de detección de intrusos de Ordenador

HUELLA IDENTIFICATIVA: Información muy precisa que permite identificar un sistema o una red en concreto. En inglés, fingerprinting.

ICMP: Internet Control Message Protocol.

IDS (Intrusion Detection System): Sistema de detección de Intrusos. Es una herramienta que permite monitorear el comportamiento y el uso que se le da a los recursos en una máquina y detectar si alguien está haciendo algo indebido.

IEEE: Tecnología desarrollada por Apple Computer en 1986 que permite transmitir información a alta velocidad. Fue adoptado como estándar en 1995 y es similar al puerto USB.

IEEE 802.3 Proporciona una LAN estándar desarrollada originalmente por Xerox y ampliada. Define dos categorías: banda base (específica una señal digital) y banda ancha (especifica una señal analógica). IEEE define únicamente una especificación para la categoría de banda ancha. Sin embargo, la restricción de la máxima longitud del cable puede cambiar usando dispositivos de red tales como repetidores o puentes.

INTEGRIDAD: Los datos reflejen la realidad y que correspondan con lo que debe ser y no ha sido modificadas indebidamente.

INTERNET CONTROL MESSAGE PROTOCOL (ICMP): Protocolo encargado de realizar el control de flujo de los datagramas IP que circulan por la red.

INTERNET PROTOCOL (IP): Protocolo para la interconexión de redes.

INTEGRIDAD: Los datos reflejan la realidad y que correspondan con lo que debe ser y no ha sido modificadas indebidamente.

LAN: Una red de área local, red local o LAN (del inglés Local Área Network) es la interconexión de varios ordenadores y periféricos. Su extensión está limitada físicamente a un edificio o a un entorno de hasta 200 metros.

MAC: Media Access Control

MODEM ADSL: Modula las señales enviadas desde la red local para que puedan transmitirse por la línea ADSL y demodula las señales recibidas por ésta para que los equipos de la LAN puedan interpretarlos. De hecho, existen configuraciones formadas por un módem ADSL y un router que hacen la misma función que un router ADSL.

MTU: Maximum Transfer Unit

MySQL: Es un sistema de gestión de base de datos relacional y multiusuario ubica las tablas en ficheros diferenciados, es recomendable para desarrollos que necesiten manejar numerosos registros y sesiones simultaneas.

NIDS: Sistema de detección de intrusos de Red.

NMAP: Programa de código abierto que abierto que sirve para efectuar rastreo de puertos TCP y UDP. Se usa para evaluar la seguridad de sistemas informáticos así como para descubrir servicios o servidores en una red informática.

NMS (Network Management Station): Estación de red encargada de gestionar varios dispositivos de red.

OID (Object ID): Identifica de manera única cada objeto representado en la MIB y es una secuencia de números enteros no negativos separados por un punto.

OSI: El Modelo OSI es un lineamiento funcional para tareas de comunicaciones y, por consiguiente, no especifica un estándar de comunicación para dichas tareas.

PCAP: Interfaz de una aplicación de programación para captura de paquetes

PDU (Protocol Data Unit): Define la estructura de la información que va a ser enviada por la red.

PHP: Es Un lenguaje de programación interpretado, diseñado originalmente para la

creación de páginas web dinámicas. Es usado principalmente en interpretación del lado del servidor (server–side scripting) pero actualmente puede ser utilizado desde una interfaz de línea de comandos o en la creación de otros tipos de programas incluyendo aplicaciones con interfaz gráfica.

PUERTA DE ENLACE: Es La que proporciona salida hacia el exterior a una red local.

ROUTER: Es un dispositivo que permite conectar uno o varios equipos o incluso una red de área local (LAN) a Internet a través de una línea telefónica con un servicio ADSL.

RRDTOOL: Herramienta que trabaja con una base de datos que maneja planificación según Round-Robin. Esta técnica trabaja con una cantidad de datos fija, definida en el momento de crear la base de datos, y un puntero al elemento actual. Su finalidad principal es el tratamiento de datos temporales y datos seriales como temperaturas, transferencias en redes, cargas del procesador, entre otros.

SNMP: (Simple Network Managment Protocol): usado para administrar la configuración de dispositivos de red desde una estación de trabajo.

SMI (Structure of Management Information): define agrupaciones, nombres, tipos de datos permitidos y la sintaxis para escribir MIB's.

SYN: Bit de control dentro del segmentoTCP.

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

TCP: Transmission Control Protocol.

TCPDUMP: Herramienta en línea de comandos cuya utilidad principal es analizar el tráfico que circula por la red

TCP/IP: Es la base de Internet, y sirve para enlazar computadoras que utilizan diferentes sistemas operativos, incluyendo PC, minicomputadoras y computadoras centrales sobre redes de área local (LAN) y área extensa (WAN)

TIA: Telecommunications Industry Association.

TRANSMISSION CONTROL PROTOCOL (TCP): Protocolo de transporte de la arquitectura de protocolos TCP/IP.

TTL: Time To Live

UDP: User Datagram Protocol.

USER DATAGRAM PROTOCOL (UDP): Protocolo de transporte de la arquitectura.

USM (User-based Security Model): Modelo de seguridad utilizado por SNMPv3 para administrar el envío de mensajes SNMPv3.

VACM (View-based Access Control Model): Modelo de control de acceso que permite administrar quien tiene acceso a qué información en la MIB.

WINPCAP: Motor de captura de paquetes y filtrado de muchas de las herramientas de red comerciales y de código abierto, incluyendo analizadores de protocolos, monitores de red, sistemas de detección de intrusos de red.

XPLICO: Librerías y daemon tipo de licencia GPL para manipulación de datos de tráfico capturado.

10. CONCLUSIONES

El desarrollo del proyecto pudo determinar la viabilidad de filtrado de direcciones IPs falseadas ocasionadas por ataques masivos en ataques de Denegación de Servicios (DDoS). Queda evidenciado que el proceso sigue dependiendo de librerías y de aplicaciones que se deben ejecutar conjuntamente y en u proceso ordenado. Deficiencia en muchos IDS que no implementan estos filtros por defecto y que serían de gran ayuda para la administración de sistemas de redes seguros o para administradores de red.

Las interfaces GUI (interfaces gráficas) que genera xplico no permiten de manera amigable identificar muchos filtros a aplicar, sigue siendo por ambientes de consola la prieta opción para identificar muchas reglas de detección filtrado en ataques distribuidos.

Se ha apropiado en conocimiento del funcionamiento básico e instalación de un sistema de detección de intrusos. Pero así mismo de la forma tan global y no específica en las alertas que pueda generar para ciertas vulnerabilidades, entre ellas las de la presencia de IPs falsas masivas.

Las acciones realizadas en este proyecto no Filtran los bloqueos por IPs masivas falsas. El IDS filtra todas las denegaciones de servicio y las muestra y evidencia. Queda por indagar como estos sistemas con esos archivos capturados podrían proactivamente y eficientemente, bloquear peticiones solo a IPs falseadas y permitir servicios a IPs legítimas.

Sigue siendo evidente que la Pila TCP para IP V4.0 es vulnerable y manipulable en muchos aspectos. Pero que así mismo hay muchas formas de implementar soluciones y barreras de optimización en seguridad para el protocolo.

11. REFERENCIAS BIBLIOGRAFICAS

WILLIAM. W. Detecting Flood-based Denial-of-Service Attacks with SNMP/RMON. USA, Caroina Uiversity. 14p. Paper.

Guerrero, S. Sistema de alertas para detección de un Ataque DDoS. Uiversidad Autónoma de Bucaramanga. p. 188

Blacker. W.J. Ataques de denegación de servicio distribuído. Técnicas de defensa. Computer Society. 2007, 12 p.

VERDEJO. G. Seguridad en redes IP. Uiversidad Autónoma de Barcelona. Departamento de informática. p. 234

CARVAJAL, A. Introducción a las técnicas de ataque e investigación Forense, Un enfoque pragmático. Colombia, Agosto de 2007, seg Ed, p. 245

CRUZ, A y TORRES, P. Sistema para Generar Gráficas a Partir de Logs Tcpdump usando Hadoop. Escuela Superior Politécnica del Litoral. Guayaquil, Ecuador, p. 59

TCP Variables. Disponible en Internet:
<http://www.frozentux.net/ipsysctl-tutorial/chunkyhtml/tcpvariables.html >

CARVAJAL, A. Introducción a las técnicas de ataque e investigación Forense, Un enfoque pragmático. Colombia, Agosto de 2007, seg Ed, p. 245

BOTERO, N. Modelo de Gestión de Seguridad con soporte a SNMP. Pontificia Universidad Javeriana. Facultad de ingeniería, Tesis. Bogotá Junio, 2005, p. 149.

Manual de Xplico. Disponible en internet: <http://www.xplico.org/docs>. Con acceso desde sep 210. Actualizado con licencia GNU/GPL.

TCPDUMP, pcap, web proyect. Disponible en internet desde
<http://www.tcpdump.org/pcap.htm>

ANEXO 2
Reglas típicas que usan los IDS.
Archivos configurables

Reglas DoS RULES

Archivo DDoS.rules

```
# (C) Copyright 2001-2004, Martin Roesch, Brian Caswell, et al.
#    All rights reserved.
# $Id: ddos.rules,v 1.23.2.2 2005/02/10 01:11:14 bmc Exp $
#-----------
# DDOS RULES
#-----------

alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"DDOS TFN Probe"; icmp_id:678;
itype:8; content:"1234"; reference:arachnids,443; classtype:attempted-recon; sid:221; rev:4;)
alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"DDOS tfn2k icmp possible
communication"; icmp_id:0; itype:0; content:"AAAAAAAAAA"; reference:arachnids,425;
classtype:attempted-dos; sid:222; rev:2;)
alert udp $EXTERNAL_NET any -> $HOME_NET 31335 (msg:"DDOS Trin00 Daemon to Master
PONG message detected"; content:"PONG"; reference:arachnids,187; classtype:attempted-recon;
sid:223; rev:3;)
alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"DDOS TFN client command BE";
icmp_id:456; icmp_seq:0; itype:0; reference:arachnids,184; classtype:attempted-dos; sid:228; rev:3;)


alert tcp $HOME_NET 20432 -> $EXTERNAL_NET any (msg:"DDOS shaft client login to handler";
flow:from_server,established; content:"login|3A|"; reference:arachnids,254;
reference:url,security.royans.net/info/posts/bugtraq_ddos3.shtml; classtype:attempted-dos; sid:230;
rev:5;)
alert udp $EXTERNAL_NET any -> $HOME_NET 18753 (msg:"DDOS shaft handler to agent";
content:"alive tijgu"; reference:arachnids,255; classtype:attempted-dos; sid:239; rev:2;)
alert udp $EXTERNAL_NET any -> $HOME_NET 20433 (msg:"DDOS shaft agent to handler";
content:"alive"; reference:arachnids,256; classtype:attempted-dos; sid:240; rev:2;)
# alert tcp $HOME_NET any <> $EXTERNAL_NET any (msg:"DDOS shaft synflood"; flow:stateless;
flags:S,12; seq:674711609; reference:arachnids,253; reference:cve,2000-0138; classtype:attempted-
dos; sid:241; rev:10;)

alert udp $EXTERNAL_NET any -> $HOME_NET 31335 (msg:"DDOS Trin00 Daemon to Master
message detected"; content:"l44"; reference:arachnids,186; classtype:attempted-dos; sid:231; rev:3;)
alert udp $EXTERNAL_NET any -> $HOME_NET 31335 (msg:"DDOS Trin00 Daemon to Master
*HELLO* message detected"; content:"*HELLO*"; reference:arachnids,185;
reference:url,www.sans.org/newlook/resources/IDFAQ/trinoo.htm; classtype:attempted-dos; sid:232;
```

rev:5;)
alert tcp $EXTERNAL_NET any -> $HOME_NET 27665 (msg:"DDOS Trin00 Attacker to Master default startup password"; flow:established,to_server; content:"betaalmostdone"; reference:arachnids,197; classtype:attempted-dos; sid:233; rev:3;)
alert tcp $EXTERNAL_NET any -> $HOME_NET 27665 (msg:"DDOS Trin00 Attacker to Master default password"; flow:established,to_server; content:"gOrave"; classtype:attempted-dos; sid:234; rev:2;)
alert tcp $EXTERNAL_NET any -> $HOME_NET 27665 (msg:"DDOS Trin00 Attacker to Master default mdie password"; flow:established,to_server; content:"killme"; classtype:bad-unknown; sid:235; rev:2;)
alert udp $EXTERNAL_NET any -> $HOME_NET 27444 (msg:"DDOS Trin00 Master to Daemon default password attempt"; content:"l44adsl"; reference:arachnids,197; classtype:attempted-dos; sid:237; rev:2;)
alert icmp $HOME_NET any -> $EXTERNAL_NET any (msg:"DDOS TFN server response"; icmp_id:123; icmp_seq:0; itype:0; content:"shell bound to port"; reference:arachnids,182; classtype:attempted-dos; sid:238; rev:6;)

alert udp $EXTERNAL_NET any -> $HOME_NET 6838 (msg:"DDOS mstream agent to handler"; content:"newserver"; classtype:attempted-dos; sid:243; rev:2;)
alert udp $EXTERNAL_NET any -> $HOME_NET 10498 (msg:"DDOS mstream handler to agent"; content:"stream/"; reference:cve,2000-0138; classtype:attempted-dos; sid:244; rev:3;)
alert udp $EXTERNAL_NET any -> $HOME_NET 10498 (msg:"DDOS mstream handler ping to agent"; content:"ping"; reference:cve,2000-0138; classtype:attempted-dos; sid:245; rev:3;)
alert udp $EXTERNAL_NET any -> $HOME_NET 10498 (msg:"DDOS mstream agent pong to handler"; content:"pong"; classtype:attempted-dos; sid:246; rev:2;)
alert tcp $EXTERNAL_NET any -> $HOME_NET 12754 (msg:"DDOS mstream client to handler"; flow:to_server,established; content:">"; reference:cve,2000-0138; classtype:attempted-dos; sid:247; rev:4;)
alert tcp $HOME_NET 12754 -> $EXTERNAL_NET any (msg:"DDOS mstream handler to client"; flow:to_client,established; content:">"; reference:cve,2000-0138; classtype:attempted-dos; sid:248; rev:4;)
alert tcp $EXTERNAL_NET any -> $HOME_NET 15104 (msg:"DDOS mstream client to handler"; flow:stateless; flags:S,12; reference:arachnids,111; reference:cve,2000-0138; classtype:attempted-dos; sid:249; rev:8;)
alert tcp $HOME_NET 15104 -> $EXTERNAL_NET any (msg:"DDOS mstream handler to client"; flow:from_server,established; content:">"; reference:cve,2000-0138; classtype:attempted-dos; sid:250; rev:4;)
alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"DDOS - TFN client command LE"; icmp_id:51201; icmp_seq:0; itype:0; reference:arachnids,183; classtype:attempted-dos; sid:251; rev:3;)

alert icmp 3.3.3.3/32 any -> $EXTERNAL_NET any (msg:"DDOS Stacheldraht server spoof"; icmp_id:666; itype:0; reference:arachnids,193; classtype:attempted-dos; sid:224; rev:3;)
alert icmp $HOME_NET any -> $EXTERNAL_NET any (msg:"DDOS Stacheldraht gag server response"; icmp_id:669; itype:0; content:"sicken"; reference:arachnids,195; classtype:attempted-dos; sid:225; rev:6;)
alert icmp $HOME_NET any -> $EXTERNAL_NET any (msg:"DDOS Stacheldraht server response"; icmp_id:667; itype:0; content:"ficken"; reference:arachnids,191; classtype:attempted-dos; sid:226; rev:6;)

alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"DDOS Stacheldraht client spoofworks"; icmp_id:1000; itype:0; content:"spoofworks"; reference:arachnids,192; classtype:attempted-dos; sid:227; rev:6;)
alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"DDOS Stacheldraht client check gag"; icmp_id:668; itype:0; content:"gesundheit!"; reference:arachnids,194; classtype:attempted-dos; sid:236; rev:6;)
alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"DDOS Stacheldraht client check skillz"; icmp_id:666; itype:0; content:"skillz"; reference:arachnids,190; classtype:attempted-dos; sid:229; rev:5;)
alert icmp $EXTERNAL_NET any <> $HOME_NET any (msg:"DDOS Stacheldraht handler->agent niggahbitch"; icmp_id:9015; itype:0; content:"niggahbitch"; reference:url,staff.washington.edu/dittrich/misc/stacheldraht.analysis; classtype:attempted-dos; sid:1854; rev:7;)
alert icmp $EXTERNAL_NET any <> $HOME_NET any (msg:"DDOS Stacheldraht agent->handler skillz"; icmp_id:6666; itype:0; content:"skillz"; reference:url,staff.washington.edu/dittrich/misc/stacheldraht.analysis; classtype:attempted-dos; sid:1855; rev:7;)
alert icmp $EXTERNAL_NET any <> $HOME_NET any (msg:"DDOS Stacheldraht handler->agent ficken"; icmp_id:6667; itype:0; content:"ficken"; reference:url,staff.washington.edu/dittrich/misc/stacheldraht.analysis; classtype:attempted-dos; sid:1856; rev:7;)

Results 1 - 1