Universidad Internacional de La Rioja
Máster universitario en Seguridad Informática

Detección de APT con herramientas de seguridad de carácter libre

Trabajo Fin de Máster

presentado por: Besteiro Calvo, Luis

Director/a: Bermejo Higuera, Javier

Ciudad: Lugo
Fecha: 18 de septiembre de 2016

Índice de contenido

Índice de ilustraciones

Índice de tablas

Resumen

Estudio sobre el contexto y características de las Amenazas Persistentes Avanzadas (APT), así como del uso de herramientas de carácter libre para detectarlas. Se busca profundizar en el conocimiento de las APT, haciendo especial hincapié en los canales encubiertos y en las herramientas utilizadas en sus ataques, así como evaluar la existencia de herramientas del tipo *monitorización de seguridad de red* (NSM) y *gestión de información y eventos de seguridad* (SIEM) de carácter libre para la detección de APT de manera eficaz, con la intención de por un lado apoyar la mejora de la ciberseguridad en las organizaciones, y por otro lado que esta mejora no suponga un coste que lleve a los responsables a frenar su implantación.

Palabras Clave: APT, Canal Encubierto, Software Libre, NSM.

Abstract

Study on the context and characteristics of the Advanced Persistent Threats (APT) and the use of free software tools to detect them. It is the author's goal to deepen in the subject of the APT, focusing on covert-channels and the tools used in the attacks, as well as assessing the existence of Free Software tools for network security monitoring (NSM) and security information and event management (SIEM) for detecting APT effectively, in order to one support an improvement in organizations cibersecurity avoiding high costs slowing down their implementation.

Keywords: APT, Covert Channel, Free Software, NSM.

1. INTRODUCCIÓN

1.1. Motivación

Las Amenazas Persistentes Avanzadas (APT) tienen ya una larga historia como amenazas de facto para la sociedad, bien sea afectando a empresas, a infraestructuras críticas, o a la privacidad y seguridad de las personas.

Desde el punto de vista de la ***motivación personal***, me ha parecido de gran interés el ahondar en la lucha contra dichas amenazas ya que si bien por ahora solo se han traducido en pérdidas económicas o de privacidad, pueden terminar por producir daños personales e incluso pérdidas de vidas si dichas amenazas llevan a cabo ataques contra infraestructuras críticas, bien sea una presa, una central nuclear o una planta de tratamiento de aguas con el objetivo de ocasionar un desastre.

Desde el punto de vista de la ***motivación profesional***, éste es un tema de gran recorrido a la par que interesante. Por un lado, las APT son un fenómeno complejo, dinámico por su constante evolución en el tiempo y adaptabilidad, y que requerirá de su estudio continuo por parte de los profesionales de la ciberseguridad, así como la continua evolución de los métodos y herramientas utilizados por éstos, lo cual conllevará además el hecho de que será un campo en el que se necesitarán más y más profesionales de manera continua, abriendo así una oportunidad laboral. Por otro lado, las características anteriormente mencionadas junto con el hecho de toda la tecnología y tipo de instalaciones afectadas o involucradas con un gran componente tecnológico involucrado (centrales generadoras de energía de distinto tipo, infraestructuras de transporte, industrias...), hacen que sea un tema muy interesante a nivel intelectual.

1.2. Planteamiento de trabajo

1.2.1 Metodología

La ***metodología*** que se seguirá para la realización del presente Trabajo de Fin de Máster (TFM) constará de tres fases generales:

1. Estudio del estado del arte.
2. Creación del entorno de trabajo o laboratorio.
3. Prueba de Concepto o Proof of Concept (PoC).

La primera fase, el ***estudio del contexto y estado del arte***, tiene como objetivo el de alcanzar un dominio en la materia suficiente como para lograr los objetivos planteados en este TFM. Por ello, las etapas seguidas para adquirir dicho dominio consistirán en los distintos aspectos que conforman, describen y contextualizan las APT:

1. Descripción de las APT.
2. Características técnicas de las APT.
3. Historia de las APT.
4. Herramientas APT.
5. Grupos de atacantes conocidos.
6. Grupos gubernamentales.
7. Empresas.
8. Detección de APT
9. Herramientas de carácter libre para detección de APT.

La segunda fase, de ***creación del laboratorio****,* será aquella en la cual se decidirá el *dónde*, el *cómo* y el *qué*, es decir, el software y sistema que utilizaremos como entorno de pruebas, los procedimientos a seguir, y el malware cuyo funcionamiento analizar.

La última fase, de ***PoC***,será aquella en la que llevaremos a cabo el análisis del funcionamiento del distinto malware considerado para este TFM, englobado dentro de las herramientas APT.

1.2.2 Objetivos

El **objetivo general** que se ha establecido para este TFM consiste en aportar una mejora a la situación actual en la lucha contra las APT. Este objetivo general, viene definido por unos objetivos específicos que definen de una manera más precisa lo que se pretende lograr con la realización del presente TFM. A continuación se enumeran dichos **objetivos específicos**:

1. Realizar el ***estudio del arte*** sobre las APT, identificando y comprendiendo sus características, así como su contexto e historia.
2. Identificar y estudiar las ***metodologías de un ataque*** APT.
3. Identificar y estudiar de las ***metodologías de detección*** de APT
4. Identificar los distintos ***actores*** en el escenario de las APT.

5. Identificar y analizar las ***herramientas de carácter libre*** que sean útiles para la detección de ataques APT.
6. Implementar un ***entorno de prueba*** que nos permita simular un entorno real para comprobar el funcionamiento tanto de las herramientas APT como de las herramientas de carácter libre para detección de APT.
7. Buscar una ***detección*** lo más ***eficaz y eficiente*** posible.
8. Identificar y analizar ***tendencias futuras*** en el campo de las APT.

1.3. Estructura del trabajo

El presente trabajo se desarrolla con una estructura tipo introducción-cuerpo-desenlace con la intención de que por un lado su lectura sea cómoda, por otro lado genere interés en el lector, además de lograr que los conceptos sean introducidos en el momento adecuado y con la profundidad adecuada para que el lector no se vea ni abrumado por una cantidad excesiva de información no necesaria en el apartado, ni desubicado por la falta de de ésta.

Para comenzar, se pondrá en contexto técnico al lector, llevando a cabo primeramente la descripción de las APT con sus características técnicas, para pasar a continuación al contexto histórico-social. Después de poner al lector en contexto, tanto técnico como histórico, se pasará a exponer las herramientas, metodologías y actores principales en el campo de las APT. A mayores serán mencionados grupos gubernamentales que luchan contra estas amenazas, así como empresas que se dedican al desarrollo de herramientas para protección de sistemas. Finalmente se enumerarán y describirán las técnicas de detección de APT y las herramientas de carácter libre que nos permitirán detectarlas.

Como “cuerpo” del TFM tendremos en primer lugar la descripción de los objetivos y la metodología del trabajo, que indicarán y describirán el Por qué, el Cómo y el Dónde qué. A continuación, tendremos el desarrollo específico, es decir, una descripción detallada del experimento, la presentación de los resultados y la discusión sobre éstos.

Para finalizar, se mostrará un análisis personal sobre el TFM realizado, se mencionarán las contribuciones de dicho trabajo al estado del arte, y ya como broche final se comentarán posibles líneas de trabajo futuro.

A continuación se enumeran y describen los distintos **capítulos** de que consta el documento:

1 Introducción

1.1 Motivación

En este apartado se comentará la razón de por qué se ha elegido el tema tratado en el TFM, por qué creo que es un tema interesante a tratar y qué creo que puedo aportar.

1.2 Planteamiento del trabajo

Aquí se llevará a cabo una introducción al TFM, haciendo un breve resumen de cómo se afrontará la búsqueda de una solución para el problema y qué objetivos se esperan cumplir al terminar la investigación.

1.3 Estructura del trabajo

En esta parte se hará una breve narrativa de cómo transcurrirán los siguientes capítulos, introduciendo así un resumen del transcurrir de este documento.

2 Contexto y estado del arte

2.1 Descripción de las APT

En este apartado se llevará a cabo una introducción a las Amenazas Persistentes Avanzadas, describiéndolas, comentando la situación actual, y en resumen, estableciendo el contexto de este TFM.

2.2 Características técnicas de las APT

En este apartado se abordará de una manera más técnica y detallada las características de los ataques de tipo APT.

2.2.1 Metodología de un ataque

2.2.2 Covert Channel

2.3 Historia de las APT - Noticias – Ataques conocidos

Para dar un contexto a este TFM, se llevará a cabo en este apartado un resumen sobre la historia de las APT, aportando noticias que han protagonizado, ataques conocidos, etc.

2.4 Herramientas APT

Ya que este trabajo trata de cómo detectar las APT, es importante comentar qué herramientas existen hoy en día para lidiar con ellas, nombrando no solo las utilizadas en este proyecto, que serán de carácter libre, sino también las de carácter comercial.

4.2 Presentación de los resultados

En este capítulo se detallarán los resultados obtenidos, y se transformarán en gráficos y tablas que permitan tener una referencia clara y una comprensión rápida del producto del experimento.

4.3 Discusión de los resultados

Este será el apartado en el cual se valorarán los resultados obtenidos en el experimento, aportando una justificación del producto, una explicación para los distintos datos obtenidos resaltando aquellos que sean relevantes, para finalmente explicar la relevancia de los resultados.

5 Conclusiones y trabajo futuro

5.1 **Análisis sobre el TFM**

Se llevará a cabo un resumen del problema tratado y de qué se ha obtenido como producto.

5.2 Contribuciones del trabajo

En este capítulo se comentará qué aporta el producto obtenido a la situación actual de las APT, y se llevará a cabo un análisis o relación sobre el objetivo y inicial y los resultados obtenidos finalmente.

5.3 Líneas de trabajo futuro

En esta parte final del TFM se llevará a cabo una valoración de cómo podría seguir investigándose en este campo a partir de los resultados obtenidos así como de qué usos tendría el producto obtenido herramienta en un entorno real de operación.

2. OBJETIVOS Y METODOLOGÍA DE TRABAJO

2.1. Objetivo general

El objetivo general establecido para este TFM consiste en ***contribuir a la mejora en la lucha contra las APT***, y esto será llevado a cabo mediante la consecución de unos objetivos específicos que pretenden por un lado analizar las APT, y por otro conseguir la implementación eficaz y eficiente de un sistema que, constituido por ***herramientas de seguridad de carácter libre*** de modo que sea asequible para cualquier organización, nos permita detectar ataques APT.

2.2. Objetivos específicos

Los pasos que se seguirán para poder cumplir con el objetivo general serán aquellos que nos permitan llevar a cabo un análisis pormenorizado del estado del arte para comprender en profundidad el problema, y así poder después establecer una metodología para luchar eficazmente contra él.

Los objetivos específicos establecidos para este TFM son los siguientes:

1. Realizar el ***estudio del arte*** sobre las APT, identificando y comprendiendo sus características, así como su contexto e historia.
2. Identificar y estudiar las ***metodologías de un ataque*** APT.
3. Identificar y estudiar de las ***metodologías de detección*** de APT
4. Identificar los distintos ***actores*** en el escenario de las APT.
5. Identificar y analizar las ***herramientas de carácter libre*** que sean útiles para la detección de ataques APT.
6. Implementar un ***entorno de prueba*** que nos permita simular un entorno real para comprobar el funcionamiento tanto de las herramientas APT como de las herramientas de carácter libre para detección de APT.
7. Buscar una ***detección*** lo más ***eficaz y eficiente*** posible.
8. Identificar y analizar ***tendencias futuras*** en el campo de las APT.

2.3. Metodología del trabajo

Para llevar a cabo este trabajo, la metodología a seguir consiste en acercarse al objetivo general por dos flancos diferentes: por un lado la búsqueda de información o estudio del estado del arte, y por otro lado, y ya que es una trabajo experimental, la implementación de un entorno controlado de pruebas y la realización de éstas.

Estudio del Estado del Arte

Para la realización de cualquier proyecto de una manera correcta hay que comenzar por el estudio del arte, y este se puede dividir en las siguientes etapas:

1. Desarrollo de la estructura del documento.
2. Búsqueda de fuentes de información.
3. Búsqueda de información.
4. Lectura y análisis de la información.
5. Redacción del documento.

Una vez llevadas estas tareas a cabo, se procederá al diseño del entorno de pruebas, el cual vendrá dado por la necesidad de reproducir en un entorno controlado la situación tratada, es decir, una APT.

Implementación del Entorno de Pruebas

En lo relativo a la implementación del entorno de pruebas, se tienen que abordar dos puntos diferentes: la estructura o configuración de la red, y la configuración de los equipos de la red.

- **Estructura de la red**:
  La estructura de la red elegida para implementar el entorno de pruebas será aquella que integre todos los elementos necesarios para llevar a cabo el experimento, evitando cualquiera que no sea explícitamente necesario, para evitar así errores, e información en los análisis no necesaria que pueda desviar la atención de lo realmente importante. Además, deberá ser un *entorno aislado* de otras redes de modo que se asegure la no propagación del malware o sus efectos.
- **Configuración de los equipos de la red**:
  Los equipos de los que constará la red serán aquellos que representen:

- Atacante: Será el equipo que represente al atacante que implanta el malware en el equipo víctima y que actúa como C&C para la conexión remota del malware a través de un canal encubierto.
- Víctima: Entendido como el equipo de usuario infectado por el malware y desde el que se produce la exfiltración de información.
- Monitorización: Este será el equipo que lleve a cabo la monitorización y análisis de la red, utilizado para detectar posibles amenazas, en este caso un ataque APT.

Los **pasos** a seguir para la implementar el entorno de pruebas serán:

1. Elegir el software de virtualización.
2. Determinar la configuración del software de virtualización.
   a) Configuración de Red.
   b) Configuración de las máquinas virtuales.
3. Elegir el software para los siguientes equipos:
   a) Víctima.
   b) Atacante.
   c) Monitorización y análisis de Red.
4. Determinar la configuración adecuada de los equipos para las pruebas.
5. Determinar el malware RAT a utilizar para simular el ataque APT.
6. Instalar el software de virtualización.
7. Configurar el software de virtualización.
8. Crear las máquinas virtuales con los Sistemas Operativos elegidos.
9. Configurar los equipos.

Metodología de Realización de las Pruebas

Una vez implementado el entorno de pruebas, es hora de llevar a cabo el experimento. Para ello, se han de seguir unas pautas que permitan llevar a cabo un ciclo continuo de pruebas objetivas en las que obtener resultados fiables.

Las etapas a seguir para la realización de las pruebas es la siguiente:

1. Entorno de trabajo sin malware.
   (1) Simular un entorno de funcionamiento normal, en el cual no se produce un ataque APT.

(2) Analizar la actividad de la red.
(3) Establecer los indicadores base.

2. Determinar el tipo de ataque a realizar y el malware-RAT a utilizar.
3. Entorno de trabajo con malware.
   (1) Determinar el malware-RAT a instalar en el equipo víctima.
   (2) Instalar el malware-RAT en el equipo víctima.
   (3) Configurar el malware para que se produzca la comunicación con el atacante.
   (4) Seleccionar los archivos a extraer.
   (5) Realizar la extracción de la información.
   (6) Mediante el software de monitorización y análisis de red:
      1. Analizar la variación de los valores de monitorización con respecto a los indicadores base establecidos.
      2. Analizar la respuesta del sistema de monitorización y análisis para averiguar si es capaz de detectar el ataque.
      3. Captar el tráfico de red para su posterior análisis.
   (7) Parar el funcionamiento del malware.
4. Evaluar los resultados de la prueba.
5. Realizar las modificaciones pertinentes en el software de monitorización y análisis de red para mejorar la eficiencia y eficacia de la detección.
6. Repetir los pasos 3.2 a 5 hasta que no se consiga mejorar la capacidad de detección del ataque APT por parte del software de monitorización y análisis de red.

3. CONTEXTO Y ESTADO DEL ARTE

3.1. Descripción de las APT

Las Advanced Persistent Threat (APT), o Amenazas Persistentes Avanzadas, constituyen uno de los desafíos de seguridad más importantes y peligrosos que deben afrontar hoy en día las organizaciones, y consisten en ataques de red de elevada sofisticación en los que una persona o ente con amplios recursos consigue acceso a una red o sistema y permanece sin ser detectado durante un largo periodo de tiempo. ***Aprovechan*** para acceder a su objetivo la falta de concienciación de los usuarios y ponen en juego también las debilidades TIC (defectos conocidos, implementaciones inadecuadas, vulnerabilidades zero-day, código malicioso de diseño específico…) y las de las arquitecturas de seguridad (Mariscal Fragoso, C., Candau Romero, J., Septiembre 2014). Tienen por ***finalidad*** conseguir acceso a sistemas de diversa índole, mantener el acceso para un futuro uso y control, y obtener información del objetivo, y/o modificarla con la intención de llevar a cabo un sabotaje, o un ataque de "consecuencias físicas" (entendiendo éstas como la avería de maquinaria, provocar víctimas, etc).

Bejtlich (2007) explica los **componentes de la terminología**:

- **Avanzada** (Advanced): significa que el adversario está versado en herramientas y técnicas de intrusión informática y es capaz de desarrollar *exploits personalizados*.
- **Persistente** (Persistent): significa que el adversario intenta cumplir una misión. Recibe una directivas y trabaja con *metas específicas*.
- **Amenaza** (Threat): significa que el adversario está *organizado*, *financiado* y *motivado*.

Las acciones APT suelen tener como **origen** *gobiernos* o *empresas*, que haciendo uso de grandes recursos, llevan a cabo operaciones de inteligencia y/o sabotaje contra otras organizaciones, bien sea por **motivos** *económicos* (perjudicar a la competencia, robar información confidencial para obtener una ventaja competitiva…) o de *seguridad nacional* (obtener información sobre el enemigo, tecnologías militares avanzadas, vulnerabilidades…).

Los **objetivos típicos** de las APT suelen ser *políticos* (provocar desestabilización o desorganización, debilitar misiones diplomáticas...), *económicos* (robo de propiedad intelectual...), *técnicos* (credenciales, código fuente...) o *militares* (identificación de vulnerabilidades, robo de información sobre seguridad nacional…), siendo las **víctimas habituales** organizaciones en sectores con información de alto valor, tal como defensa nacional, manufactura, industria financiera...

The Anatomy of an Advanced Persistent Threat (Cutler, 2010) describe la **estrategia APT típica**:

1. El atacante consigue ***acceso*** en el sistema víctima mediante ingeniería social y malware.
2. El atacante abre una terminal de la ***shell*** del sistema víctima para averiguar si el sistema tiene mapeado algún disco en red.
3. El que el sistema víctima tenga un disco mapeado en red dará pie al atacante a iniciar un ***escaneo de puertos*** desde el sistema víctima.
4. El *atacante* ***identifica*** por consiguiente *puertos abiertos*, *servicios* corriendo en otros sistemas y *segmentos de red*.
5. Una vez mapeada la red, el *atacante se enfoca hacia* ***víctimas VIP*** con activos de elevado valor a su disposición.

3.2. Características técnicas de las APT

Las Amenazas Persistentes Avanzadas o APT poseen unas características técnicas que las definen, y que vienen dadas por las necesidades del tipo de atacante que se esconde tras ellas.

A continuación se enumera muestra de dichas características son:

- Uso de ***ingeniería social*** para lograr acceso al sistema bien sea a través de credenciales o mediante la infección del sistema con malware.
- Se hace uso de herramientas/malware del tipo ***Remote Administration Tool*** (RAT).
- El ***malware ha sido personalizado*** para que se adapte mejor a la/s víctima/s.
- Hacen uso de ***vulnerabilidades de día-cero*** para atacar a las organizaciones objetivo.

- Cuando se habla de ataques de APT se habla de ***organización***, ***premeditación***, ***persistencia***, ***sofisticación*** y ***novedad***.
- Capacidad de pasar ***desapercibidas***.
- ***Movimiento lateral*** o *pivotaje* del malware para evitar ser detectado y acceder a nuevos recursos.
- Se produce ***extracción*** *de la información*.
- Hay un ***centro de mando y control*** (C&C) que monitoriza y extrae la información de la víctima.

El **ciclo de vida de una APT** es elevado, de hecho "persistente" es una de las características intrínsecas de las APT. A continuación se muestra una comparativa entre el ciclo de vida de una APT y otras ciberamenazas:

Figura 1: Diagrama que describe el ciclo de vida por fases de los distintos tipos de amenazas informáticas, entre las que se encuentran las APT.

Imagen recuperada de
https://commons.wikimedia.org/wiki/File:Advanced_persistent_threat_lifecycle.jpg

A continuación se muestra una tabla comparativa entre los tres tipos de ataques mostrados en la figura anterior:

Tabla 1: Comparativa entre tipos de ataques.

	APT	Amenaza Avanzada	Hacktivismo
Selección del objetivo	Objetivo específico	Aleatorio	Objetivo específico
Motivación	Financiera, espionaje, robo de propiedad intelectual, obtener control de una infraestructura crítica, sabotaje	Ganancia financiera	Política: libertad de expresión, derechos humanos, promoción de una ideología determinada, protesta contra una situación determinada, contra una empresa, gobierno...
Vectores de ataque	Combinación de múltiples vectores de ataque y herramientas. Herramientas desarrolladas específicamente para el ataque. Técnicas de inteligencia. Intervención telefónica. Robo físico.	Troyanos Vulnerabilidades “zero-day” Exploits conocidos	Troyanos Vulnerabilidades “zero-day” Exploits conocidos
Solución	Aun si se detectan y se corrigen los nodos infectados en una red, es muy posible que los atacantes tengan planes de contingencia y nodos redundantes que les permitan continuar sus operación.	Técnicas de virtualización. Parches de vulnerabilidades. Sistemas de autenticación robustos. Software antimalware.	Técnicas de virtualización. Parches de vulnerabilidades. Sistemas de autenticación robustos. Software antimalware.

3.2.1 Metodología de un ataque

Podemos decir que las etapas en que se divide un ataque son las siguientes:

1. **Selección de la víctima**.
   La primera etapa de cualquier ataque APT consiste en escoger la víctima, sea ésta una organización, un personaje público, etc. Algunos atacantes escogen primero la víctima y luego llevan a cabo un trabajo de recolección de información sobre dicha víctima como por ejemplo buscando currículums de los empleados, información en la red sobre la compañía, software y dispositivos utilizados y sus vulnerabilidades… Otros atacantes crean primero el malware, y luego buscan víctimas accidentales.
2. **Recolección de información**.
   Los atacantes llevan a cabo un *estudio minucioso sobre la víctima* con la intención de crear un perfil sobre los sistemas utilizados, posibles vulnerabilidades, etc. Para llevar a cabo esta fase se utilizarán técnicas de *footprinting*, *phishing*, *fingerprinting*.
3. **Infección**.
   Las APT suelen utilizar diversas vías de entrada para acceder a la organización objetivo. Una vez analizados los datos adquiridos en el paso anterior, escogerán el mejor vector de ataque con el que puedan conseguir su objetivo. Se distinguen cuatro grandes grupos de vías de infección en las campañas APT: infección por *malware*, *medios físicos*, *exploits* y *Web-based attacks*.
   El vector más común de ataque observado suele ser el de envío de correos electrónicos maliciosos dirigidos (*Spear-phishing Attacks*) usando técnicas de ingeniería social, normalmente combinándolos con exploits 0-day, URLs o documentos maliciosos anexados.
4. **Escalada de privilegios**.
   En un comienzo, el atacante recolecta *credenciales de acceso* de los usuarios o dispositivos comprometidos (usuario, dominio, contraseña, cuentas…). A continuación, se lleva a cabo el escalado de privilegios en usuarios no administradores dentro de los sistemas objetivo para, de seguido, hacer lo mismo pero en objetivos de alto valor, como expertos de procesos, administradores de IT, servidores…
   Para conseguir las credenciales de identificación, los atacantes recurren a *keyloggers*, *ARP spoofing*, y herramientas de acceso o "*hooking tools*". Una muestra de herramientas utilizadas para obtener credenciales: *Pwdump*, *Windows Credential*

Editor, *Mapiget*, *Lslsass*, *Gsecdump* y *CacheDump*. Otras técnicas para hacerse con las credenciales serían los ataques de fuerza bruta y de diccionario.

5. **Comunicación con el *centro de mando y control***.
   Una vez dentro de la organización objetivo, las APTs son habitualmente dirigidas de manera remota, vía comunicaciones de "mando y control" (C&C) entre los sistemas infiltrados y los propios atacantes. Este canal de comunicación será utilizado para abrir y manipular puertas traseras de acceso a la red para descubrir y exfiltrar la información deseada.
6. **Pivoting o movimiento lateral**.
   Es un proceso que consiste en que cuando un atacante compromete un sistema, usa dicho sistema para explorar otros sistemas dentro de la misma red, para posteriormente infectarlos y conseguir *sobrepasar el perímetro de seguridad*.
7. **Descubrimiento de activos y persistencia**.
   Diversas técnicas como el *escaneo de puertos* y el *análisis de red* son usadas para identificar servidores valiosos y servicios que alojan información de interés. Además, el escaneo de puertos también es utilizado para poder crear una *conexión tunelizada* entre el sistema comprometido y el sistema atacante, y así poder evadir el firewall.
8. **Exfiltración de la información**.
   Consiste en la transferencia no autorizada de información sensible del sistema objetivo al sistema externo controlado por los atacantes. Normalmente, después de descubrir la información de interés, la APT reúne la información en un archivo y luego lo comprime y cifra, para así mantenerlo oculto de los análisis profundos de paquetes y de las técnicas de prevención de pérdida de información. Una vez realizado esto, el siguiente paso consistirá en la extracción de la información del sistema víctima.
   Con la intención de que el tráfico exfiltrado no levante sospechas en los sistemas de seguridad, éste se "camufla" mediante distintas ***técnicas*** como la *esteganografía*, la *compresión* y el *cifrado* acompañados de su *troceado*.
   Las herramientas utilizadas son copiadas en las máquinas infectadas y en muchas ocasiones, posteriormente eliminadas para no dejar rastro.
9. **Encubrimiento de las huellas**.
   Una vez que el o los atacantes han cumplido con su cometido, buscarán no dejar ninguna huella que dé pistas sobre sus operaciones encubiertas.

En la siguiente imagen podemos ver una representación de las distintas etapas del ataque:

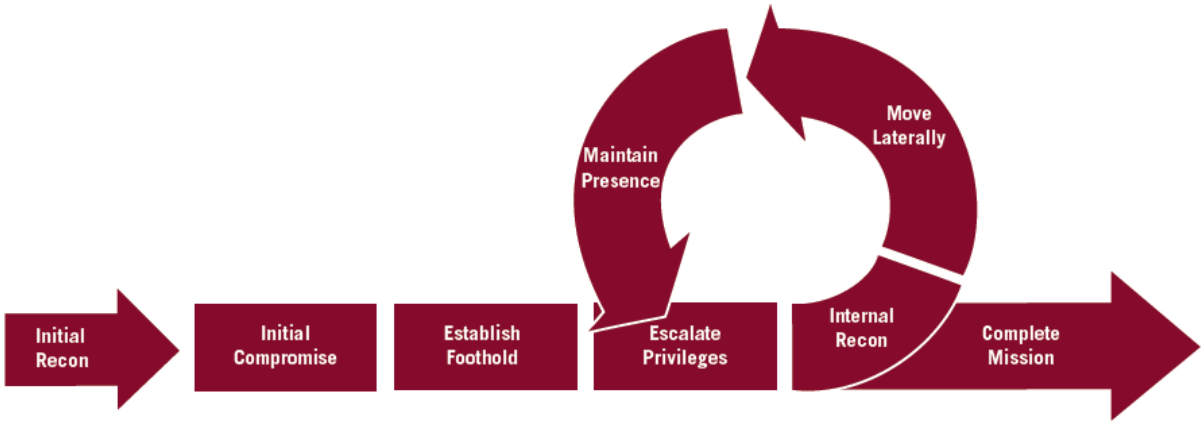


Figura 2: Modelo Mandiant del ciclo de vida de una APT.

(Mandiant, 2013)

A continuación, se muestra una representación más gráfica del ataque APT:

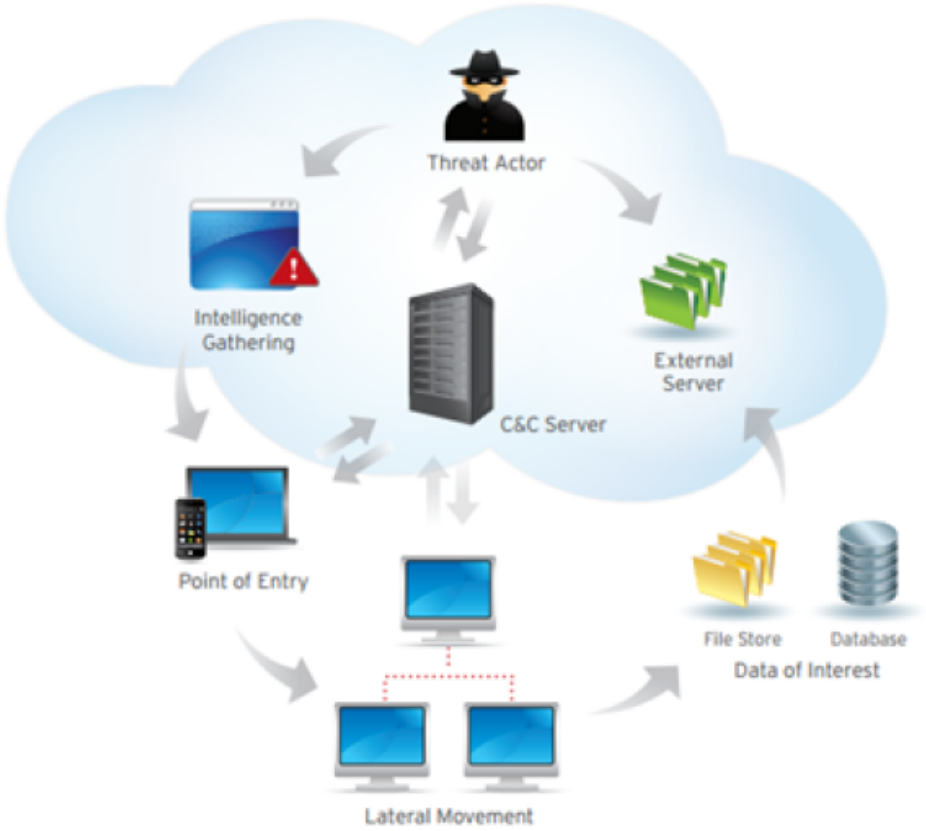


Figura 3: Ilustración de una operación APT.

Recuperada de http://resources.infosecinstitute.com/anatomy-of-an-apt-attack-step-by-step-approach/

Una vez que el atacante obtiene la información, entonces lleva a cabo diversas actividades criminales como:

- *Vender* la información.
- *Coaccionar* a través de la amenaza de la revelación pública de la información.
- Pedir a la víctima que pague un *rescate*.

A continuación se describen en mayor profundidad algunas de las etapas del ataque.

3.2.2 Exfiltración de la información

A continuación se describen dos técnicas características de la exfiltración de información en APT y cuya comprensión es de vital importancia para poder detectar estos ataques.

3.2.2.1 Redes Fast-Flux

La técnica fast-flux es utilizada cuando ***el atacante desea dificultar el seguimiento de la exfiltración de la información*** (Binde, McRee, O'Connor, 2011).

Las redes fast-flux han sido documentadas por el Honeynet Project en *Know Your Enemy: Fast-Flux Services Networks* [HoneyNet Project, 2007]: “La **meta** de la técnica fast-flux es, para un dominio dado (tal como www.ejemplo.com), tener múltiples (cientos o incluso miles) direcciones IP asignadas. Estas direcciones IP son intercambiadas para el flujo de datos con mucha frecuencia, utilizando una combinación de direcciones IP de round-robin y un Time-To-Live (TTL) muy bajo para cualquier DNS Resource Record (RR). Los nombres de sitios web pueden llegar a estar asociados con un nuevo grupo de direcciones IP con una periodicidad de 3 minutos. De hecho, un navegador que estuviera conectándose al mismo sitio web cada 3 minutos, estaría realmente conectándose a una computadora infectada diferente cada vez. Además, los atacantes se aseguran de que los sistemas comprometidos que están utilizando tengan los mejores ancho de banda y disponibilidad del servicio posibles.”

En resumen, las redes Fast-Flux son redes formadas por equipos comprometidos a los que apuntan los registros DNS de un determinado dominio, y *actúan como proxy entre los clientes y los servidores donde se almacena el contenido*.

Tipos de Redes Fast-Flux

Dentro de este tipo de redes tenemos dos tipos: single-flux y double-flux:

- **Redes Single-flux**:

  Cuando una víctima intenta acceder a un determinado dominio, la respuesta del DNS se corresponde con una dirección IP de los equipos infectados, los cuales varían continuamente, capturando la petición del cliente y siendo ellos quienes negocien con el servidor donde se aloja el contenido.

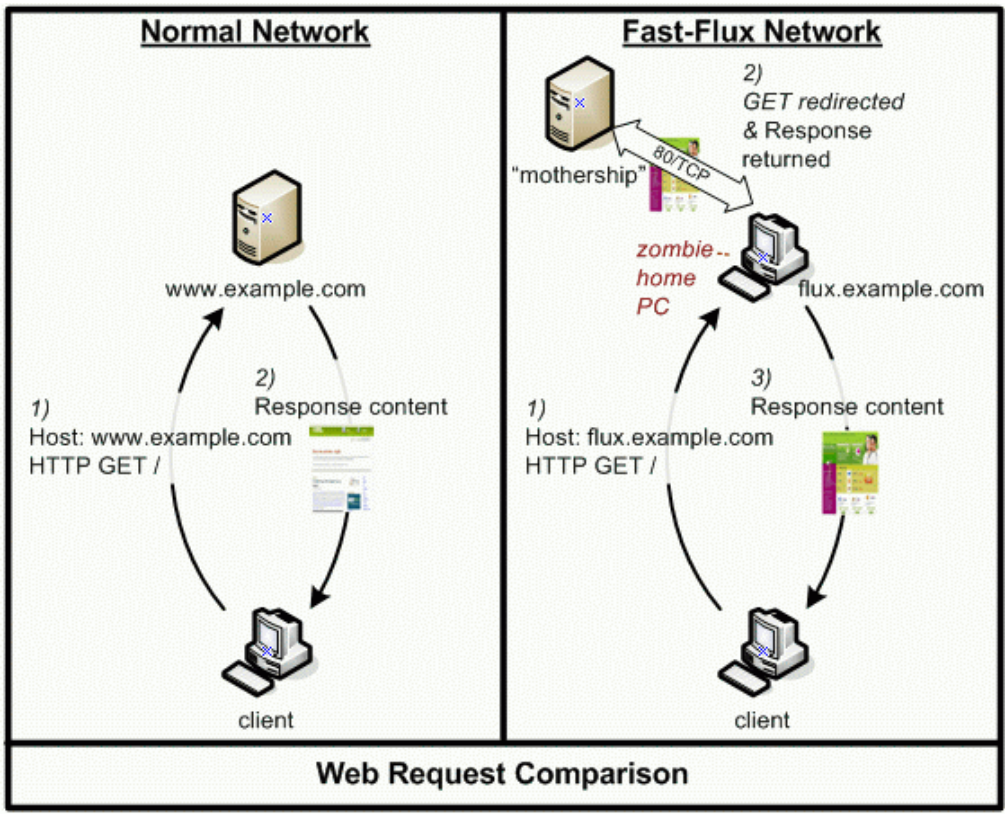


Figura 4: Comparativa entre funcionamiento de una red normal y una red Fast-Flux. Recuperado de http://www.honeynet.org/node/134

- **Redes Double-flux**:

  Mantienen el concepto de las single-flux pero añadiendo una capa más de redundancia. En este tipo de implementaciones, además de variar los registros A del DNS para que un mismo dominio resuelva direcciones IP diferentes, también varían los registros NS correspondientes a los servidores DNS autorizados. Por lo tanto, en este caso, las peticiones DNS de las víctimas son respondidas directamente por la red Fast-Flux.

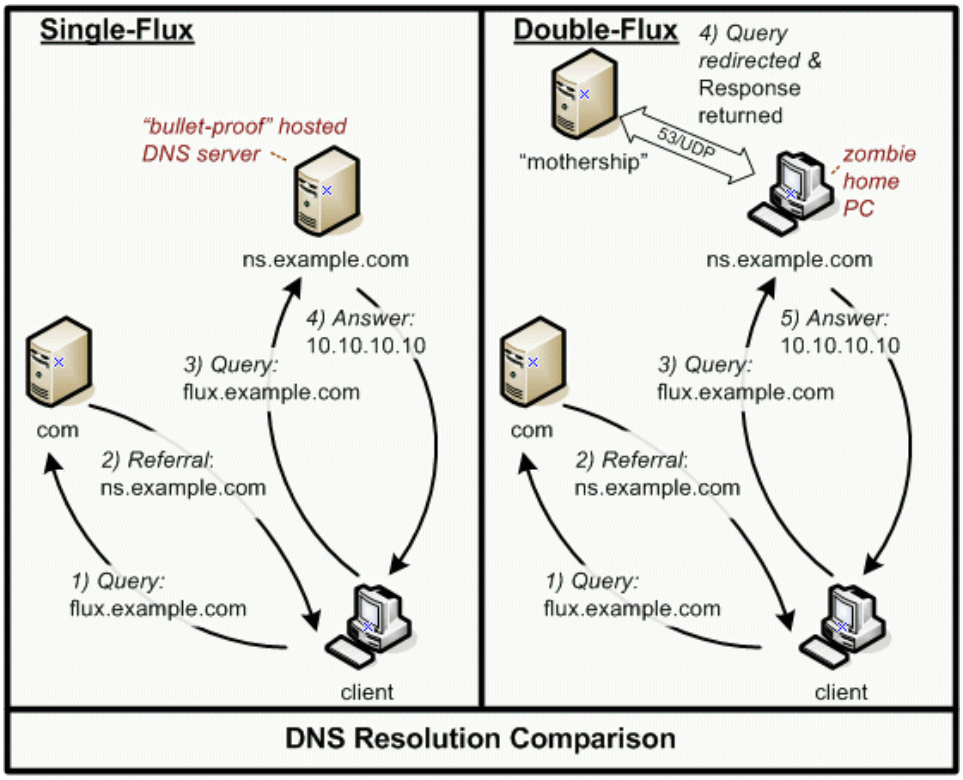


Figura 5: Comparativa entre Single-Flux y Double-Flux.

Recuperado de http://www.honeynet.org/node/136

Ventajas que obtienen los atacantes

A continuación se nombran las ventajas que obtienen los atacantes mediante el uso de este tipo de arquitecturas:

- Simplicidad: Los atacantes necesitarán menos infraestructura, además de ahorrar tiempo en mantenimiento, ya que antes sus servidores centrales estaban en "primera línea" y eran fácilmente identificables, con lo que necesitaban grandes recursos si querían persistir.
- Protección ante investigaciones: Al existir una o varias capas más de conexión es difícil rastrear las huellas y llegar a los servidores centrales. A esto se le suma que habitualmente los equipos infectados se encuentran repartidos por multitud de países con jurisprudencias diferentes y configurados con las auditorías desactivadas para no dejar evidencias.
- Extensión de la vida útil de la estafa: El dinamismo de la red, unido a las diferentes capas de conexionado tras las que se ocultan los servidores centrales donde se aloja el contenido, hacen que todo el proceso de identificación y corte del servicio sea mucho más costoso y por tanto mucho más largo.

3.2.2.2 Covert Channel

Se conoce como *covert channel,* o *canal encubierto*, a un *canal que puede ser utilizado para transferir información desde un sistema a otro, utilizando medios no destinados para este propósito por los desarrolladores del sistema*. Normalmente, para que la comunicación sea posible, suele ser necesario un *preacuerdo* entre el emisor y el receptor, para que así el mensaje sea codificado de una manera que el receptor sea capaz de interpretar y/o localizar.

Este concepto fue introducido en 1973 por Butler W. Lampson cuando estaba examinando el problema de mantener secreta la información de un proceso para otro en un sistema con seguridad multinivel (*problema de confinamiento*). Así, el autor introduce el concepto de *covert channel* para identificar los "*canales que un proceso tiene disponibles para transmitir información, a pesar de no haber sido éstos diseñados para ese propósito*". A partir de esta definición, el Departamento de Defensa de los Estados Unidos (DoD) define los *covert channels* en su TCSEC (también conocido como "libro naranja") como "*cualquier canal de comunicación que puede ser explotado por un proceso para transferir información de forma que viole la política de seguridad del sistema*".

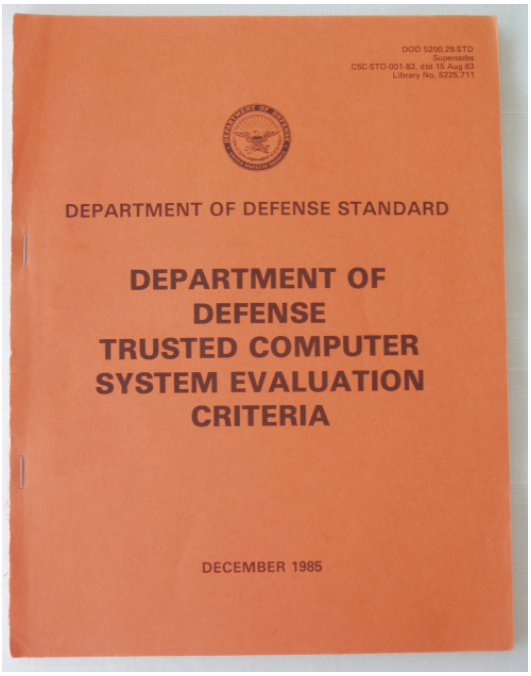


Figura 6: TCSEC (también conocido como "libro naranja").

Recuperado de https://commons.wikimedia.org/wiki/File:Orange-book-small.PNG

Posteriores definiciones como la del Common Criteria, "*un canal de señalización forzado e ilícito que permite a un usuario contravenir subrepticiamente la política de seguridad multinivel y requisitos de no-obserbabilidad del sistema*" y el "Handbook for the Computer Security Certification of Trusted System" del Laboratorio de Investigación Naval de los Estados Unidos muestran que con el paso del tiempo el concepto ha ido evolucionando desde el ámbito de los sistemas operativos y el problema del confinamiento de los procesos hasta abarcar cualquier tipo de comunicación entre entidades que utilizan canales que no han sido diseñados para ser utilizados como canal de comunicación.

Clasificación de los canales encubiertos

Los canales encubiertos se clasifican habitualmente según el *mecanismo de ocultación* en: ***canales de almacenamiento*** (*Covert Storage Channel*), ***canales de temporización*** (*Covert Timing Channel*), ***canales ocultos*** (*Side Channel*). Sin embargo, algunos autores plantean que ésta no es una clasificación correcta y que deberían de utilizarse otras (Ríos y Onieva, 2008).

Canales de almacenamiento

El receptor de la información percibe la transmisión de dicha comunicación como un *cambio en el valor de un atributo o recurso compartido*. Para tener un canal de almacenamiento, se deberán cumplir como mínimo los siguientes criterios (Kemmerer, 1983):

1. Tanto el proceso receptor como el proceso emisor deben tener acceso al mismo atributo o recurso compartido.
2. Debe haber algún medio mediante el cual el proceso emisor puede forzar al atributo compartido a cambiar.
3. Debe haber algún medio mediante el cual el proceso receptor pueda detectar el cambio en el atributo.
4. Debe haber algún mecanismo para iniciar la comunicación entre los procesos emisor y receptor y para la correcta secuenciación de los eventos. Para ello podría usarse otro canal de un ancho de banda menor."

Ejemplos de dichos canales serían:

- Ciertos campos en la cabecera de los paquetes de red que bien están en desuso o que su modificación no afecta al correcto funcionamiento del protocolo.
- Mediante el uso de zonas de memoria compartida, como bloqueos de ficheros del disco duro o la modulación de la ocupación del espacio de disco.

Canales de temporización

Este tipo de canales puede definirse como aquel en el que el receptor de la información la percibe por medio de un *cambio en el tiempo requerido por el destinatario para detectar cierta acción*. Para tener un canal de temporización, se deberán cumplir como mínimo los siguientes criterios (Kemmerer, 1983):

1. Tanto el proceso receptor como el proceso emisor deben tener acceso al mismo atributo o recurso compartido.
2. Los procesos emisor y receptor ha de tener acceso a una *referencia temporal*, tal como un reloj de tiempo real.
3. El emisor debe ser capaz de modular la respuesta temporal del receptor para detectar un cambio en el atributo compartido.
4. Hay un mecanismo para iniciar el proceso y para secuenciar los eventos.

Ejemplos de este tipo de canales serían:

- Tiempo de uso de la CPU.
- Tiempo de comienzo de un proceso.
- Demanda de ejecución de determinada tarea observable por el otro usuario.

Canales ocultos

Los casos en que, al contrario que en los casos previamente mencionados, ***emisor y receptor no acuerdan cómo realizar la comunicación***, y el emisor envía información que el receptor debe descubrir cómo interpretar, se conocen como *canales ocultos* o *side channel*. En estos sistemas el desafío del receptor es descubrir y decodificar el canal oculto para obtener la información. Los *canales ocultos más habituales* son aquellos que están relacionados con las emanaciones que de forma no intencionada emiten los dispositivos electrónicos.

Otras clasificaciones

Gligor (1993) propone dos nuevas clasificaciones de los covert channels:

- **En función de la cantidad de ruido que afecta al canal** (fiabilidad del canal):
  - Canales sin ruido (*Noiseless Channels*): los símbolos que son transmitidos por el emisor son los mismos que los recibidos por el receptor con probabilidad 1.
  - Canales con ruido (*Noisy Channels*): existe cierta probabilidad de que cualquier símbolo enviado por el emisor sea recibido de manera incorrecta por el receptor.
- **En función de la cantidad de procesos que se comunican simultáneamente mediante la utilización de zonas de memoria** (variables):
  - Canales desagrupados (*Non-Aggregated Channels*): las variables utilizadas para creación de un canal encubierto son accedidas únicamente por un par de procesos, el emisor y el receptor.
  - Canales agrupados (*Aggregated Channels*): creados por varias parejas de procesos que acceden a un mismo conjunto de variables. Estos pueden ser en *serie* (en cola), en *paralelo* (simultáneos) y *mixtos*.

Por su parte Ríos y Onieva (2008) plantean dos posibles criterios de clasificación:

- **Según el número de procesos que intervienen en la comunicación**:
  - Canales desagrupados: creados para el envío de información entre únicamente dos equipos de la red.
  - Canales agrupados en serie, paralelo o mixtos: aquellos en los que la información transmitida puede ser aprovechada por más de una pareja de equipos de la red.
- **Según el medio que se utiliza para ocultar la información**:
  - Canales de almacenamiento: utilizan una zona dentro del paquete, bien sea la cabecera o bien la zona de datos.
    - *Basados en valor*.
    - *Basados en transición*.
  - Canales de temporización: Ocultan la información en los patrones de llegada de los paquetes al receptor. La recepción o no de paquetes está regida por relojes, lo cual codifica la información.
    - *Canales de contabilización*: codifican la información en la cantidad de eventos que tienen lugar durante un determinado periodo de tiempo.
  - Canales de ordenación: aquellos que ocultan la información en el orden de llegada de los paquetes.

- Canales combinados o híbridos: resultan de la combinación de las técnicas empleadas por algunos de los tipos anteriores.

Covert channels en los protocolos de red

Aunque prácticamente cualquier protocolo puede ser utilizado como canal encubierto para transmitir información, existen razones prácticas para enfatizar algunos en concreto, y es que la gran mayoría de la investigación realizada sobre covert channels se ha focalizado en protocolos de las capas 3 y 4 (Red y Transporte) tales como ICMP, IP y TCP, y protocolos de la capa 7 (Aplicación) tales como HTTP y DNS, del modelo de referencia *Open System Interconnection* (OSI). El ***factor principal en la selección de estos protocolos*** no es solo su prevalencia en Internet, sino que también el hecho de que sea *práctica común permitir por defecto su tráfico a través de dispositivos de protección de red*, para permitir aplicaciones cliente legítimas (Couture, E., 2010).

Tanto los protocolos IP, TCP, UDP como los protocolos a nivel de aplicación **pueden ser explotados para Covert Channels** de tipo *almacenamiento* y de tipo *temporización* **debido básicamente a**:

- Una *definición pobre* o indefinición de los protocolos.
- Cabeceras o porciones del paquete que no se utilizan.
- La conducta inherente del routing basado en el destino.
- A la conducta de los protocolos a nivel de aplicación.

Las **técnicas** para transmitir información por medio de covert channels de tipo *almacenamiento* son las siguientes (Couture, E.,19 de agosto de 2010):

- **Header bit modulation** (modulación de bit de cabecera):
  - La información encubierta puede ser codificada en un bit que no es crítico para la correcta transmisión del protocolo del host.
  - Ancho de banda de 1 bit.
  - Método ineficiente para usos prácticos.
- **Header bit crafting** (confección de bit de cabecera):
  - La información encubierta se codifica en varios bits de la cabecera o incluso en bytes completos en ciertos casos.
  - Ancho de banda de varios bits.
  - Método de mayor eficiencia que el “header bit crafting”.

- **Optional header extension** (extensión opcional de cabecera):
  - Se utilizan protocolos que permiten campos de cabecera opcionales, los cuales son añadidos a la cabecera y diseñados para transportar información o instrucciones adicionales.
  - Mayor ancho de banda (bytes).
  - Método de mayor eficiencia.

Tabla 2: Ratio de eficiencia teórica.

	Covert bytes/packet	Packet Size	Efficiency
Modulation	1 bit	20	0.63%
Crafting	1 byte	20	5.00%
Hdr Options	100 bytes	260	38.46%
	65515 bytes	65535	99.97%

Extraída de Couture, E.,19 de agosto de 2010, pp. 7

A continuación se exponen ejemplos de Covert Channels que se centran en protocolos de red, en sus diferentes capas.

Almacenamiento: ICMP protocol

Internet Control Message Protocol (ICMP) es un protocolo que trabaja en la capa 3 del modelo OSI y TCP/IP, la capa de red, y que tiene como objetivo el control y la notificación de errores, siendo su descripción detallada en el RFC792. ICMP opera en base a datagramas, de manera similar al protocolo UDP, y por ello no posee mecanismos para el control de transmisión o garantía de recepción (Doug, 2008).

Los paquetes ICMP están encapsulados dentro de los paquetes IP, comenzando la cabecera ICMP (de 8 bytes) una vez ha finalizado la cabecera IP, correspondiéndose el primer octeto de la porción de datos del datagrama al tipo de mensaje ICMP. La **cabecera** está formada por los siguientes campos:

- Type (1byte): Tipo de mensaje.
- Code (1byte): Subtipo.
- Checksum (2 bytes): Para la comprobación de errores del mensaje.
- Resto de la cabecera (4 bytes): El contenido de este campo depende del tipo de mensaje.

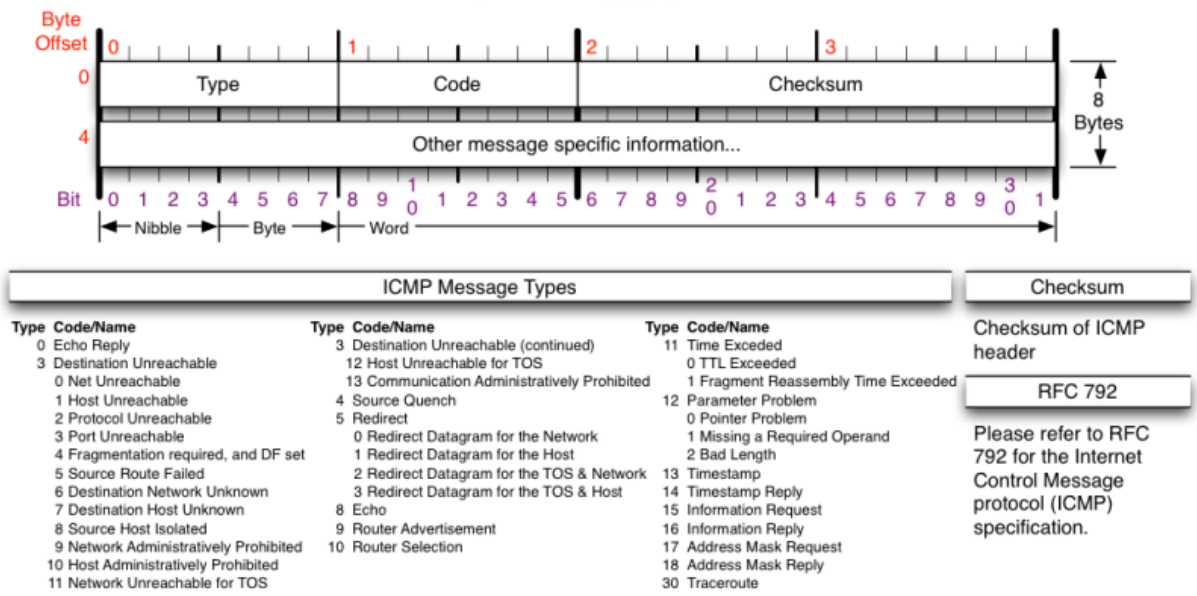


Figura 7: Cabecera del protocolo ICMP.

Recuperado de http://anand.prakshal.com/bizFloat/563444244ec0a40c6c8ec327/The-Internet-Control-Message-Protocol-ICMP-is-one-of-the-main-protocols-of-the-Internet-Protocol-Suite-It-is-used-by-network-devices-like-routers-to-send-error-messages-indicating-for-example-that-a-requested-s

A continuación se procederá a comentar un ejemplo de aplicación utilizando el tipo de mensaje ICMP más conocido y utilizado: **echo** y **echo reply** (tipo 8 y 0 respectivamente). La diferencia entre ambos mensajes es el campo "Type".

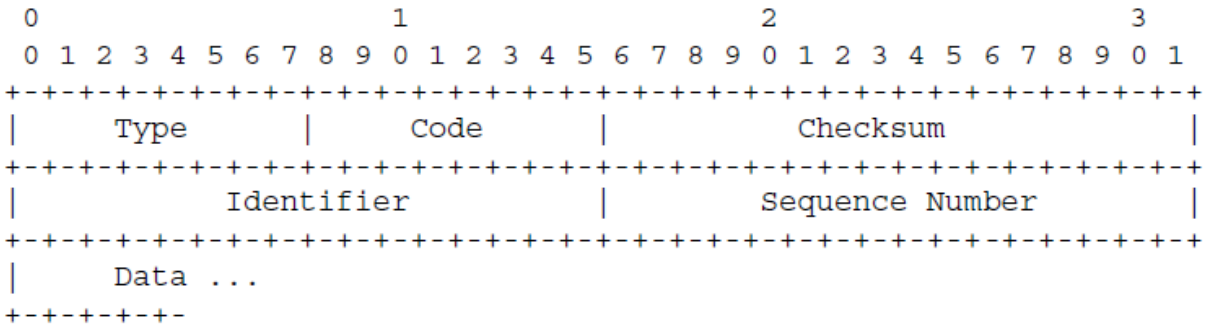


Figura 8: Cabecera de un mensaje echo / echo reply. RFC 792.

En sistemas **Linux** (figura 4 de la Figura 9), el campo "Data" del mensaje *Echo Request* se rellena con 56 bytes de dígitos numéricos. Por contra, en sistemas **Windows** (figura 6 de la

Figura 9), se rellena con 32 bytes de caracteres alfanuméricos (Gregg, 2007). Esta variación de tamaño nos indica que dependiendo de la implementación, es posible disponer de tamaños diferentes, añadiendo complejidad a la detección de la “anomalía” producida a consecuencia del covert channel. Además, en Linux, utilizando el comando *ping -p*, se puede introducir información arbitraria en la payload. Asimismo, la *cantidad máxima de información* que puede ser almacenada en el campo de información añadida o payload puede ser calculado como el tamaño máximo de un paquete IP menos las cabeceras de IP e ICMP: 65535 – 20 – 8 = 65507 bytes.

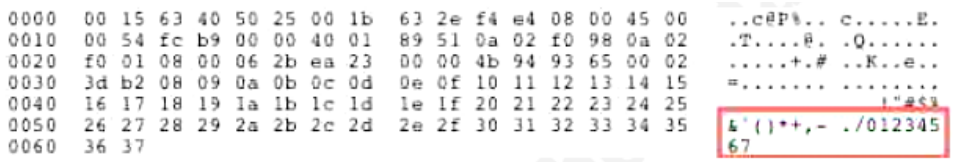


Figure 4 - Linux Ping Request

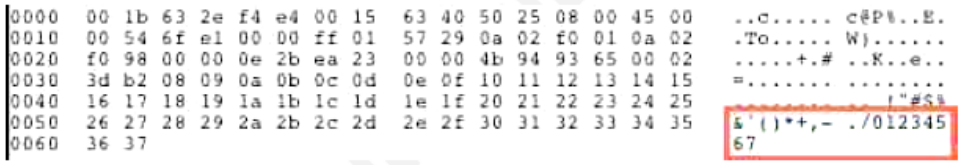


Figure 5 - Linux Ping Reply

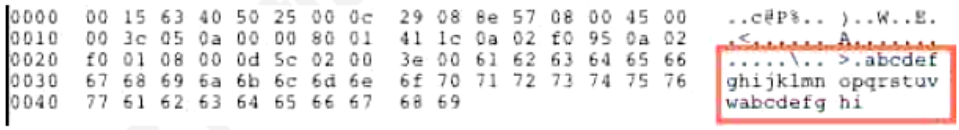


Figure 6 - Windows Ping Request

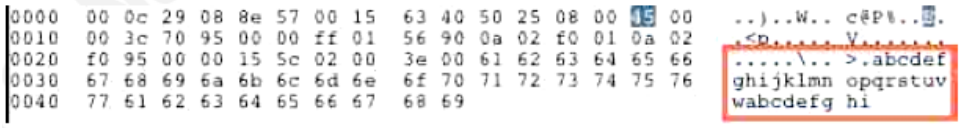


Figure 7 - Windows Ping Reply

Figura 9: Comparativa del contenido de la payload en los mensajes Echo y Echo Request. (Couture, E.,19 de agosto de 2010, p.10)

También cabe destacar, que tal y como se indica en el RFC:792, el contenido del campo de datos del mensaje *ICMP echo* debe encontrarse también en el campo de datos del mensaje ICMP *echo reply*.

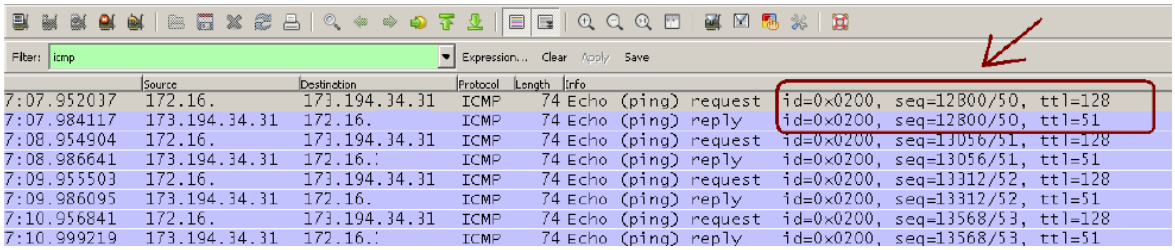


Figura 11: Captura con Ethereal del envío de mensajes echo request y echo reply.

(Holguín, J. M., Moreno, M., Merino, B.,Mayo 2013, p.158)

Exploit ICMP

Diversas **herramientas** para llevar a cabo ICMP tunnels se encuentran disponibles de manera pública, como es el caso de (Couture, 2010):

- **ptunnel**: (http://www.cs.uit.no/~daniels/PingTunnel/). Software de tunelado ICMP que soporta múltiples conexiones concurrentes y autenticación por contraseña. Se puede utilizar tanto en Linux como en Windows.
- **Loki2**: (http://phrack.org/issues/51/6.html). Programa para crear puertas traseras ICMP. Originalmente lanzado en 1996 en la revista hacker 'Phrack'.
- **007Shell**: (https://packetstormsecurity.com/groups/s0ftpj/007shell.tgz). Aplicación similar a Loki, pero que divide cada paquete en múltiplos de 64 bytes para que el túnel parezca ser tráfico legítimo.
- **ICMP Backdoor**: (https://packetstormsecurity.com/UNIX/penetration/rootkits/icmp-backdoor.tar.gz). Programa que solo utiliza paquetes ping reply. Algunos sistemas IDS lo pueden detectar fácilmente debido a que no rellena mensajes cortos ni divide mensajes largos.
- **B0CK**: (http://www.s0ftpj.org/bfi/bfi7.tar.gz). Variante que utiliza mensajes IGMP multicast para mejorar el trabajo hecho por los creadores de Loki y 007Shell. También codifica el campo de direcciones para mayor encubrimiento.
- **Hans**: (http://code.gerade.org/hans/). Solución IP sobre ICMP. Emplea dispositivos TUN/TAP para, entre otras cosas, poder operar de manera fiable si el firewall bloquea múltiples *echo replies* por petición.

A continuación se muestra un **ejemplo** de un caso de covert-channel de almacenamiento de una comunicación TCP a través de paquetes ICMP utilizando la aplicación ***ptunnel*** (Couture, 2010, pp 12-16):

El laboratorio de prueba para este ejemplo será como el de la figura 12, mostrada a continuación:

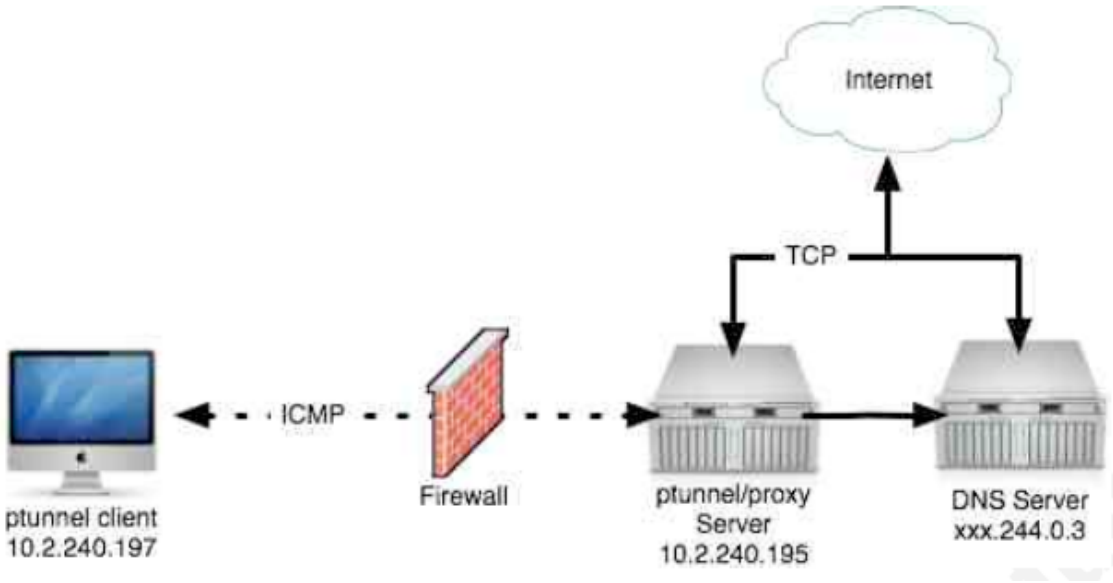


Figura 12: Configuración del laboratorio ptunnel.

(Couture, E.,19 de agosto de 2010, p.10)

El servidor es activado al ejecutar el comando *ptunnel* desde la línea de comandos. Además, y aunque no sea necesario, se ha habilitado un proxy privoxy en el servidor para redirigir todo el tráfico recibido hacia Internet. Si el usuario simplemente quisiera acceder a un recurso local, este último paso podría ser omitido.

La instancia del túnel se lleva a cabo en el cliente a través del siguiente comando:

#ptunnel -p *sIPaddress* -lp *localport* -da *localhost* -dp *destport*

Donde:

- **-p *sIPaddress***: dirección IP del listener/server.
- **-lp *localport***: puerto local del cliente, a través del cual el tráfico será tunelizado.
- **-da *localhost***: dirección IP de *loopback* en el servidor para redirigir el tráfico al web-proxy (privoxy).
- **-dp *destport***: puerto de redirección del web-proxy privoxy del servidor.

A continuación se muestra cómo se llevaría a cabo, en una terminal de comandos en un entorno Linux, una comunicación sencilla *(Holguín, J. M. et al., 2013. p.161)*:

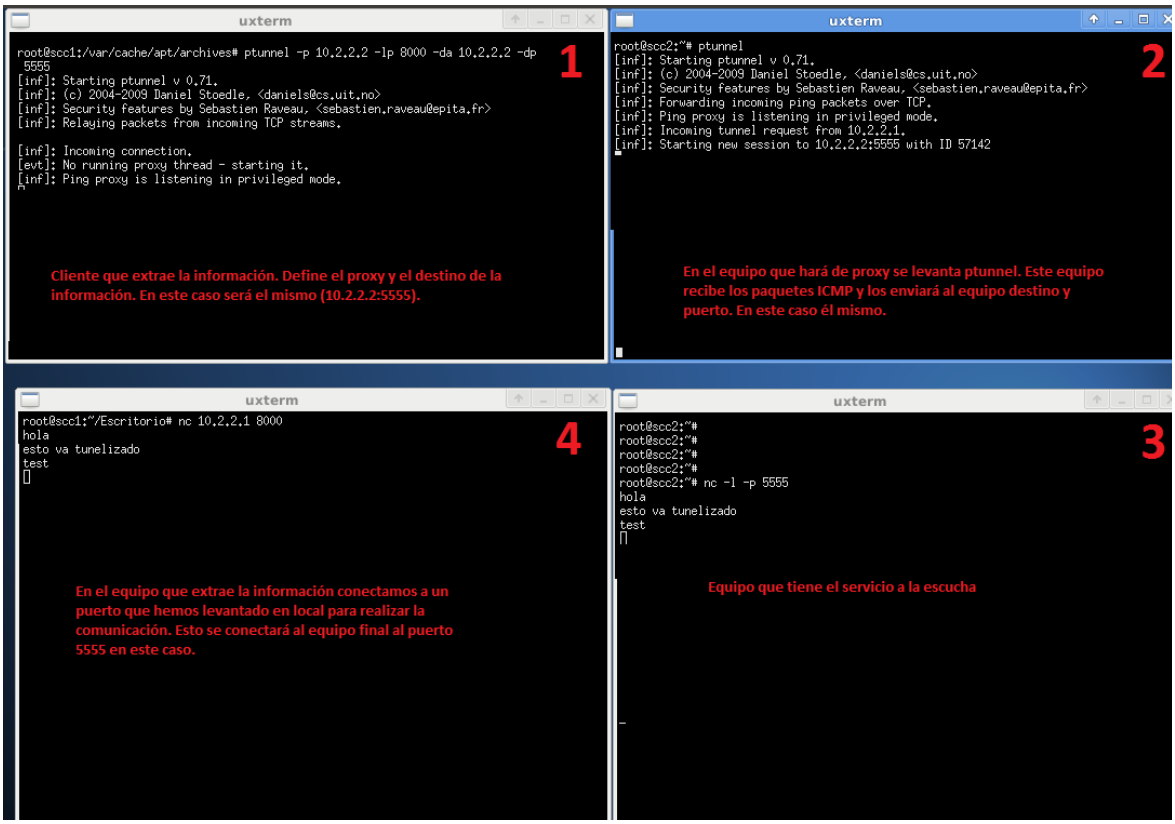


Figura 13: Cómo encapsular una comunicación TCP a través de paquetes ICMP ptunnel. (Holguín, J. M., Moreno, M., Merino, B.,Mayo 2013, p.161)

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

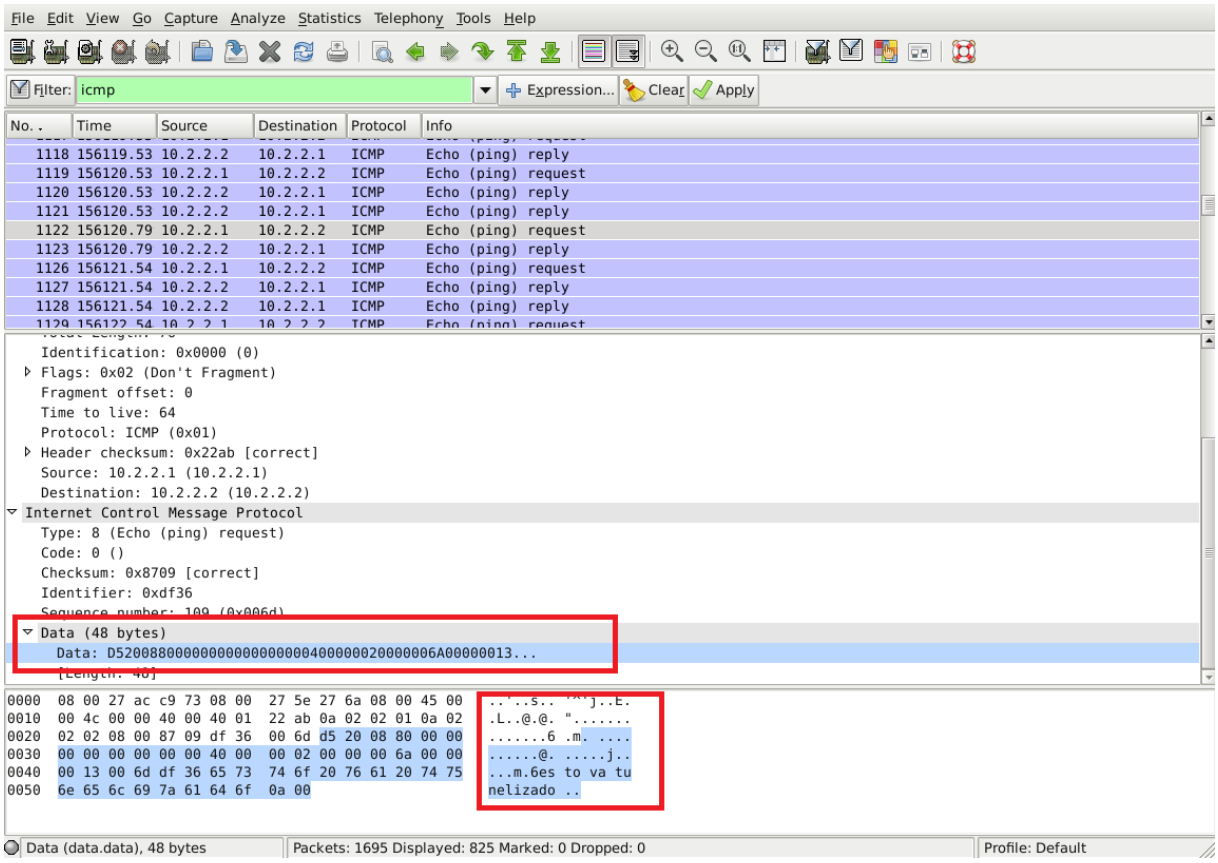


Figura 14: Paquete ICMP con información.

(Holguín, J. M., Moreno, M., Merino, B.,Mayo 2013, p.162)

Para que este encapsulamiento funciones correctamente, el *protocolo ping tunnel* ha sido diseñado de manera similar al protocolo TCP/IP, añadiéndole un campo extra denominado "magic number" (número mágico), que actúa como identificador para separar el tráfico *echo* del tráfico *echo* regular (Doug, 2008).

A continuación se muestran y describen los campos de la cabecera del protocolo ptunnel (Stødle, 2005):

Figura 15: Formato de paquete usado para el intercambio de mensajes entre cliente y proxy.

(Stødle, 2005).

Campos:

- **Magic**: Identificador único que sirve para diferenciar el tráfico *Echo* modificado del tráfico *Echo* normal.
- **IP – Port**:: Utilizados únicamente en los paquetes que salen del cliente hacia el proxy. Indican donde el cliente quiere que los paquetes recibidos sean redirigidos. Son utilizados, normalmente, solo una vez, cuando el proxy recive el primer mensaje con el código de estado *kProxy_start*).
- **State**: Tiene un doble propósito: Indicar qué tipo de mensaje está siendo recibido, e indicar quién envió el mensaje.
- **Length field**: Indica la longitud de la porción de información (campo *Data*) del paquete.

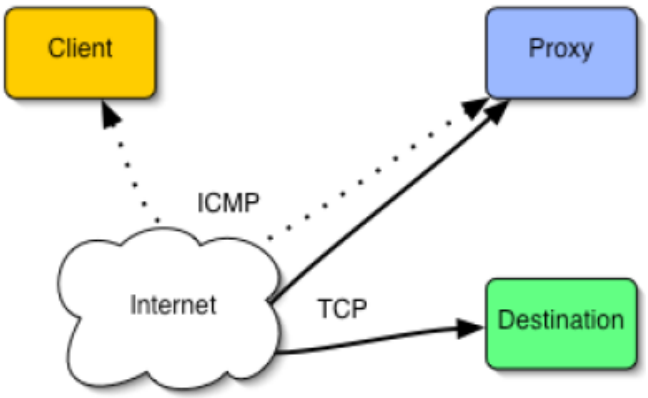


Figura 16 : Configuración de la red para el funcionamiento de ptunnel. (Stødle, 2005).

Covert Channels Storage: IP protocol

El Internet Protocol (IP) forma parte de la suite TCP/IP y su descripción se encuentra en el RFC 791. Fue diseñado en 1981 para su uso en sistemas interconectados de redes de comunicaciones de conmutación de paquetes, para proveer las funciones necesarias para enviar un paquete de bits (datagrama de Internet) desde una fuente a un destino a través de un sistema interconectado de redes.

A continuación se muestra la cabecera del protocolo IP:

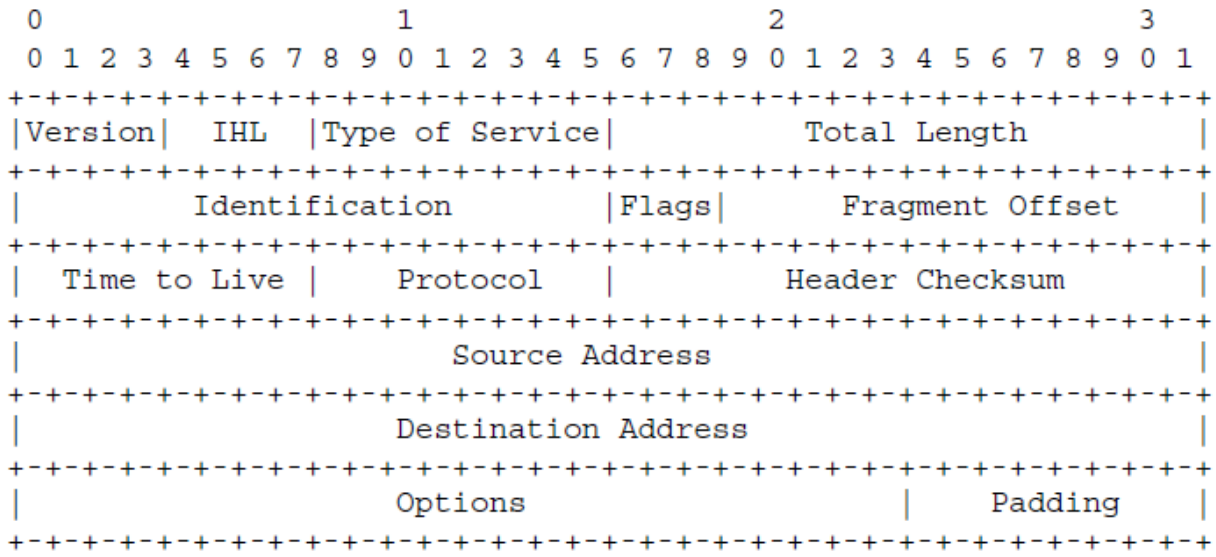


Figura 17: Cabecera del protocolo IP. RFC791

Thyer (2008) nos muestra que es posible utilizar la cabecera del protocolo IP a modo de covert channel para transmitir información.

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 (Holguín, J. M. et al., 2013. p.163). Este campo, conformado por 16 bits, es un valor identificativo asignado por el emisor para ayudar en el ensamblado de los fragmentos de un datagrama. Lo que convierte a este campo en objetivo para la creación de covert channels, es que ***cuando los paquetes no se fragmentan, se puede interpretar como se considere***.

Exploit IP

Para llevar a cabo una demostración acerca del uso de la cabecera IP para la creación de covert channels, Thyer (2008) escribió un programa conocido como **SUBROSA**, el cual a su vez está basado en el programa conocido como ***covert_tcp*** (Rowland, 1996), y se puede descargar desde la página web del propio Thyer: http://www.packetheader.net/.

Características de **SUBROSA** (Thyer, 2008, pp.10-11):

- Escrito en lenguaje C.
- Implementado utilizando los raw sockets del kernel de Linux.
- Codifica información ASCII dentro de cabeceras del protocolo IP y envía paquetes IP que imitan tráfico de red legítimo.
- Evita el uso de campos reservados de cabecera fácilmente detectables.
- Utiliza campos de cabecera con valores vagamente definidos y sujetos a interpretación.
- Implementa cifrado simétrico para ofuscar la información, ya que la información codificada en ASCII podría ser detectada como comunicaciones anormales.
- Necesita privilegios *root* para manipular y enviar paquetes de información de cabecera modificados.
- Utilizando la opción "-w" haremos 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 un rango de tiempo determinado.

A continuación se muestra una prueba de concepto (Holguín, J. M. et al., 2013. p.163-164):

1. **Envío de una cadena de texto desde *Cliente (emisor)* a *Servidor (destinatario)***:
   - **Comandos** (Thyer, 2008, pp.12):
     - Cliente# subrosa -s*SourceIP DestinationtIP destinationPort datos*
     - Servidor# subrosa -sSource*IP* -lp*Localport datos*

       El puerto del emisor, a no ser que se especifique con "-p", es escogido de manera aleatoria con un valor superior a 1024.

- **Prueba de Concepto** (Holguín, J. M. et al., 2013. p.163-164):
  - En la primera imagen (figura 18) se aprecia cómo el cliente envía la parte “ho” de la cadena “hola”, que es el mensaje a transmitir con el receptor:

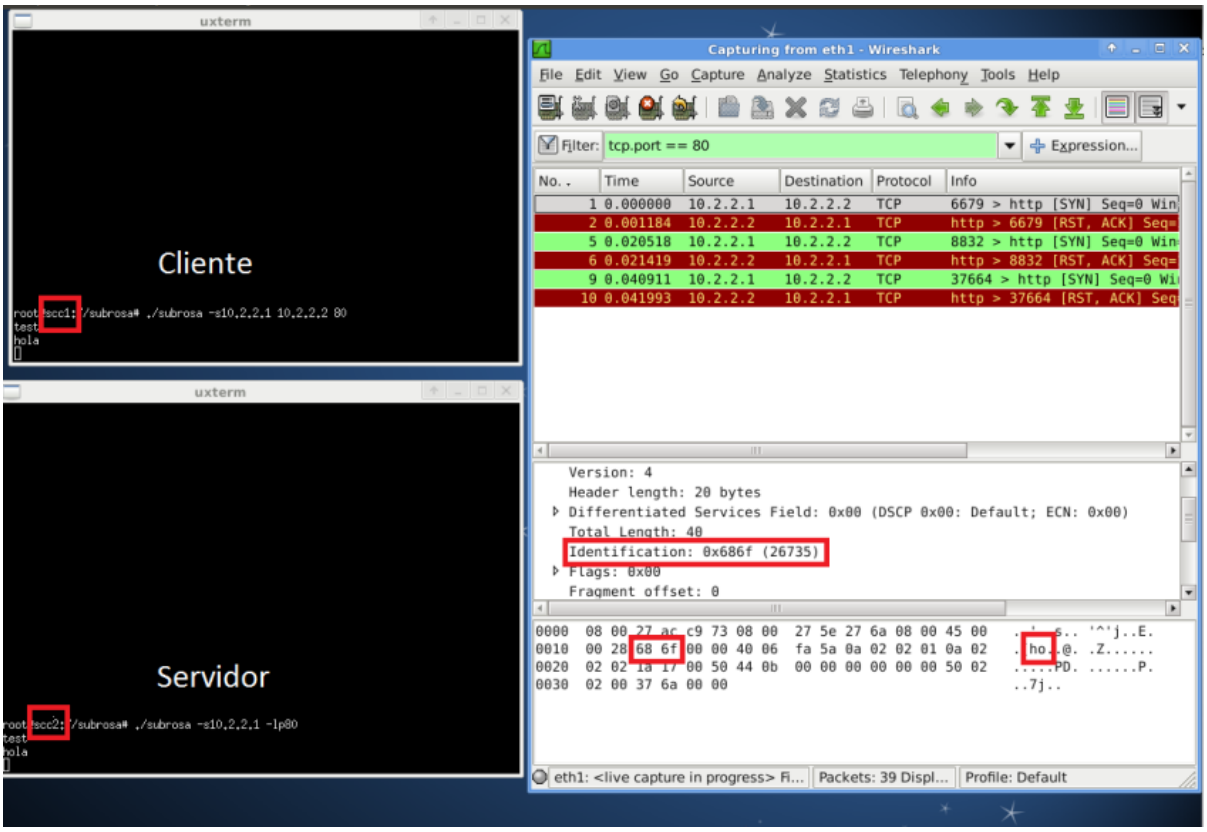


Figura 18: Ejemplo de envío en el campo IPID, fragmento “ho”.

(Holguín, J. M., Moreno, M., Merino, B.,Mayo 2013, p.163)

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

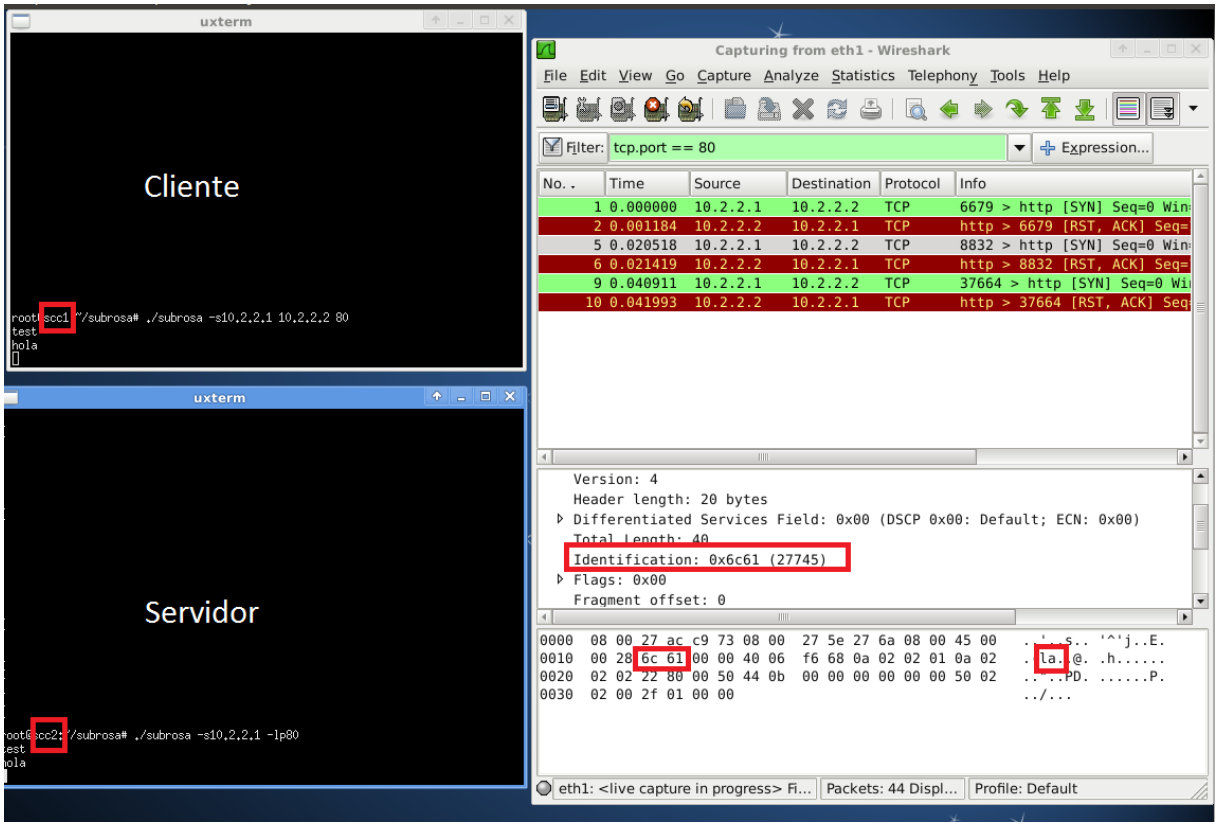


Figura 19: Ejemplo de envío de información, fragmento “la”.

(Holguín, J. M., Moreno, M., Merino, B.,Mayo 2013, p.163)

2. **Extracción de un documento confidencial**:
   - **Comandos:**
     - Cliente # cat *documento_confidencial*.doc | ./subrosa -s*SourceIP DestinationIP destinationPort*
     - Servidor # ./subrosa -s*SourceIP* -lp*LocalPort* > *documento_confidencial*.doc.

Covert Channels Storage: TCP protocol

El Transport Control Protocol (TCP) es descrito en el RFC793, enfocado en la robustez en la presencia de irregularidades e indisponibilidad en las comunicaciones, y diseñado para encajar en una jerarquía por capas de protocolos.

A continuación se muestra la cabecera del protocolo TCP:

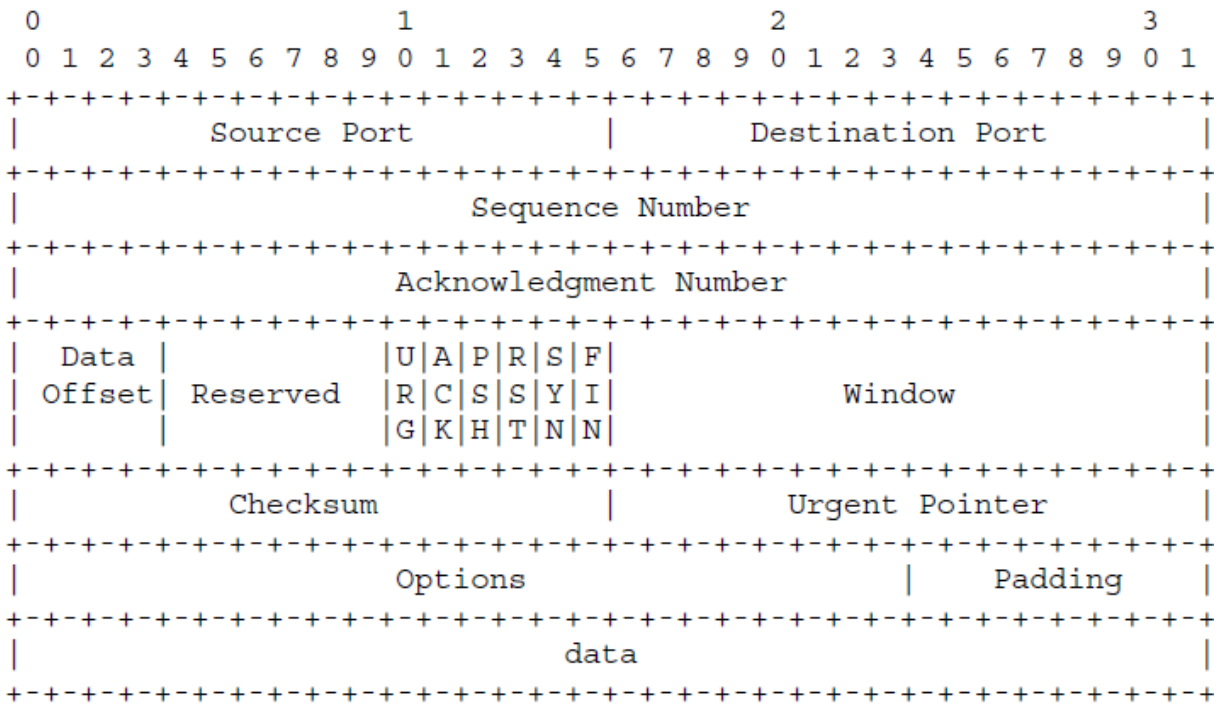


Figura 20: Cabecera del protocolo TCP. RFC793.

Algunas de las técnicas utilizadas para llevar a cabo la implementación de covert-channels sobre el protocolo TCP consisten en:

- Utilizar los campos "*Sequence Number*" o "*Acknowledgment Number*" (Rowland, 1997).
- Explotar los "*ptype handlers del kernel de Linux*" para crear *covert channels pasivos* (PCC), utilizando el programa **NUSHU** cuyo código fuente se encuentra en el artículo "*The Implementation of Passive Covert Channels in the Linux Kernel*" (Rutkowska, J., 2004).

A continuación se muestra un ejemplo donde un equipo está sacando información a través del puerto 80 escondida en el campo "Número de Secuencia" de la cabecera TCP con la

herramienta **covert_tcp** (Holguín et al., 2013. p.165-166), cuyo código fuente se encuentra como anexo en el artículo "*Covert Channels in the TCP/IP Protocol Suite*" (Rowland, C. H., 1997) :

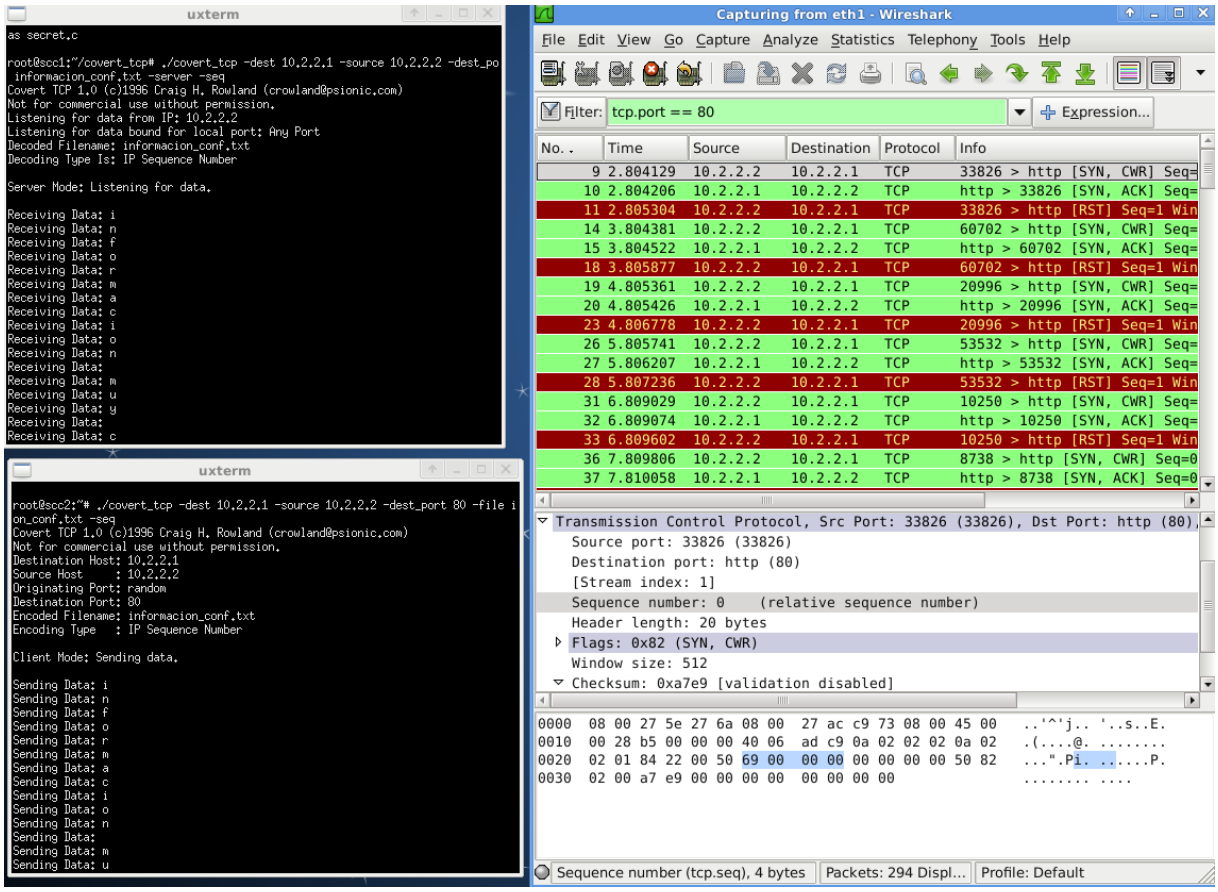


Figura 21: Ejemplo de covert_tcp.

(Holguín, J. M., Moreno, M., Merino, B.,Mayo 2013, p.166)

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.

Comandos utilizados (Rowling, 1997):

1. Server# ./covert_tcp -dest *destIP* -source *sourceIP* -dest_port *destPort* -file *filename* -server -seq
2. Cliente# ./covert_tcp -dest *destIP* -source *sourceIP* -dest_port *destPort* -file *filename* -seq

Covert Channels Storage: UDP protocol

El User Datagram Protocol (UDP) se encuentra definido en el RFC768, y forma parte de la suite de protocolos TCP/IP. Este protocolo provee de un procedimiento para programas de la capa de aplicación para enviar mensajes a otros programas con un mecanismos de protocolo mínimo. Además, es un protocolo orientado al intercambio, y la entrega no está garantizada (es un protocolo no fiable, al contrario que el protocolo TCP).

A continuación se muestra la cabecera del protocolo UDP:

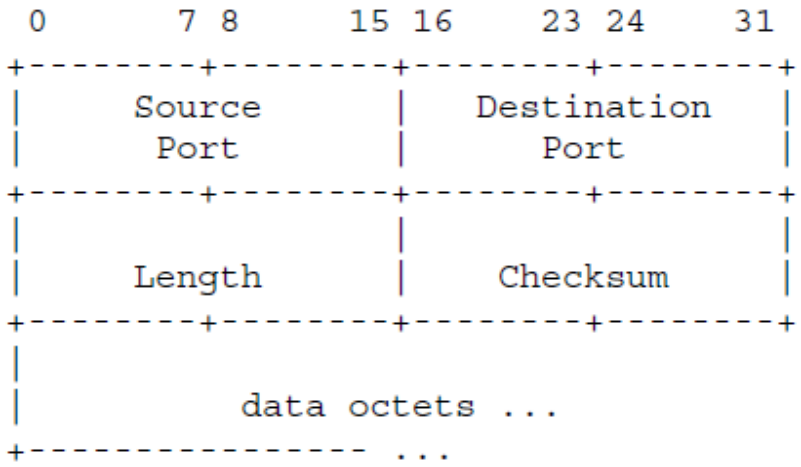


Figura 22: Cabecera del protocolo UDP. RFC768.

Algunas investigaciones (Bidou, Raynal, 2005) han demostrado que la cabecera del protocolo UDP es aprovechable para el establecimiento de canales encubiertos en la red.

En este caso son tres los campos de la cabecera UDP que pueden usarse para transferir información e intentar pasar desapercibidos: *Source Port*, *Length* y *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 (Holguín et al., 2013).

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.

- Domain Name System (DNS).
- Hypertext Transfer Protocol (HTTP).
- Dynamic Host Configuration Protocol (DHCP).

DNS protocol

El Domain Name System (DNS), descrito en los RFC1034 y RFC1035 es el protocolo que traduce localizadores de recursos legibles a direcciones IP enrutables.

A continuación se muestra la cabecera del protocolo DNS:

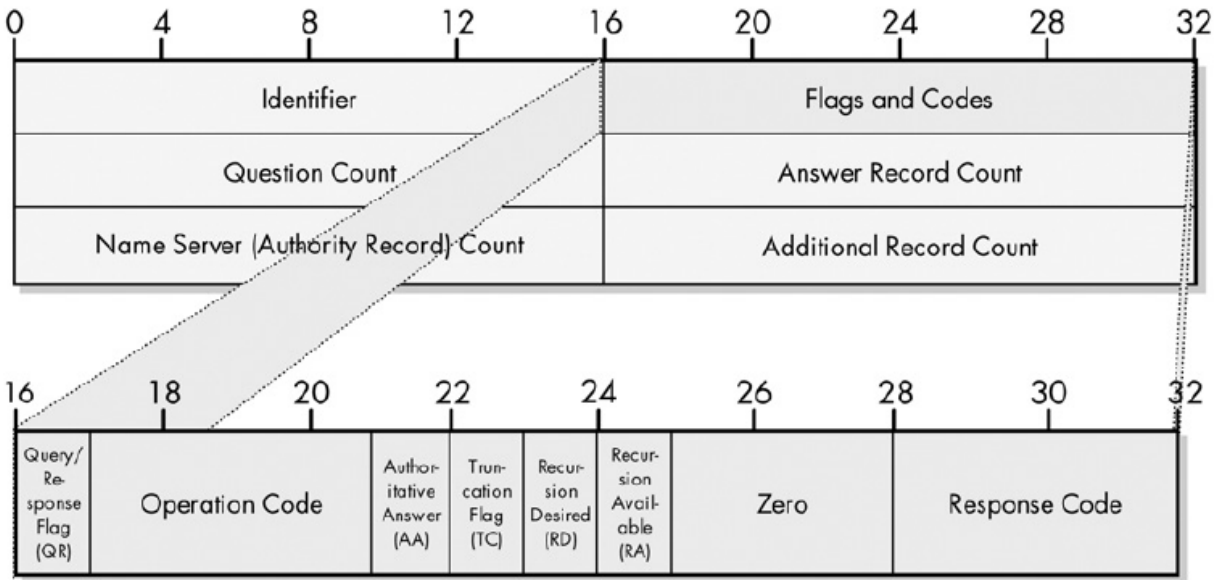


Figura 23: Cabecera del protocolo DNS.

(Kozierok, 2005, p.934)

Como Couture (2010) indica, los componentes clave de la cabecera del protocolo DNS están principalmente localizados en los bits 16 a 32 de dicha cabecera (figura 23). Entre otras cosas, los flags sirven para indicar si el mensaje DNS en retorno es el mensaje acreditado, o si una recursividad adicional es necesaria para localizar la respuesta para la petición. La petición viajará a través del sistema DNS partiendo del secundario hacia los

sistemas primario y 'root', en búsqueda de la respuesta acreditada para la petición, la cual será finalmente devuelta al host originario, tal como se muestra en la figura 24.

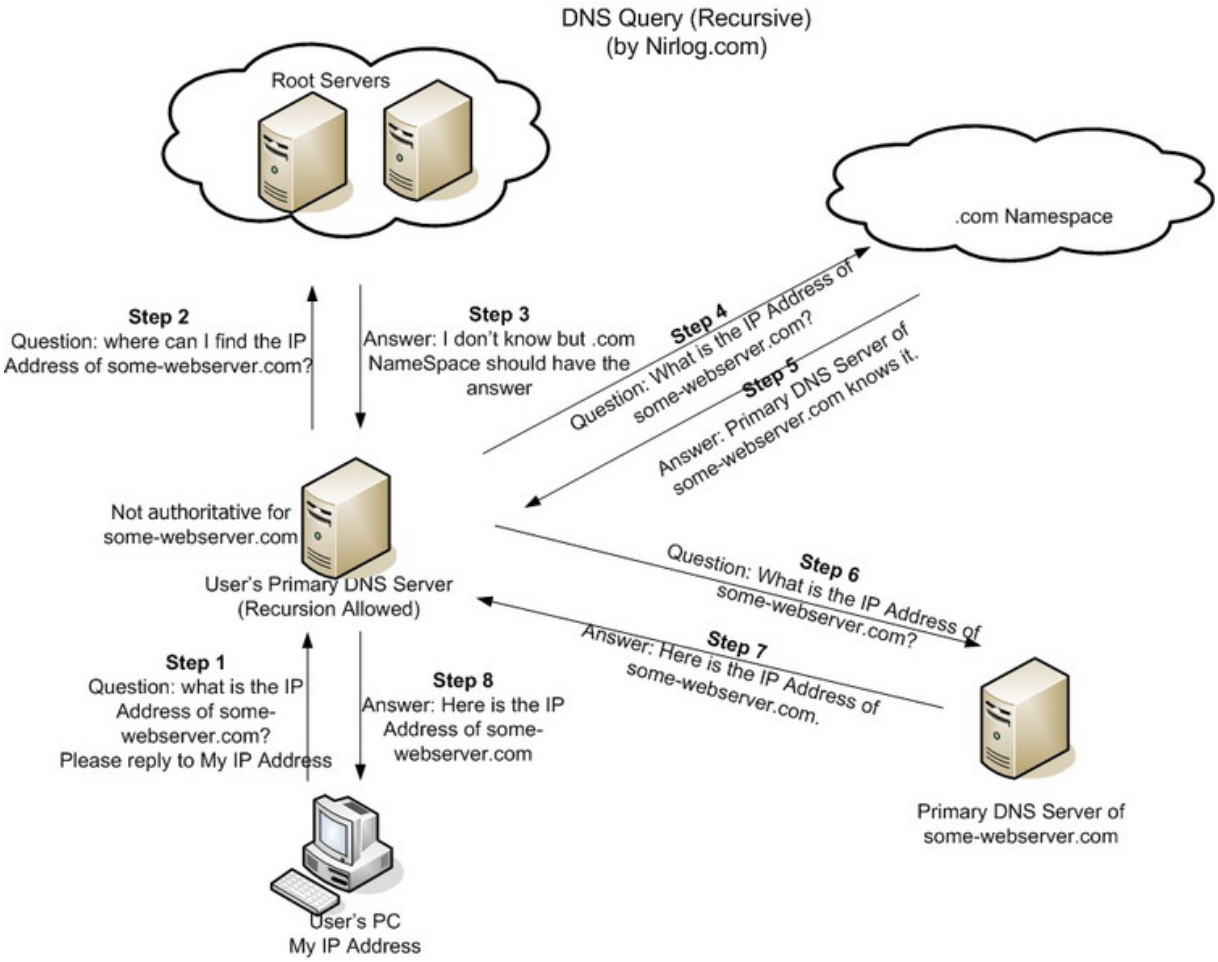


Figura 24: Proceso recursivo de petición DNS.

Recuperado el 4 de agosto de 2016 de http://nirlog.com/2006/03/28/dns-amplification-attack/

Los exploits para crear canales encubiertos utilizando el protocolo DNS aprovechan su bidireccionalidad, recursividad y su naturaleza insegura, haciendo posible exfiltrar información desde dentro de una red segura, acceder recursos restringidos o proveer de un medio para el control del malware residente (Couture, 2010).

Herramientas que permiten para crear este tipo de canal encubierto, descritas en el artículo de Couture (2010):

- **OzymanDNS**: (https://dankaminsky.com/2004/07/29/51/). Scripts escritos en Perl, desarrollados y popularizados por el gurú DNS Dan Kaminsky.

- **NSTX**: (http://savannah.nongnu.org/projects/nstx/). La Name Server Transfer Protocol es una de las aplicaciones originales de tunelado DNS-IP. No actualizada en más de 7 años, aunque todavía muy funcional.
- **dns2tcp**: (http://www.hsc.fr/ressources/outils/dns2tcp/). Muy similar a OzymanDNS. Crea un túnel DNS utilizando una redirección DNS "IN", permitiendo el tráfico arbitrario.
- **Iodine**: (http://code.kryo.se/iodine/). Una variante actual con código fuente UNIX y binarios Windows. Ofrece protección por contraseña y múltiples opciones concurrentes de usuario.

HTTP protocol

El Hypertext Transfer Protocol Version HTTP, es un protocolo de comunicación de red perteneciente a la capa de aplicación de los modelos OSI y TCP/IP, y que fue desarrollado por el World Wide Web Consortium (W3C) y la Internet Engineering Task Force (IETF). Su versión más actual (HTTP/2.0) se encuentra definida en el RFC7540. Define la sintaxis y la semántica que utilizan los elementos de software de la arquitectura web para comunicarse, siendo un protocolo sin estado (no guarda ninguna información sobre conexiones anteriores). Debido a que las aplicaciones web necesitan frecuentemente mantener el estado, se recurre a las ***cookies***, siendo éstas una herramienta utilizada, como se verá en los siguientes párrafos, para llevar a cabo el canal encubierto.

Tal y como exponen Holguín et al. (2013) HTTP es el protocolo más utilizado hoy en día como transporte de información, y por ello las organizaciones permiten su uso en cortafuegos, lo cual hace que sea elegido por los atacantes como medio de transporte para la extracción de la información.

Una de las técnicas utilizada para extraer información y que resulta bastante habitual en APTs es **COVCOM** (Hiddeen Comments for Covert Communication), que utiliza los comentarios del protocolo HTTP para introducir información encubierta.

Un ejemplo de **PoC** sobre cómo utilizar la **cabecera *HTTP Cookie*** está descrito en el artículo "How to cook a covert channel" (Gray World Team, 2006), donde los autores crean una herramienta para crear un Covert Channel con esta cabecera (Holguín et al., 2013).

Otro ejemplo de **PoC** de este tipo de canales encubiertos es el creado ***sobre Facebook*** por José Selvi en su investigación para el SANS Reading Room (Selvi, 2012), utilizando la herramienta desarrollada en la misma investigación, **FaceCat**, la cual se puede descargar desde http://www.giac.org/paper/gcih/10163/covert-channels-social-networks/117979. Con esta herramienta, un atacante puede extraer información utilizando como lugar de almacenamiento la red social Facebook, lugar en el cual un analista únicamente visualizará tráfico hacia la red social, no viendo dominios extraños, ni tráfico hacia países sospechosos para la actividad del usuario, etcétera (Holguín et al., 2013).

A continuación se enlazan dos ***videotutoriales*** del autor del estudio acerca del uso de la herramienta *FaceCat* para crear canales encubiertos a través de Facebook:

- FaceCat (FaceBook Cat): https://www.youtube.com/watch?v=flZUuRK2R-k
- FaceCat + Poison Ivy: https://www.youtube.com/watch?v=C_c8KNvVSVg

DHCP

El Dynamic Host Configuration Protocol (DHCP), definido en el RFC 2131, provee un framework para compartir información de configuración a los dispositivos en una red TCP/IP.

Ríos y Onieva (2008) analizan en su estudio "*Clasificación de canales encubiertos. Un nuevo canal:Covert_DHCP*" la implementación de canales encubiertos utilizando el protocolo DHCP.

La primera de las implementaciones se basa en el campo **xid** de la estructura de mensajes DHCP. El cliente DHCP modificado permite que el usuario pueda decidir si utilizar el canal encubierto en la solicitud de una configuración de red. El principal *inconveniente* es su reducido ancho de banda, y su mayor *ventaja* es que utiliza un campo de naturaleza aleatoria y de amplio uso.

La segunda de las implementaciones se basa en el campo **Options**. En este caso se ha optado por la utilización de aquellas opciones que se encuentran definidas dentro del protocolo como opciones de uso privado, es decir, las pertenecientes al rango 224 – 254. Como inconveniente principal está en su detectabilidad, y como ventaja se tiene un mayor ancho de banda que en la anterior implementación.

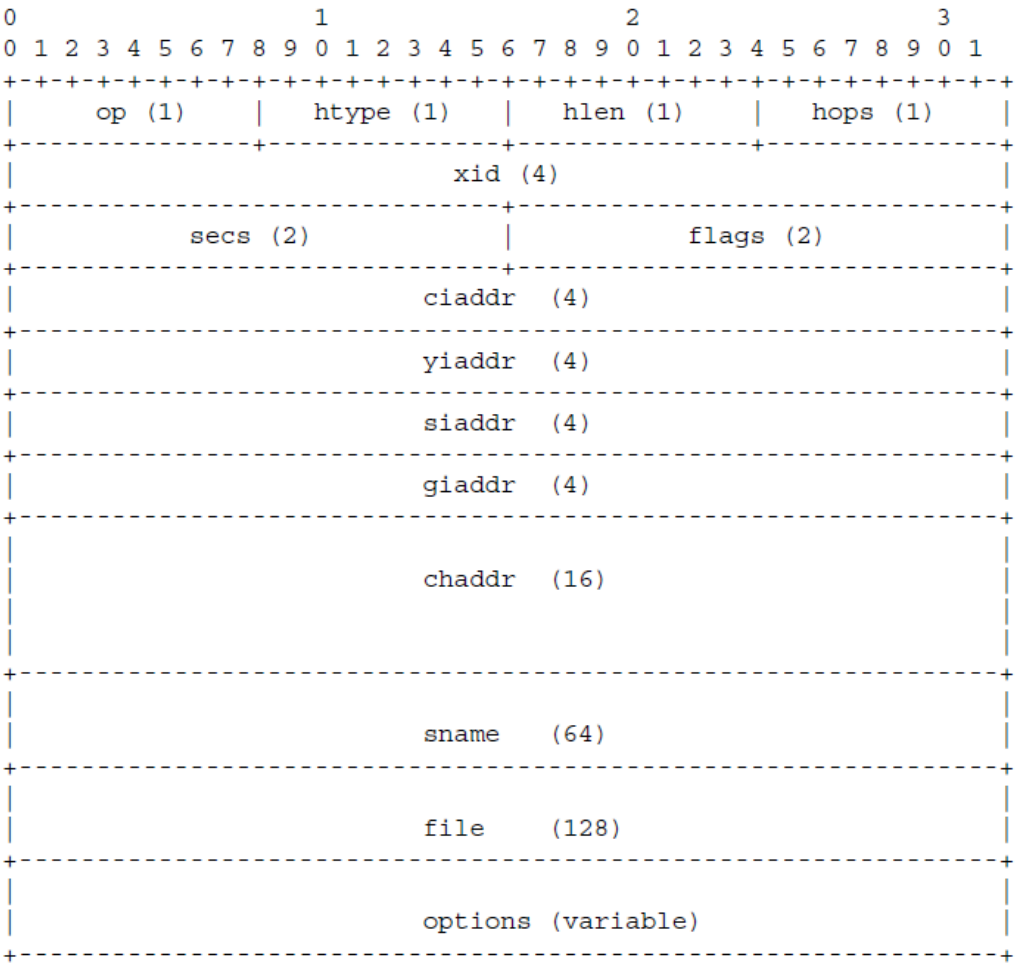


Figura 25: Formato de un mensaje DHCP. RFC 2131.

SNMP

El Simple Network Management Protocol (SNMP) es un protocolo de capa de aplicación que facilita el intercambio de información de administración entre dispositivos de red, permitiendo a los administradores supervisar el funcionamiento de la red, buscar y resolver sus problemas, y planear su crecimiento. Viene descrito en el RFC 1157 (SNMP, 1990) y en el RFC 3410 (SNMPv3, 2002).

Esteban (2016) en su artículo "*SNMPChat: Chatear en un canal encubierto sobre SNMP*", muestra una Prueba de Concepto de cómo crear un canal encubierto sobre el protocolo SNMP utilizando la herramienta **SNMPChat**, cuyo código se puede descargar del siguiente link: https://github.com/jevalenciap/snmp_chat/blob/master/snmp_chat2.py.

Otros Tipos de Covert Channels

Canales de Temporización: Jitterbug

En agosto de 2006 ve la luz un estudio que crea canales de temporización a partir de un dispositivo al que se bautiza con el nombre de ***Jitterbug***, el cual puede ser implementado tanto por software como por hardware, y cuya finalidad es la de reconocer información sensible y modularla a través de la red. Son mecanismos semipasivos que no generan nueves eventos sino que utilizan otros para transmitir la información (Ríos, Onieva, 2008).

Shah, Molina y Blaze (2006) desarrollan un ejemplo basado en un jitterbug hardware para teclado. Éste se aprovecha de que la mayoría de las aplicaciones de red interactivas (ejm. Telnet) envían un paquete de datos tras cada pulsación de teclado realizada. Así pues, jitterbug añadirá pequeños retrasos, inapreciables a nivel de usuario, a las pulsaciones de teclado, para retrasar a su vez el envío de los respectivos paquetes. Estos retrasos serán observados por la entidad receptora con el fin de recuperar la información oculta.

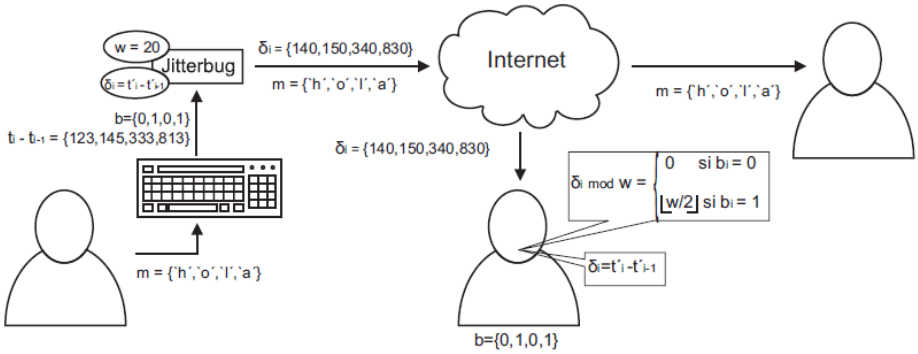


Figura 26: Ejemplo de uso de un dispositivo jitterbug para teclado.
(Ríos, Onieva, 2008, p.5)

Covert Channels en videojuegos multijugador

Zander (2010) en su tesis "*Performance of Selected Noisy Covert Channels and Their Countermeasures in IP Networks*", capítulo 5, muestra una posible implementación de canal encubierto utilizando un videojuego multijugador online del tipo shooter en primera persona, denominado por él mismo FPS Covert Channel (FPSCC).

FPSCC es un canal entre clientes del juego escondido de los operadores del servidor de juego y de los jugadores que no son participantes de la comunicación producida a través del canal encubierto. La transmisión de la información se llevará a cabo utilizando los movimientos de inclinación y de movimiento de mandíbula de los personajes de los

jugadores, pero de tal manera que no sea evidente, utilizando una captura de movimiento de entre varias enviadas.

Figura 27: Mensajes intercambiados entre servidor y cliente, y movimientos del personaje utilizados para transmitir la información.

(Zander, 2010, p. 110)

Covert Channels basados en temperatura

Zander (2010) también presenta una implementación de canal encubierto basado en la temperatura de la CPU. El el ejemplo, el emisor modula la carga de la CPU de un host intermedio conectado a la red (por ejemplo un servidor público) por medio de la variación del ratio de peticiones enviadas, en función de los bits encubiertos a codificar. La variación en la carga de la CPU modifica su temperatura, que a su vez induce cambios en la señal de reloj, la desviación en referencia al tiempo real. El receptor mide la deformación de la señal de reloj del host intermedio obteniendo timestamps dicho reloj y comparando estas con las del reloj local. Entonces decodifica los bits encubiertos mediante la estimación de las deformaciones en la señal de reloj.

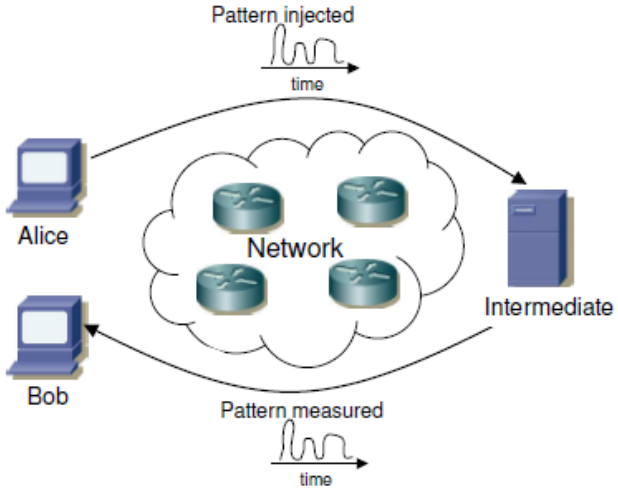


Figura 28: Canal encubierto basado en temperatura.

(Zander, 2010, p 140)

3.3. Ejemplo de ataques conocidos

En el documento "*Assessing Outbound Traffic to Uncover Advanced Persistent Threat" de* Binde et al. (2011), se describen dos ataques APT que describen de manera clara los componentes clave de un ataque APT.

Operación Aurora

Este ciberataque llevado a cabo contra diversas tecnológicas, de seguridad y defensa comenzó a mediados de 2009 y continuó hasta diciembre del mismo año. Los atacantes eligieron como blanco los sistemas Software-Configuration Management (SCM) que contenían información propiedad de Google, Adobe y otras compañías. La anatomía del ataque entra dentro de la categoría de un ataque APT clásico:

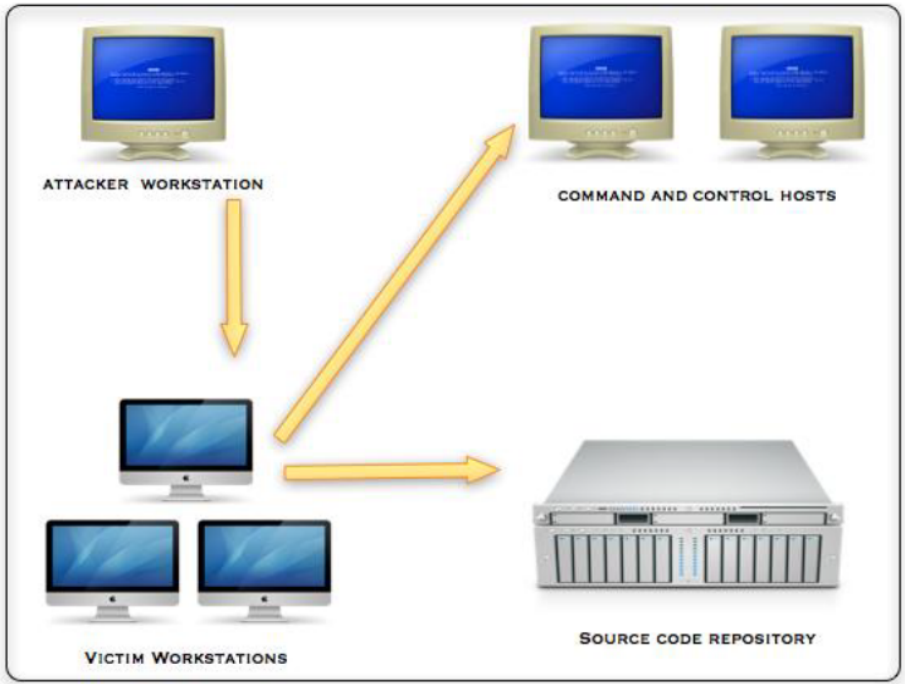


Figura 29: Anatomía del ataque Operación Aurora.

(Binde et al., 2011)

El ataque era "**avanzado**" ya que lo atacantes llevaron a cabo, con la intención de ganar acceso a las redes víctima, un ataque *Spear Phishing dirigido* contra la compañía. El email recibido por los empleados contenía un link a un sitio web que alojaba código JavaScript malicioso. Este malware explotaba una vulnerabilidad de Internet Explorer. Una vez explotado, el sistema de la víctima se conectaba a una serie de servidores C&C utilizando el puerto 443 TCP, asociado a tráfico cifrado, y por lo tanto difícil de inspeccionar.

El ataque era "**persistente**" ya que después de ganar infiltrarse dentro de las compañías objetivo, los atacantes emplearon los terminales infectados para comprometer otros recursos internos empleando el *pivoting* o *movimiento lateral* para evitar firewalls y otras restricciones.

El ataque era una "**amenaza**" ya que los adversarios demostraron alta motivación, financiación, y formaban parte de una organización estructurada. No solo tenían la intención de acceder a información sensible, además tenían la capacidad. Esto lo apoya el hecho de que los ataques fueron rastreados hasta dos escuelas chinas.

RSA Breach

En marzo de 2011, RSA reconoció un ataque exitoso contra su red. Los métodos utilizados por este ataque ilustran los componentes típicos de un ataque APT.

Objetivo: El ataque comenzó con una campaña de phishing, con la intención de introducir el malware en la red de la compañía, eligiendo como blanco dos pequeños grupos de empleados. En este caso los atacantes utilizaron una vulnerabilidad zero-day de Adobe Flash para, a través de una hoja de Microsoft Excel, instalar de forma un *kit de acceso remoto* o *Remote Access Toolkit* (RAT) en la máquina víctima conocido como *Poison Ivy (PI-RAT)*.

Movimiento lateral hacia recursos internos para recabar información: Una vez conseguido el acceso a la red interna utilizando PI-RAT, los atacantes comenzaron a moverse lateralmente hacia otros recursos internos. Una vez que recabaron la información para exfiltrar, ésta fue colocada en diversos servidores internos.

Exfiltración de la información: Para ocultar la retirada masiva de información, ésta fue colocada dentro de archivos RAR comprimidos y protegidos por contraseña, utilizando FTP para transferirlos a un servidor intermedio infectado, siendo posteriormente retirándos de éste.

3.4. Historia de las APT – Noticias – Ataques conocidos

ORIGEN

Con la publicación del informe "*APT1 Exposing One of China's Cyber Espionage Units*" de MANDIANT en 2013, 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 (Holguín et al., 2013). No obstante, ya en 2006, los analistas de la Fuerza Aérea de Estados Unidos (USAF) habían acuñado el término APT para facilitar la discusión sobre actividades intrusivas con sus contrapartes civiles difusas (Binde et al., 2011).

En cuanto al ámbito nacional, en 2010 a través de una entrevista a los medios de comunicación, un alto cargo del Centro Criptológico Nacional (CCN-CNI) informaba de que se habían registrado en 2009 más de 40 ataques cibernéticos categorizados como 'graves' contra instituciones, organizaciones, e incluso en el mismo centro, siendo en todos los casos patrón común el malware sofisticado y creado dirigido a la víctima (Holguín et al., 2013).

HISTORIAL DE ATAQUES APT

A continuación se muestra una tabla con los acontecimientos más importantes relacionados con las APT.

Tabla 3: Cronología de las APT

Año	Nombre	Motivación	Descripción
1982	The Farewell Dossier	Político/militar (USA→ URSS)	Oficiales norteamericanos modifican el software del equipamiento informático adquirido por los soviéticos para operar un gaseoducto, causando la explosión de éste (O'Harrow, Linch, 2012).
1986	The Cuckoo's Egg	Político/militar (URSS → USA)	En 1986 se descubrió la intrusión de un hacker, que vendía la información obtenida a la KGB, en las redes de instituciones académicas, militares y gubernamentales de Estados Unidos. Fue la primera investigación de su tipo, y donde

			por primera vez se puso en práctica el concepto de Honeypot (Stoll,1988).
2003	Titan Rain	Político/militar (? → USA)	Un grupo de hackers, supuestamente apoyados por China (el ataque se realizó desde este país, aunque no se sabe si utilizado a modo de proxy por otras potencias) (Graham, 2005).
2006	APT1	Político/militar (China → USA)	Mandiant (2013), describe en su informe "*APT1. Exposing One of China's Cyber Espionage Units*" los ataques llevados a cabo desde 2006 por un grupo de atacantes al que se bautizó como APT1, contra cerca de 150 organizaciones.
	Operación Shady RAT	Político/militar	Ciberataque de 5 años de duración orientado al robo de información en el que se vieron afectadas hasta 72 organizaciones de todo el mundo (ElMundo, 2011).
2008	Buckshot Yankee	Político/militar (? → USA)	La brecha de seguridad más importante en la seguridad informática de Estados Unidos se produjo cuando, supuestamente, alguien trabajando en el Comando Central del Pentágono introdujo una memoria USB infectada en un portátil militar en una base en oriente medio. El código dañino se esparció sin ser detectado en sistemas tanto clasificados como no clasificados, transmitiendo la información recabada a servidores bajo control extranjero (O'Harrow, Linch, 2012).
2009	Operación Aurora	Propiedad Intelectual (China → Empresas)	Primer gran ciberataque reconocido contra múltiples empresas (Google, Adobe Systems, Juniper Networks) con la intención de recopilar información sobre Propiedad Intelectual. (McAfee Labs, McAfee

			Foundstone Professional Services, 2010)
	Operación GhostNet	Político/militar (China → Dalai Lama)	Red de unos 1295 dispositivos infectados en 103 países, siendo un 30% considerados objetivos de alto valor, incluidos ministerios de exteriores, embajadas, organizaciones internacionales, medios de información y ONGs. Las evidencias de penetración en sistemas que contenían información sensible sobre el Dalai Lama y otros objetivos Tibetanos. (TheSecDevGroup, 2009)
2010	Stuxnet	Político/militar (USA/Israel → Irán)	Ciberataque concebido por los servicios de espionaje de Estados Unidos e Israel para sabotear los sistema SCADA de la central iraní de enriquecimiento de uranio Natanz. Si bien la intención era que el malware afectara solo a dicha instalación, finalmente se expandió por otro países debido a la conexión a Internet de dispositivos portátiles infectados (Anderson, 2012).
	Operación Night Dragon	Propiedad Intelectual (China → Empresas Sector Energético)	Ataque focalizado en el sector energético con el objetivo de robar información confidencial (McAfee Labs y McAfee Foundstone, 2011).
	Red October	Político/militar (Rusia)	Campaña de espionaje a agencias gubernamentales y diplomáticas internacionales entre 2010 y 2012 (Wangen, 2015).
2011	Nitro	Propiedad Intelectual	Campaña de ciberespionaje industrial y enfocada al robo de información de grandes empresas químicas y del sector de la defensa (Holguín et al., 2013).
	Duqu	Político/militar	Malware posiblemente de los

		(USA → Europa/Oriente Medio)	mismos creadores de Stuxnet con fines de espionaje de PLCs y sistemas de control industriales para preparar posteriores ataques (Wangen, 2015).
2012	Flame	Político/militar (? → Oriente Medio)	Malware modular utilizado para espiar a países de Oriente Medio. No hay consenso de cuanto tiempo llevaba produciéndose el ataque (Wangen, 2015).
	Mahdi	Político/militar (Irán → Oriente Medio)	Malware entre cuyas víctimas se incluían infraestructuras críticas de compañías, servicios financieros y embajadas diplomáticas localizadas en Irán, Israel y otros países de Oriente Medio (Wangen, 2015).
	Shamoon	Hacktivismo	Ataque asumido por el grupo hacker “Cutting Sword of Justice” que infectó alrededor de treinta mil PCs de la compañía saudí “Saudi Aramco” probocando elevados daños a sus sistemas de información (Wangen, 2015).
	Gauss		Malware enfocado al ciberespionaje basado en Flame, diseñado con el objetivo de robar tanta información como sea posible del sistema infectado. Afectó principalmente al Líbano (Wangen, 2015).
2014	Careto	Político/militar (España →)	Malware avanzado de espionaje con posible origen en España que tenía como objetivos instituciones gubernamentales, embajadas, industria energética, compañías privadas, institutos de investigación y activistas mayoritariamente de Marruecos, Brasil y Gibraltar (Virus News, 2014).
	Dragonfly	Político/militar (Europa del este)	Descrito por Symantec como el ataque más ambicioso donde los ataques habían comprometido

			sistemas legítimos de fabricantes ICS/SCADA. Afectó a las industrias de aviación, defensa, y más recientemente energía. Kaspersky, en su análisis, considera que la motivación del ataque era la de reunir información sobre futuros objetivos potenciales (Wangen, 2015).
2016	DNC Hack	Político/militar (Rusia → USA)	Infiltración en los sistemas del Comité del Partido Demócrata de Estados Unidos, y de los servidores de correo utilizados por la candidata a la presidencia Hillary Clinton, por parte de dos grupos de origen ruso conocidos como “Cozy Bear” y “Fancy Bear” (Alperovitch, 2016).
	Project Sauron / Strider	Político/militar	Una de las APT más avanzadas conocidas hasta la fecha, que se cree ha estado activa desde 2011 en más de 30 organizaciones (gobierno, ciencia, militar, telecomunicaciones, finanzas) de Rusia, Irán, Ruanda, y posiblemente en países de habla italiana. Los servidores C&C estaban localizados en los Estados Unidos y en diversos países europeos. El objetivo de los atacantes era información sobre los sistemas de cifrado propios de las víctimas, así como documentación sensible y claves de cifrado (GREAT, 2016).

Se puede encontrar amplia documentación como documentos públicos, whitepapers y artículos sobre campañas APT en el siguiente repositorio de GitHub, APTnotes: https://github.com/kbandla/APTnotes.

3.5. El Ciberespionaje- Seguridad Nacional

Los ciberataques son una amenaza con la que terroristas, el crimen organizado, empresas, Estados o individuos aislados podrían poner en peligro infraestructuras críticas, vitales para el funcionamiento de un país. Entre otas 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, infraestructuras relacionadas con el suministro de agua potable, y el transporte de mercancías y personas, tecnologías y elementos relacionados con los sectores financieros, o cualquier otro servicio o activo que sea crítico para el funcionamiento de un país (Holguín et al., 2013).

Se han dado numerosos precedentes de de cómo un país puede sufrir serios daños ante un ciberataque, o cómo puede utilizarse en un acto de terrorismo (Holguín et al., 2013):

- **1985 – Ataque terrorista – grupo Middle Core Faction**: ataque al sistema de control de los ferrocarriles de alta velocidad en Japón. Resultado: 6,5 millones de usuarios afectados y coste económico de 6 millones de dólares.
- **Años 90** – **Conflictos bélicos** - robo de información estratégica – manipulación de la información.
- **2007** – Estonia es víctima del **primer ataque a gran escala contra un estado**. Resultado: sitios gubernamentales, medios de comunicación, bancos y diversas organizaciones vieron su seguridad de la información afectada.
- **2011** – **Intrusiones en bases de información del Gobierno** de Canadá con datos altamente confidenciales.
- **2012 – Recopilación y robo de información estratégica** a estados de Oriente Medio.

3.6. EL MERCADO DEL CIBERCRIMEN

Los criminales siempre encuentran la manera de obtener un rédito económico a partir de la desgracia ajena, bien sea directamente llevando ellos a cabo el robo o bien indirectamente vendiendo a otros delincuentes las herramientas necesarias para llevar a cabo el delito, y en el caso de los cibercriminales no es diferente.

Con el auge de la sociedad de la comunicación, ha habido un crecimiento paralelo del mercado del cibercrimen, el cual se nutre de la comercialización de malware hecho a medida, vulnerabilidades 0-day, exploits, ataques DDoS u otro tipo de sabotajes o ciberespionaje.

La posibilidad de llevar a cabo estas transacciones y ciberdelitos en el anonimato, gracias a las criptomonedas como Bitcoin en el caso de las transacciones, y gracias a sitios ubicados en la DarkNet accedidos a través del software libre de navegación cifrada y anónima TOR.

En el artículo “Top 20 Countries Found to Have the Most Cybercrime” (Sumo3000, 2010) podemos ver una lista con los 20 países con mayor ciberdelincuencia desarrollada por la empresa de seguridad informática Symantec. En ella podemos ver que los países que se encuentran en el top3 son Estados Unidos, China y Alemania.

3.7. Herramientas APT - Malware

Una de las características de las APT es que infectan el sistema víctima con software especialmente desarrollado para por un lado analizar la red y sistemas infectados, y por otro lado tomar su control, permitiendo la comunicación entre atacante y víctima sin que ésta sea consciente del hecho y adquiriendo un control total, tal y como si se tuviera acceso físico al dispositivo. Dicho software es definido como un tipo de *malware* conocido por las siglas **RAT**, que son el acrónimo de *Remote Administration Tool* , o lo que es lo mismo en español, *Herramienta de Administración Remota* (Siciliano, 2015).

Algunas de las herramientas **RAT** más conocidas son:

- AndroRAT.
- Back Orifice.
- NetBus.
- LuminosityLink.
- PoisonIvy.
- Sub7.
- DarkComet.
- Beast.
- Bitfrost.
- Blackshades.
- Optix Pro.

Además, para llevar a cabo la infección se utiliza un software conocido como **WebKits** o **Exploits**, cuyo funcionamiento básico consiste en utilizar servidores Web donde alojar un grupo de exploits para intentar aprovechar diversas vulnerabilidades a partir de navegadores y plugins de los clientes conectados; estos Webkits constan de una gran sofisticación y son frecuentemente actualizados con los últimos exploits y técnicas de cifrado, ofuscación y packing con los que evadir los dispositivos de seguridad (Holguín et al., 2013).

Para lograr que las víctimas descarguen en su equipo este *malware* existen diversos métodos, como campañas de Spear Phishing con emails que contienen links maliciosos, o la inclusión de código malicioso en páginas legítimas de modo que los usuarios sean redirigidos a las páginas en las que se descargará el malware.

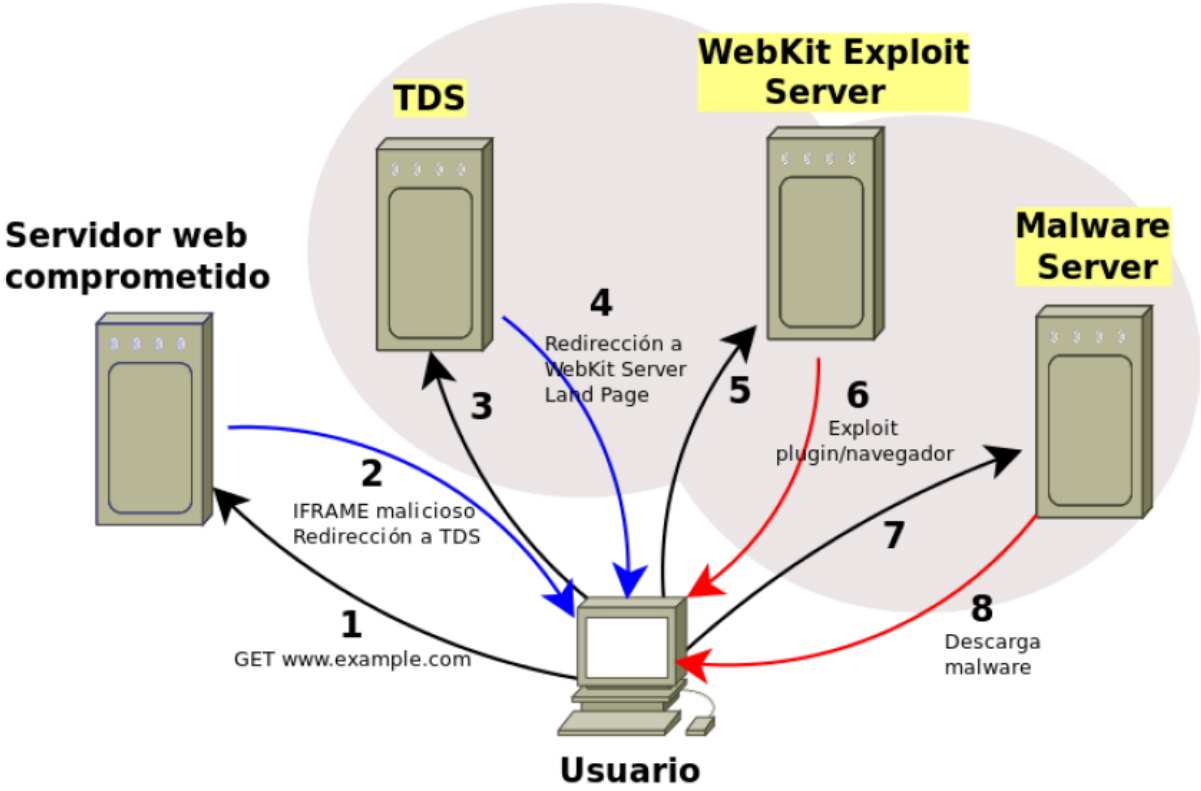


Figura 30: Proceso de infección llevado a cabo desde que un usuario accede a una página comprometida hasta que se descarga y ejecuta malware en su equipo.

(Holguín et al., 2013)

Algunos de los **WebKits** o **Exploits** más utilizados son (Holguín et al., 2013) :

- Blackhole.
- Phoenix.
- Unique.
- Eleonore.
- Liberty.

Otros WebKits recientemente detectados (Segura, 2016):

- Angler EK.
- RIG EK.
- Nuclear EK.
- Neutrino EK.
- Magnitud EK.

3.8. Actores

3.8.1 Grupos de atacantes conocidos

Algunos grupos de atacantes conocidos por sus actividades, pero sin estar directamente ligados a un estado (lo cual no quita que puedan estar esponsorizados o apoyados por un estado):

- Energetic Bear
- Turla.
- The Shadow Brokers
- Patchwork

3.8.2 Grupos de atacantes gubernamentales

En lo referente a grupos de acciones APT relacionados con agencias gubernamentales, se suelen mencionar los siguientes en los distintos estudios sobre ataques:

- APT1 (Unidad 61398) – China.
- APT28 (Fancy Bear) - GRU – Inteligencia militar rusa.
- APT29 (Cozy Bear) - FSB – Servicio Federal de seguridad de Rusia .
- The Equation Group - NSA TAO – Agencia de Seguridad Nacional EEUU.

3.8.3 Grupos gubernamentales de Ciberseguridad y Ciberdefensa

En cuanto a instituciones gubernamentales dedicadas a la ciberseguridad y ciberdefensa podemos encontrar, en España:

- Centro Criptológico Nacional (CCN).
- Instituto Nacional de Ciberseguridad (INCIBE).
- Mando Conjunto de Ciberdefensa (MCCD).

3.8.4 Empresas de Ciberseguridad

Algunas de las empresas con mayor prestigio a nivel internacional que suelen elaborar informes que son tomados como referencia en el mundo de la ciberseguridad:

- FireEye – Mandiant.
- Kaspersky.
- Symantec.

3.9. Metodologías de Detección de APT a través del tráfico de exfiltración

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 (Holguín, J. M., Moreno, M., Merino, B., 2013).

Si bien, incluso la mejor metodología de monitorización no puede garantizar la detección del código de la APT atacante, debido a que ésta dependen del control y acceso remoto, la actividad de red asociada con dicho control puede ser identificada, contenida e interrumpida a través del análisis del tráfico de red saliente (Binde et al., 2011).

Entre las distintas **metodologías** que se pueden implementar para detectar APT se pueden encontrar:

- Conjuntos de reglas basadas en firmas.
- Métodos estadísticos y de correlación.
- Aproximaciones Manuales.
- Bloqueo automático de exfiltración de información.

3.9.1 Reglas de Firmas

La metodología de detección de APT por reglas de firmas, comúnmente utilizada en Sistemas de Detección de Intrusiones (IDS), que consiste en buscar patrones de coincidencia del tráfico de red en relación a un patrón preestablecido (firma) que pueda alertar de una anomalía o un posible ataque.

Algunos ejemplos de utilización de esta metodología para la prevención del ataque RSA serían (Binde et al., 2011):

- Identificación de campañas de phishing.
- Reconocimiento y bloqueo de tráfico malicioso tal como el asociado con PI-RAT.
- Monitorización del Registro de Windows en búsqueda de entradas peligrosas conocidas.

A continuación se muestra un ejemplo de firma producida por Avert Labs para IDS que permita detectar el tráfico del C&C de la APT Aurora:

```
alert tcp $HOME_NET any -> $EXTERNAL_NET 443 (msg:"ET TROJAN Aurora C&C
Checkin"; flow:established,to_server; content:"|ff ff ff ff ff ff 00 00
fe
ff ff ff ff ff ff ff ff ff 88 ff|"; offset:0; depth:20;
classtype:trojan-activity; reference:url,
www.avertlabs.com/research/blog/index.php/2010/01/18/an-insight-into-
the-aurora-communication-protocol/;
sid:10000000001; rev:1;)
```

Figura 31: Regla para IDS para detectar el comportamiento del malware Aurora. (Binde et al., 2011)

Desventaja

Debido a que está basada en reglas que detectan una amenaza previamente conocida, no permite detectar aquellas amenazas que no han sido previamente detectadas, analizadas y documentadas, por lo que sería inútil para cualquier ataque en el que se utilizase nuevo malware.

3.9.2 Métodos estadísticos y de correlación

Por "**métodos estadísticos y de correlación**" se entiende la búsqueda de anomalías en el tráfico de red respecto a lo que sería el tráfico habitual y esperable en la instalación a proteger. Una de las *dificultades* para llevar a cabo su implementación es establecer la línea base: ¿cuales son los parámetros "normales?. Y es que éstos pueden diferir en función de la red, el tipo de tráfico que transporta, etc. Una errónea configuración de los filtros podría dar lugar a múltiples falsos positivos (Couture, 2010).

A continuación se definen una serie de *indicadores* que ayuden a detectar de una manera rápida si algo anómalo sucede en nuestra red (Holguín et al., 2013):

Capa de Red. Geolocalización:

La monitorización de la geolocalización del tráfico de red nos permite establecer un nuevo indicador para la detección de APT. Y es que conociendo la organización cuales son los patrones habituales de conexiones que realizan sus equipos hacia

según qué países, se puede establecer la ubicación geográfica de las conexiones salientes como indicador de referencia para conocer si se está produciendo una exfiltración de información ilegítima. Por ejemplo, si se detecta que el equipo de un usuario establece conexiones empleando un protocolo seguro SSL hacia Rusia, país con el que la empresa no tienen ninguna relación, debería de llevarse a cabo una comprobación del motivo de la conexión para descartar que fuera ilícita.

Capa de Transporte:

- **Relación TCP SYN, TCP SYN/ACK y RST**:
  En condiciones normales, el número de TCP SYN debe encontrarse próximo al de TCP SYN/ACK, por lo que si existe una diferencia pronunciada entre ambos valores, esto podría indicar una anomalía en nuestra red, que podría ser debida a ataques de denegación de servicio DDOS de tipo TCP SYN flood o bien a la existencia de *Covert Channels* para exfiltración de información. Además, un número de TCP RST respecto a TCP SYN divergente puede ser indicativo de la extracción de información mediante una técnica de Covert Channels.

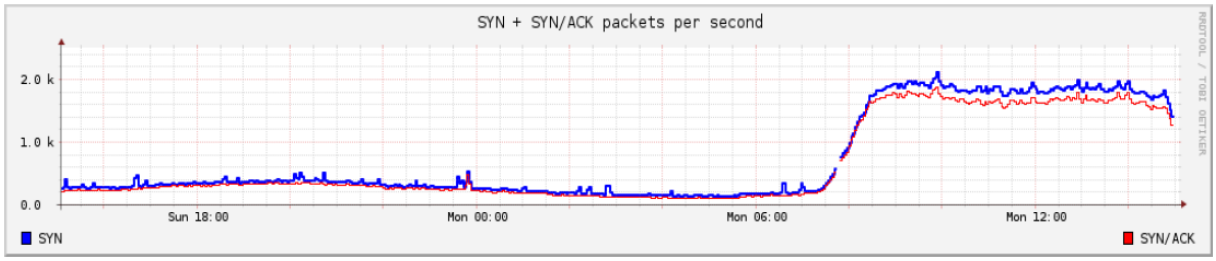


Figura 32: Comportamiento normal TCP/SYN – TCP/SYN/ACK.

(Holguín et al., 2013)

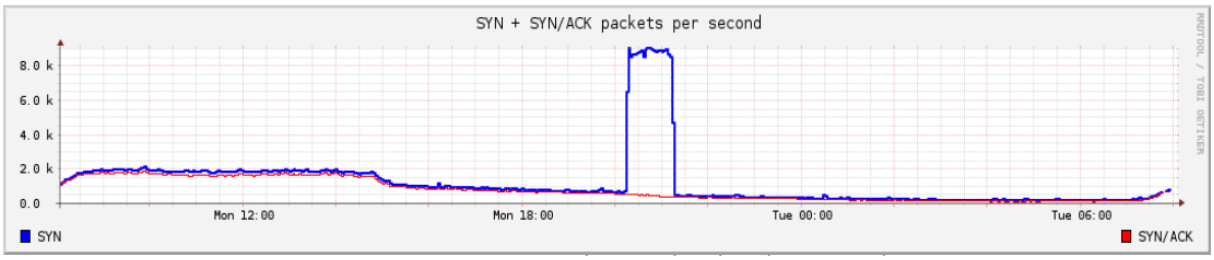


Figura 33: Comportamiento anormal TCP/SYN-TCP/SYN/ACK.

(Holguín et al., 2013)

- **Tamaño de los paquetes**:
  Conociendo el tamaño medio de los paquetes UDP o TCP que entran y salen hacia Internet, podemos tener un indicador general donde observar si se produce una variación significativa que podría deberse a:
  - Si *aumenta* el valor medio: extracción de información.
  - Si *disminuye* el valor medio: nuevas vías de comunicación hacia el exterior.

- **Número de paquetes por puerto TCP**:
  Conociendo el número de paquetes por puerto TCP que se han generado, se puede establecer un indicador si éstos se analizan de manera periódica. Si se produce una variación para un rango horario, debería generarse una alerta en los sistemas de seguridad de la organización. Un ejemplo de indicadores sería la media diaria de paquetes enviados al puerto 80/TCP en horario laboral, y la media diaria de paquetes enviados al puerto 80/TCP fuera del horario laboral.

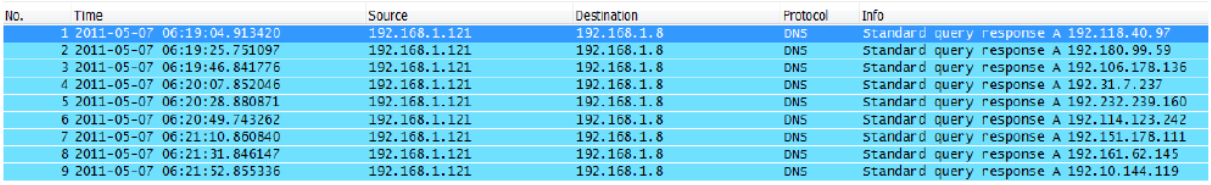
- **Número de conexiones TCP**:
  Otro indicador a tener en cuenta es el número de conexiones TCP que se producen de manera periódica en la organización. Si se aprecia un incremento en la media por periodo de tiempo, éste ha de ser controlado por el equipo de seguridad.

- **Volumen de tráfico:**
  Definiendo una serie de umbrales de cantidad de tráfico por periodo de tiempo, de manera que sea más restrictiva en determinadas ventanas temporales, se obtendrá un indicador más. Un aspecto importante a contemplar en horario no laboral, es la variación de tráfico que viene determinada por las copias de seguridad, ya que podría provocar falsos positivos.

- **Distribución del tráfico por servicio**:
  Para establecer este indicador, es necesario que la organización tenga definida una política que defina los servicios que pueden utilizarse y aquellos que no. Se buscarán anomalías en los servicios utilizados y el periodo en que han sido utilizados, de modo que si un servicio determinado es monitorizado generando tráfico a una hora en que no debería de hacerlo, saltará una alarma para que el administrador del sistema lo investigue.

Detección de redes Fast-Flux

Tal como se ha descrito en el apartado 2.2.2.1, las redes Fast-Flux consisten en que para un mismo servidor nodriza se tendrán múltiples direcciones IP proxy a través de máquinas infectadas en un periodo de tiempo muy corto. Por ello, un indicador a tener en cuenta como alerta de que se está produciendo una incidencia de seguridad sería el que se produjeran una cantidad excesiva de respuestas con direcciones IP únicas para un mismo dominio.



No.	Time	Source	Destination	Protocol	Info
1	2011-05-07 06:19:04.913420	192.168.1.121	192.168.1.8	DNS	Standard query response A 192.118.40.97
2	2011-05-07 06:19:25.751097	192.168.1.121	192.168.1.8	DNS	Standard query response A 192.180.99.59
3	2011-05-07 06:19:46.841776	192.168.1.121	192.168.1.8	DNS	Standard query response A 192.106.178.136
4	2011-05-07 06:20:07.852046	192.168.1.121	192.168.1.8	DNS	Standard query response A 192.31.7.237
5	2011-05-07 06:20:28.880871	192.168.1.121	192.168.1.8	DNS	Standard query response A 192.232.239.160
6	2011-05-07 06:20:49.743262	192.168.1.121	192.168.1.8	DNS	Standard query response A 192.114.123.242
7	2011-05-07 06:21:10.860840	192.168.1.121	192.168.1.8	DNS	Standard query response A 192.151.178.111
8	2011-05-07 06:21:31.846147	192.168.1.121	192.168.1.8	DNS	Standard query response A 192.161.62.145
9	2011-05-07 06:21:52.855336	192.168.1.121	192.168.1.8	DNS	Standard query response A 192.10.144.119

Figura 34: Tráfico Fast-Flux.

(Binde et al., 2011)

3.9.3 Aproximaciones manuales

En determinadas ocasiones, cuando el ataque APT utiliza una vulnerabilidad de día cero, la amenaza puede permanecer sin ser detectada. Por ello, puede darse la situación en que el ataque solo pueda ser detectado por medio de la *monitorización* y el examen de los *logs* del sistema.

Algunos ejemplos de comportamiento anómalo qué podemos detectar mediante aproximaciones manuales (Binde et al., 2011):

- Tráfico de ingreso extraño, iniciado por la organización víctima en lugar de por el atacante.
- Logs DNS.
- Tráfico anómalo comparado con las líneas base.
- Movimiento lateral.

Por ello, es importante afinar correctamente los sistemas de presentación de información recogida para que el analista pueda distinguir un comportamiento anómalo. Como ejemplo, se muestra a continuación una imagen en la que se puede observar como el host víctima 192.168.1.6 está sondeando su vecindario en la red buscando servicios conocidos que el atacante pueda explotar en caso de presentar vulnerabilidades:

Figura 35: Visualización en Squert/AfterGlow del escaneo de sondeo de un host de pivotaje. (Binde et al., 2011)

3.9.4 Bloqueo automático de exfiltración de información

Los sistemas IDS pueden bloquear la exfiltración de información por medio de la detección de las características del tráfico de salida. Por ejemplo, se podría bloquear la transmisión de archivos en formato que permiten ocultar la información exfiltrada, como por ejemplo el formato de compresión RAR.

A continuación se muestran ejemplos de métodos de bloqueo automático de exfiltración de información (Binde et al., 2011):

- **Detección de la exfiltración de archivos RAR**
  En el ataque a RSA Security la información fue ofuscada mediante el uso de archivos RAR, lo que ha llevado al desarrollo de reglas para detectar las firmas de ficheros RAR que salen de la red. Se llevarían a cabo los siguientes procesos:
  - Configurar el firewall (iptables) para bloquear el tráfico basado en la firma.
  - Configurar el NIDS (Snort) para la detección.

```
alert tcp $HOME_NET any -> $EXTERNAL_NET any (msg:"ET POLICY RAR
File Outbound"; flow: established; content:"|52 61 72 21|";
offset: 0; depth: 4; tag: session; classtype: not-suspicious;
sid: 2001950; rev:3;)
```

Figura 36: regla Snort de detección de archivos RAR salientes en base a su firma. (Binde et al., 2011)

- **OSSEC Active Response**

  Permite la ejecución automática de comando o respuestas cuando un evento específico, o un conjunto de eventos, suceden. Además, puede ser gestionado de manera escalable permitiendo la ejecución preventiva de comando tanto en el lado del servidor como en el del cliente. En definitiva es un gran elemento disuasorio contra escaneo de puertos, ataques de fuerza bruta y algunos otros tipos de ataques de fingerprinting. No obstante hay que tener cuidado en su afinado, ya que si no puede dar lugar a diversos falsos positivos.

- **Uso de Proxy**

  Otra manera de prevenir que el atacante tenga éxito en la exfiltración de la información consiste en integrar un proxy en el entorno. Esto permitiría bloquear tanto el intento de conexión a páginas web fuera de la “lista blanca” o ACL de la

organización, como bloquear cualquier intento de realizar una conexión en la capa de aplicación sin pasar a través del Proxy.

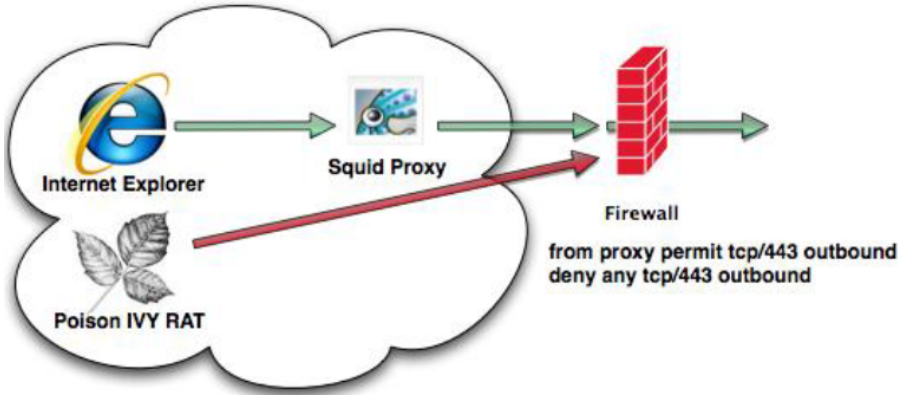


Figura 37: Protección perimetral a través de Proxy.

(Binde et al., 2011)

3.10. Herramientas de carácter libre para detección de APT

La detección de ataques de Amenazas Persistentes Avanzadas son el fruto de la integración de diversos sistemas de análisis y detección, ya que por su propia naturaleza las APT constan de diversas características que, en conjunto, las identifican, por lo que para poder detectarlas a tiempo necesitamos emplear diversas herramientas que nos permitirán analizar el estado del sistema desde distintos puntos de vista. A esto se suma el hecho de que sean en gran medida personalizadas para cada ataque, lo cual lleva a que no porque conozcamos la firma y características de una APT seamos capaces de detectar automáticamente o con una sola herramienta una nueva amenaza.

Este documento tiene como objetivo el llevar a cabo dicha detección mediante el uso de herramientas de carácter libre, sin coste de adquisición para el usuario u organización, de modo que a continuación se procede a enumerar y describir brevemente algunas de las que se pueden encontrar para integrar en un sistema de detección de ataques APT.

Analizadores de Red

- **Wireshark** https://www.wireshark.org/):
  Herramienta multiplataforma de carácter libre (licencia *GPL*) de análisis de protocolos que mediante su interface gráfica permite ver todo el tráfico que pasa a través de una red. Entre las funcionalidades incluidas se encuentran:
    - Inspección profunda de protocolos.
    - Captura de paquetes en tiempo real y análisis offline.
    - Análisis de VoIP.
- **Nmap** (https://nmap.org/):
  Herramienta de carácter libre y de licencia *GPL* para llevar a cabo escáneres de red y auditoría de seguridad. Mediante el uso de paquetes IP nos permite determinar, entre muchas otras cosas:
    - Dispositivos disponibles en la red.
    - Servicios ofrecidos por los dispositivos.
    - Sistema Operativo de los dispositivos (y su versión).
    - Tipo de firewalls utilizados.

- **Nfdump** (http://nfdump.sourceforge.net/):
  Herramienta de código abierto para capturar tráfico de red, almacenar la información en archivos, y mostrar estadísticas y diversas información acerca de la información capturada.
  - **Nfsen** (http://nfsen.sourceforge.net/):
    Frontend gráfico web para representar de una manera intuitiva y visual los datos creados por la herramienta Nfdump.
- **Ntopng** (http://www.ntop.org/):
  Herramienta de código abierto (que podemos encontrar en dos versiones, una de pago, ntopng Pro, y otra gratuita, ntopng Community) de monitorización de tráfico de red de alto rendimiento y baja demanda de recursos, diseñada como sustituta de ntop.

Proxy web

- **Squid** (http://www.squid-cache.org/):
  Servidor proxy web con caché de carácter libre (licencia GPL), multiplataforma, que permite:
    - Mejorar el rendimiento de las conexiones a Internet.
    - Acelerar el acceso a un servidor web.
    - Añadir seguridad realizando filtrado de tráfico en capa de aplicación del modelo OSI.
- **SquidGuard** (http://www.squidguard.org/):
  Herramienta de redireccionado de URLs utilizado con el proxy Squid.
- **DansGuardian** (http://dansguardian.org/):
  Filtro de contenido web de código abierto para entornos UNIX/LINUX, que filtra el contenido de las páginas web en función de diversos métodos incluyendo coincidencias de frases, filtro de imágenes y filtrado de URLs. No filtra únicamente en función de una lista de sitios baneados como la mayoría de filtros comerciales.

Firewalls

- **Modsecurity** (https://www.modsecurity.org/):
  Firewall, de código abierto, de monitorización en tiempo real *aplicaciones web* (WAF), que se ejecuta como módulo del servidor web Apache, y provee protección contra diversos ataques hacia aplicaciones Web, y permite monitorizar tráfico HTTP.

- **Netfilter / Iptables** (https://www.netfilter.org/):
  Netfilter es un framework disponible en el núcleo de Linux que permite interceptar y manipular paquetes de red. Iptables es un cortafuegos que pasa por ser la herramienta más popular de Netfilter, permitiendo no solo filtrar paquetes, sino también realizar traducción de direcciones de red o mantener registros de log.

Sistemas de Detección de Intrusiones - IDS

- **Host-based Intrusion Detection System (HIDS)**
  - **OSSEC** (http://ossec.github.io/)**:**
    Herramienta escalable y multiplataforma de código abierto, con un poderoso motor de análisis y correlación, que integra análisis de log, chequeo de integridad de archivos, monitorización del registro de Windows, aplicación de políticas centralizadas, detección de rootkits, alertas en tiempo real y respuesta activa.
  - **Samhain Labs** (http://www.la-samhna.de/samhain/)**:**
    Software de código abierto multiplataforma para sistemas POSIX (Unix, Linux; Cygwin/Windows) que pasa por ser la principal competencia de OSSEC. Entre las diferencias encontramos por un lado que el agente posee una variedad de métodos de salida como un servidor central, Syslog, email y RDBMS. Otra diferencia importante radica en dónde el análisis tiene lugar: a diferencia de OSSEC, el procesamiento se lleva a cabo en el cliente, lo cual si bien proporciona una ventaja en términos de velocidad de procesamiento, puede tener un impacto en el rendimiento de los servidores.

- **Network Intrusion Detection System (NIDS)**
  - **Snort** (https://www.snort.org/):
    Sistema de Prevención de Intrusiones de Red basado en reglas, capaz de llevar a cabo análisis de tráfico en tiempo real y registro de paquetes en redes IP. Además, puede analizar protocolos, buscar/comparar contenido, y ser usado para detectar diversos tipos de ataques y fingerprinting, tales como buffer-overflows, escáneres de puertos, ataques CGI, etc.
  - **Suricata** (https://suricata-ids.org/)**:**
    NIDS de alto rendimiento de código abierto propiedad de la fundación sin ánimo de lucro gestionada por una comunidad Open Information Security Foundation

(OISF). A pesar de que su arquitectura es diferente de la de Snort, puede usar las mismas firmas ya que se comporta de la misma manera.

- **Bro** (https://www.bro.org/):
  NIDS de diferentes características que Snort y Suricata, ya que es al mismo tiempo un IDS basado en firmas y anomalías, cuyo motor de análisis convierte el tráfico capturado en series de eventos, que pueden ser desde la conexión de un usuario a FTP, hasta una conexión a una página web. Su poder radica en el *Intérprete de Scripts de Políticas*, el cual tiene su propio lenguaje (Bro-Script) y puede realizar tareas muy potentes y versátiles.
- **Kismet** (https://www.kismetwireless.net/)**:**
  Considerado como el referente de los denominados Wireless-IDS o WIDS or también WiFi NIDS, es un detector de redes, packet-sniffer e IDS que identifica redes recolectando paquetes de manera pasiva y detectando redes, lo cual le permite detectar redes ocultas a través del tráfico de red.

Sistemas de Monitorización de Seguridad de Red (NSM)

- **Sguil** (http://www.sguil.net):
  Colección de componentes de software libre para Monitorización de Seguridad de Red y análisis de alertas de IDS por medio de eventos. Su principal componente es una interface gráfica (GUI) que permite visualizar en tiempo real eventos, información de sesión, y paquetes capturados.
- **Squert** (http://www.squertproject.org/):
  Aplicación web utilizada para consultar y ver información almacenada en una base de datos Sguil, mediante su interface gráfica, aportando contexto adicional a los eventos, por medio del uso de metadatos, representaciones de series temporales y conjuntos de resultados agrupados de manera lógica.
- **Enterprise Log Search and Archive (ELSA)** (https://github.com/mcholste/elsa):
  Software receptor, archivador e indexador de logs y frontend web para syslog.

Herramientas de Análisis Forense de Red (NFAT):

- **Xplico** (http://www.xplico.org/):
  Herramienta de código abierto, para entornos Linux, que permite extraer de una captura del tráfico de Internet la información sobre las aplicaciones contenidas.
- **NetworkMiner** (http://www.netresec.com/?page=NetworkMiner):

Herramienta NFAT de carácter libre (aunque con una versión de pago más completa) multiplataforma, que que facilita el llevar a cabo Análisis de Tráfico de Red Avanzado (NTA) al presentar los datos en una interface de usuario intuitiva. Puede ser utilizado como:

- Herramienta de captura de paquetes o network sniffer para detectar sistemas operativos, sesiones activas, nombres de dispositivos, puertos abiertos, etc.
- Tratar ficheros PCAP para análisis online y reconstrucción de archivos y certificados transmitidos.

- **OpenIOC** (http://www.openioc.org/):
  Open Indicators of Compromise es un framework que, mediante un esquema xml extensible para la descripción de características técnicas que identifican un peligro conocido, una metodología de ataque u otra evidencia de compromiso, permite compartir información de amenazas entre organizaciones. A continuación se enumeran algunas herramientas de carácter libre desarrolladas por FireEye para gestionar IOC:
  - **IOC Editor** (https://www.fireeye.com/services/freeware/ioc-editor.html) :
    Interface para gestionar información y manipular estructuras lógicas de Indicadores de Compromiso.
  - **IOC Finder** (https://www.fireeye.com/services/freeware/ioc-finder.html):
    Herramienta para recolectar información de sistema del host y reportar la presencia de IOCs.
  - **IOC Writer** (https://github.com/mandiant/ioc_writer):
    Librería de Python que permite la creación y edición básica de objetos OpenIOC.

Detectores de Malware

- **VirusTotal** (https://www.virustotal.com/es/):
  VirusTotal es una empresa subsidiaria de Google que ofrece un servicio online gratuito que analiza archivos y URLs permitiendo identificar virus, gusanos, troyanos y otro tipo de malware. Además puede ser utilizado para detectar falsos positivos, como por ejemplo archivos o URLs detectados como maliciosos por otros escáneres.
- **ClamAV** (https://www.clamav.net/):
  Antivirus de código abierto multiplataforma para detectar virus, troyanos y demás malware. Uno de sus muchos usos es el de escáner de virus de email en el lado del servidor.

Sistemas de Gestión de Incidentes y Eventos – SIEM

- **Security Onion** (https://securityonion.net/):
  Distribución de Linux basada en Ubuntu que ofrece:
  - Detección de intrusiones.
  - Monitorización de la seguridad de la red.
  - Gestión de logs.

  Entre otras ***herramientas de seguridad incluidas*** en esta distribución se puede encontrar (https://github.com/Security-Onion-Solutions/security-onion/wiki/Tools):
  - Snort (NIDS):
  - Suricata (NIDS).
  - Bro (NIDS).
  - OSSEC (HIDS).
  - Sguil (NSM).
  - Squert (NSM).
  - ELSA (NSM).
  - Xplico (NFAT).
  - NetworkMiner (NFAT).

  Para más información acerca de esta distribución, consultar la Wiki habilitada en su página web: https://github.com/Security-Onion-Solutions/security-onion/wiki/IntroductionToSecurityOnion

- **AlienVault OSSIM** (https://www.alienvault.com/products/ossim):
  Open Source Security Information and Event Management es un producto de AlienVault que provee de un SIEM de licencia *open source* con recolección, normalización y correlación de eventos. Éste ofrece las siguientes funciones de seguridad:
  - Descubrimiento de dispositivos.
  - Evaluación de vulnerabilidades.
  - Detección de intrusiones.
  - Monitorización de comportamiento.
  - Información de seguridad y gestión de eventos.
  - Integración del sistema de compartir indicadores de amenazas *Open Threat Exchange* (OTX).

Entre otras ***herramientas de seguridad incluidas*** en esta distribución se puede encontrar:

- Suricata (NIDS).
- Bro (NIDS).
- OSSEC (HIDS).
- Kismet (NIDS).

Otras herramientas:

- **fwsnort** (http://www.cipherdyne.org/fwsnort/):
  Herramienta que traduce los archivos incluidos en Snort y contruye el conjunto de reglas equivalente de Iptables, para tantas reglas como sea posible. Además, utiliza el módulo de combinación de cadenas de iptables para detectar ataques en la capa de aplicación.
- **AfterGlow** (http://afterglow.sourceforge.net/):
  Colección de scripts incluida en Squert, y que hace uso de las librerías graphviz para generar visualizaciones gráficas en tiempo real del tráfico de red capturado y así facilitar el análisis de conjuntos de datos complejos

Scripts de seguridad

- **dnsWatch.py** (https://github.com/ctxis/DNSWatch):
  Script, escrito en Python para Scapy, que analiza el tráfico de red (sea en tiempo real o a través de un fichero PCAP) en búsqueda de un token concreto, emitiendo un informe sobre el resultado de la búsqueda..
- **ActiveState mini fake dns-server** (http://code.activestate.com/recipes/491264-mini-fake-dns-server/history/4/):
  Script escrito en Python para generar tráfico DNS falso que simula el generado por el tráfico Fast-Flux. En el se puede definir un dominio (por ejemplo “hacker.ru”), así como manipular el número de paquetes, peticiones, punteros, tipos de respuesta, y el Time To Live (TTL).
- **Barnyard2** (https://github.com/firnsy/barnyard2):
  Intérprete de código abierto para los binarios unified2 de salida de Snort incluido en Security Onion. Su principal cometido es permitir que Snort deje la tarea de traducir los binarios a distintos formatos a otro proceso, para que Snort escriba en disco de una manera más eficiente y no pierda tráfico de red.

- **PyOleScanner** (https://github.com/Evilcry/PythonScripts/blob/master/pyOLEScanner.py): Herramienta que, dado un fichero OLE2 (doc, xls, ppt), busca shellcodes, ejecutables embebidos, la presencia de APIs, aplica fuerza bruta XOR, y finalmente muestra si el archivo de Microsoft Office es malicioso o si por el contrario está limpio.
- **trackByGeo.py** (Binde et al., 2011): Script creado por Binde et al. (2011) que, haciendo uso de la API en Python *MacMind GeoIP* (https://www.maxmind.com/en/open-source-data-and-api-for-ip-geolocation), al ejecutar *trackByGeo.py <foo.pcap>* devuelve un informe de resultados incluyendo las direcciones IP de origen y destino clasificadas por países.

3.11. Prevención de APT

A continuación se enumeran tácticas eficaces para ayudar a establecer una defensa en profundidad (Couture, 2010):

- **Principio de mínimo privilegio**: Si los usuarios no tienen derechos administrativos para ejecutar software no autorizado, acceder a la consola de comandos, enviar pings ICMP, etc, será menos probable que se puedan establecer canales encubiertos.
- **Política de seguridad fuerte**: El desarrollo, mantenimiento e implantación de una política de seguridad fuerte disuadirá a los individuos menos motivados para realizar un ataque.
- **Educación y concienciación**: Cuanto mayor sea el conocimiento sobre seguridad de la información de los empleados, y mayor sea su concienciación sobre los peligros de actuar incorrectamente, mayor será su efectividad para detectarlos.
- **Hardening de los servidores**: Ayudará a minimizar la exposición a todas las vulnerabilidades.
- **Proxies web y DNS**: Su instalación fuerza que todas las peticiones externas DNS y HTTP pasen a través suyo, ayudando a combatir peticiones irregulares.
- **IDS**: Su instalación puede destacar características de troyanos, puertas traseras y malware conocidos que pudieran estar actuando en nuestra red.

- **Línea base de tráfico de red**: Permitirá que los algoritmos de detección de anomalías sean más eficaces.
- **Normalización del tráfico**: Eliminar posibles canales encubiertos mediante la sobreescritura y padding de campos explotables de los protocolos con ruido aleatorio.
- **Deshabilitar ICMP**: Ignorar los paquetes ICMP entrantes, excepto aquellos de los administradores de red.
- **Logging**: Ya que los canales encubiertos son descubiertos normalmente a través de efectos secundarios, un conjunto meticuloso de logs puede ayudar en gran medida en la investigación del origen de la brecha.
- Bloquear, o permitir solo aquellos tipos de **mensajes ICMP y DNS** requeridos para el adecuado funcionamiento.
- **Restricciones de tráfico explotable**: Si bien los canales encubiertos no pueden ser completamente evitados, sí que se puede reducir su efectividad limitando el ancho de banda, o implantando restricciones para el tráfico que explotan.

4. DESARROLLO ESPECÍFICO DE LA CONTRIBUCIÓN

Este apartado representa la culminación del trabajo realizado previamente a lo largo del TFM, de documentación y análisis de información relacionada con las APT, con el objetivo de profundizar en su comprensión y mejorar la capacidad de lucha contra ellas, en lo referido a su detección.

A continuación se expone una descripción detallada del experimento, dividida en cuatro apartados. En primer lugar se tratará la composición del entorno de trabajo o laboratorio. A continuación se indicarán los pasos seguidos para la instalación y configuración de sus distintos componentes, para posteriormente describir paso a paso la prueba realizada. Finalmente, se llevará a cabo la exposición de los resultados obtenidos.

4.1. Descripción detallada del experimento

4.1.1 Descripción del Laboratorio

En este apartado se tratará lo concerniente a la estructura y composición del laboratorio o entorno de trabajo, sin entrar todavía en la configuración o en el piloto en sí.

4.1.1.1 Descripción y Configuración del Entorno de Trabajo

El entorno de trabajo se implementará a su vez sobre un entorno virtualizado para poder aislar los equipos de prueba de la red local e Internet para por un lado evitar la propagación del malware, y por otro lado evitar interacciones no deseadas de software de otros equipos que pueda distorsionar u ocultar los datos de interés.

4.1.1.2 Software de Virtualización

El programa elegido para llevar a cabo la implementación del entorno virtualizado a sido **VirtualBox** (https://www.virtualbox.org/), el cual aúna dos características clave para formar parte del presente trabajo: ser software libre y fiabilidad, en su versión 5.1.4r110228.

VirtualBox, tal como podemos ver en su página web, es un virtualizador de propósito general para hardware de arquitectura x86, enfocado a servidores, sobremesa y uso embebido.

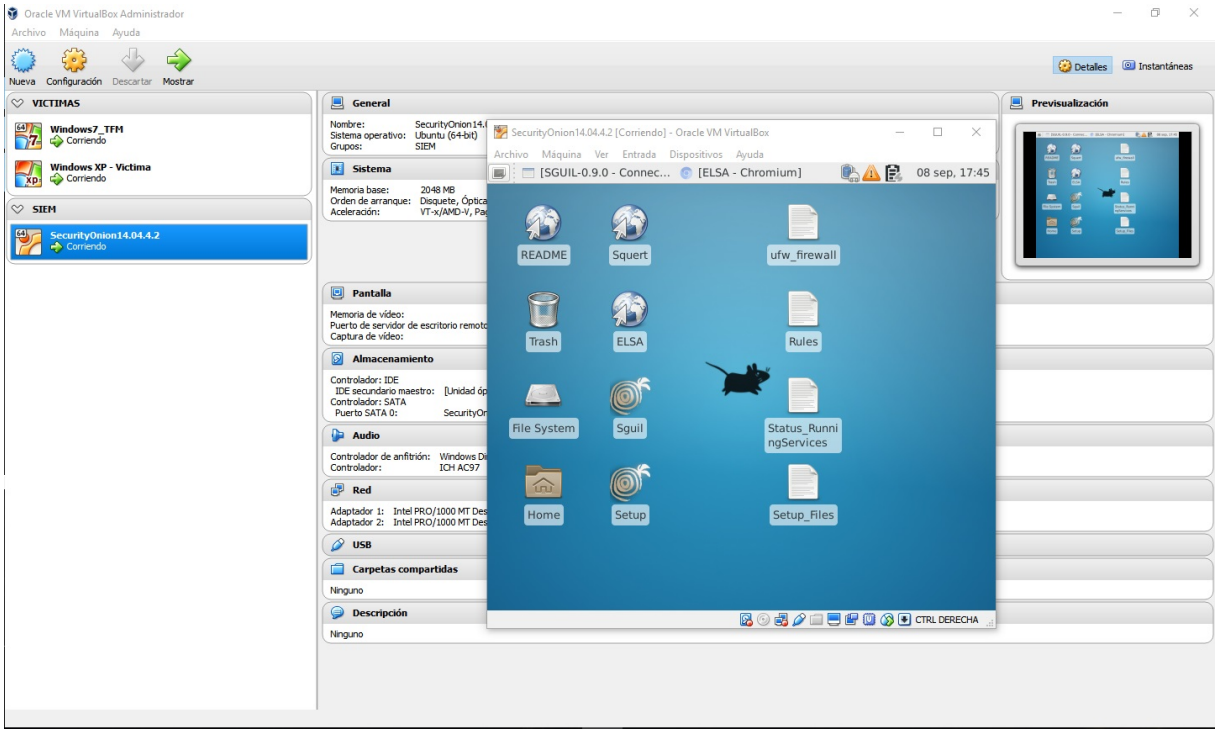


Figura 38: Máquina virtual con Security Onion corriendo sobre Windows10.

4.1.1.3 Herramientas de análisis y monitorización

Para llevar a cabo la monitorización y el análisis de la red y los sistemas se ha decidido utilizar la distribución ***Security Onion*** (https://securityonion.net/), desarrollada por Doug Burks, en su ***versión 14.04.4.2***, mencionada y descrita previamente en el apartado 3.9 de este documento, al igual que las herramientas en ella incluida. Las razones para escoger dicha distribución por encima de otras son varias:

- Es una distribución conformada enteramente por software libre.
- Este bundle de herramientas posee todas las herramientas necesarias para llevar a cabo el cometido del presente trabajo.

4.1.1.4 Equipo Víctima

Como sistema operativo para la máquina víctima se ha elegido el SO ***Microsoft WindowsXP***, ya que representa un host medio que se podría encontrar de manera habitual en el periodo en que PoisonIVY fue desarrollado y utilizado, además de poder encontrarse todavía en múltiples instalaciones de todo el mundo, que a pesar de poseer una relevancia importante, no han sido actualizadas a sistemas operativos más modernos.

4.1.1.5 Equipo Atacante

En el equipo atacante se ha optado por instalar Microsoft Windows 7 Professional SP1, ya que, por un lado representa en la actualidad el sistema operativo medio de usuario, tanto a nivel particular como profesional, y por otro lado nos permite ejecutar la herramienta PoisonIVY, diseñada para ser utilizada en Windows, tanto a nivel de cliente (atacante) como de servidor (víctima).

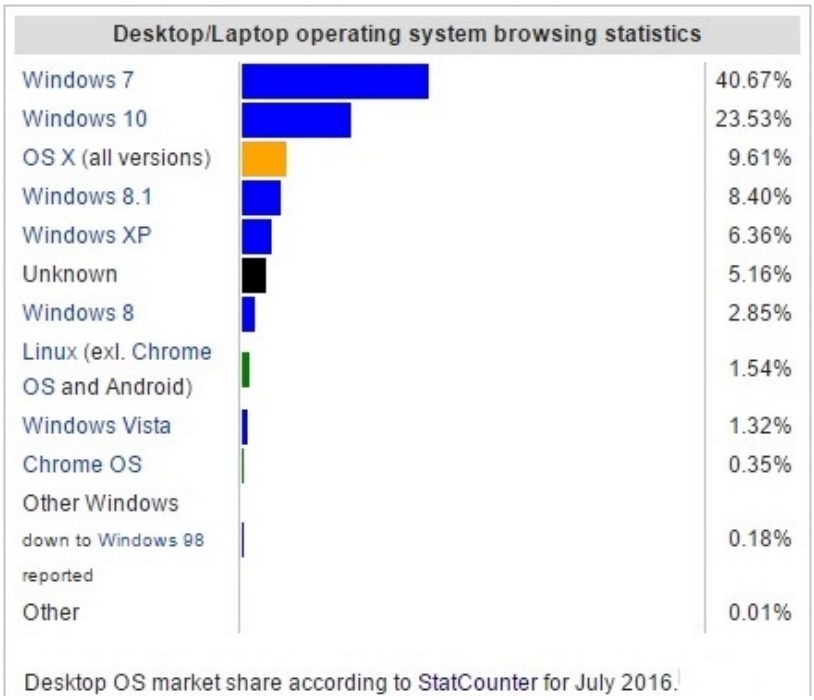


Figura 39: Cuota de mercado de sistemas operativos sobremesa/portátil.

Recuperado el 1 de septiembre de https://en.wikipedia.org/wiki/Usage_share_of_operating_systems

4.1.1.6 Equipos de Covert-Channel

Para la simulación de un canal encubierto vamos a utilizar como Cliente y Servidor dos MV Linux debido a que el software que vamos a utilizar (ptunnel) está desarrollado para dichos sistemas. Dichas máquinas virtuales serán:

- **Linux Mint Cinamon 18**.
- **Kali Linux 2016.2 64bit**.

4.1.1.7 Herramienta de APT

El **malware** que utilizado para esta prueba será el **Poison IVY-RAT** en su versión **2.3.2**, herramienta de administración remota para Windows, que ha sido utilizado en diversas campañas de malware de alto perfil, siendo los ataques más notorios, el ataque a RSA SecurID de 2011 y la campaña conocida como Nitro, también en 2011, en su versión 2.3.2, última versión de dicho malware que vio la luz por primera vez en 2005 y sufrió su última revisión en 2008 (FireEye, 2014).

Entre las funcionalidades a destacar de PoisonIvy se encuentran:

- Captura de pulsación de teclado.
- Capturas de pantalla.
- Captura de video.
- Transferencia de ficheros.
- Administración de sistema.
- Robo de contraseñas.
- Redirección de tráfico de red.

Ataque - Infección

El entorno de PoisonIVY, por medio de una GUI intuitiva, permite a los atacantes personalizar y configurar su propio *servidor*, que será enviado a la víctima, siendo el *cliente* el encargado de recibir y enviar las órdenes a las máquinas infectadas.

El funcionamiento del código de este malware, en la máquina víctima, se divide en dos partes:

- Código de inicialización y mantenimiento.
- Código de red.

El proceso de infección comienza con la inyección del *código de inicialización y mantenimiento* en el proceso *explorer.exe* de Windows, para a continuación, el *código red*, dependiendo de como el atacante lo haya configurado, inicia un proceso oculto del navegador web y se inyecta dentro de dicho proceso. Así, el código de red descarga remotamente el resto del código e información necesario para su funcionalidades.

4.1.1.8 Herramienta de Covert Channel

Para llevar a cabo la simulación de un canal encubierto en la red se va a recurrir al software **ptunnel**, el cual aparece mencionado y descrito previamente en este documento en el apartado "*3.2.2.2 Covert Channel*", que nos permitirá crear un covert-channel de tipo *almacenamiento*.

4.1.2 Instalación y configuración del laboratorio

Instalación y Configuración del Entorno de Virtualización

En primer lugar llevaremos a cabo la instalación del software de virtualización, que en este caso será Oracle VM VirtualBox 5.1.4r110228.

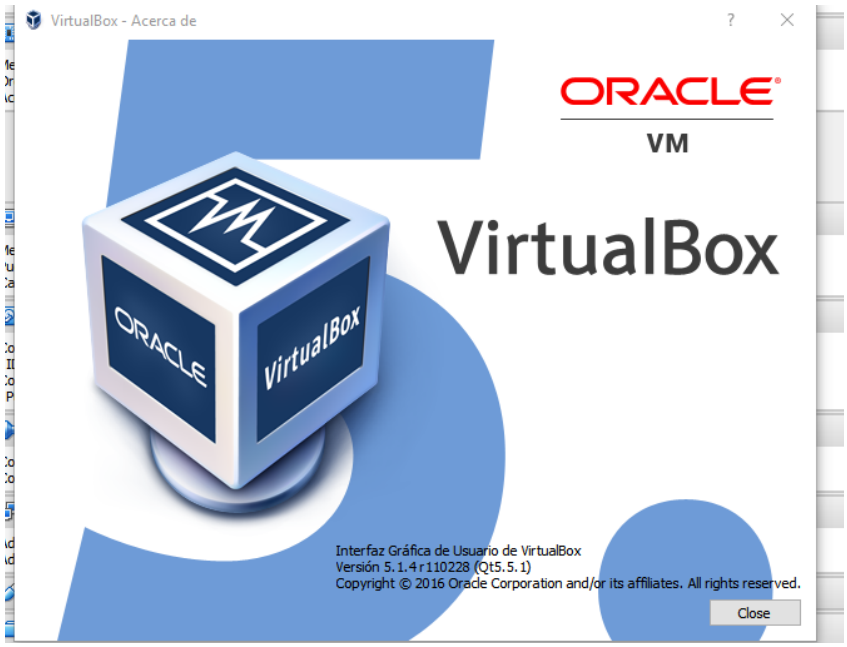


Figura 40: Pantalla de administrador de VirtualBox

Configuración de Red

Para el presente trabajo, la configuración de red será:

- **Tipo**: Red solo anfitrión (Host-Only Networking).
- **Puerta de enlace**: 192.168.99.1
- **Máscara de subred**: 255.255.255.0
- **Servidor DHCP**:

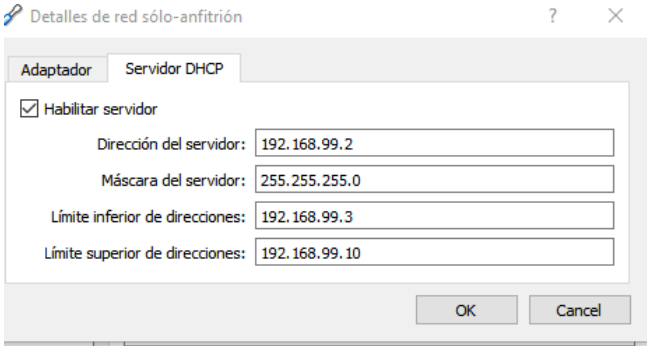


Figura 41: Configuración de red solo-anfitrión. Servidor DHCP.

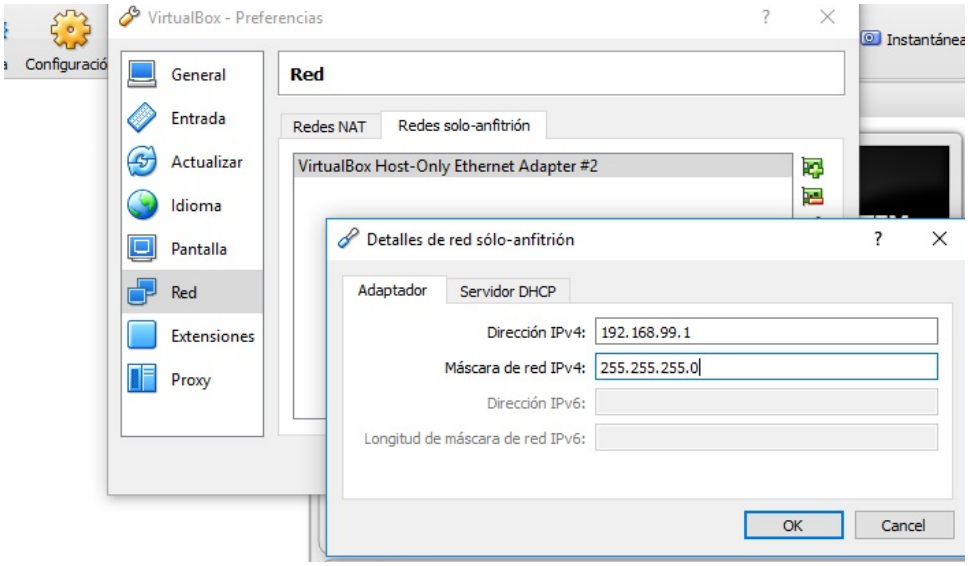


Figura 42: Configuración de red solo-anfitrión. Adaptador.

El tipo de red elegido, Red Solo Anfitrión, híbrido entre los modos “adaptador puente” y “Red Interna” que nos permite crear un entorno en el cual las máquinas virtuales pueden interactuar las unas con las otras y con el host tal como si estuvieran conectadas a través de un switch, al mismo tiempo que limitamos el alcance a dicho host, es decir, que no pueden comunicarse con “el mundo exterior” al host. Cuando este tipo de red es configurado VirtualBox crea una nueva interface software en el host, la cual aparecerá junto con las otras interfaces de red.

4.1.2.1 Windows XP - Víctima

La configuración de la máquina virtual utilizada es la siguiente:

- Disco Duro: VHD de 4GB
- RAM: 512 MB
- Red: Adaptador sólo-anfitrión.

4.1.2.2 Windows7 - Atacante

La configuración de la máquina virtual utilizada es la siguiente:

- Disco Duro: VHD de 20GB
- RAM: 1GB
- Red: Adaptador sólo-anfitrión.

4.1.2.3 Security Onion

A continuación se va a realizar la instalación de la distrubución Security Onion 14.04.4.2 en una máquina virtual. Los **pasos** a llevar a cabo serán los siguientes:

1. Crear la máquina virtual (VM).
   - **Espacio en HD**: 25GB
   - **RAM**: 2GB.
2. "Insertar" la .iso de instalación de la distribución en el menú.
3. **Configuración de red de la máquina virtual**:
   Para esta máquina virtual necesitaremos dos *interfaces*, una de mantenimiento del servidor (eth0) y otra de monitorización (eth1). Por ello tendremos que habilitar dos adaptadores de red para esta máquina, siendo el primero una interfaces normal, y la segunda una interface en *modo promiscuo* para poder captar los paquetes de la red. A mayores, y para poder actualizar la distribución, en un inicio se configurará la primera de ellas como conectada a un adaptador puente, y así tener conexión a Internet.
4. Iniciar la máquina virtual para proceder a la instalación del sistema operativo.
5. **Instalar la distribución Security Onion**. Debido a que la instalación *no finaliza correctamente al seleccionar como idioma de instalación el Español*, hay que seleccionar como idioma de instalación el Inglés.

6. Actualización del Sistema: Abriremos la consola de comandos e introduciremos y ejecutaremos lo siguiente: `sudo soup`
7. Instalación de las GuestAdditions:
    (1) Montar la iso.
    (2) Abrir consola de comandos.
    (3) `cd /media/luisbc/VBOXADDITIONS_5.1.2_108956`
    (4) `sudo ./VBoxLinuxAdditions.run`
    (5) reiniciar.
8. **Configuración de las interfaces**
    (1) Pulsamos en el icono "Setup" del escritorio.
    (2) eth0 -> management. → static → 192.168.99.18
    (3) eth1 -> sniffing
    (4) local domain name → tfmUNIR.com
    (5) Reiniciar.

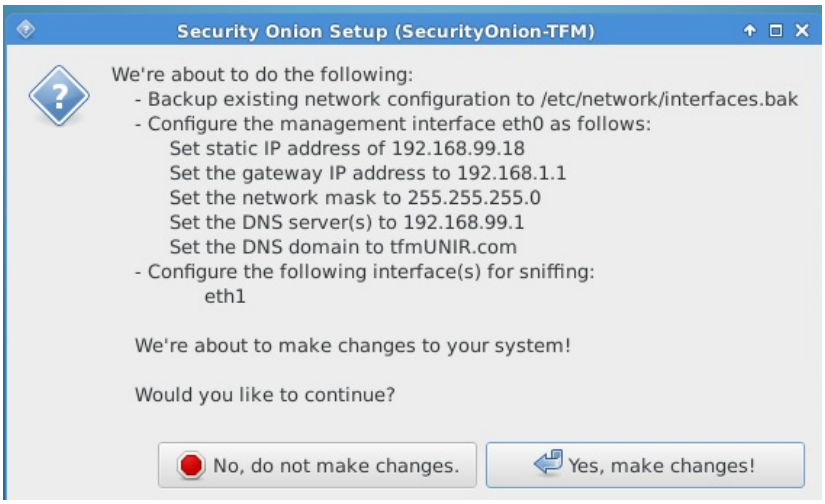


Figura 43: Ventana de confirmación de configuración de las interfaces.

9. Configuración del **modo de funcionamiento**
    (1) Seleccionamos *Producction Mode* (ya que es el modo real/profesional).
    (2) Seleccionamos *Standalone* (funciona como servidor y sensor a la vez).
    (3) Seleccionamos *BestPractices.*
10. **Usuario y Contraseña** para *Sguil*, *Squert* y *ESLA*.
11. Seleccionamos **NIDS** por defecto: *Snort*.
12. **IDS ruleset**: Emerging Threats Open.

13. **PF_RING min_num_slots** -> por defecto, 4096.
14. Seleccionamos la **interface de red que queremos monitorizar**: eth1.
15. **HOME_NET**: Dejar por defecto, se refiere a redes tipo A, B y C.

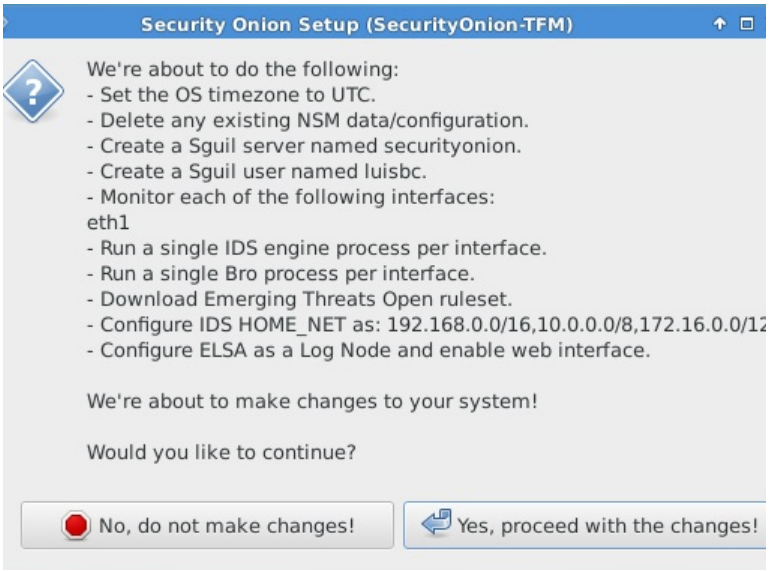


Figura 44:Ventana de confirmación de configuración.

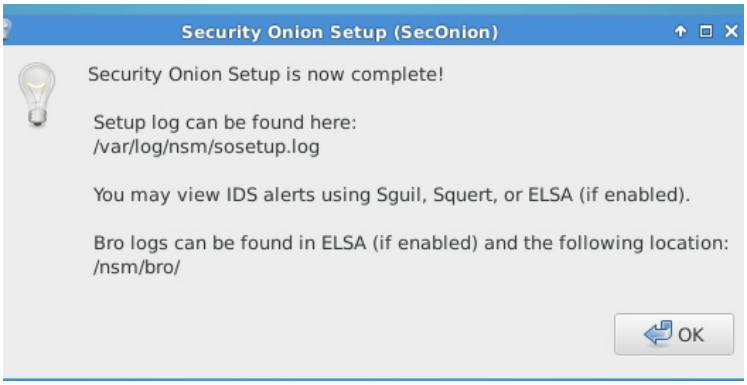


Figura 45: Finalización de la configuración de Security Onion.

16. **Configuración Horaria**:
    - Por defecto SecurityOnion y sus diversos componentes están configurados en zona horaria UTC/GMT, (https://github.com/Security-Onion-Solutions/security-onion/wiki/TimeZones).
    - Squert
        I. Click en "time interval" (INTERVAL). Click en flechas esquina superior derecha tipo "forward".
        II. Deseleccionamos "UTC". Establecemos nuestra zona horaria (TZ OFFSET).

III. Guardamos pulsando sobre el botón "save TZ".

- ELSA: Por defecto coge la zona horaria del navegador, por lo que aparece de manera correcta.

17. **Ubicación de las Reglas**:
    - Descargadas por Pulledpork:
      `/etc/nsm/rules/downloaded.rules`
    - Reglas locales:
      `/etc/nsm/rules/local.rules`
    - Para que PulledPork modifique las reglas descargadas:
      `/etc/nsm/pulledpork/`
    - Las reglas se actualizan cada mañana, y pueden ser descargadas de:
      `sudo rule-update`
    - Los sensores se pueden modificar a través de los ficheros almacenados en:
      `/etc/nsm/NAME-OF-SENSOR/`

18. **Archivos de Configuración**:
    `/var/log/nsm/sosetup.log`
19. **Revisar si los servicios están funcionando correctamente**:
    (1) Miramos si están los *servicios* funcionando correctamente:
    `sudo service nsm status`
    (2) Si no estuvieran los servicios activos, los iniciamos:
    `sudo service nsm start`
    (3) Si es necesario conectar OSSEC agents, dispositivos syslog, o VMs analistas, tendremos que, debido a que el Firewall por defecto en el Setup inicial solo tiene abierto el puerto 22, utilizar la utilidad "so-allow".

4.2. Desarrollo de la prueba APT

A continuación se procederá a iniciar las tres máquinas virtuales para proceder con el inicio del piloto, siendo las distintas etapas de que consta las siguientes:

1. Simulación de entorno de trabajo normal.
2. Análisis del ataque a realizar y el malware a utilizar.
3. Simulación de entorno de trabajo con la máquina víctima infectada.
4. Evaluación de la prueba.
5. Propuesta de mejoras.

4.2.1 Simulación de entorno de trabajo normal

Esta etapa de la prueba consistirá en simular un entorno normal de funcionamiento de la máquina víctima (Windows XP) para analizar la actividad de la red y poder establecer las líneas base contra las que comparar posteriormente el análisis del sistema infectado.

Por otro lado comentar que el periodo de captura de datos será el suficiente para poder captar el tráfico estándar.

A continuación se muestran capturas de pantalla de Sguil y Squert posteriores a la instalación y configuración de Security Onion, estando las otras dos máquinas virtuales también activas aunque sin proceso de ataque aun.:

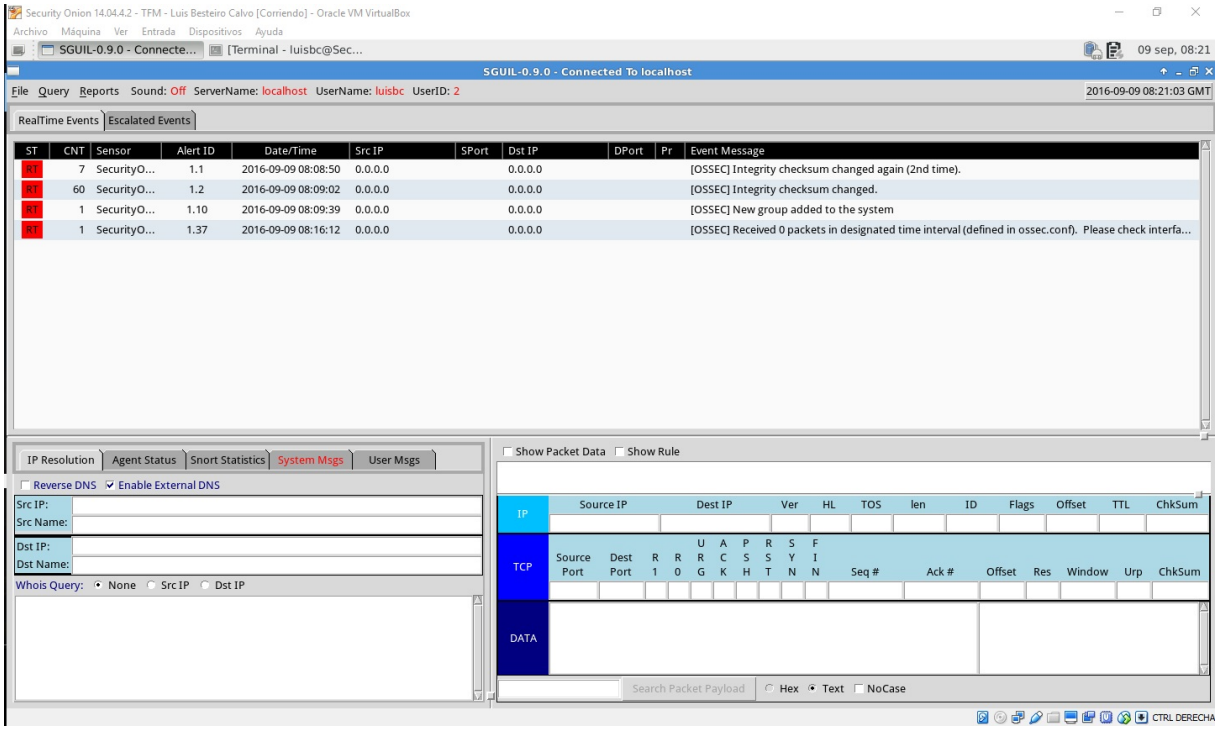


Figura 46: Monitorización mediante Sguil de entorno no comprometido.

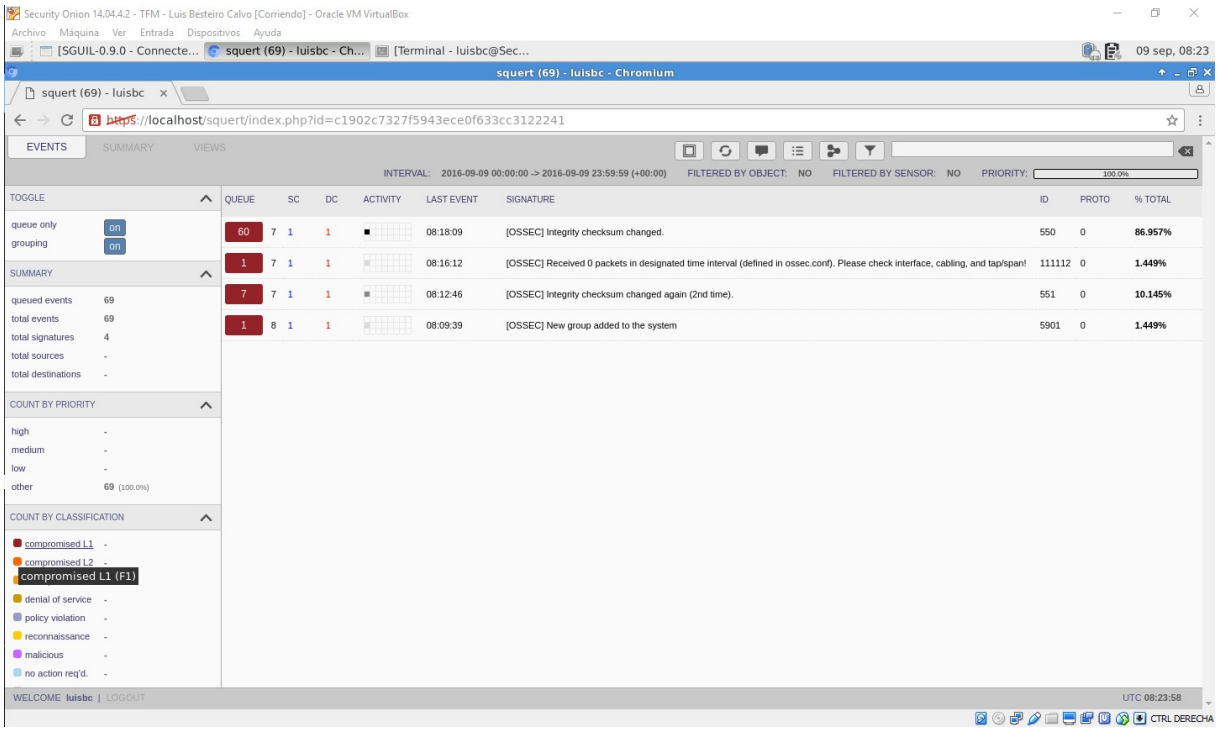


Figura 47: Monitorización mediante Squert de entorno no comprometido.

4.2.2 Análisis del ataque a realizar

El tipo de ataque a realizar será el de infectar la máquina objetivo (siendo en este piloto una máquina virtual con Windows XP) con un archivo generado con PoisonIVY conocido como "servidor". Ya que el método de infección queda fuera de los objetivos de este TFM, dicha infección se llevará a cabo generando el servidor en la máquina víctima, en la cual también dispondremos de la aplicación PoisonIVY, para facilitar el proceso.

Una vez "infectada" la víctima, se llevará a cabo una sesión remota desde el cliente (atacante), mediante la cual llevaremos a cabo lo siguiente:

1. Búsqueda de un archivo pdf.
2. Extracción de dicho archivo.
3. Análisis con el keylogger.

Una vez realizada la extracción llevaremos a cabo el borrado del proceso de infección para finalizar la comunicación del servidor con el cliente.

4.2.3 Ataque

Para llevar a cabo el ataque con PoisonIVY seguiremos los siguientes pasos:

CONFIGURACIÓN

1. **Establecimiento de los parámetros base**:
   En primer lugar es necesario establecer ciertos parámetros que nos servirán tanto para la configuración del servidor como del cliente, y que tienen que ser los mismos para ambos:
   - Dirección IP del C&C (cliente), o dominio: Si bien podemos introducir una IP, para un caso real de ataque sería mejor utilizar por ejemplo un dominio asociado a una IP dinámica. Un ejemplo sería crear un dominio en el servicio online "NO-IP" (http://www.noip.com).
   - Puerto de escucha del C&C: El que trae por defecto la aplicación, 3460, es válido, pero no obstante, ya que es que que trae por defecto conllevará que sea más sencillo el ser detectados.
   - Contraseña: Esta será la contraseña a introducir en la configuración tanto en cliente como en servidor para evitar que sesiones ajenas o no deseadas puedan conectarse a nuestro C&C.

2. **Creación del servidor**: Haciendo doble-click sobre el ejecutable de PoisonIVY accedemos al framework.

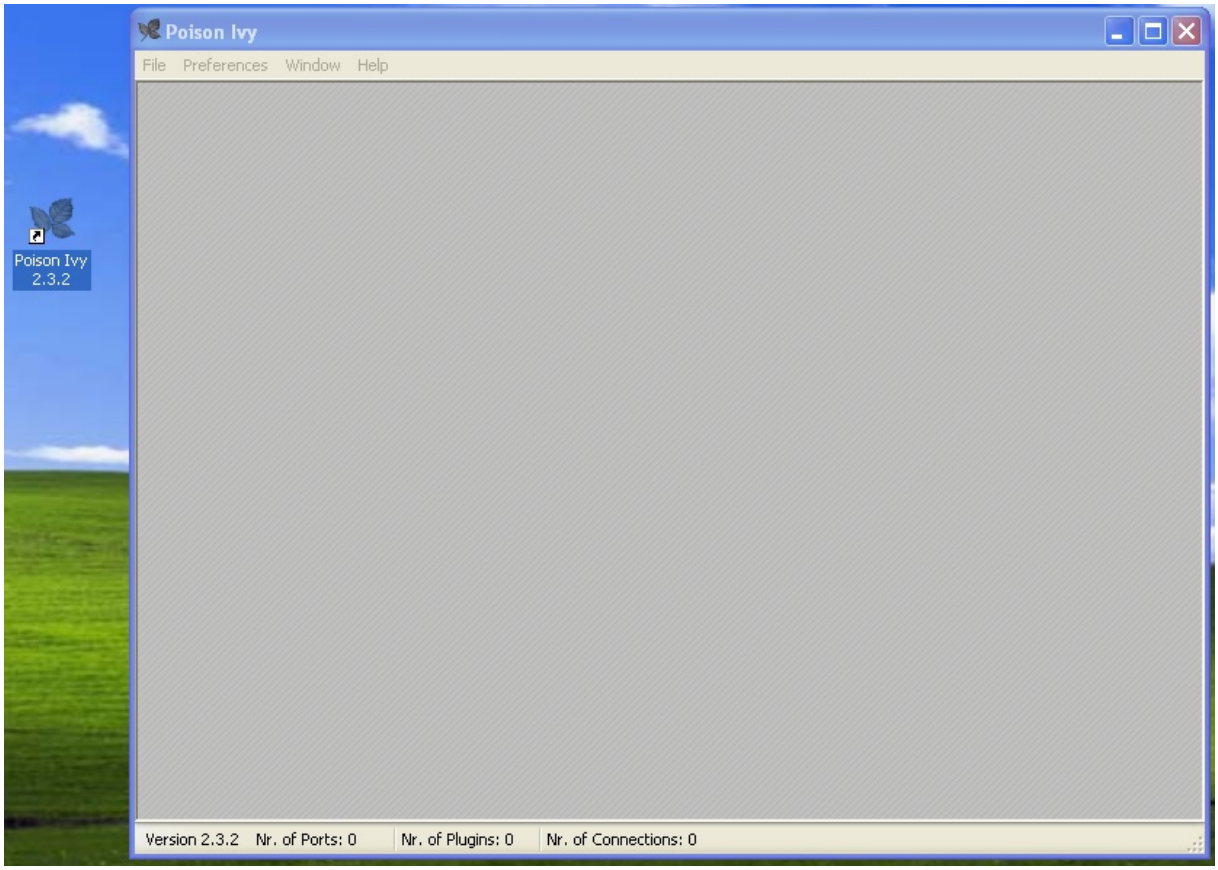


Figura 48: Ventana de inicio de PoisonIvy.

(1) A continuación creamos el Servidor en la máquina víctima: *File → New Server*.

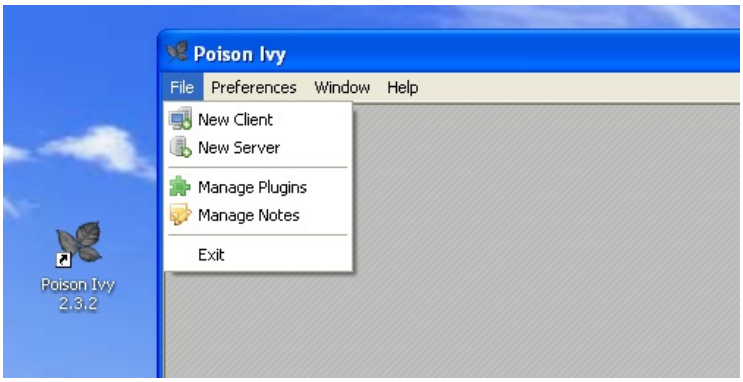


Figura 49: Menú de fichero.

(2) **Creamos un perfil**:

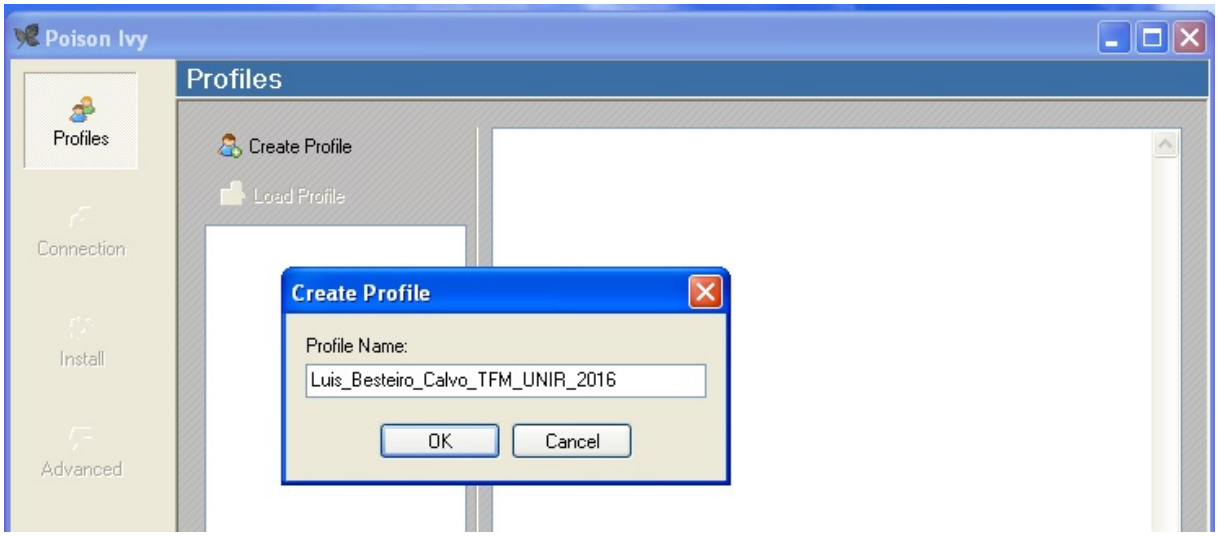


Figura 50: Creación del perfil en PoisonIvy.

(3) **Ajustes de configuración**:

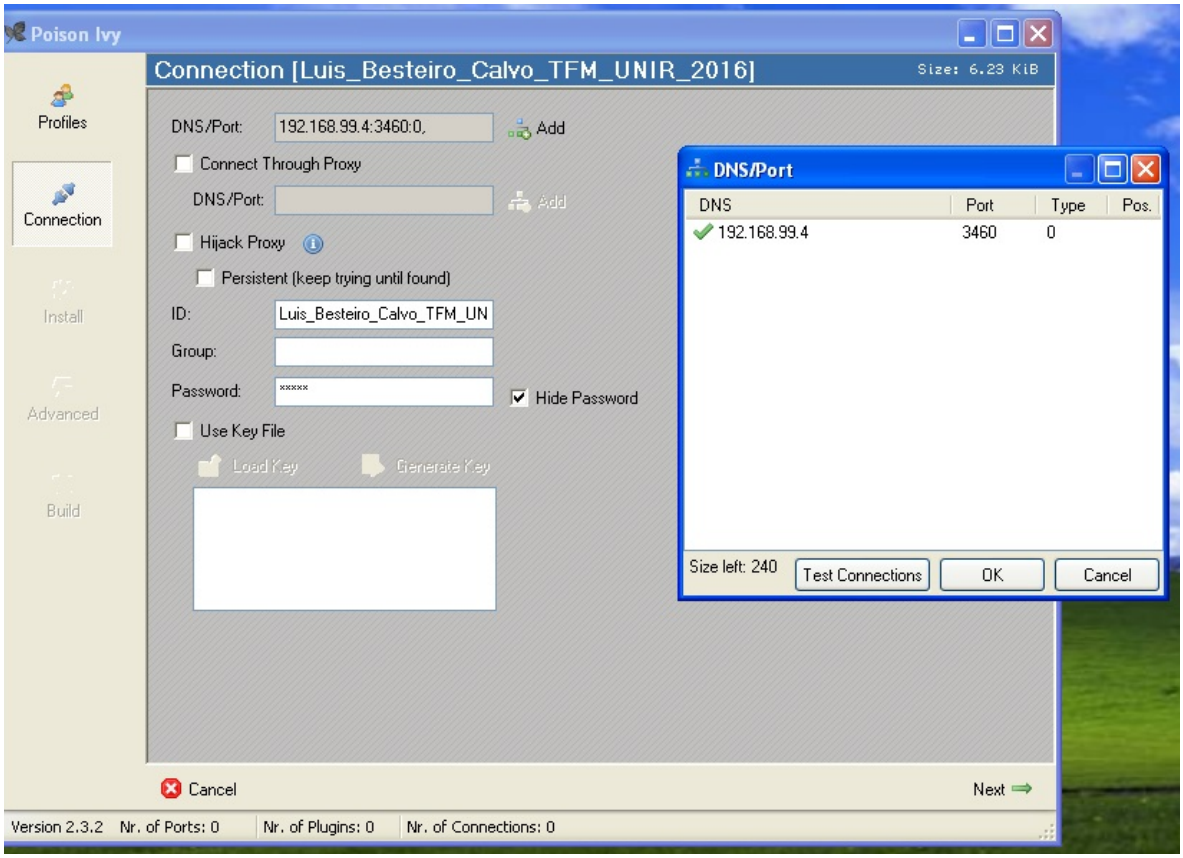


Figura 51: Ajustes de conexión del servidor PoisonIvy.

Después de introducir los campos, pulsamos en “Next” en la esquina inferior derecha y pasamos a configurar la instalación.

(4) **Ajustes de instalación**: en esta ventana procederemos a introducir los ajustes que decidirán cómo se comportará el instalador:

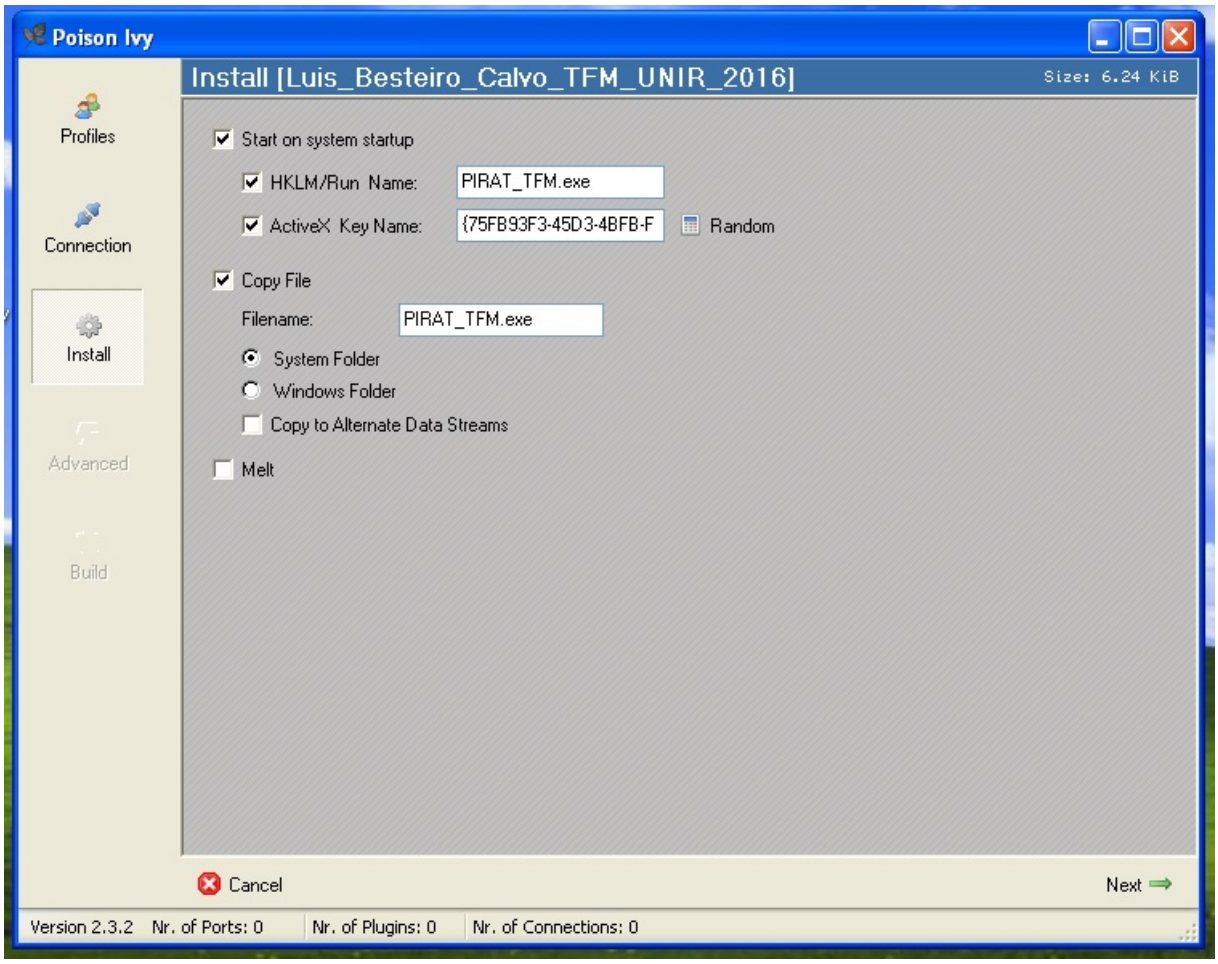


Figura 52: Ajustes de instalación del servidor PoisonIvy.

(5) **Ajustes avanzados**:

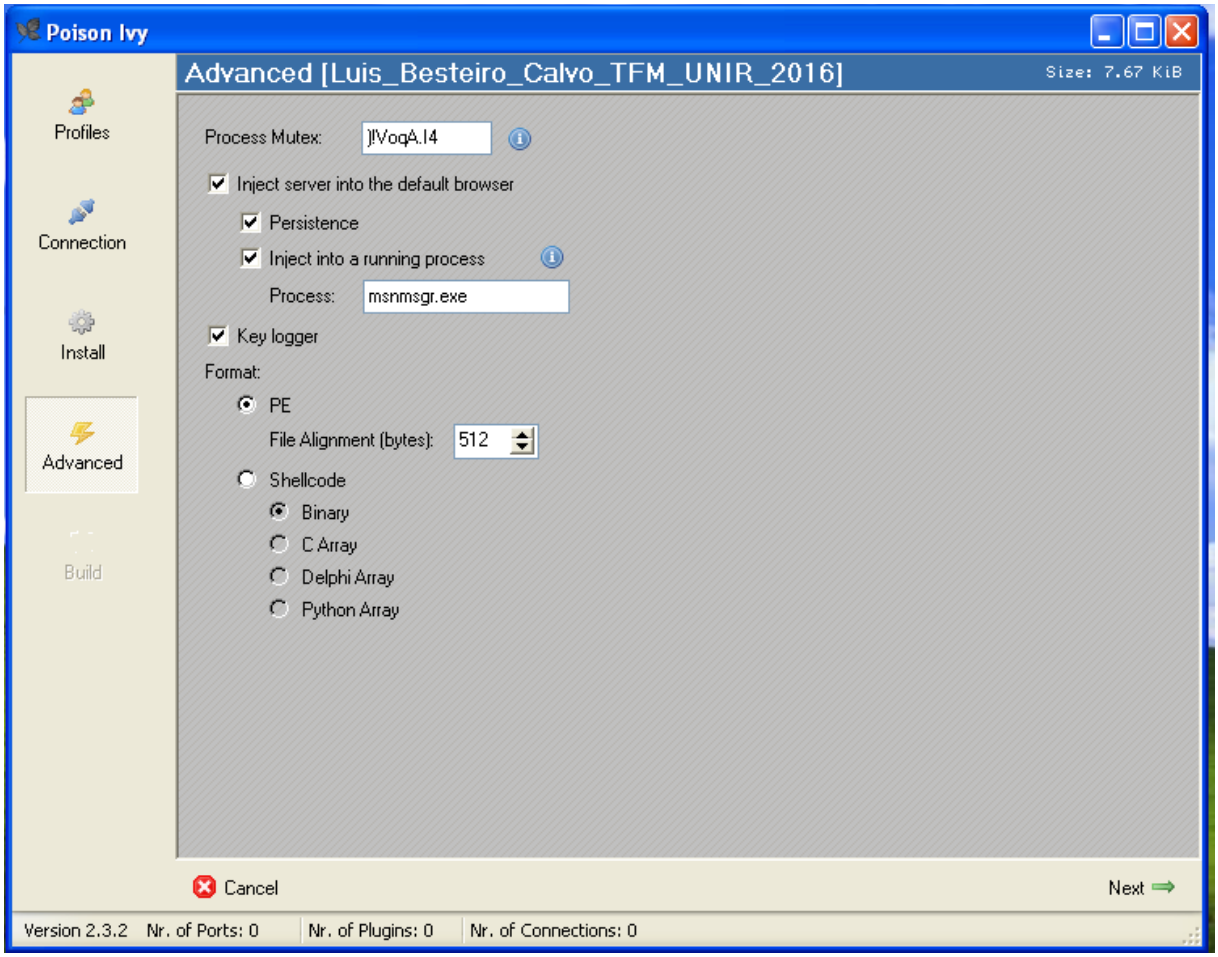


Figura 53: Ajustes avanzados del servidor PoisonIvy.

(6) **Compilación del servidor**:

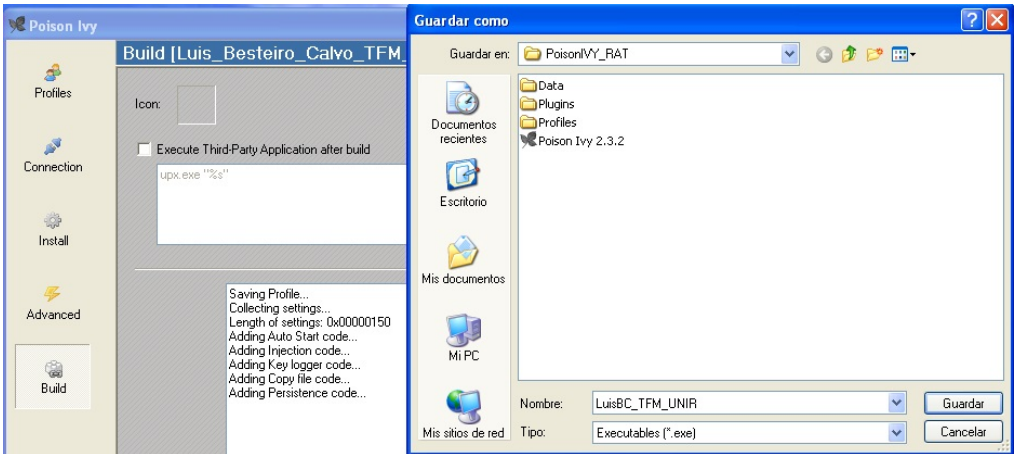


Figura 54: Compilación del servidor.

3. **Creación del cliente**:

   (1) Para crear el cliente abriremos en la máquina atacante (Windows 7) el archivo PoisonIvy2.3.2 y seleccionamos en la ventana el menú:
   File → New Client.

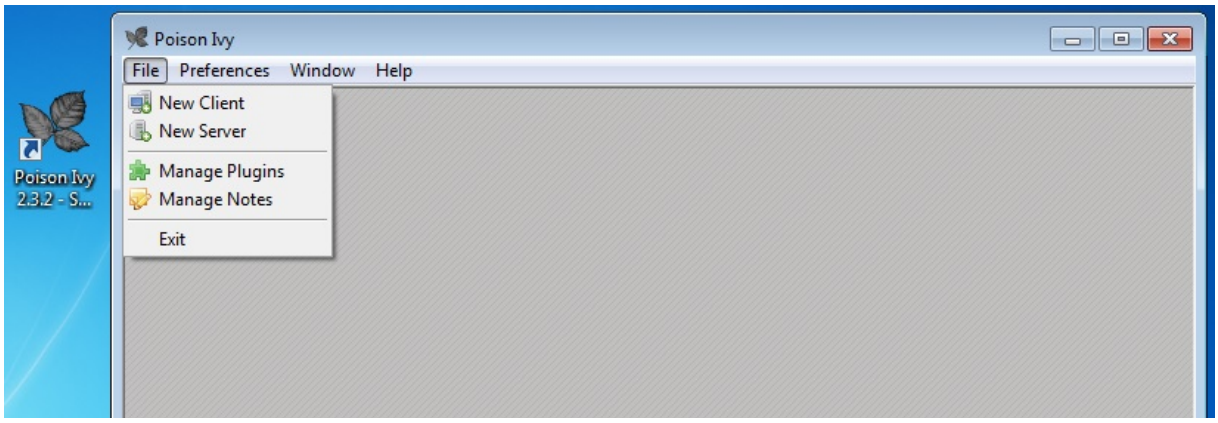


Figura 55: Creación de un nuevo cliente PoisonIvy.

   (2) A continuación **configuramos el cliente** con los datos decididos en el primer apartado de "*Establecimiento de los parámetros base*" y pulsamos sobre el botón "Start" para comenzar a escuchar a la espera de que el servidor se ponga en contacto con su C&C.

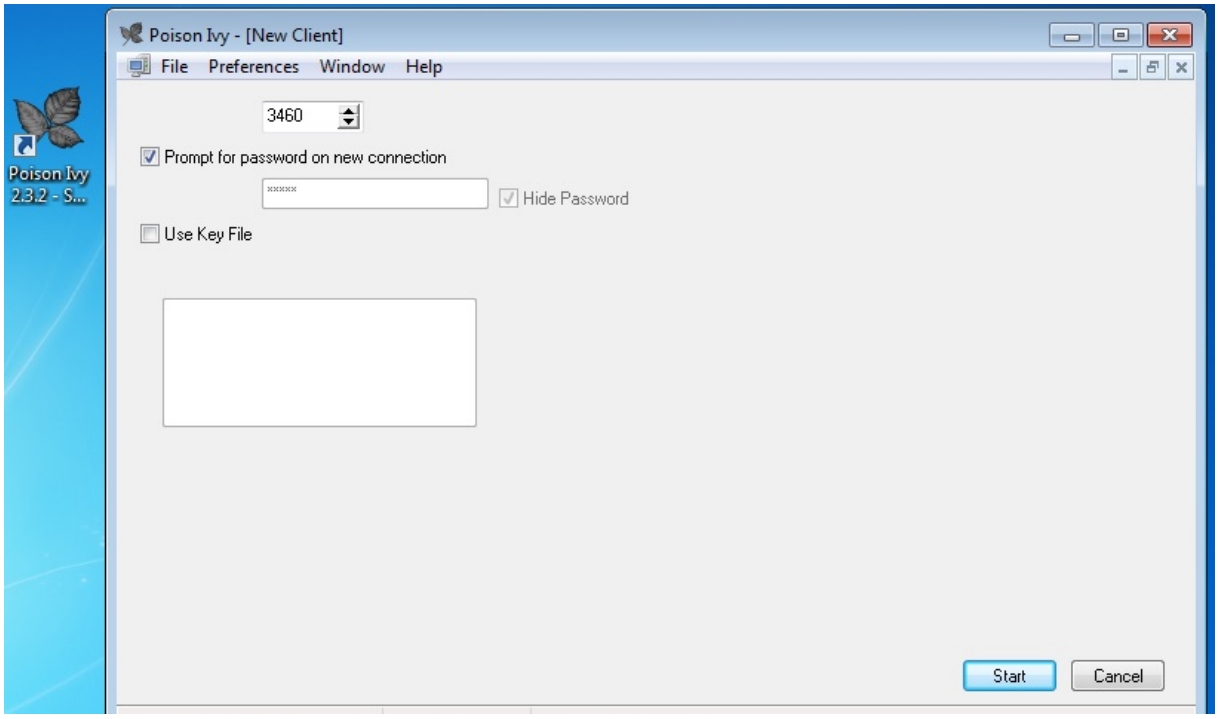


Figura 56: Configuración del cliente PoisonIvy.

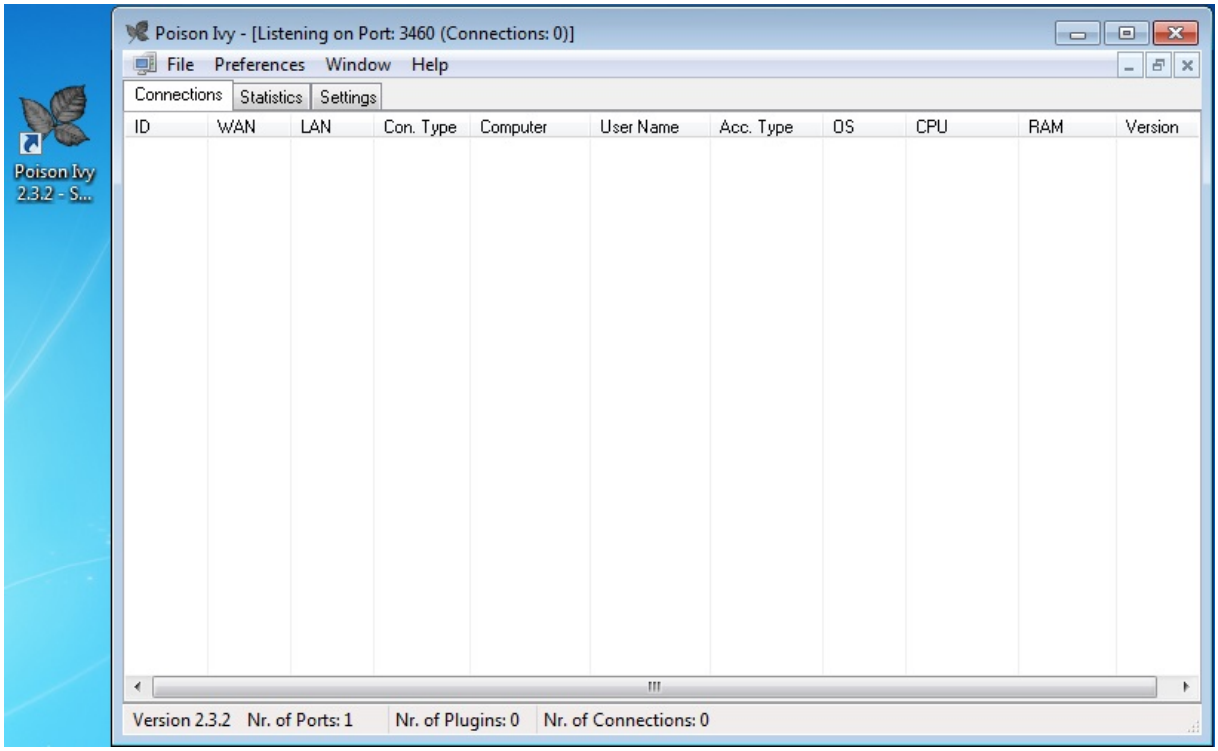


Figura 57: Consola del cliente PoisonIvy.

INFECCIÓN

1. Abrimos el archivo “servidor” generado en la máquina víctima, siendo dicho archivo introducido en un ataque APT vía email, USB o de cualquier otra manera que pueda introducir dicho fichero en la máquina objetivo.

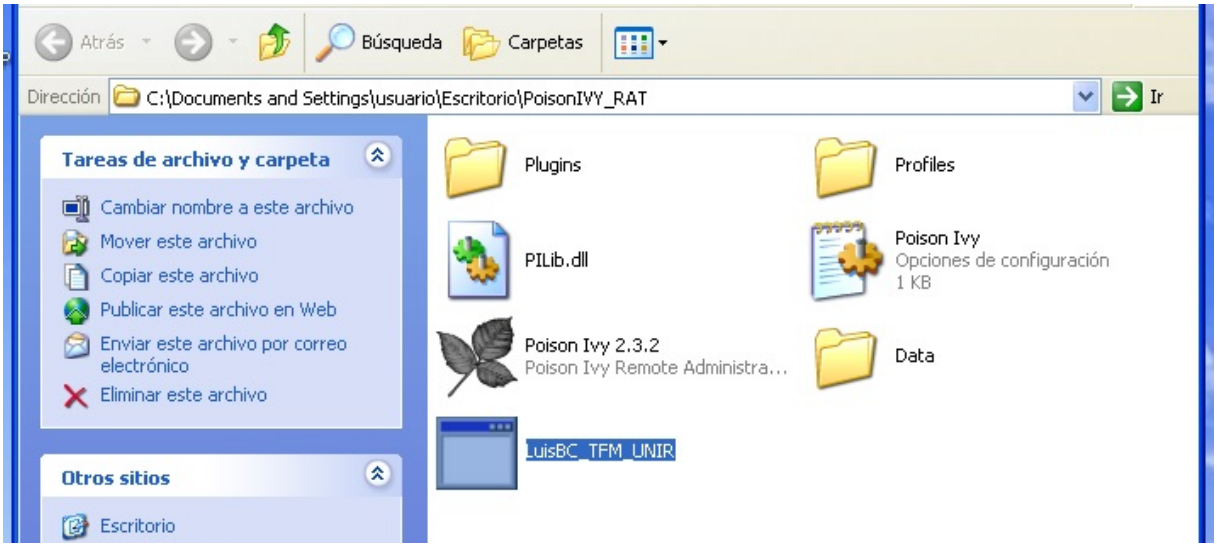


Figura 58: Archivo ejecutable del servidor PoisonIvy.

2. Esperamos a que aparezca en el cliente la sesión para dicho servidor.
3. Pulsamos sobre el icono de la sesión para comenzar la infiltración en el sistema víctima.

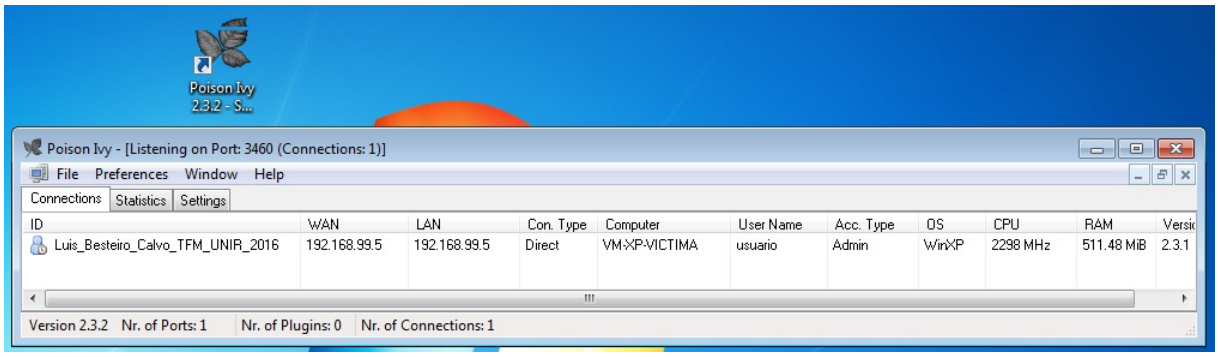


Figura 59: Inicio del control remoto.

4.2.4 Análisis de entorno de trabajo con la máquina víctima infectada

Mediante el software de monitorización y análisis de red llevamos a cabo la captura de los datos de tráfico en la red, para ver así qué es lo que observamos en las distintas etapas de la infección:

1. Monitorización de la la red tras realizarse la infección.

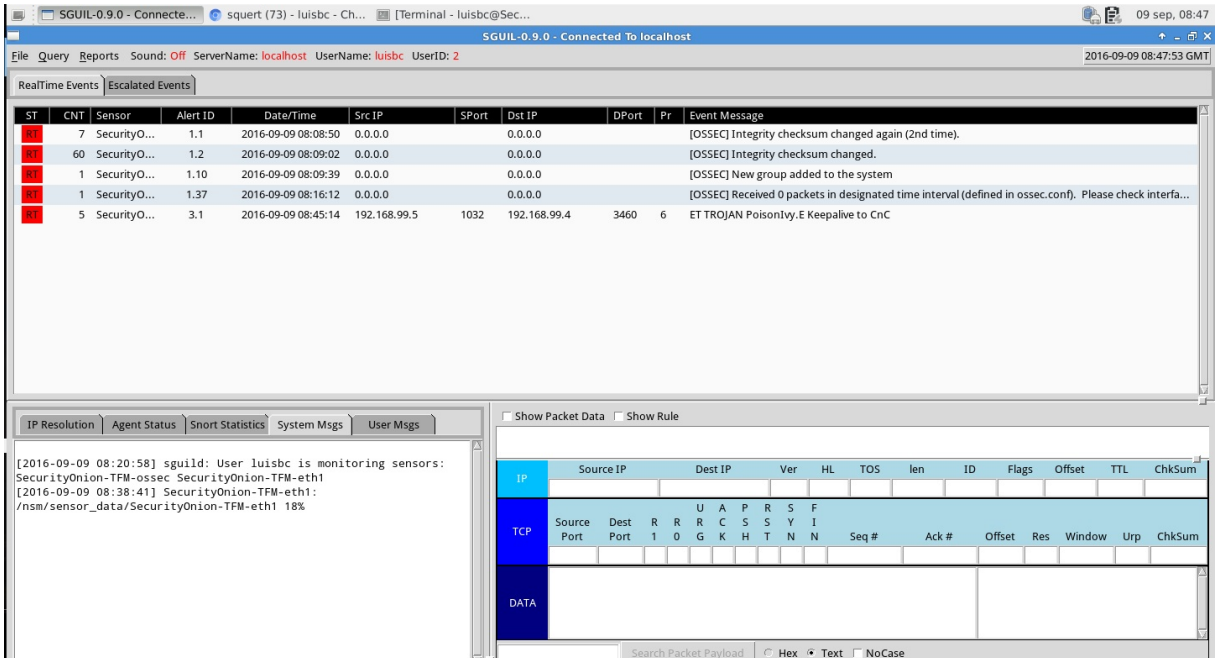


Figura 60: Captura con Sguil de conexión del servidor PoisonIvy con el C&C.

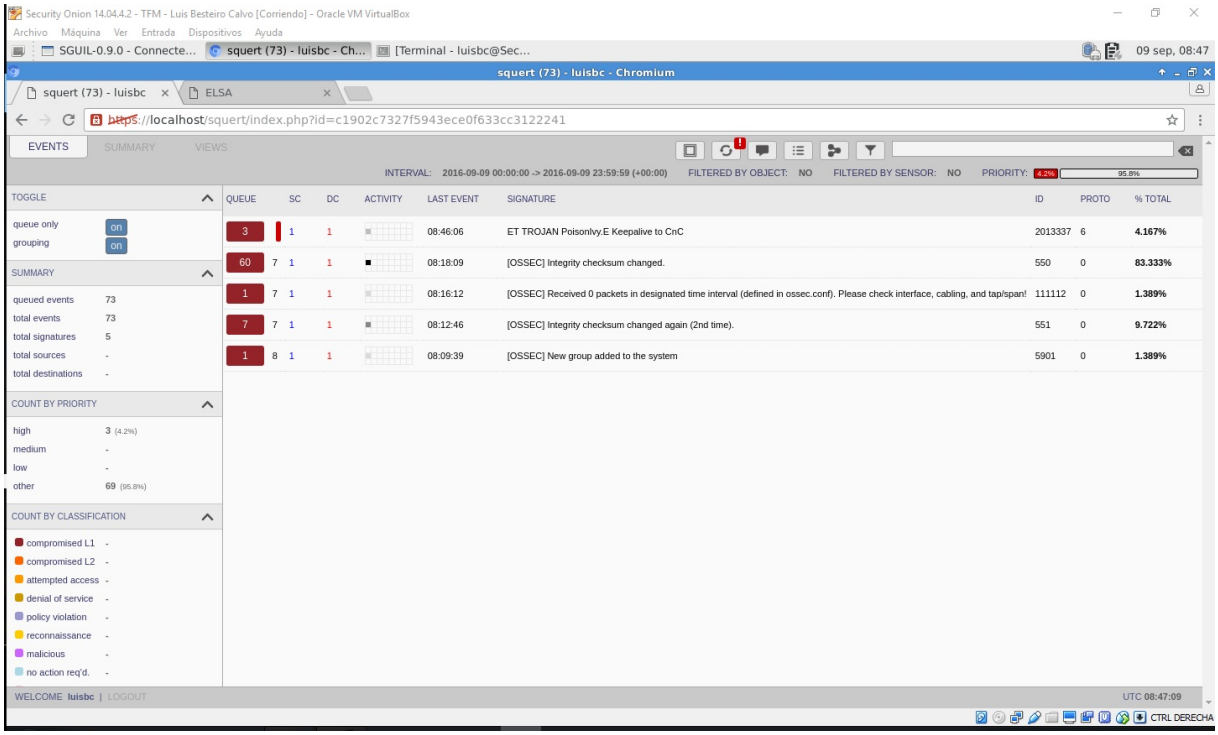


Figura 61: Captura con Squert de conexión del servidor PoisonIvy con el C&C.

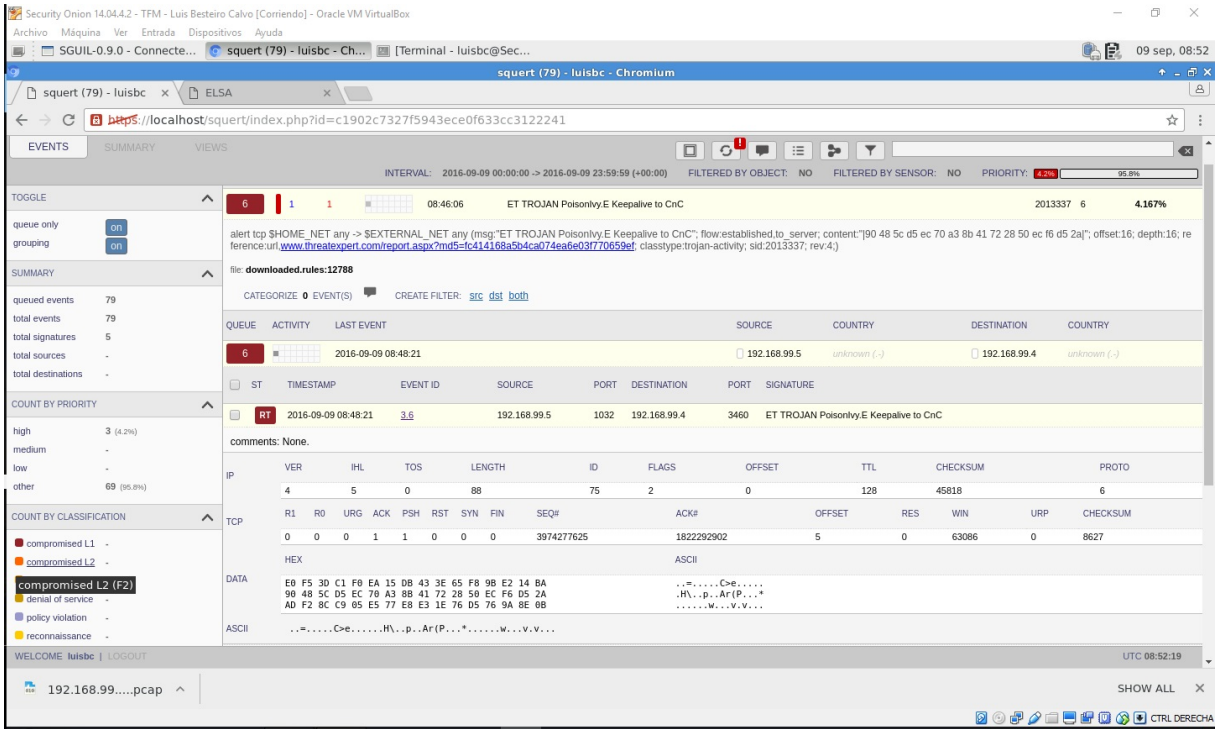


Figura 62: Detalles con Squert de conexión del servidor PoisonIvy con el C&C.

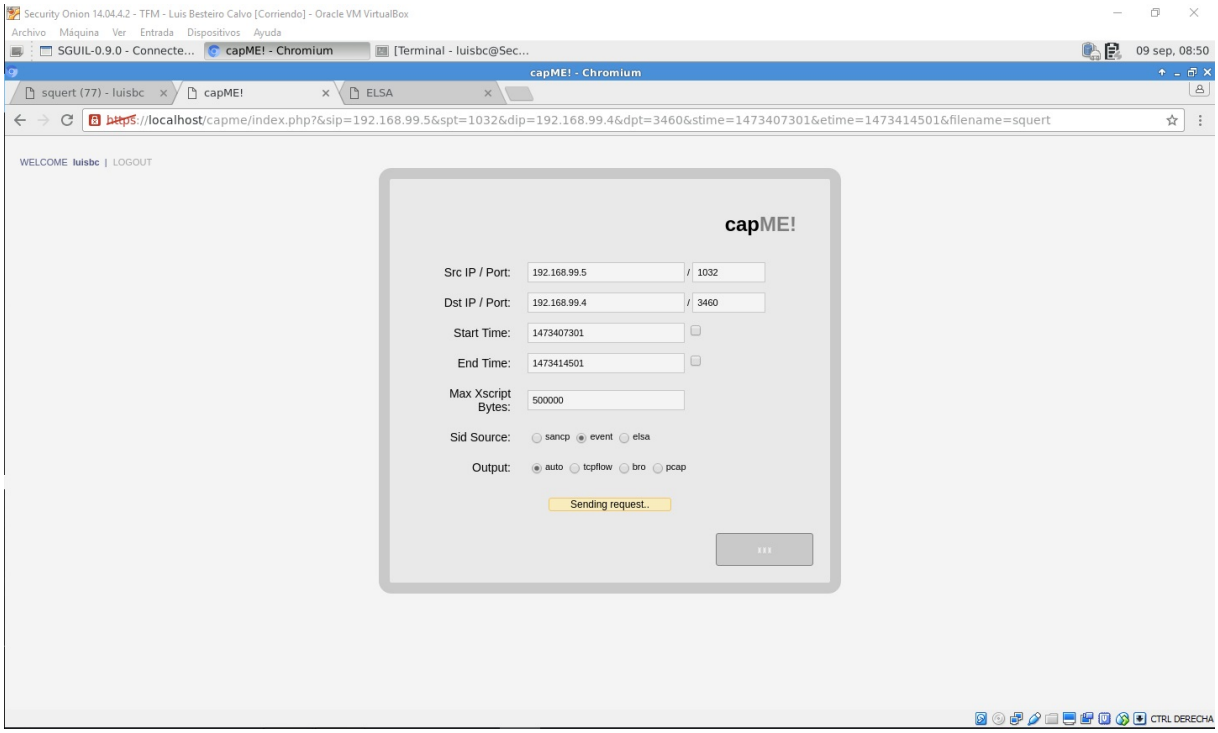


Figura 63: Procesado con capME! de conexión del servidor PoisonIvy con el C&C.

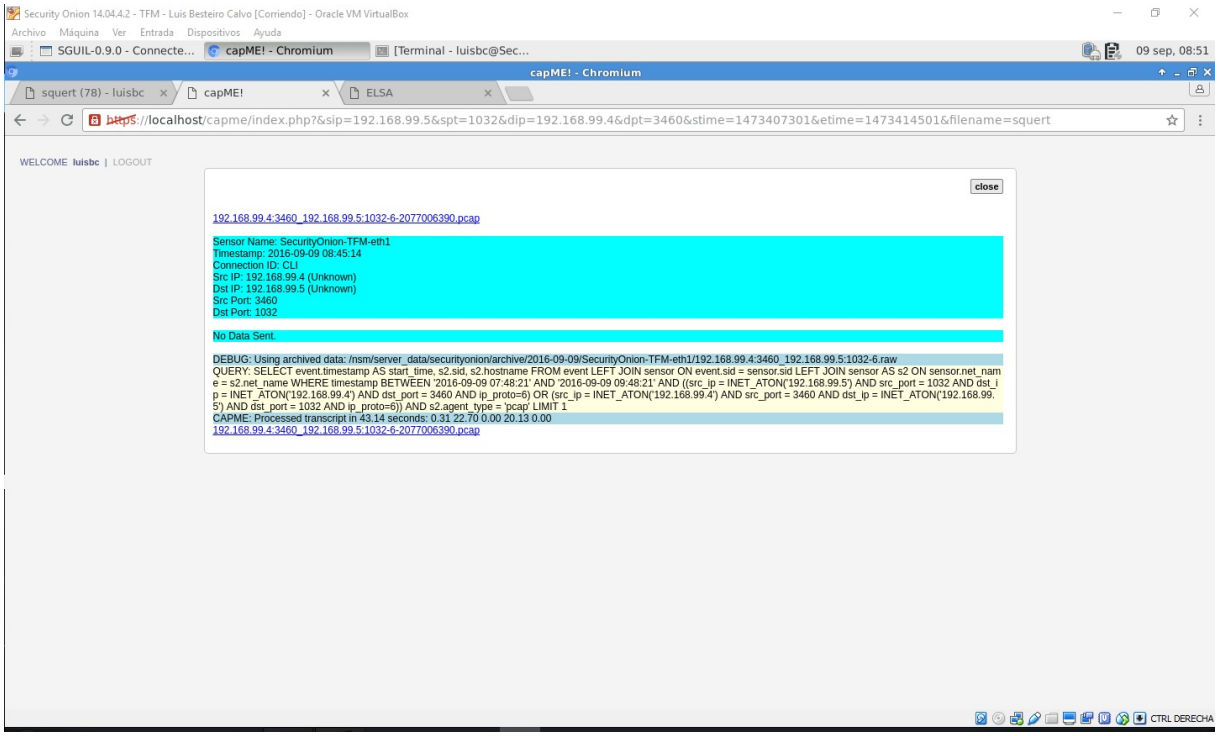


Figura 64: Detalles con capME! de conexión del servidor PoisonIvy con el C&C..

2. Llevar a cabo la infiltración desde la máquina atacante

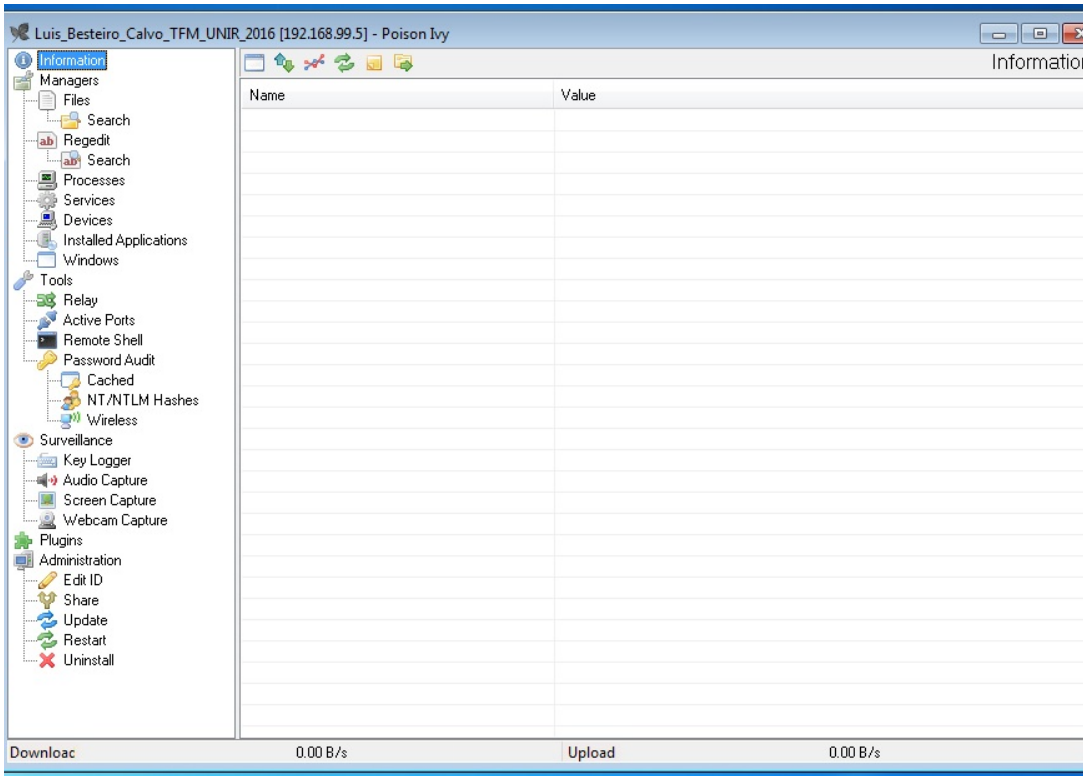


Figura 65: Panel de Control del cliente de PoisonIvy.

3. Buscar un fichero.

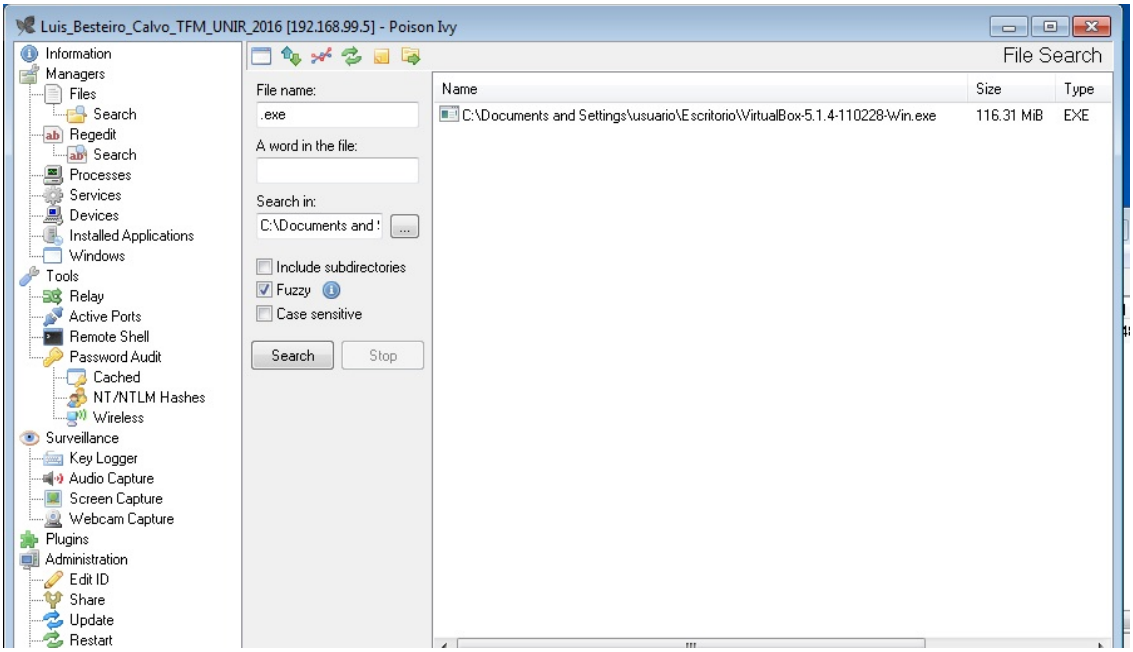


Figura 66: Búsqueda de archivo con el Panel de Control del cliente de PoisonIvy.

4. Exfiltración de un fichero de elevado tamaño

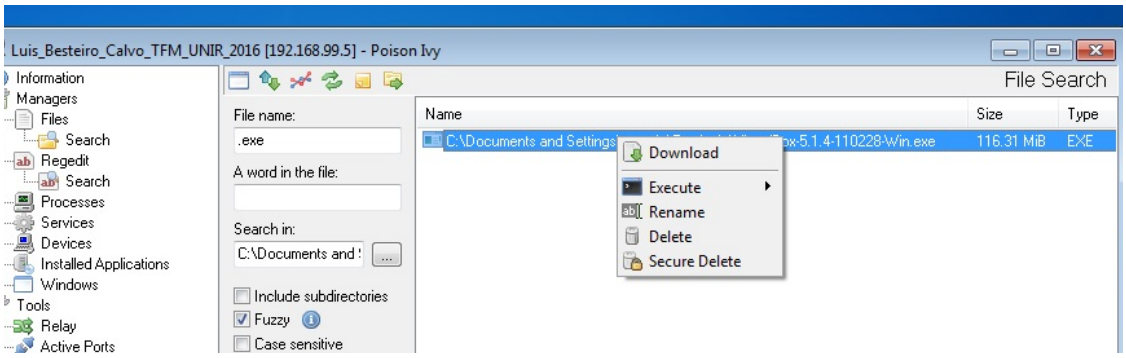


Figura 67: Exfiltración de archivo con el Panel de Control del cliente de PoisonIvy.

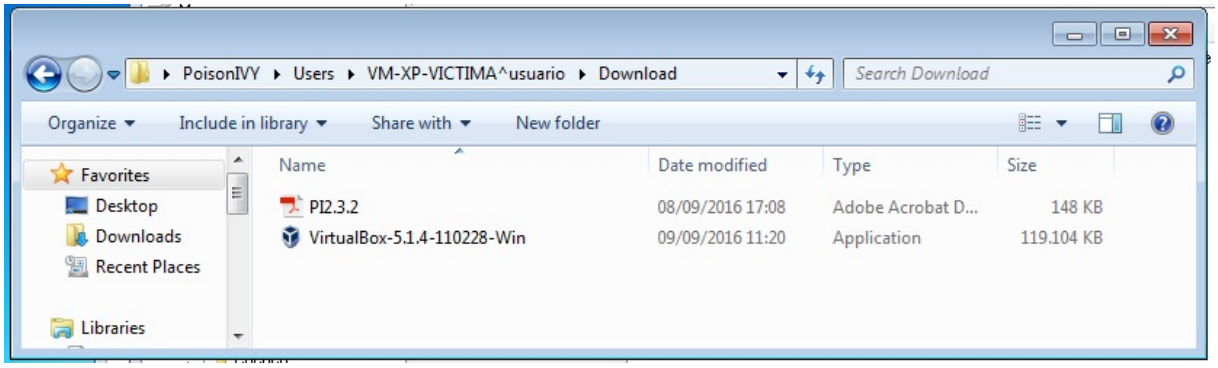


Figura 68: Archivo exfiltrado con PoisonIvy.

5. Hacer captación de pulsación de teclas. Keylogger.

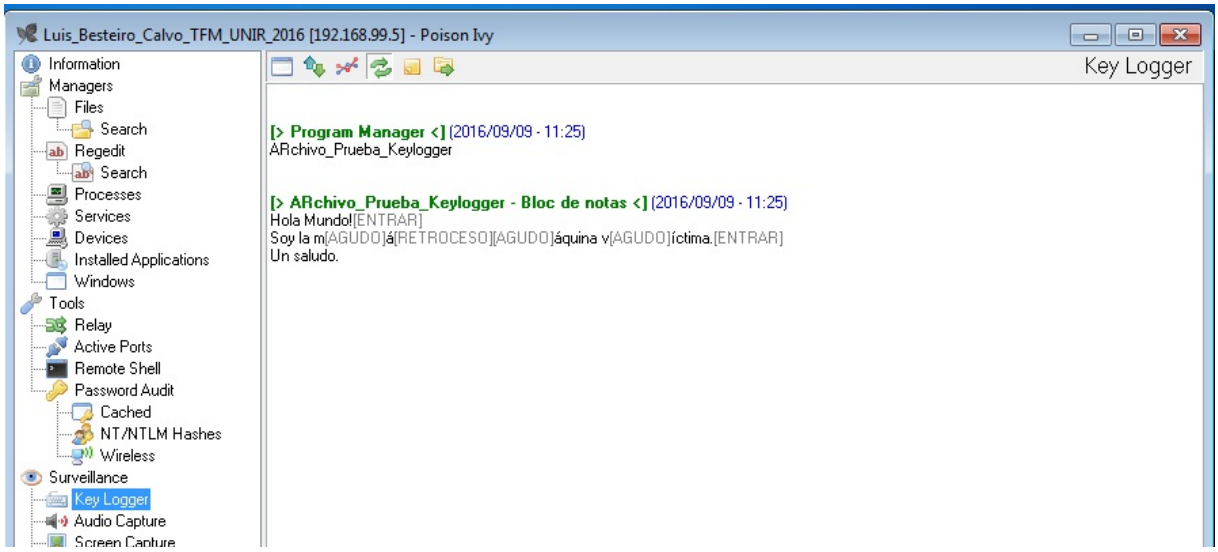


Figura 69: Captación de pulsaciones de teclado con PoisonIvy.

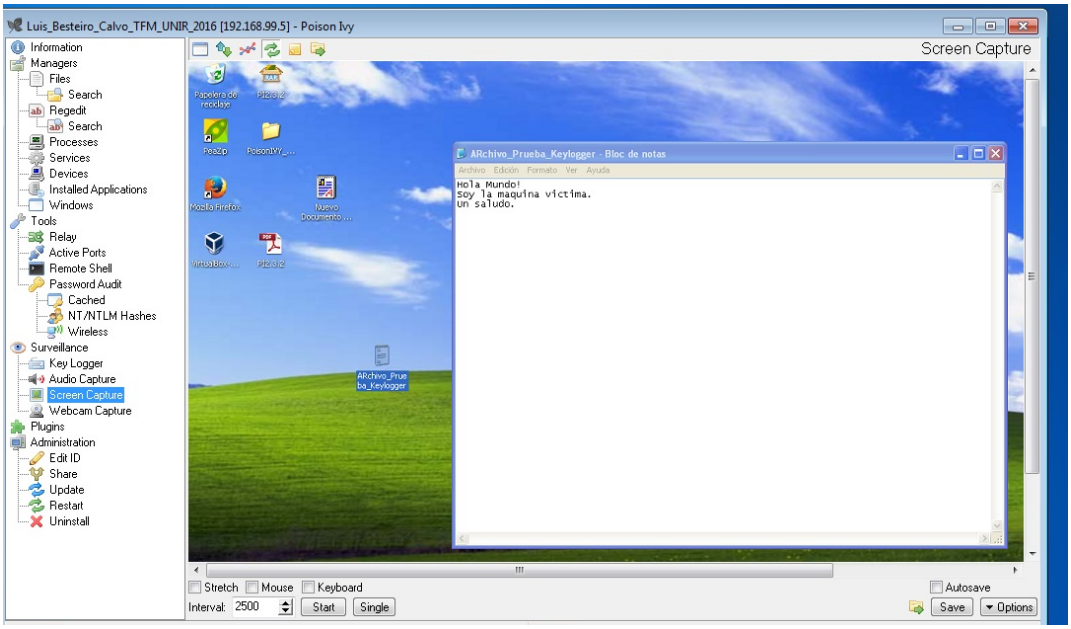


Figura 70: Captura de pantalla con PoisonIvy.

5. Descargar el fichero de texto.

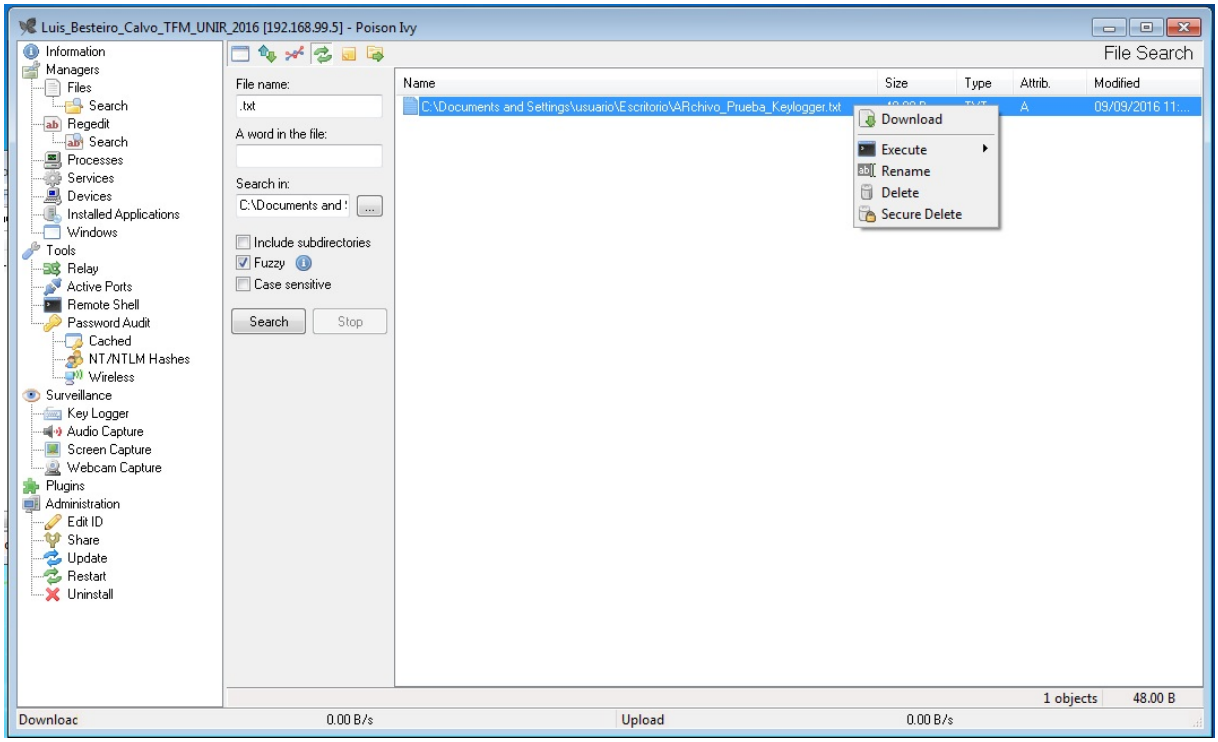


Figura 71: Descarga del fichero de texto creado por la víctima.

6. Descargar el fichero de texto comprimido.

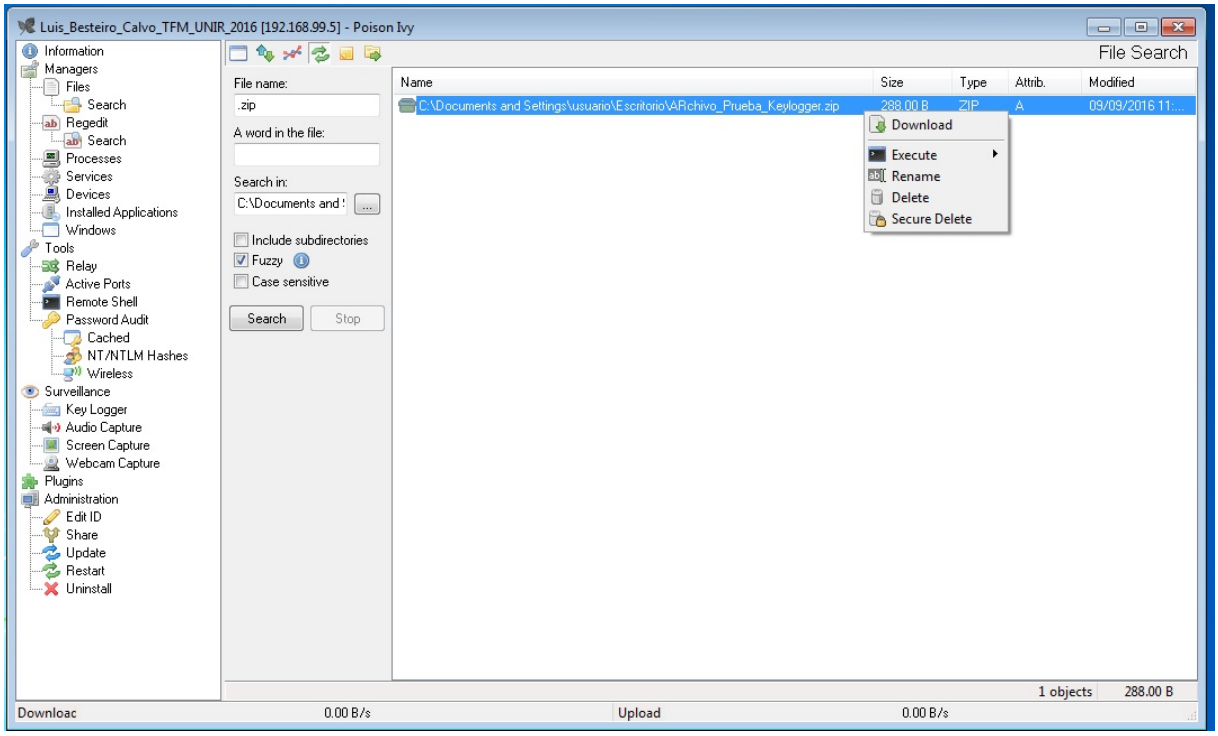


Figura 72:Descarga del fichero de texto comprimido en .zip con PoisonIvy.

6. Eliminamos el servidor.

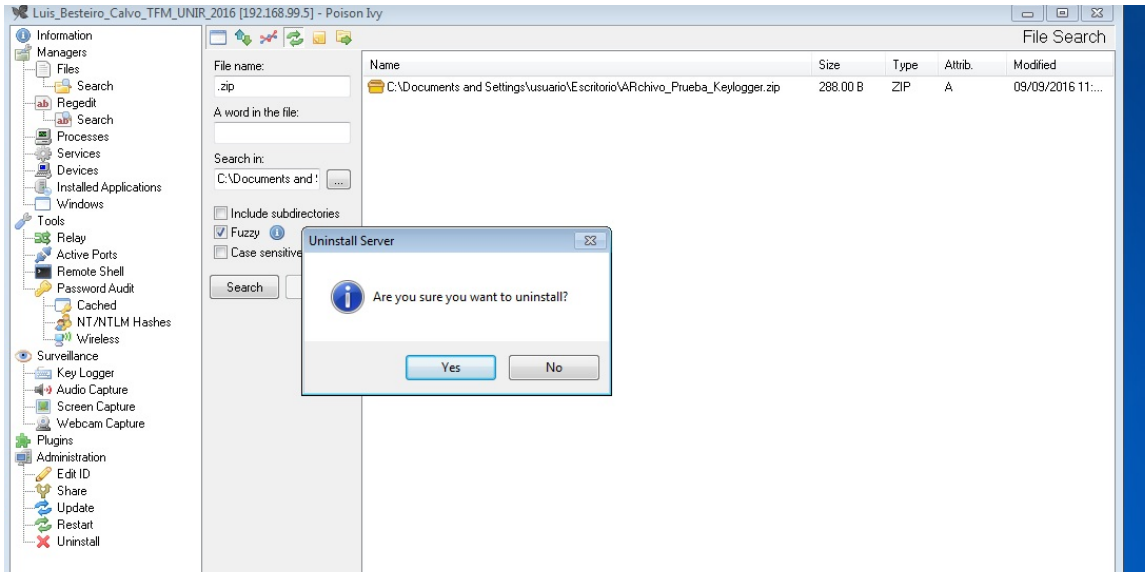


Figura 73: Desinstalación servidor víctima con panel de control de PoisonIvy.

4.2.5 Evaluación de la prueba

SGUIL

A continuación se procede a mostrar la información obtenida a través de Sguil:

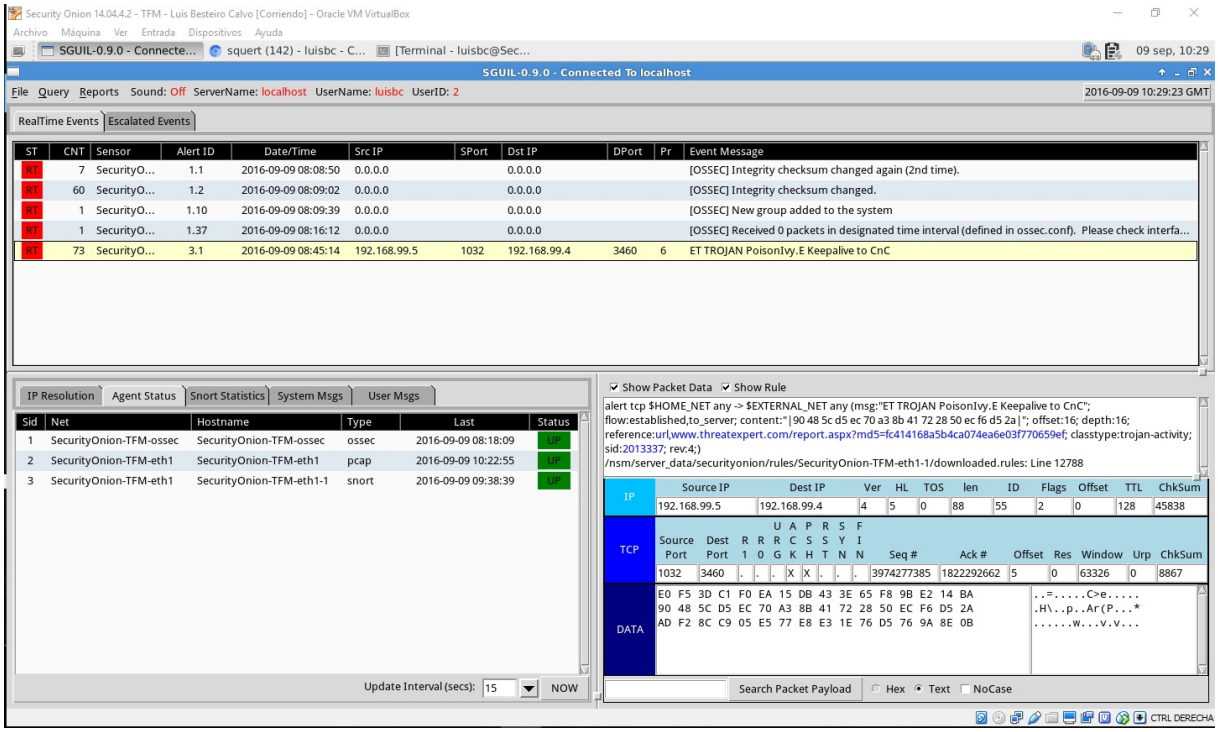


Figura 74: Resultados del análisis en Sguil.

Para cada una de las alertas, podemos ver los eventos relacionados:

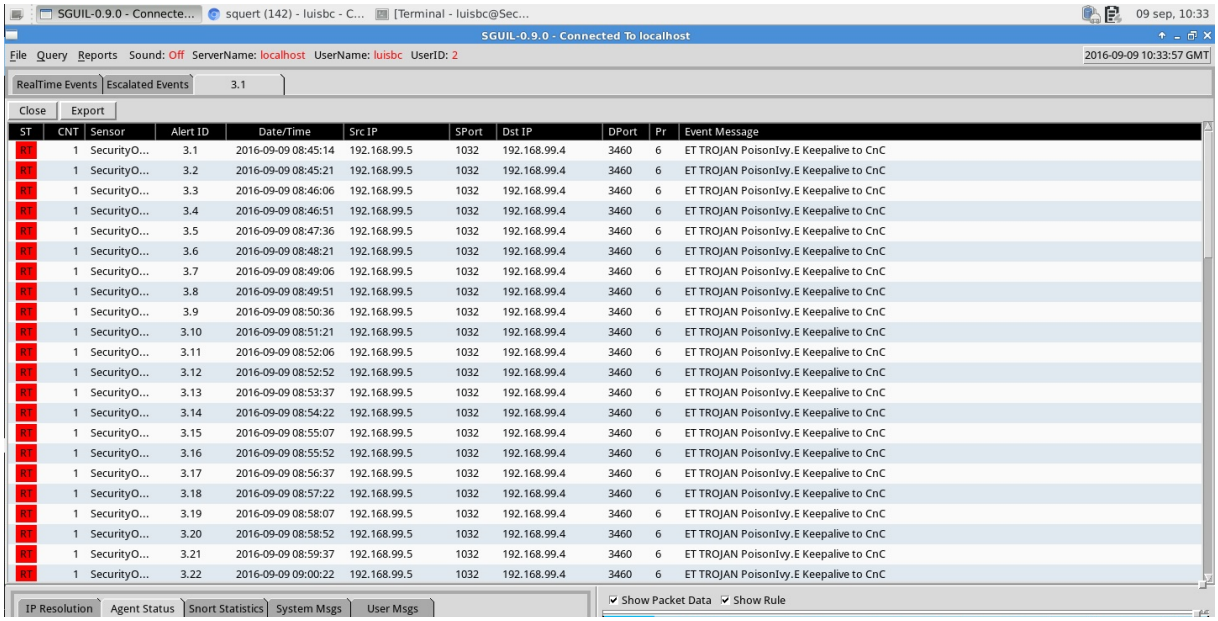


Figura 75: Eventos correlacionados para el ataque con PoisonIvy en Sguil.

Sguil es una potente herramienta que nos permite procesar la información captada a través de varias herramientas, como Wireshark o NetworkMiner. A continuación se muestran los análisis llevados a cabo con dichas herramientas:

WIRESHARK

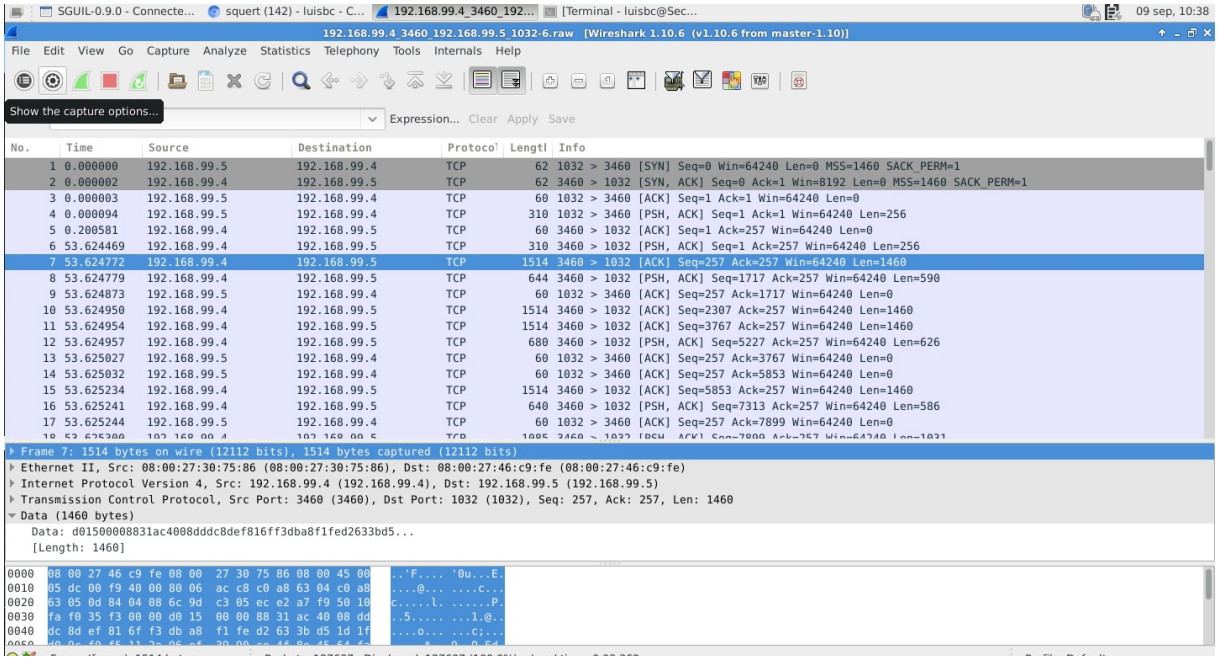


Figura 76: Captura de tráfico generado por PoisonIvy en Wireshark a través de Sguil.

NETWORKMINER

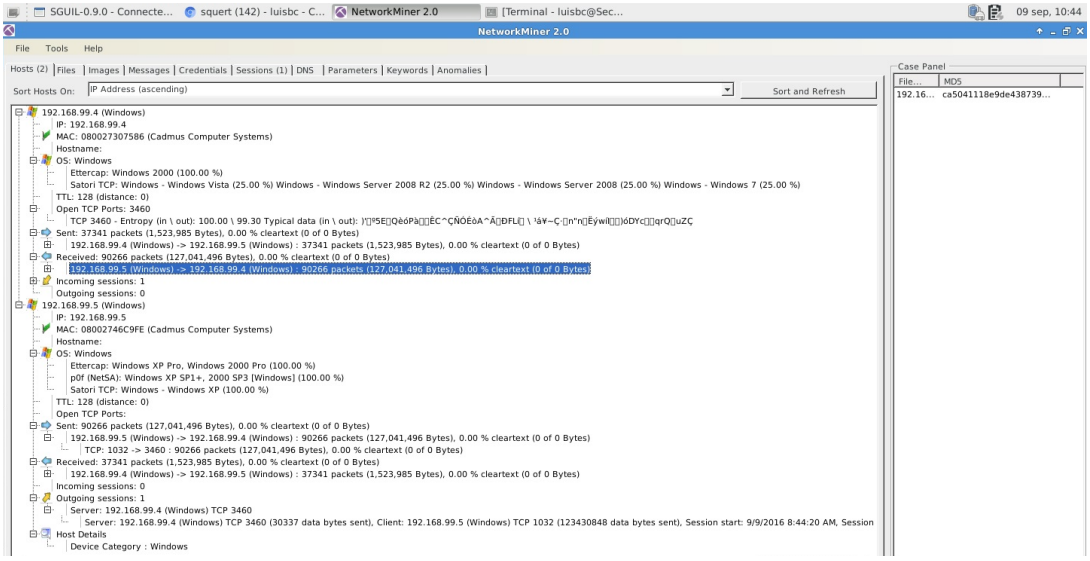


Figura 77: Datos analizados por NetworkMiner.

SQUERT

Squert es un NSM al igual que Sguil, y nos permite correlacionar datos captados por los distintos sensores.

A continuación se muestra la la alerta generada por la actividad de PoisonIvy generada en Snort:

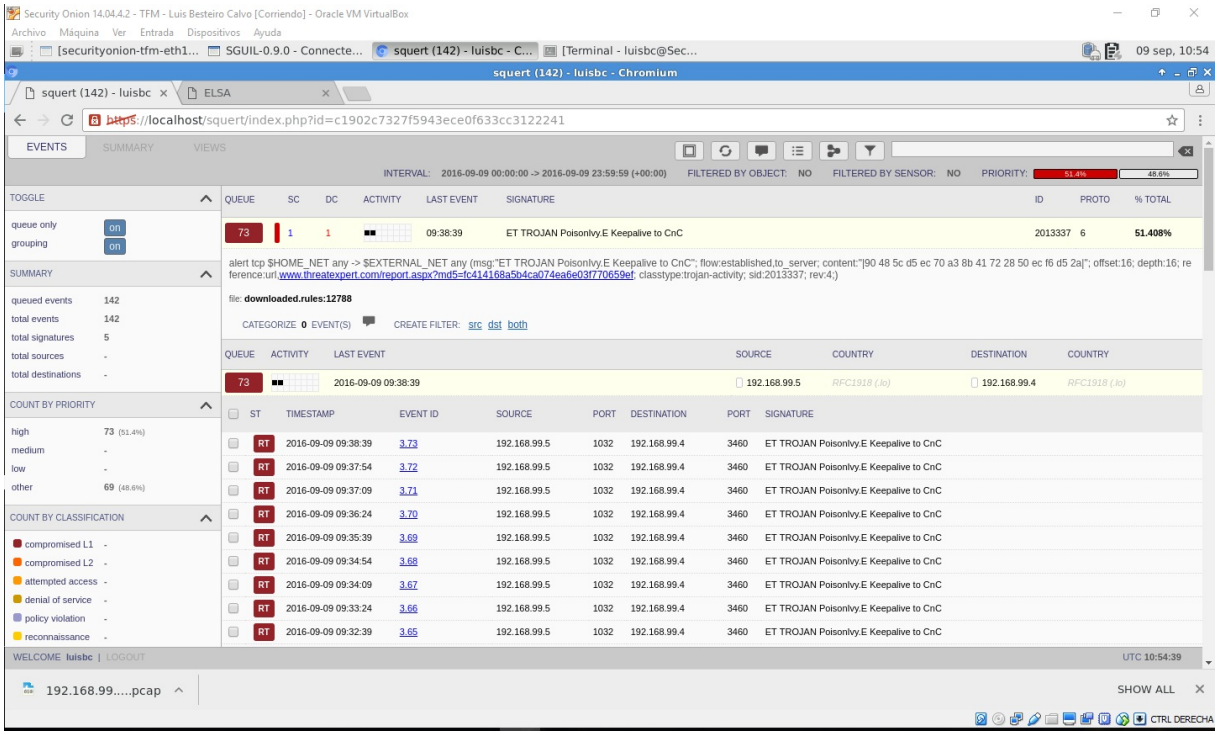


Figura 78: Alertas Snort mostradas en Squert.

Squert, al igual que Sguil, nos permite analizar los datos a través de la perspectiva de otras herramientas, tales como capME!, o ELSA.

A continuación podemos ver el análisis de una alerta de Snort por la actividad de PoisonIvY a través de capME!:

CAPME!

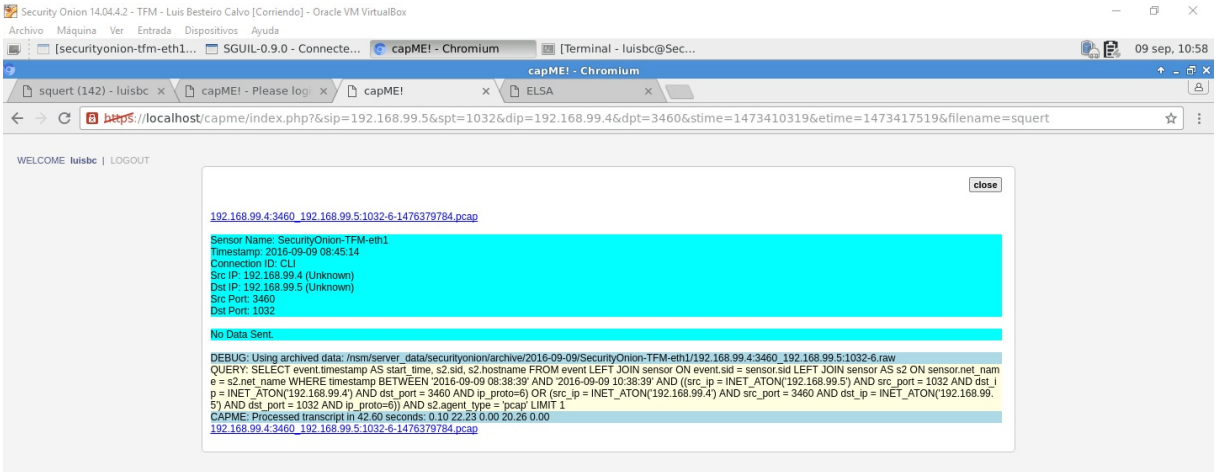


Figura 79: Análisis de una transmisión de PoisonIvy con capME!.

ELSA

Elsa nos permite llevar a cabo análisis pormenorizados a través de los datos de los logs almacenados. A continuación se muestra una gráfica en la que se agrupan las conexiones agrupadas por origen. Se puede ver que la máquina atacante (192.168.99.4) destaca sobre la víctima (192.168.99.6).

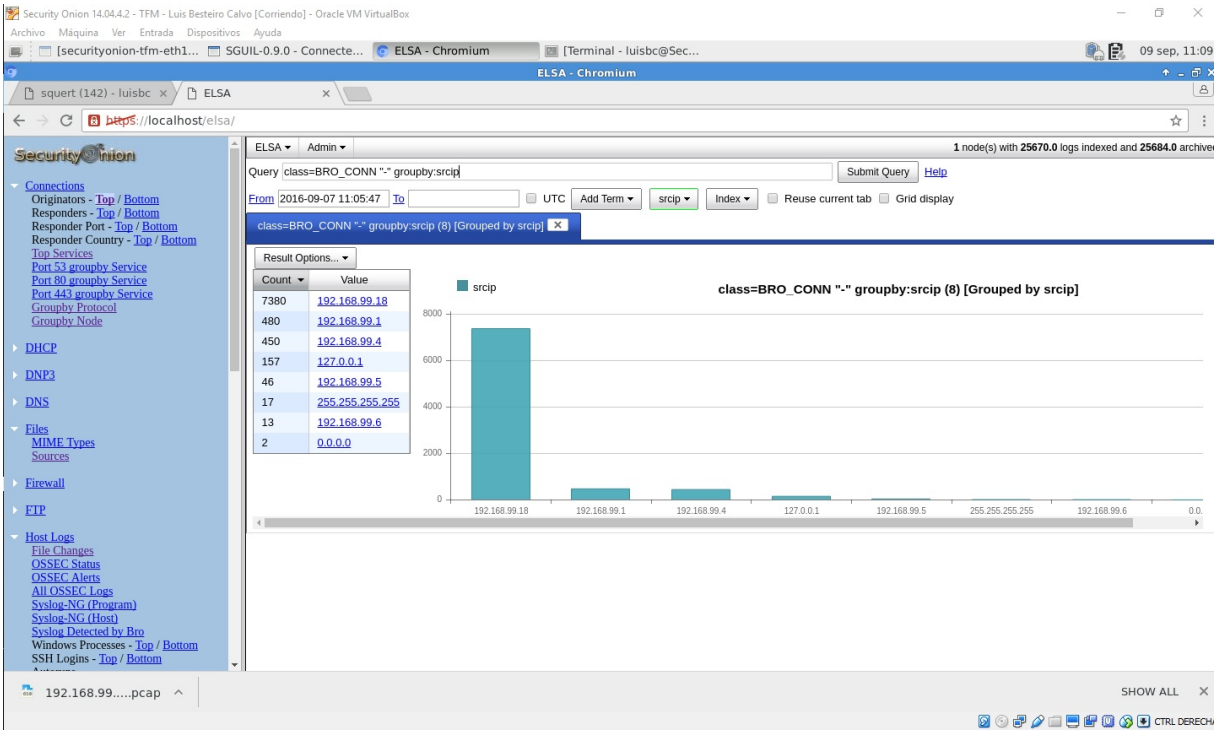


Figura 80: Análisis de conexiones por origen con ELSA.

4.2.6 Propuesta de mejoras

Debido a las limitaciones de medios y tiempo de un TFM, no es posible llevar a cabo un análisis elevadamente exhaustivo de todas las posibilidades posibles en el campo de la detección de APT, y por ello a continuación planteo propuestas que, a mi entender, completarían y mejorarían lo realizado en el presente trabajo, y deberían de ser llevadas a cabo para complementarlo:

- **Uso de más herramientas de detección (SIEM o NSM)**:
  Una de las mejoras a implementar sería la de realizar las mismas pruebas para 2 o más sistemas de monitorización. En mi opinión sería interesante repetir las pruebas con AlienVault OSSIM en lugar de Security Onion para comparar resultados.
- **Mayores muestras de malware**:
  Para evaluar mejor la eficacia de las herramientas de monitorización, es importante utilizar el mayor número de muestras posibles de malware, tanto si hablamos de muestras completamente diferentes (por ejemplo DarkComet) o modificaciones más actuales de PoisonIvy que mejoren su eficacia.
- **Mayor número de iteraciones con cada muestra de malware**:
  Adicionalmente a lo mencionado en el anterior apartado, para valorar con más detalle la eficacia de las herramientas de monitorización, sería interesante el realizar un mayor número de iteraciones, ya no solo con cada herramienta, sino con cada una de sus múltiples configuraciones posibles.
- **Realizar las modificaciones** pertinentes en el software de monitorización y análisis de red para mejorar la eficiencia y eficacia de la detección.
- **Un mayor número de máquinas virtuales** simulando distintos servicios y así poder simular en tiempo real junto con la infección una mayor cantidad de tráfico que permita generar un entorno más realista.

4.3. Desarrollo de la prueba Covert Channel

Para la realización de esta prueba partimos del estado de la prueba realizada previamente, por lo que no sería necesario repetir el apartado 4.2.1 de Simulación de entorno de trabajo normal. A continuación se muestra el cómo se realizó la prueba y los resultados obtenidos.

4.3.1 Análisis de la prueba a realizarse

Esta prueba consiste en la creación de un covert channel entre dos máquinas dentro del mismo entorno de red para comprobar la eficacia de la distribución Security Onion en su detección.

A modo de recapitulación de los componentes que entrarán en juego para esta prueba:

- **Máquinas virtuales utilizadas para la comunicación a través del covert channel**:
  - Linux Mint Cinamon 18 64bit.
  - Kali Linux 2016.2 64bit.
- **Máquina virtual de monitorización y detección**:
  - Security Onion 14.04.4.2.
- **Software para creación del covert channel**:
  - ptunnel.

4.3.2 Creación del Covert Channel

A continuación se va a describir el procedimiento a seguir para crear el covert channel por medio de ptunnel.

1. **Instalamos ptunnel** en la MV cliente (Linux Mint). En Kali Linux no sería necesario ya que viene incluido en la distribución.

   `sudo apt-get install ptunnel`

2. Abrimos **Wireshark** en la **máquina cliente** para analizar la conexión tunelizada. Hay que *abrirlo como root* para que capte el tráfico.

   `sudo /usr/bin/wireshark`

3. Creamos el **proxy** en la MV proxy, es decir, en la Kali Linux.

   `Ptunnel`

4. Creamos la **conexión tunelizada** en la **máquina cliente** con el proxy.

   sudo ptunnel -p *IPproxy* **-lp** *PuertoCliente* **-da** *127.0.0.1* **-dp** *PuertoProxyServidor*

5. Ponemos en **modo escucha** el **servidor**-proxy.

   nc -l -p *PuertoProxyServidor*

6. **Establecemos la comunicación** en el **cliente** con el proxy.

   nc *127.0.0.1 PuertoClientePTunnel*

7. **Escribimos** en la terminal para llevar a cabo el ejemplo de *transmisión a través de Covert-Channel*.
8. **Cerramos la aplicación** pulsando en la terminal:

   Ctrl+C

9. **Cerramos Wireshark** guardando la captura de tráfico.

A continuación se muestra el procedimiento realizado mediante capturas de pantalla en las máquinas virtuales que se comunican a través del covert channel:

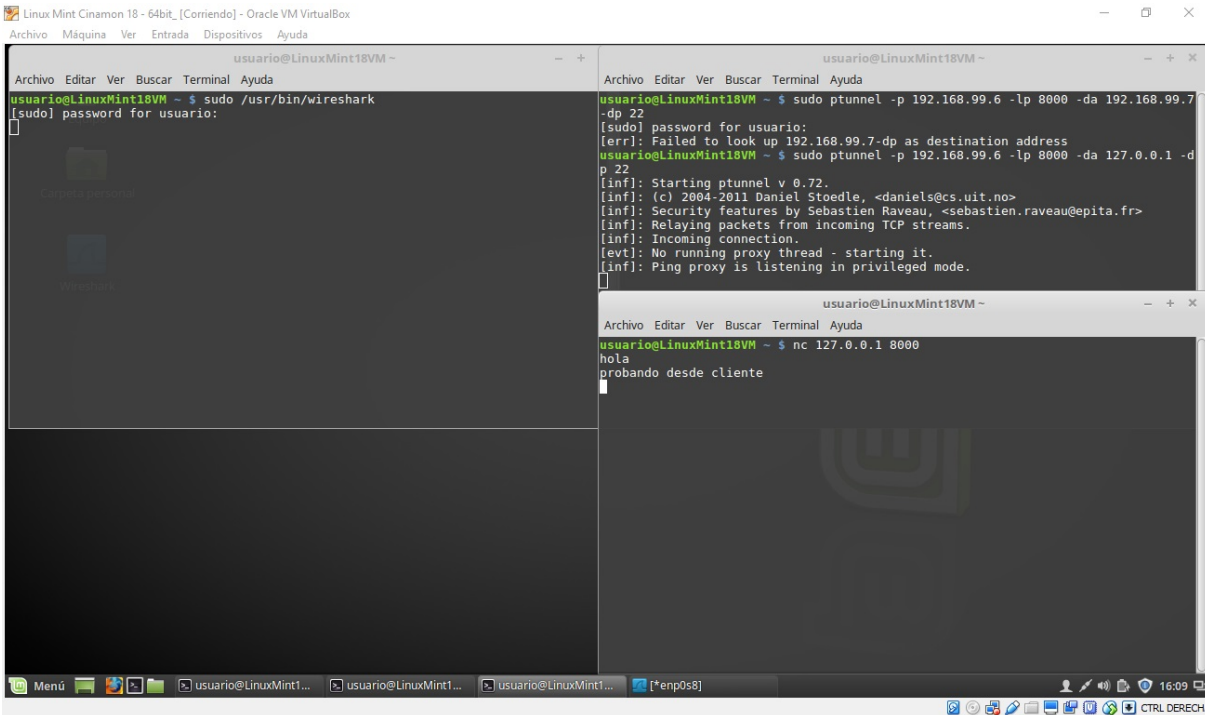


Figura 80: Creación del ptunnel en máquina cliente Linux Mint.

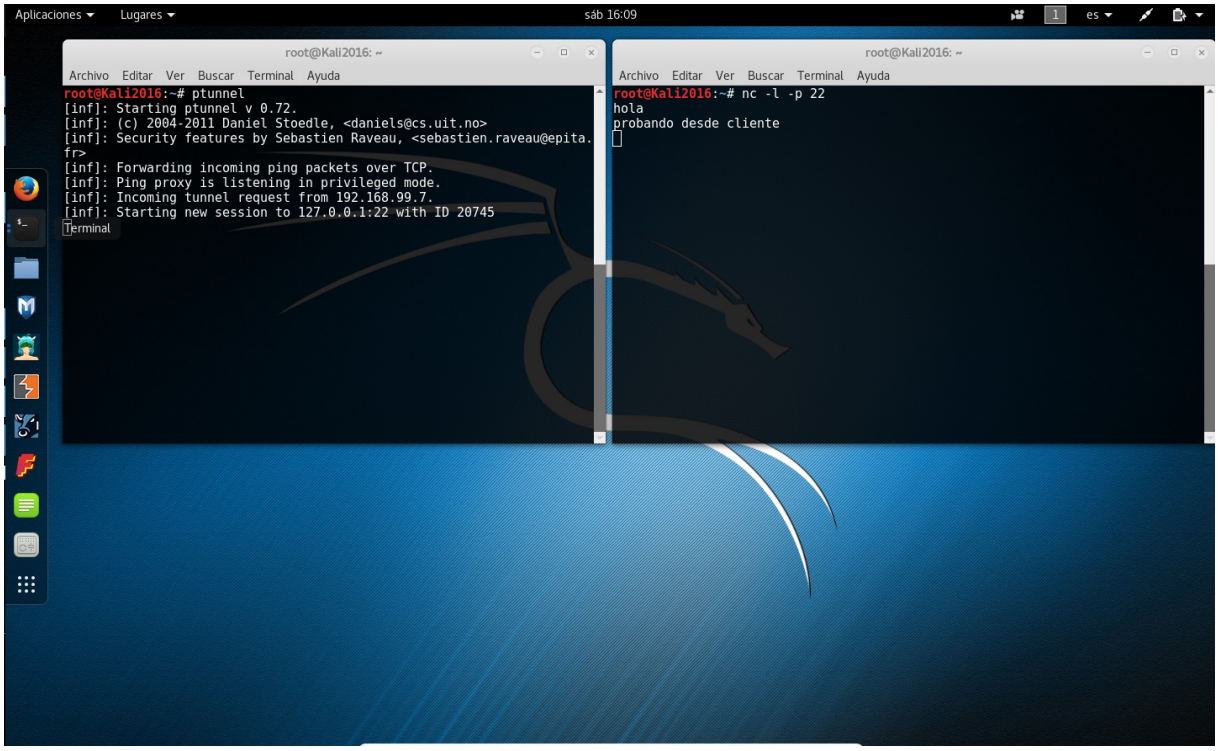


Figura 81: Creación del ptunnel en máquina proxy-servidor Kali linux.

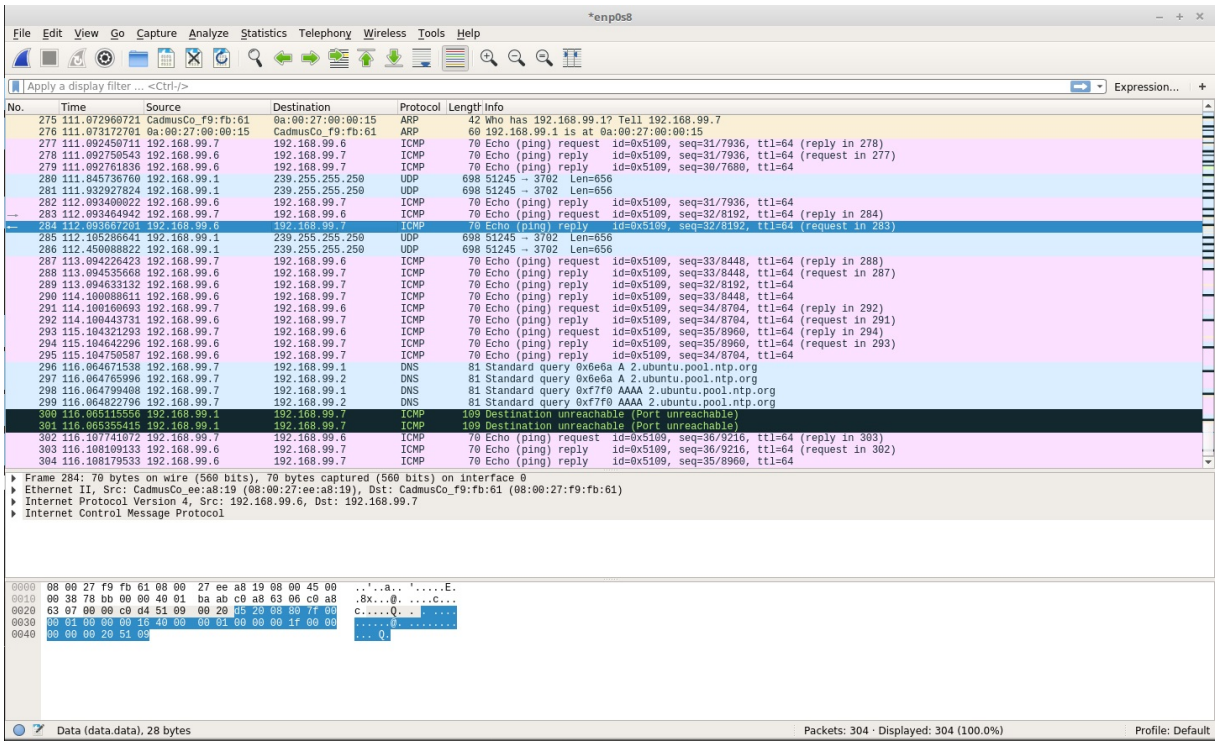


Figura 82: Captura de paquetes con Wireshark en la máquina cliente.

4.3.3 Análisis de entrono de trabajo con Covert Channel

A continuación se muestra la información obtenida a través de las herramientas de visualización de eventos Sguil y Squert de Security Onion:

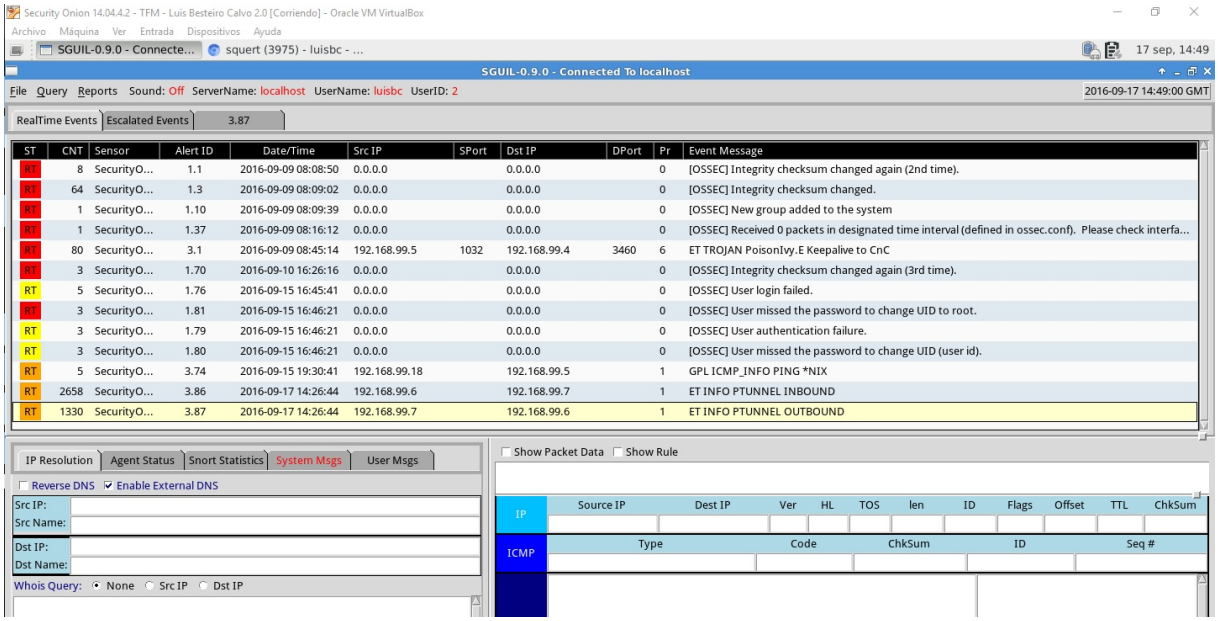


Figura 83: Captura de pantalla de Sguil en la que se puede ver la detección del tráfico de entrada y salida del covert channel.

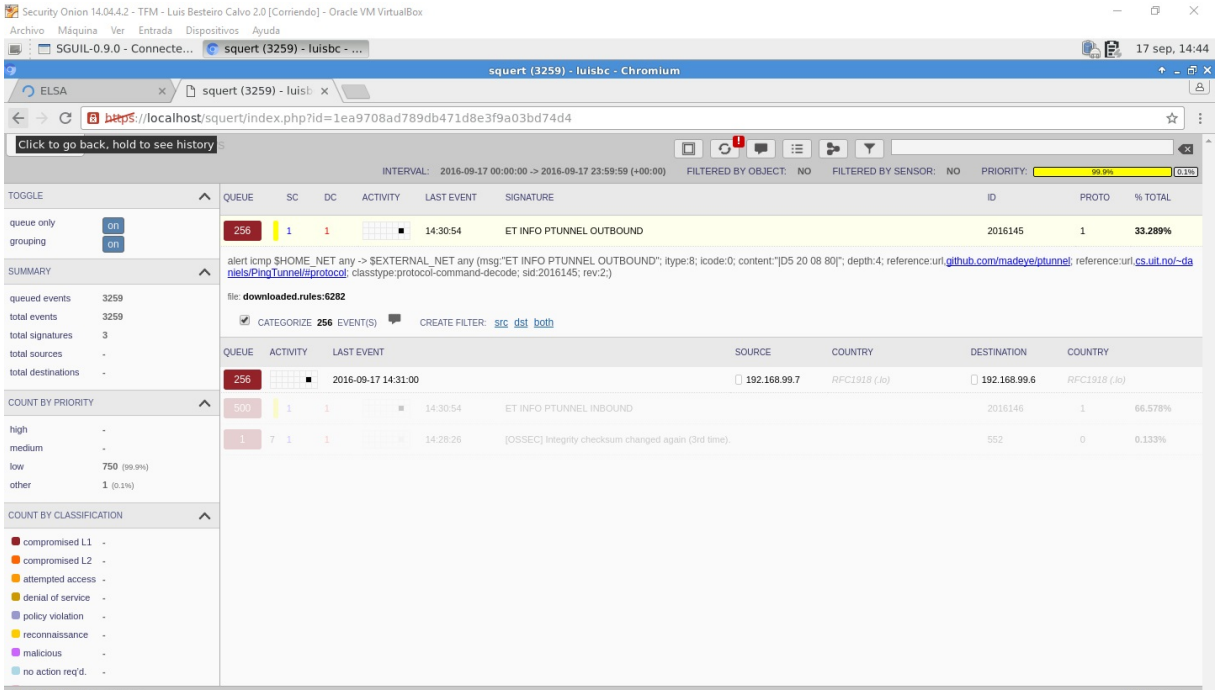


Figura 84: Captura de pantalla de Squert con la regla de Snort para detectar el covert channel.

4.3.4 Evaluación de la prueba

Mediante la distribución Security Onion llevamos a cabo la monitorización del la red y observamos si ésta es capaz de detectar el covert channel y qué datos nos ofrece sobre.

Como se puede observar en las capturas de pantalla que se muestran a continuación, las distintas herramientas de visualización del estado de la red nos avisan del covert channel. En la figura 83, podemos ver que Sguil nos muestra como las últimas alarmas la detección del covert channel, indicando además que ha sido implementado utilizando la herramienta ptunnel. En la figura 84 podemos observar que Squert nos muestra a mayores la regla de Snort utilizada para la detección.

4.3.5 Propuesta de mejoras

Tal como se ha comentado para este mismo apartado en la prueba anterior, el piloto tiene como limitaciones las limitaciones intrínsecas a un TFM. Personalmente, sería interesante implementar adicionalmente las siguientes propuestas:

- **Uso de más herramientas de detección (SIEM o NSM**).
  Una de las mejoras a implementar sería la de realizar las mismas pruebas para 2 o más sistemas de monitorización.
- **Implementación de diversos tipos de covert channel**.
  Para evaluar mejor la eficacia de las herramientas de monitorización, sería interesante el llevar a cabo la implementación de distintos tipos de covert channel, ya que cada uno de ellos conllevas reglas de detección diferentes.
- **Utilización de diversas herramientas de implementación de covert channel**.
  El uso de diferentes herramientas implica diferentes puntos de vista, diferentes maneras de implementar un mismo concepto, y ello ayudaría en la evaluación de la eficacia de las herramientas de detección.

4.4. Presentación de los resultados

La prueba piloto realizada ha sido satisfactoria en la medida que la herramienta de monitorización ha sido capaz de identificar correctamente el ataque mediante el análisis del tráfico de red, no solo mostrando una alerta acerca del ataque en si, sino también identificando el tipo de malware utilizado.

4.4.1 Ataque APT

¿Se ha detectado la comunicación entre máquina víctima y C&C?
Sí. Security Onion, mediante la alerta generada por herramienta NIDS **Snort**, y a través de las interfaces **Sguil** y **Squert**, nos ha alertado de la APT.

Regla de Snort con que ha detectado el ataque y ha generado la alerta:
alert tcp $HOME_NET any -> $EXTERNAL_NET any (msg:"ET TROJAN PoisonIvy.E Keepalive to CnC"; flow:established,to_server; content:"|90 48 5c d5 ec 70 a3 8b 41 72 28 50 ec f6 d5 2a|"; offset:16; depth:16; reference:url,www.threatexpert.com/report.aspx?md5=fc414168a5b4ca074ea6e03f770659ef; classtype:trojan-activity; sid:2013337; rev:4;)

Identificación del tipo de malware:
Sí. Snort ha identificado correctamente el tipo de malware (troyano) y el nombre por el que es conocido, PoisonIvy.

Número de alertas:
El número de alertas generadas en SGUIL por la acción de PoisonIvy es de 73 (Alert ID 3.1 a 3.73).

Correcta identificación de máquinas relacionadas con el ataque:
Sí. La identificación de direcciones IP y puertos ha sido correcta.

Alertas con falsos positivos:
No.

4.4.2 Covert Channel

¿Se ha detectado la comunicación a través de covert chanel?
Sí. La distribución Security Onion contiene ya por defecto las reglas necesarias no solo para detectar que se está produciendo una comunicación encubierta sino también para conocer qué software se ha utilizado para crearla.

Regla de Snort con que ha detectado el covert channel y ha generado la alerta:
Regla para tráfico de extracción:
alert icmp $HOME_NET any -> $EXTERNAL_NET any (msg:"ET INFO PTUNNEL OUTBOUND"; itype:8; icode:0; content:"|D5 20 08 80|"; depth:4; reference:url,github.com/madeye/ptunnel; reference:url,cs.uit.no/~daniels/PingTunnel/#protocol; classtype:protocol-command-decode; sid:2016145; rev:2;)

Regla para tráfico hacia el cliente:
alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"ET INFO PTUNNEL INBOUND"; itype:0; icode:0; content:"|D5 20 08 80|"; depth:4; reference:url,github.com/madeye/ptunnel; reference:url,cs.uit.no/~daniels/PingTunnel/#protocol; classtype:protocol-command-decode; sid:2016146; rev:3;)

Identificación del tipo de software con que se ha creado el covert channel:
Sí. Snort ha identificado correctamente la herramienta utilizada para crear el covert channel, es decir, nos ha notificado que ésta era ptunnel.

Correcta identificación de máquinas relacionadas con el covert channel:
Sí. La identificación de direcciones IP y puertos ha sido correcta.

Alertas con falsos positivos:
No.

4.5. Discusión de los resultados

Siendo el objetivo del presente trabajo el averiguar si existen herramientas de carácter libre que nos ayuden a luchar contra las APT de una manera eficaz, se puede decir que la distribución Security Onion ha cumplido con creces, ya que no solo ha detectado una intrusión, sino que además ha identificado qué tipo de intrusión era y su denominación, dejando claro cual era la amenaza a la que el sistema estaba haciendo frente. No cabe más que alabar el trabajo realizado Doug Burks, ya que no solo ha sabido hacer converger una gran suite de herramientas, sino que además resulta extremadamente eficaz con la configuración por defecto, incluyendo una gran cantidad de reglas.

Si bien para conocer si sería capaz de detectar una amenaza de última generación habría que hacer unas pruebas mucho más exhaustivas mediante un laboratorio con mayores medios, sí que se puede afirmar con un elevado grado de confianza que manteniendo este conjunto de herramientas actualizado, no solo a nivel de software, sino también de firmas, se puede tener una monitorización exhaustiva, eficaz y eficiente.

Además, la elevada comunidad de usuarios existente alrededor de las herramientas incluidas en la distribución Security Onion, permitirá tener un nivel de recursos que será tan elevado en cantidad como en calidad.

5. CONCLUSIONES Y TRABAJO FUTURO

5.1. Análisis sobre el TFM

Las APT suponen un problema tan complejo y peligroso como apasionante. No solo intervienen cuestiones técnicas, sino también políticas y activistas, por lo que profundizar en su conocimiento y motivaciones nos lleva a un contexto rico en matices.

A lo largo de este trabajo fin de máster recorremos desde las distintas etapas de un ataque APT hasta la metodología para exfiltrar la información del sistema víctima, los covert-channels o canales encubiertos, metodología rebosante de creatividad e ingenio para lograr extraer la información sin que usuarios, administradores o sistemas de seguridad sean conscientes del ataque que se está produciendo.

En lo referente a las herramientas de monitorización para detectar ataques APT, por un lado se han nombrado y descrito, y por otro lado se han puesto a prueba de la mano de la distribución Security Onion para comprobar su eficacia y así la posibilidad de llevar a cabo la instalación de un sistema de monitorización de seguridad en la red de una manera asumible por la gran mayoría de las organizaciones.

Si bien la distribución de herramientas de carácter libre **Security Onion** dista de ser una distribución con una corta curva de aprendizaje, ya que cuenta con un elevado número de herramientas y la interface no es excesivamente intuitiva, es una distribución suficientemente amigable y completa como para ser implantada en cualquier organización de cualquier tamaño, dejando claro que sí se puede llevar a cabo una defensa efectiva contra APT mediante software libre. Cabe añadir que al ser Security Onion un compendio de herramientas de carácter libre, existe una gran comunidad de contribuyentes a su alrededor que proveen de mejoras del rendimiento de este sistema a través de nuevas alertas, firmas y recomendaciones de configuración y gestión.

Sin duda, dedicándole los recursos humanos y técnicos adecuados, a través de herramientas de carácter libre se puede ayudar a las distintas organizaciones a luchar contras las APT sin tener que llevar a cabo una elevada inversión en lo referente a software.

5.2. Contribuciones del trabajo

Este trabajo realiza un extenso y minucioso análisis sobre las APT y sus metodologías de exfiltración de datos por medio de los canales encubiertos, que puede ser tomado como referencia tanto para el estudio de las APT como para trabajos que pretendan lograr nuevas metas en el campo.

Al mismo tiempo es una descriptiva guía sobre como llevar a cabo paso a paso la elaboración de un laboratorio virtual de pruebas de ataques APT y así como sobre llevar a cabo la configuración de la distribución de monitorización de seguridad en red Security Onion y la configuración de las herramientas APT PoisonIvy y ptunnel para llevar a cabo la simulación de un APT.

5.3. Líneas de trabajo futuro

En cuanto a **cómo debería de continuarse el presente trabajo**, se tendría que comenzar por lo comentado previamente en el apartado 4.2.6:

- Uso de más herramientas de detección (SIEM o NSM).
- Mayores muestras de malware.
- Mayor número de iteraciones con cada muestra de malware.
- Realizar las modificaciones pertinentes en el software de monitorización y análisis de red para mejorar la eficiencia y eficacia de la detección.
- Un mayor número de máquinas virtuales simulando distintos servicios y así poder simular en tiempo real junto con la infección una mayor cantidad de tráfico que permita generar un entorno más realista.

En lo referente a **cómo puede evolucionar, o cómo debería evolucionar este campo**, en opinión del autor:

- Creación de *algoritmos* para llevar a cabo predicciones de cómo puede evolucionar un malware concreto y así poder detectar posibles futuras variantes.
- Empleo de la *Inteligencia Artificial* (IA) para llevar a cabo un filtrado y exposición de datos más funcional para el administrador de sistemas, e incluso sustituirlo. Este es el caso de la plataforma de ciberseguridad basada en IA “AI2” desarrollada por investigadores del MIT, que puede bloquear un 85% de los ciberataques con una gran precisión (Khandelwal, 2016).

6. LISTA DE REFERENCIAS

Alperovitch, D. (2016). Bears in the Midst: Intrusion into the Democratic National Committee. *Crowdstrike Blog*. Recuperado el 10 de agosto de 2016 de https://www.crowdstrike.com/blog/bears-midst-intrusion-democratic-national-committee/

Anderson, N. (2012). Confirmed: US and Israel created Stuxnet, lost control of it. arsTECHNICA. Recuperado el 9 de agosto de 2016 de http://arstechnica.com/tech-policy/2012/06/confirmed-us-israel-created-stuxnet-lost-control-of-it/

Bejtlich, R. (2010). Understanding the advanced persistent threat. Recuperado el 15 de junio de 2016 de http://searchsecurity.techtarget.com/magazineContent/Understanding-the-advanced-persistent-threat

Belshe, M., BitGo, Peon, R. (mayo de 2015). Hypertext Transfer Protocol Version 2 (HTTP/2). RFC 7540. *Internet Engineering Task Force (IETF)*. Recuperado el 2 de agosto de 2016 de https://tools.ietf.org/pdf/rfc7540.pdf

Bidou, R., Raynal, F. (2005). Covert Channels. Recuperado el 2 de agosto de 2016 de http://www.iv2-technologies.com/CovertChannels.pdf

Binde, B. E., McRee, R. M., O'Connor, T. J. (5/22/2011). Assessing Outbound Traffic to Uncover Advanced Persistent Threat. *SANS Technology Institute*. Recuperado de https://www.sans.edu/student-files/projects/JWP-Binde-McRee-OConnor.pdf

Butler, W. L..(1972). A Note on the Confinement Problem. *Xerox Palo Alto Research Center*. Recuperado de http://research.microsoft.com/en-us/um/people/blampson/11-confinement/acrobat.pdf

Couture, E.. (19 de agosto de 2010). Covert Channels. *SANS Institute InfoSec Reading Room*. Recuperado de: https://www.sans.org/reading-room/whitepapers/detection/covert-channels-33413

Cutler, T. (2010). The anatomy of an advanced persistent threat. *Wired Business Media*. Recuperado el 15 de junio de 2016 de http://www.securityweek.com/anatomy-advanced-persistent-threat

Department of Defence. (26 de diciembre de 1985). Department of Defense Trusted Computer System Evaluation Criteria. Recuperado el 19 de junio de 2016 de http://csrc.nist.gov/publications/history/dod85.pdf

Doug. (2008). Using ICMP tunneling to steal Internet. *Neverfear*. Recuperado el 20 de julio de 2016 de http://www.neverfear.org/blog/view/9/Using_ICMP_tunneling_to_steal_Internet

ElMundo.es. (2011). McAfee denuncia una cadena de ciberataques contra 72 grandes organismos. Recuperado el 9 de agosto de 2016 de http://www.elmundo.es/elmundo/2011/08/03/navegante/1312346768.html

FireEye. (2014). POISON IVY: Assessing Damage and Extracting Intelligence. Recuperado el 6 de septiembre de 2016 de https://www.fireeye.com/content/dam/fireeye-www/global/en/current-threats/pdfs/rpt-poison-ivy.pdf

Gligor, V. D..(1993). A Guide to Understanding Covert Channel Analysis of Trusted Systems. The Light Pink Book. *U.S. National Computer Security Center, Tech. Rep*. Recuperado el 19 de junio de 2016 de http://fas.org/irp/nsa/rainbow/tg030.htm

González Pérez, P. (2014). *Ethical Hacking. Teoría y práctica para la realización de un pentesting*. Móstoles (Madrid): Edición 0xWORD Computing S.L.

Graham, B. (2005). Hackers Attack Via Chinese Web Sites. The Washington Post. Recuperado el 9 de agosto de 2016 de http://www.washingtonpost.com/wp-dyn/content/article/2005/08/24/AR2005082402318.html

GREAT (Global Research and Analysis Team). (2016). The ProjectSauron APT. Kasperskylab. Recuperado el 10 de agosto de 2016 de https://securelist.com/files/2016/07/The-ProjectSauron-APT_research_KL.pdf

Gregg, M. (2007). Hack the Stack: Using Snort and Ethereal to Master The 8 Layers of An Insecure Network. *Syngress Publishing*.

Gray World Team. (2006). How to cook a covert channel. *Haking9 Magazine.* Recuperado el 4 de agosto de 2016 de http://gray-world.net/projects/papers/cooking_channels.txt

Holguín, J. M., Moreno, M., Merino, B.,(Mayo 2013). Detección de APTs. *INTECO*. Recuperado de: https://www.incibe.es/extfrontinteco/img/File/intecocert/EstudiosInformes/deteccion_apt.pdf

HoneyNet Project. (2007). Know your enemy: Fast-flux service networks. Recuperado el 15 de junio de 2016 de http://www.honeynet.org/papers/ff

Information Sciences Institute. University of Southern California. (Septiembre de 1981). RFC:791. Internet Protocol. DARPA Internet Program Protocol Specification. *DARPA*. Recuperado el 19 de julio de 2016 de https://tools.ietf.org/pdf/rfc791.pdf

Information Sciences Institute. University of Southern California. (Septiembre de 1981). RFC:793. Transmission Control Protocol. DARPA Internet Program Protocol Specification. *DARPA*. Recuperado el 19 de julio de 2016 de https://tools.ietf.org/pdf/rfc791.pdf

Kemmerer, R. A. (Agosto de 1983). Shared resource matrix methodology: A practical approach to identifying covert channels. *ACM Transactions on Computer Systems*. Recuperado de http://www.cs.unm.edu/~crandall/491591spring10/kemmerer.pdf

Khandelwal, S. (19 de abril de 2016). MIT builds Artificial Intelligence system that can detect 85% of Cyber Attacks. *The Hacker News. Security in a serious way*. Recuperado el 10 de septiembre de 2016 de http://thehackernews.com/2016/04/artificial-intelligence-cyber-security.html?utm_source=feedburner&utm_medium=feed&utm_campaign=Feed%3A+TheHackersNews+%28The+Hackers+News+-+Security+Blog%29&_m=3n.009a.1220.gn0ao07oxa.pbl

Kozierok, C. M. (2005). *The TCP/IP GUIDE. A Comprehensive, Illustrated Internet Protocols Reference*. Recuperado de http://index-of.es/Magazines/hakin9/books/No.Starch.TCP.IP.Guide.Oct.2005.pdf

Mandiant. (2013). APT1. Exposing One of China's Cyber Espionage Units. *MANDIANT*. Recuperado de: http://intelreport.mandiant.com/Mandiant_APT1_Report.pdf

Mariscal Fragoso, C. y Candau Romero, J., (Septiembre 2014). APTs. Durmiendo con el enemigo. *SiC*. Recuperado de:https://www.ccn-cert.cni.es/publico/dmpublidocuments/SIC111_073-083.pdf

McAfee Labs y McAfee Foundstone Professional Services. (2010). Protecting Your Critical Assets. Lessons Learned from “Operation Aurora”. *Wired*. Recuperado el 9 de agosto de 2016 de http://www.wired.com/images_blogs/threatlevel/2010/03/operationaurora_wp_0310_fnl.pdf

O’Harrow, R. y Linch, G. (2012). Timeline: Key events in cyber history. *The Washington Post*. Recuperado el 9 de agosto de 2016 de http://www.washingtonpost.com/wp-srv/special/investigative/zeroday/cyber-history-timeline/index.html

Postel, J. (Agosto de 1980). RFC:768.User Datagram Protocol. Recuperado el 2 de agosto de 2016 de https://tools.ietf.org/pdf/rfc768.pdf

Postel, J. (Septiembre de 1981). RFC:792. Internet Control Message Protocol. DARPA Internet Program Protocol Specification. *Network Working Group*. Recuperado el 19 de julio de 2016 de https://tools.ietf.org/pdf/rfc792.pdf

Ríos, R., Onieva, J. A. (2008). Clasificación de canales encubiertos. Un nuevo canal: Covert_DHCP. *NICS Lab. Publications*. Recuperado el 17 de junio de 2016 de https://www.nics.uma.es/pub/papers/Rios2008.pdf

Rowland, C. (1997). Covert Channels in the TCP/IP Protocol Suite. Recuperado el 1 de agosto de 2016 de http://www.firstmonday.org/ojs/index.php/fm/article/view/528/449

Rutkowska, J. (Diciembre de 2004). The Implementation of Passive Covert Channels in the Linux Kernel. *Chaos Communication Congress*. Recuperado el 2 de agosto de 2016 de https://events.ccc.de/congress/2004/fahrplan/files/223-passive-covert-channels-linux.pdf

Segura, J. (2016). Top Exploit Kits Round Up | March Edition. MalwarebytesLabs. Recuperado el 10 de agosto de 2016 de https://blog.malwarebytes.com/threat-analysis/exploits-threat-analysis/2016/03/top-exploit-kits-round-up-march-edition/

Selvi, J. (20 de marzo de 2012). Covert Channels Over Social Networks. *SANS Institute. Global Information Assurance Certification Paper.* Recuperado el 4 de agosto de 2016 de http://www.giac.org/paper/gcih/10163/covert-channels-social-networks/117979

Shah, G., Molina, A., Blaze, M. (2006). Keyboards and Covert Channels. *USENIX-SS'06: Proceedings of the 15th Conference on USENIX Security Symposium*. Berkeley, CA, USA. USENIX Association, pp. 59-75.

Siciliano, R. (2015). What is a Remote Administration Tool (RAT)?. McAfee Consumer Blog. Recuperado el 10 de agosto de 2016 de https://blogs.mcafee.com/consumer/what-is-rat/

Stødle, D. (2005). ptunnel – Ping Tunnel. Recuperado el 27 de julio de 2016 de http://www.mit.edu/afs.new/sipb/user/golem/tmp/ptunnel-0.61.orig/web/

Stoll, C. (1988). Stalking The Wily Hacker. *Communication of the ACM, Vol. 31, No. 5*. Recuperado de http://pdf.textfiles.com/academics/wilyhacker.pdf

Zander, S. (mayo de 2010). *Performance of Selected Noisy Covert Channels and Their Countermeasures in IP Networks*. (Tesis de Doctorado). Center for Advanced Internet Architectures. Swinburne University of Technology, Melbourne. Recuperado el 5 de agosto de 2016 de http://caia.swin.edu.au/cv/szander/thesis/thesis.html

The Common Criteria Project. (Enero 2004). Common Criteria for Information Technology Security Evaluation Part 3. Technical Report CCIMB-2004-01-003. Recuperado el 16 de junio de 2016 de http://www.commoncriteriaportal.org/public/files/ccpart3v2.2.pdf

TheSecDevGroup. (2009). Tracking GhostNet: Investigating a Cyber Espionage Network. *Information Warfare Monitor*. Recuperado el 9 de agosto de 2016 de https://www.f-secure.com/weblog/archives/ghostnet.pdf

Thyer, J. S. (10 de enero de 2008). Covert Data Storage Channel Using IP Packet Headers. *SANS Institute InfoSec Reading Room*. Recuperado el 31 de julio de 2016 de https://www.sans.org/reading-room/whitepapers/covert/covert-data-storage-channel-ip-packet-headers-2093

Valencia, J. E.(28 de mayo de 2016). SNMPChat: Chatear en un canal encubierto sobre SNMP. *UN INFORMÁTICO EN EL LADO DEL MAL*. Recuperado el 5 de agosto de 2016 de http://www.elladodelmal.com/2016/05/snmpchat-chatear-en-un-canal-encubierto.html

Virus News. (2014). Kaspersky Lab Uncovers “The Mask”: One of the Most Advanced Global Cyber-espionage Operations to Date Due to the Complexity of the Toolset Used by the Attackers. *Kaspersky lab*. Recuperado el 9 de agosto de 2016 de http://www.kaspersky.com/about/news/virus/2014/Kaspersky-Lab-Uncovers-The-Mask-One-of-the-Most-Advanced-Global-Cyber-espionage-Operations-to-Date-Due-to-the-Complexity-of-the-Toolset-Used-by-the-Attackers

Wangen, G. (2015). The Role of Malware in Reported Cyber Espionage: A Review of the Impact and Mechanism. Recuperado el 9 de agosto de 2016 de http://www.mdpi.com/2078-2489/6/2/183/pdf

Weissman, C. (1 de enero de 1996). Handbook for the Computer Security Certification of Trusted Systems. Naval Research Laboratory. Recuperado el 16 de junio de 2016 de http://www.dtic.mil/cgi-bin/GetTRDoc?AD=ADA390673

7. BIBLIOGRAFÍA

Anatomy of an Advanced Persistent Threat (APT) Group. FireEye, Inc. (18 diciembre 2014). [https://www.youtube.com/watch?v=SZCE677ijMU] YouTube.

Ashiq, J. A., (13 de mayo de 2015). Anatomy of an APT Attack: Step by Step Approach. EXPLOIT DEVELOPMENT, GENERAL SECURITY. *INFOSEC INSTITUTE*. Recuperado el 15 de junio de 2016 de http://resources.infosecinstitute.com/anatomy-of-an-apt-attack-step-by-step-approach/

Bermejo, J. (2013). *Tema 4: Análisis de malware*. Material no publicado. Recuperado el 15 de junio de 2016 de https://campusingenieria.unir.net/access/lessonbuilder/item/1630059/group/rep-musi/per7/musi011/pdf/apuntesprofesort4.pdf

Burks, D. (2016). Security Onion. Peel Back the Layers of Your Network. Recuperado el 8 de septiembre de 2016 de http://blog.securityonion.net

CCN-CERT (2016). CCN-CERT IA-09/16. Ciberamenazas 2015/Tendencias 2016. Resumen Ejecutivo. Recuperado el 8 de agosto de 2016 de https://www.ccn-cert.cni.es/informes/informes-ccn-cert-publicos/1483-ccn-cert-ia-0916-ciberamenazas-2015-tendencias-2016-resumen-ejecutivo/file.html

Dunne, R. (10 de diciembre de 2014). Covert_TCP File Transfer for Dummies using Linux. *Dunnesec*. Recuperado el 1 de agosto de 2016 de https://dunnesec.com/category/tools/covert_tcp/

ElevenPaths. (2016). Abuso de alias de Gmail para la exfiltración de información. *Eleven Paths Discovers*. Recuperado el 11 de agosto de 2016 de https://www.elevenpaths.com/wp-content/uploads/2016/07/ElevenPaths%20Discover_Abuso-alias-gmail-para-exfiltracion-informacion-v1_1.pdf

FaceCat (FaceBook Cat). Jose Selvi. (2 de octubre de 2011). [https://www.youtube.com/watch?v=flZUuRK2R-k] YouTube.

FaceCat + Poison Ivy. Jose Selvi. (2 de octubre de 2011). [https://www.youtube.com/watch?v=C_c8KNvVSVg] YouTube.

Falliere, N., Murchu, L. y Chien, E. (2011). W32.Stuxnet Dossier. *Symantec Security Response*. Recuperado el 9 de agosto de 2016 de https://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/w32_stuxnet_dossier.pdf

FireEye. *CYBER THREAT INTELLIGENCE REPORTS*. Recuperado el 8 de agosto de 2016 de https://www.fireeye.com/current-threats/threat-intelligence-reports.html

Gestión de la Seguridad Unificado vs. SIEM. Alien Vault. Recuperado el 27 de agosto de https://www.alienvault.com/docs/whitepapers/alienvault-vs-siem-white-paper-es.pdf

Intrusion Detection Systems (IDS) Best Practices. (2 de mayo de 2016). AlienVault. Cybrary. Recuperado el 29 de agosto de 2016 de https://www.cybrary.it/channelcontent/intrusion-detection-systems-ids-best-practices/

Instituto Español de Estudios Estratégicos Instituto Universitario General Gutiérrez Mellado. (2010). Ciberseguridad. Retos y Amenazas a la seguridad Nacional en el Ciberespacio.

kbandla. APT Notes. *GitHub*. Recuperado el 18 de junio de 2016 de https://github.com/kbandla/APTnotes

Ministerio de Defensa. Cuadernos de Estrategia. Recuperado el https://www.cni.es/comun/recursos/descargas/Cuaderno_IEEE_149_Ciberseguridad.pdf

Moreno, J. (26 de octubre de 2010). Covert Channels. *Security Artwork*. Recuperado el 16 de junio de 2016 de http://www.securityartwork.es/2010/10/26/covert-channels/

McHugh, J. (16 de diciembre de 1995). Covert Channel Analysis: A Chapter of the handbook for the Computer Security Certification of Trusted Systems. *University of North Carolina for*

the Naval Research Laboratory. Recuperado de http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.51.1438&rep=rep1&type=pdf

Oracle VM VirtualBox User Manual Version 5.1.4. Recuperado 16 de agosto de 2016 de http://www.virtualbox.org

Ramos, A. (13 de enero de 2010). TUNELIZANDO DNS, OTRA OPCIÓN CON IODINE 0.5.X. *Security By Default*. Recuperado el 4 de agosto de 2016 de http://www.securitybydefault.com/2010/01/tunelizando-dns-otra-opcion-con-iodine.html

Scott, S.J. (07 de noviembre de 2011). Amenazas persistentes avanzadas. *Magazcitum*. Recuperado el 18 de julio de 2016 de http://www.magazcitum.com.mx/?p=1547

Schreiber, J. Beginner's Guide to Open Source Intrusion Detection Tools. *Alien Vault*. Recuperado el 27 de agosto de 2016 de http://learn.alienvault.com/beginner-s-guide-to-opensource-ids-lookbook/asset

Security Onion Setup Phase 1. (23 septiembre 2013). [https://www.youtube.com/watch?v=D6IibAfPPD4] YouTube.

Security Onion Wiki. IntroductionToSecurityOnion. *Github*. Recuperado el 3 de agosto de 2016: https://github.com/Security-Onion-Solutions/security-onion/wiki/IntroductionToSecurityOnion

Thyer, J. *Packetheader.net*. Recuperado el 1 de agosto de 2016 de http://www.packetheader.net/

What's New in AlienVault OSSIM v5.3: Enhanced Insider Threat Detection and Improved Usability. Alien Vault. Recuperado el 27 de agosto de 2016 de https://www.alienvault.com/forms/webcast-thank-you/whats-new-in-ossim-v53-enhanced-insider-threat-detection

Wikipedia, La enciclopedia libre. Hypertext Transfer Protocol. Recuperado el 5 de agosto de 2016 de https://es.wikipedia.org/wiki/Hypertext_Transfer_Protocol

Wikipedia, La enciclopedia libre. Simple Network Management Protocol. Recuperado el 5 de agosto de 2016 de https://es.wikipedia.org/wiki/Simple_Network_Management_Protocol

Wikipedia, The Free Encyclopedia. Hacktivism. Recuperado el 18 de julio de 2016 de https://en.wikipedia.org/wiki/Hacktivism

Wikipedia, The Free Encyclopedia. Remote administration software. Recuperado el 10 de agosto de 2016 de https://en.wikipedia.org/wiki/Remote_administration_software#cite_note-12

Wikipedia, The Free Encyclopedia. The Cuckoo's Egg. Recuperado el 9 de agosto de 2016 de https://en.wikipedia.org/wiki/The_Cuckoo%27s_Egg

Wikipedia, The Free Encyclopedia. Usage share of operating systems. Recuperado el 1 de Septiembre de https://en.wikipedia.org/wiki/Usage_share_of_operating_systems

Zander, S. (mayo de 2010). Performance of Selected Noisy Covert Channels and Their Countermeasures in IP Networks. Doctorado. Centre for Advanced Internet Architectures. Faculty of Information and Communication Technologies Swinburne University of Technology, Melbourne. Recuperado el 18 de julio de 2016 de http://caia.swin.edu.au/cv/szander/thesis/thesis.html

Zetter, K. (2011). How digital detectives deciphered Stuxnet, the most menacing malware in history. *Wired.com*. Recuperado el 9 de agosto de 2016 de http://arstechnica.com/tech-policy/2011/07/how-digital-detectives-deciphered-stuxnet-the-most-menacing-malware-in-history/