UNIVERSIDAD INTERNACIONAL DE LA RIOJA

Universidad Internacional de La Rioja
Máster universitario en Seguridad Informática

Aplicación de metodología de Análisis de Malware al caso de estudio de la Amenaza Avanzada Persistente (APT) "Octubre Rojo"

Trabajo Fin de Máster

Presentado por: Abad Aramburu, Carlos

Director/a: Vázquez Poletti, José Luis

Ciudad: San Sebastián
Fecha: 20/02/2015

Agradecimientos

A mi Madre y mi Hermano por estar ahí una vez más.

A mis Amig@s porque, sin ser expertos en la materia, me han apoyado incondicionalmente.

A Don José Luis Vázquez Poletti por su apoyo durante la realización de este trabajo.

A Don Javier Bermejo por darme la oportunidad de contribuir a su trabajo y descubrirme un área realmente apasionante.

Resumen

La necesidad de adaptar la seguridad de las comunicaciones y los sistemas de información a la realidad de la evolución continua de las amenazas y la sofisticación de los ataques, ha forzado la aparición de una nueva disciplina denominada **Ciberdefensa**. Dicha disciplina de carácter puramente proactivo complementa la ya existente disciplina de Ciberseguridad, de naturaleza reactiva. La presente memoria se centra en una de las principales capacidades de Ciberdefensa, el **Análisis de Malware**, concentrada en analizar y comprender el funcionamiento del **código malicioso**. Para desarrollar dicha capacidad es necesaria la utilización de metodologías de análisis predefinidas. La realización de evaluaciones prácticas de estas metodologías genera una base de conocimiento sólida y garantizan su validez y efectividad. El presente piloto experimental se basa en la metodología de "Análisis e ingeniería inversa de código malicioso" desarrollada por Don Javier Bermejo Higuera, aplicándola sobre la **Amenaza Persistente Avanzada** (**APT)** conocida con el nombre de "**Octubre Rojo**".

Palabras Clave: Ciberdefensa, análisis de malware, código malicioso, Amenaza Avanzada Persistente (APT), Octubre Rojo.

Abstract

The birth of the new discipline known as **Cyberdefense** can be regarded as a direct result of the necessity to adapt security in the communications as well as information systems to the reality of the constant evolutions of threats and sophistication in cyber attacks. This new discipline of purely proactive character complements the already existing discipline of Cybersecurity, which is of reactive nature. This report mainly deals with one of the main abilities of Cyberdefense, **Malware Analysis**, which focuses on analysing and understanding the functioning of the **malicious code**. In order to develop this capacity it is necessary the use of predefined analytical methodologies. The execution of practical assessments of those methodologies creates a solid knowledge base and warrantees its validity and effectiveness. The present experimental pilot is based on the methodology of "Inverted analysis and engineering of malicious code" developed by Mr. Javier Bermejo Higuera, applying it to the **Advanced Persistent Threat** (**APT)** known as "**Red October**".

Keywords: Cyberdefense, Malware Analysis, malicious code, Advanced Persistent Threat (APT), Red October.

ÍNDICE DE CONTENIDOS

ÍNDICE DE ILUSTRACIONES

ÍNDICE DE TABLAS

1. Introducción

La seguridad de los sistemas de tecnologías de la información y comunicación (TIC) así como de la propia información es una necesidad real cuya demanda se encuentra en continuo crecimiento. Con el objetivo de cubrir dicha necesidad, desde el punto de vista conceptual, se presenta la Ciberseguridad como una disciplina, de naturaleza reactiva, que actualmente se desarrolla en todos los elementos, basados en sistemas TIC, de la cadena de valor del desarrollo de la sociedad como son empresas privadas de tipo tecnológico, instituciones de investigación y desarrollo, infraestructuras críticas y gobiernos. Tanto la dependencia actual de los sistemas de información y comunicación como el incremento del valor de la misma convierten al ciberespacio en un entorno con un alto grado de oportunidad para la comisión de distintos tipo de delitos con distintos tipos de fines.

El ciberespacio se define como la composición de tres capas, física, lógica y social.

- Capa física incluye el componente geográfico de los elementos y componentes de la red. En este entorno, los límites geopolíticos se pueden cruzar fácilmente a una tasa cercana a la velocidad de la luz.
- Capa Lógica incluye las conexiones lógicas entre nodos.
- Capa social que comprende el aspecto humano y cognitivo, e incluye la componente de personalidad virtual y física. La personalidad virtual está directamente relacionada con la dificultad de atribución de responsabilidad.

Las características de del ciberespacio son las siguientes:

- Facilidad para explotar el anonimato basado en la utilización de la identidad lógica en lugar de la física.
- No existen fronteras definidas y la regulación no es efectiva debido a la facilidad de traspaso de límites geográficos utilizando de las redes de comunicación.
- Escaso coste de las acciones de ataque en relación con otros dominios de la delincuencia gracias a la alta capacidad de impacto con la utilización de pocos recursos técnicos y económicos.
- Alcance para todas las personas de manera sencilla gracias a la penetración de la tecnología y el acceso a internet.
- Construido a base de tecnologías inseguras en su diseño ocasionado principalmente por la evolución exponencial del sector.

La complejidad del medio así como sus vulnerabilidades, en su dimensión más amplia, se encuentran en constante evolución lo que implica la necesidad de desarrollar disciplinas que permitan implementar mecanismos de seguridad de una forma proactiva. Esta necesidad es la que origina el desarrollo de una nueva disciplina conocida como Ciberdefensa complementaria a la Ciberseguridad y de carácter proactivo.
La disciplina de Ciberdefensa se desarrolla en el ámbito de operación, lo que le hace susceptible de trabajar bajo mecanismos de normalización que optimicen su desarrollo. Esta disciplina se divide en distintas áreas, donde el análisis de *malware* es una de las más complejas y que requieren mayor desarrollo.

El análisis de *malware* tiene como objetivo analizar y obtener información de los diferentes códigos maliciosos que se desarrollan y ante los que los sistemas de información y comunicaciones están continuamente expuestos. La diversidad del código malicioso existente obliga a su clasificación. Dentro de esta clasificación, se identifica un tipo determinado de *malware* o código malicioso denominado Amenaza Persistente Avanzada, comúnmente conocido por su acrónimo APT (*Advanced Persistent Thread*).

Las APTs actualmente se presentan como los ataques más sofisticados a los que un sistema complejo puede estar expuesto, convirtiéndose en un verdadero desafío tanto para las organizaciones privadas como las administraciones públicas.

Las amenazas avanzadas persistentes requieren un grado de especialización y unos recursos que no están al alcance de todos. El factor que determina las capacidades necesarias para explotar esta metodología o tipología de ciberdelito es la finalidad que se persigue. Generalmente, este tipo de *malware* está relacionado con las infraestructuras y sectores estratégicos de un país lo que indica un interés tanto económico como político.

De acuerdo a los estudios existentes sobre el origen y desarrollo de APTs, la mayoría, por no decir la totalidad, concluyen la necesidad de un grupo de expertos con alta capacidad tecnológica, que acometen desarrollos de largo plazo y que se encuentran bien organizados.

La posibilidad de disponer de metodologías que garanticen la efectividad en su aplicación permitiendo el conocimiento de la estructura y el funcionamiento del *malware*, en detalle, aportará información de gran valor que permitirá en primer lugar el desarrollo de contramedidas eficaces y en segundo lugar, ampliar el conocimiento sobre el origen del código malicioso y sus vectores de ataques.

Estos y no otros son los principales elementos motivadores para el desarrollo del presente piloto experimental cuyo objetivo principal es la prueba empírica de la metodología desarrollada y propuesta por Don Javier Bermejo en su Tesis Doctoral [1].

Esta metodología se fundamente en cuatro procedimientos que se presentan a continuación:

- **Acciones iniciales**, basadas en la creación del escenario de prueba y obtención de la línea base de la víctima
- **Clasificación**, basado en la obtención de información inicial sobre la APT a analizar
- **Análisis estático y dinámico de código**.
- **Análisis de comportamiento** fundamentado en la ejecución y estudio del comportamiento del código malicioso en tiempo real.

El presente piloto experimental tiene como objetivo implementar un escenario de prueba virtual sobre el que se analizará, de acuerdo a la metodología propuesta por Don Javier Bermejo Higuera, el funcionamiento de una de las APTs más complejas y recientemente detectadas como es “Octubre Rojo”.

De origen presuntamente ruso y aunque se confirma que estaba operativo desde el año 2007, “Octubre Rojo” es una APT detectada en enero de 2013. Esta herramienta de código malicioso tiene como perfil objetivo las organizaciones gubernamentales y servicios diplomáticos. Su funcionamiento se centra en el robo de información de calidad con fines relacionados con la mejora de los servicios de inteligencia de los atacantes. Este código malicioso se desplegó principalmente en países de Europa del Este y Asia Central. El vector de ataque principal era el correo electrónico y su ejecución por parte de la víctima se traducía en la apertura de una puerta trasera en su equipo.

Tal y como comenta Don Javier Bermejo Higuera en su Tesis Doctoral [1], el proceso sistemático y metodológico de análisis de *malware* que desarrolla debe ser de carácter flexible, permitiendo adaptarlo a los distintos tipos de *malware* que existen así como a sus tecnologías de desarrollo. Gracias a esta piloto experimental, se contará con un caso adicional al expuesto en la Tesis utilizada como base teórica para este piloto y que a su vez, presenta un análisis empírico de la APT conocida como “Flame”.

Junto con el desarrollo de un nuevo caso de aplicación de esta metodología de análisis de código malicioso, se plantea un objetivo más ambicioso como es el contribuir a la mejora de dicha metodología identificando posibles áreas de mejora y/o evolución de la misma.

El presente trabajo de Fin de Máster cuenta con la siguiente estructura de contenidos

- **Capítulo 1: Introducción.** En este apartado se describe el marco en el que se circunscribe el presente trabajo, destacando los conceptos principales, el interés y los objetivos que se persiguen.
- **Capítulo 2: Contexto y Estado del arte**. En este apartado se describe qué es un APT, la relevancia que tiene dentro del campo de la seguridad de la información, los casos más importantes detectados y la justificación de la necesidad de una metodología que permita detectar y prevenir este tipo de amenaza
- **Capítulo 3: APT, "Octubre Rojo"**. En este apartado se presenta el APT utilizada como fuente para el piloto experimental. Esta información se complementa con datos generales relativos a su operación, alcance e impacto de la infección así como su evolución, con objetivo de mejorar la comprensión del lector es este tipo de amenazas con un caso real.
- **Capítulo 4: Descripción de la metodología**. En este apartado se describe de forma detallada la metodología utilizada, con el objetivo de que el lector, independientemente de su perfil, pueda comprender su alcance.
- **Capítulo 5: Objetivos**. En este apartado se describen los objetivos, de forma detallada, que se persiguen con la realización del presente piloto experimental.
- **Capítulo 6: Definición del escenario.** En este apartado se describe el laboratorio general para desarrollar un análisis y de *malware* y se particulariza en el utilizado para la realización de las pruebas.
- **Capítulo 7: Desarrollo de pruebas y resultados**. En este apartado se describen las pruebas de forma detallada, presentado información y documentación gráfica de su ejecución así como los resultados obtenidos
- **Capítulo 8: Conclusiones**. Una vez finalizadas las pruebas del piloto experimental y analizados los resultados, se presentan las conclusiones obtenidas, comparando dichos resultados con los objetivos iniciales.
- **Capítulo 9: Futuras líneas de desarrollo**. Este capítulo presenta aquellas líneas que, identificándose de interés para su desarrollo, podrían ser utilizadas como base para futuros trabajos en este campo.

2. Contexto y Estado del Arte

Tal y como se ha comentado en la introducción, la seguridad de los sistemas de información es una de las mayores prioridades de los departamentos de seguridad en estos momentos. El ciberespacio se ha convertido en un entorno cuasi ideal para la comisión de delitos, comúnmente tildados como delitos informáticos. Cierto es, que este tipo de prácticas ha sufrido una evolución directamente relacionada con el objetivo en su comisión. Si bien inicialmente se fundamentaban en una búsqueda de notoriedad o simples actos puntuales de protesta, en la actualidad se ha añadido, principalmente, un interés económico. Este factor es el principal garante de la evolución de los tipos de ataques y la explotación de las características del ciberespacio que ponen de manifiesto las vulnerabilidades de las víctimas.

Desde el punto de vista de la persecución de este tipo de actividades, se realiza una clasificación básica por la cual se identifican actividades basadas en el uso de las tecnologías de la información para la comisión de delitos previamente existentes o tradicionales (estafas como *phising* o *smishing*) y actividades basadas en el ataque contra los propios sistemas TI centrados en dañar su confidencialidad, integridad y disponibilidad.

Es en este segundo grupo en el que se centra el desarrollo de este piloto experimental. El desarrollo de código malicioso, comúnmente llamado *malware*, ha sufrido una evolución exponencial que obliga la búsqueda continua de soluciones de seguridad y la mejora de las ya existentes. Tal es esta carrera que, como se ha mencionado con anterioridad, se ha desarrollado una disciplina nueva y complementaria a la Ciberseguridad, conocida como Ciberdefensa.

Para poder comprender la importancia del desarrollo de técnicas de análisis de *malware* y, en particular, metodologías para analizar el tipo de *malware* más complejo identificado actualmente y denominado como Amenaza Persistente Avanzada (APT), es necesario previamente y de forma sencilla comprender y modelar en mundo del *malware* tal y como se realiza en este apartado.

2.1 Definición de *Malware*

En primer lugar y antes de analizar de forma detallada las características de las APTs, es importante comenzar definiendo el concepto de *malware* tal y como se presenta en la siguiente definición: “Término genérico utilizado para referirse a cualquier tipo de software malicioso o molesto que puede instalarse en los sistemas informáticos para llevar a cabo acciones sin el conocimiento del usuario” [2].

De acuerdo a la anterior definición, puede afirmarse que *malware* es cualquier programa o código malicioso que se ejecuta en un sistema informático sin conocimiento y permiso del usuario, provocando algún tipo de perjuicio. Estos perjuicios pueden agruparse en daños, deterioro, alteración, supresión o inaccesibilidad de los archivos junto con el acceso no autorizado a la máquina y sus recursos.

La identificación de las acciones que lleva a cabo un código malicioso es fundamental para poder detectarlo y combatirlo. A continuación se listan [3] aquellas acciones que la máquina víctima suele sufrir por la acción de *malware.*

- Envío de correo a gran escala.
- Eliminación de archivos, ocasionando problemas en el arranque y correcto funcionamiento del sistema.
- Modificación de archivos.
- Modificación de claves del registro u otro tipo de datos de configuración, actuando sobre otras aplicaciones, como por ejemplo anti-virus o firewall, o buscando una escalada de privilegios para una mayor explotación y uso no autorizado de los recursos.
- Degradación del rendimiento general del sistema pudiendo afectar a la ejecución de todo tipo de procesos como por ejemplo las comunicaciones.
- Inestabilidad del sistema.
- Robo de información corporativa y confidencial.
- Modificación de la configuración de seguridad.

Si bien las anteriores acciones están centradas en equipos informáticos tradicionales (servidores, workstations, equipos portátiles,…), en la actualidad es imprescindible tener en cuenta la expansión que están sufriendo los terminales móviles, principalmente Smartphone y tabletas.

Esta evolución ha abierto un nuevo campo de explotación de *malware* para los ciberdelincuentes, centrándose en la búsqueda de los siguientes resultados.

- Envío de mensajes de texto a servicios de tarificación adicional.
- Captura de información bancaria procesada desde los terminales móviles.
- Robo de información particular (contactos, fotografías, cuentas de usuario).
- Secuestro virtual del terminal.
- Control remoto y unión a *redes botnet*[1].

Las primeras identificaciones de *malware* datan de finales de la década de los 90 con "Melisa" y principios del año 2000 con la detección del *malware* "LoveLetter", ambos basados en el correo electrónico como medio de propagación y un archivo adjunto infectado como vector de ataque. Desde esa época hasta la actualidad, el desarrollo ha sido exponencial y se calcula que actualmente existe *malware* en un orden de magnitud de millones. A continuación se muestra un gráfico publicado por la empresa Microsoft[2] en el informe [4] en el que se muestra la evolución del *malware* entre los años 1991 y 2011

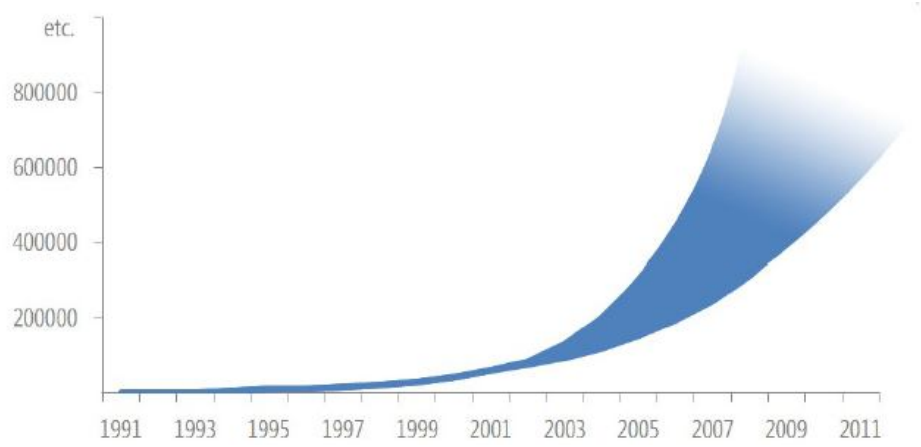


Ilustración 1 Crecimiento aproximado de *Malware* 1991-2001 [4]

Es importante destacar que la evolución del *malware* está directamente asociada con la mayor utilización de los sistemas de información en el desarrollo de la mayoría de las actividades de la sociedad. Por ello, no debe olvidarse que este crecimiento se deriva de un incremento de vulnerabilidades explotables en las herramientas disponibles y mayormente utilizadas. El mismo informe [4] muestra la evolución y distribución de las vulnerabilidades en la misma escala de tiempo, tal y como se muestra en la siguientes imágenes.

[1] http://es.wikipedia.org/wiki/Botnet
[2] http://www.microsoft.com/es-es/default.aspx

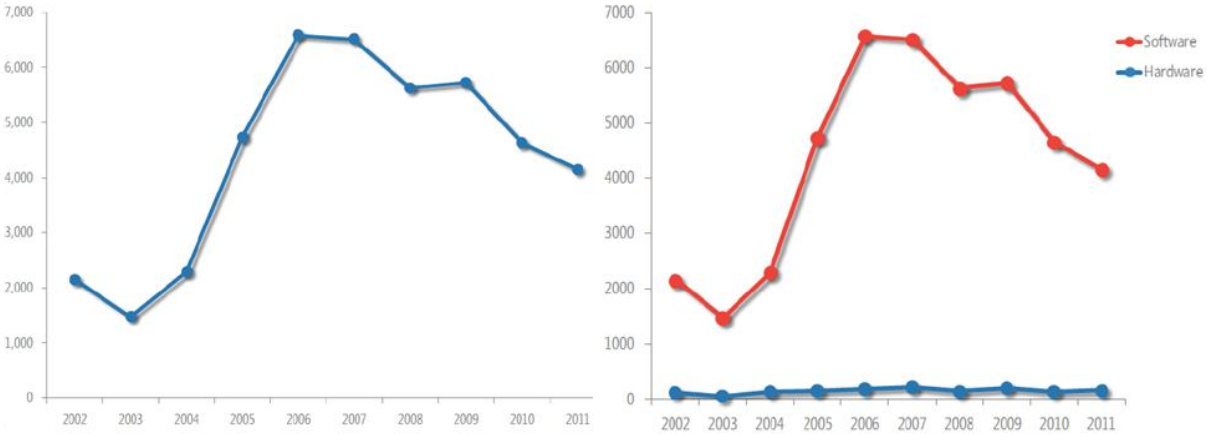


Ilustración 2 Evolución de las vulnerabilidades de la industria [4]

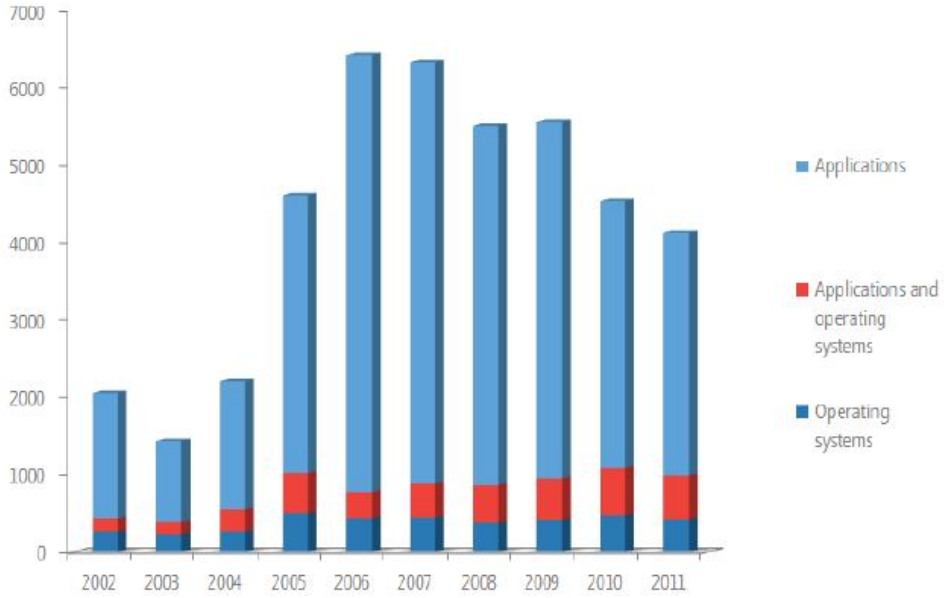


Ilustración 3 Distribución de las vulnerabilidades SW [4]

Las anteriores gráficas nos permiten identificar la ventana de oportunidad que el desarrollo de *malware* ofrece a los ciberdelincuentes, así como los sus principales objetivos que no son otros que las aplicaciones SW, reforzando el interés en el desarrollo de metodologías de análisis de *malware* y en particular de sus variantes más peligrosas.

2.2 Tipos de *malware*

El concepto de *malware*, como cabe esperar tras su definición, abarca distintos tipos de códigos maliciosos. Su clasificación es importante desde el punto de vista de su defensa ya que permite establecer un grado de coherencia entre todos los *stakeholders* implicados en el desarrollo de soluciones de seguridad. Desde el punto de vista teórico es sencillo respaldar la anterior afirmación, sin embargo, desde el punto de vista práctico, la clasificación se torna complicada debido a la rápida evolución y modificación a la que continuamente está sometida.

A continuación se realiza una breve enumeración de los tipos de *malware* que servirá como introducción al capítulo específico sobre el *malware* conocido como APT, núcleo del desarrollo de este trabajo. En el sector de la seguridad informática y en particular en el tratamiento de código malicioso, la empresa Kaspersky Lab[3] es un referente que se autodefine como el mayor operador de seguridad IT del mundo. Kaspersky Lab realiza una clasificación del *malware* en 10 familias tal y como se muestra a continuación.

- **Virus clásicos**. Programas que infectan a otros programas añadiendo o modificando su código para tomar el control tras la infección. El objetivo principal de un virus es infectar.
- **Gusanos de red**. Este tipo de *malware* usa los recursos de red para distribuirse. Su está relacionado con la similitud con el animal, en su capacidad de penetración a un equipo. La penetración se realiza por medio del correo electrónico, sistemas de mensajes instantáneos, redes de archivos compartidos (P2P) o canales IRC entre otros. Su velocidad de propagación es muy alta, superior a la de los virus.
- **Troyanos**. Esta clase de programas maliciosos incluye una gran variedad de programas que efectúan acciones sin que el usuario se dé cuenta y sin su consentimiento. Recolectan datos para enviarlos a los ciberdelicuentes. También pueden destruir o alterar información con intenciones delictivas, causando desperfectos en el funcionamiento del ordenador. Pueden hacer uso de los recursos del equipo infectado para fines criminales, como envíos masivos de correo electrónico. No son virus clásicos porque no infectan otros programas. Los troyanos no pueden penetrar a los equipos por sí mismos, sino que se propagan utilizando otras aplicaciones. Son nivel de peligrosidad es mayor que la de los virus clásicos.

[3] http://www.kaspersky.es

- ***Spyware***. Software que permite recolectar la información sobre un usuario/organización de forma no autorizada. Su presencia puede ser completamente invisible para el usuario.
- ***Adware***. Su objetivo es la recolección y envío de datos personales del usuario. La mayoría de programas *adware* son instalados mediante software de distribución gratuito.
- ***Riskware***. No son programas maliciosos pero contienen una amenaza potencial. En ciertas situaciones ponen los datos en peligro. Un ejemplo pueden ser los programas de administración remota.
- **Bromas**. Este grupo incluye programas que no causan ningún daño directo a los equipos que infectan aunque muestran advertencias falsas sobre supuestos daños ocurridos o por ocurrir.
- ***Rootkits***. Un *rootkit* es una colección de programas usados por un hacker para evitar ser detectado mientras busca obtener acceso no autorizado a un ordenador. Esto se logra de dos formas: reemplazando archivos o bibliotecas del sistema o instalando un módulo de *kernel*[4].
- ***Spam***. Los mensajes no solicitados de remitente desconocido enviados en cantidades masivas de carácter publicitario, político o de propaganda entre otros.
- **Otros programas maliciosos**. Son una serie de programas que no afectan directamente a los ordenadores pero que se usan para crear virus, troyanos o para realizar actividades ilegales como ataques DoS y penetrar en otros ordenadores.

Estas 11 familias recogen casi la totalidad del *malware* clasificado aunque sea interesante ampliar esta lista tal y como recoge Don Javier Bermejo en su Tesis Doctoral [1].

- **Puertas Traseras o *Backdoors***. Software malicioso que crea un canal de entrada que permite al ciberatacante conectarse a la máquina víctima, controlarla, espiar e instalar otros tipos de *malware*.
- ***Keyloggers***. Programas cuyo objetivo es la captura del uso del teclado por parte del usuario, permitiendo registrar y enviar las teclas utilizadas en cualquier proceso de entrada de datos por parte del usuario.

[4] http://es.wikipedia.org/wiki/N%C3%BAcleo_%28inform%C3%A1tica%29

- ***Botnet***. Código malicioso que permite que la víctima sea controlada remotamente por un sistema de mando y control. El ciberdelincuente puede concentrar equipos infectados pudiendo crear redes amplias de gran capacidad para cometer acciones maliciosas como el envío de *spam*, ataques distribuidos de denegación de servicio (*DDoS*[5]) o el robo de información de los propios equipos.
- **Bomba Lógica**. Código que se utiliza para que, en unión con otras piezas de código malicioso, poder llevar a cabo actividad en periodos de tiempo determinados, garantizando un periodo de residencia en la víctima con pocas probabilidades de detección.
- ***Ransomware***. *Malware* cuyo objetivo es el chantaje del usuario mediante el bloqueo de algún recurso del equipo.
- ***Rogueware o Scareware***. Código que se sirve de la ingeniería social para instalar el *malware* y obtener información del usuario.
- ***Advanced Persistent Threat o APT***. *malware* definido como amenaza avanzada persistente, se desarrolla en la siguiente sección.

Para concluir este apartado se considera importante reforzar la clasificación de los tipos de *malware* con su presencia y evolución en la actualidad. Para poder desarrollar esta información, se utiliza como referencia el informe [5], emitido anualmente por la empresa internacional de origen español, Panda Security[6].

Este informe arroja datos tan interesantes como que durante el año 2013, han aparecido 30 millones de nuevas muestras de *malware* lo que equivale a 82.000 muestras diarias. PandaLabs afirma tener registradas más de 145millones de muestras de *malware* y estima que, teniendo en cuenta la tendencia creciente, el 20% de todo el *malware* que ha existido en la historia, se desarrolló en 2013.

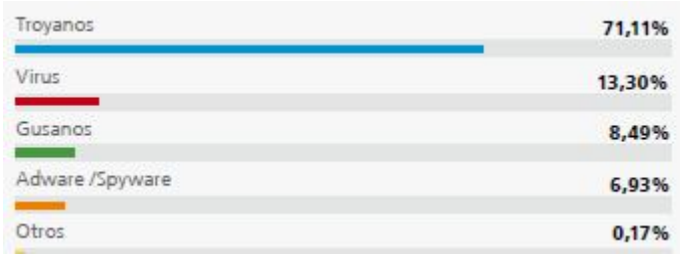


Ilustración 4 *Malware* creado en 2013 [5]

[5] http://es.wikipedia.org/wiki/Ataque_de_denegaci%C3%B3n_de_servicio
[6] http://www.pandasecurity.com

Desde el punto de vista de la Ciberseguridad y Ciberdefensa es importante disponer de información sobre la distribución de las infecciones por tipos de *malware* así como su distribución geográfica. El informe desarrollado por Panda Security ofrece los siguientes resultados, de alto interés para el desarrollo de este trabajo teniendo en cuenta una posterior aplicación o desarrollo de este tipo de metodologías en función de la sensibilidad real que pueda existir.

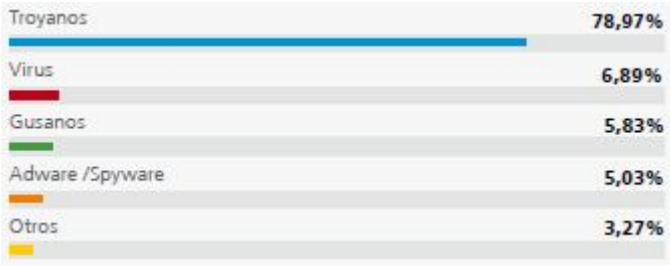


Ilustración 5 Infección por tipo de *malware* en 2013 [5]

Desde una perspectiva geográfica, se afirma que el porcentaje global de ordenadores infectados en el mundo es de 31,53% no siendo lineal su distribución. Los países en los existe un mayor porcentaje de infección se muestran en la siguiente ilustración. Siendo Asia y Latinoamérica las regiones con mayores infecciones, el resto de países que se encuentra por encima de la media mundial son Uruguay (33,64%), Chile (33,51%), España (32,72%) y Colombia (32,22%).

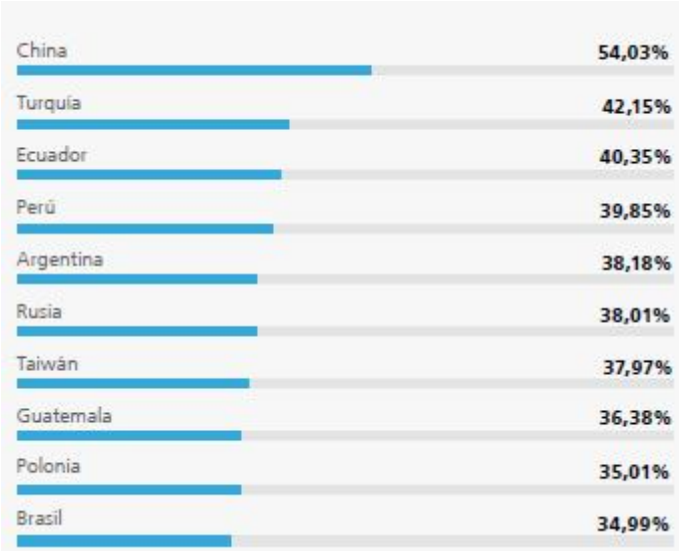


Ilustración 6 Países con mayor índice de infección [5]

Los países con menor porcentaje de infección se muestran en la siguiente ilustración del mismo modo que se ha realizado en el supuesto anterior. Para completar la lista de países con un inferior porcentaje de infección a la media mundial, habría que añadir Portugal (25,28%), Francia (25,68%), Australia (26,84%), Austria (27,69%), Canadá (27,82%), Estados Unidos (28,96%), Venezuela (29,83%), Hungría (30,96%), México (31,00%), Italia (31,47%) y Costa Rica (31,50%).

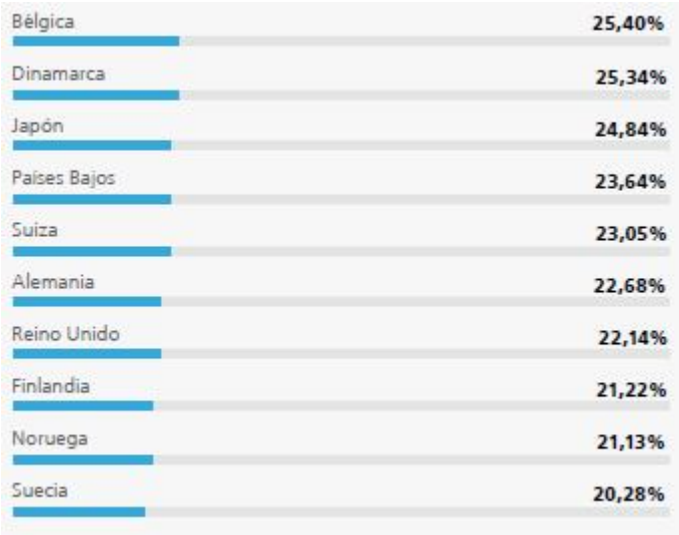


Ilustración 7 Países con menor índice de infección [5]

Este informe también pone de manifiesto el protagonismo que han tenido los diferentes gobiernos por sus operaciones de ciberespionaje a nivel mundial, algo poco usual hasta la fecha y donde China ha liderado esta posición. Es importante tener en cuenta, que el *malware* conocido como APT, tal y como se comenta en el apartado siguiente, requiere de altos recursos para su desarrollo, lo que lo relaciona directamente con los estados y grandes organizaciones. De acuerdo a la información que Panda Security muestra en su informe anual, el incremento de actividad de los estados en la lucha del ciberespionaje posiblemente tenga asociado un desarrollo de este tipo de *malware*.

2.3 APT: Advanced Persistent Thread

Hasta el presente apartado se han mencionado y descrito las principales características de los *malware* más comunes. Sin embargo, la evolución del *malware* tuvo un punto de inflexión en el momento en el que se empezaron a implementar técnicas avanzadas capaces de generar códigos polimórficos.

Este hito es uno de los factores fundamentales en el desarrollo de un nuevo tipo de *malware*, conocido como Amenaza Avanzada Persistente (APT), de alta complejidad cuyo objetivo es ser utilizado en un ataque sofisticado a un objetivo concreto y en un periodo dilatado en el tiempo. Como se observa en la anterior afirmación, las principales características que definen una APT son su complejidad, la persistencia en el tiempo y el haber sido desarrollado para un objetivo concreto.

Desde el punto de vista del atacante, aunque se trata de *malware* de alta complejidad, es importante destacar que los vectores de ataque no difieren en gran medida de los utilizados por otros tipos de *malware*, basándose en la explotación de vulnerabilidades de los sistemas TIC de la organización objetivo o en la explotación de vulnerabilidades humanas mediante técnicas de ingeniería social. Este tipo de *malware* está principalmente orientado al ciberespionaje o robo de información privilegiada para su posterior comercialización en el mercado negro, aunque no hay que olvidar que pueden existir amenazas orientadas al control de sistemas críticos de la víctima, como sistemas de generación de energía, sistemas de comunicaciones principales o redes de distribución.

Desde el punto de vista de la víctima, inicialmente se consideraban únicamente las infraestructuras críticas como potenciales víctimas, sin embargo, con la evolución y la mayor utilización de estas técnicas, se confirma que los objetivos son mucho más amplios que el sector anteriormente mencionado. En el informe [6], publicado por la empresa Symantec[7], se muestra la distribución en porcentaje de los objetivos más buscados por parte de las organizaciones criminales durante 2012.

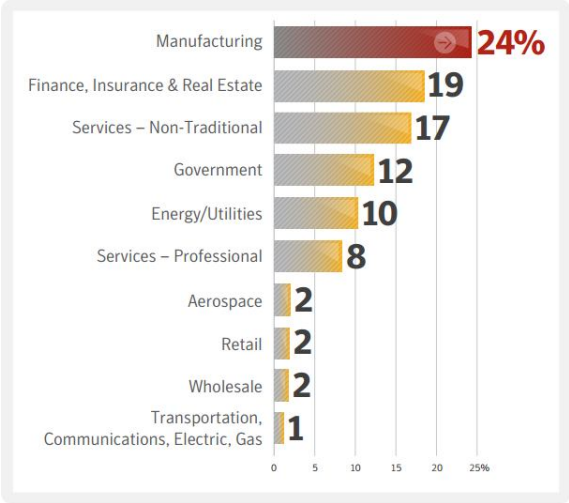


Ilustración 8 Top 10 Industrias atacadas en 2012 [6]

[7] http://www.symantec.com

Tal y como se define en la Tesis Doctoral [1], los principales objetivos de este tipo de ataque son los siguientes:

- **Políticos**. Relacionados con el mantenimiento de la estabilidad interna de un país.
- **Económicos**. Centrados en el robo de propiedad intelectual e industrial.
- **Técnicos**. Centrados en el acceso a desarrollos de seguridad y medidas defensivas con el fin de evadirlas o interrumpirlas.
- **Militares**. Centrados en la identificación de puntos débiles de la víctima.

2.3.1 APT: Características

En la introducción a las Amenazas Persistentes Avanzadas se han mencionado sus principales características que ahora son desarrolladas en este apartado.

- **Amenaza Dirigida**. Amenazas que se desarrollan específicamente para un objetivo concreto. Esto implica un estudio previo de la víctima y un alto conocimiento de sus sistemas y potenciales vulnerabilidades. Esta característica conlleva que una APT no pueda ser utilizada, en todos los casos, en objetivos diferentes, reduciendo una posible economía de escala y poniendo de manifiesto la necesidad de de un análisis inicial, profundo, del ratio coste/beneficio por parte del ciberatacante.
- **Amenaza Avanzada**. Son códigos altamente complejos con desarrollos específicos y exclusivos que permiten llevar a cabo ataques muy avanzados. Para su desarrollo es necesario contar con grandes recursos, no sólo económicos, si no tecnológicos. Desde el punto de vista de la materialización del ataque, se suelen utilizar vulnerabilidades todavía no publicadas y obtenidas en el mercado negro, se valen de métodos de ingeniería social y cualquier otro tipo de medida fuera del alcance del ciberatacante convencional.
- **Amenaza Persistente**. En línea con los recursos dedicados para el desarrollo de este tipo de amenazas, el ciberatacante, una vez logrado el acceso a los sistemas de la víctima, tendrá como objetivo mantener el *malware* durante un periodo de tiempo prolongado. Para ello se sirven de técnicas de desarrollo que dotan al código de unas cualidades polimórficas, aumentando la dificultad en la correlación de las firmas digitales sobre las que trabajan la mayoría de las medidas de seguridad existentes, dificultando su detección. Junto con lo anterior, los desarrolladores son capaces de mantener un perfil bajo durante un espacio de tiempo prolongado hasta la comisión del ataque, incrementando, una vez más, la dificultad su detección.

2.3.2 APT: El ataque

Para poder analizar en profundidad una Amenaza Persistente Avanzada es necesario comprender cómo se caracteriza un ataque de este tipo. Para ello, en primer lugar es necesario identificar los posibles métodos de ataque y complementarlo con un modelo que represente sus principales fases.

En relación a los métodos de ataque utilizados por una APT, cabe destacar que se suelen utilizar múltiples vectores de ataque tanto de forma simultánea como escalada en el tiempo. No todos los vectores de ataque utilizados tienen la misma función, si no que se combinan los más efectivos con otros a modo de señuelo, dificultando la labor de los equipos de defensa. En el momento de la ejecución de un ataque, los objetivos también son diversos, como en el caso de la APT Stuxnet[8], una de las más conocidas, la cual ataca de forma simultánea a los sistemas TIC, usuarios y administradores.

Dentro de los métodos de ataque debe realizarse una mención especial a las técnicas de ingeniería social que son ampliamente utilizadas para introducir el *malware* en la red objetivo, evitando las defensas perimetrales que puedan existir. Así mismo, explotando técnicas de ingeniería social es posible obtener información muy valiosa sobre contactos, direcciones de correo o ubicaciones geográficas dispersas, permitiendo al atacante diseñar su ataque con mayor efectividad.

Tras presentar, de manera general, los distintos métodos de ataque utilizados cuando se trata de APTs, se torna necesario representar las fases del mismo mediante un modelo. En este caso se describe el modelo del ciclo de vida de una APT que muestra la empresa Solutionary[9] en el informe [7] y que se basa en 7 fases diferenciadas.

- **Primera Fase, Planificación y Obtención de información**. Durante esta fase los esfuerzos se concentran en obtener el máximo de información del objetivo. Para ello clasifican la información en pública, obtenida mediante técnicas de *footprinting*[10], y clasificada, obtenida generalmente utilizando técnicas de ingeniería social. Se trata de un proceso metódico y continuado en el tiempo que permite establecer patrones de comportamiento de los usuarios y otros recursos de la víctima. Facilita la identificación de las condiciones óptimas para desplegar el *malware*.

[8] http://es.wikipedia.org/wiki/Stuxnet
[9] http://www.solutionary.com/
[10] Término utilizado para describir el proceso de adquisición de información no intrusivo de los sistemas TIC de una organización

- **Segunda Fase, Intrusión en los sistemas**. Tras la fase anterior, se dispone de todo lo necesario para iniciar el ataque. En esta fase, el objetivo es penetrar en los sistemas de la víctima para poder introducir el *malware*. Esta fase se puede desarrollar de dos formas, una puramente técnica basada en la explotación de alguna vulnerabilidad y rotura de la defensa perimetral de la víctima, y otra, basada en la explotación de la información obtenida en la fase anterior para hacer uso de técnicas de ingeniería social y burlar los mecanismos de defensa perimetral existentes.

- **Tercera Fase, Establecer *Backdoor***. Esta fase es una de las más importantes ya que tiene como objetivo establecer un canal de comunicación entre la APT y el exterior con el objeto de disponer de un medio para exportar la información usurpada y mantener el *malware* siempre actualizado. Para ello, es necesario que el código malicioso sea capaz de adquirir privilegios en el sistema que le permitan activar esta comunicación. Un APT está programado para que se conecte con un sistema de Mando y Control (C&C), con el que mantendrá una comunicación, muchas veces bidireccional, durante todo el proceso de infección. Para garantizar una comunicación segura, el *malware* utiliza técnicas de cifrado y ofuscación.

- **Cuarta Fase, Robo de credenciales**. El uso de los privilegios obtenidos en la anterior fase de forma continuada podría aumentar las probabilidades de detección, por ello, el objetivo de esta fase es adquirir credenciales de usuarios legítimos para llevar a cabo acciones con sus credenciales minimizando las probabilidades de ser detectado.

- **Quinta Fase, Infección y adquisición de información**. Durante las cuatro fases anteriores, el APT construye el entorno de operación desde el que pueda lograr sus objetivos. En esta quinta fase, el *malware* comienza a corromper los sistemas, instalando código malicioso, como *Keyloggers* o *sniffers*[11], que le permita seguir obteniendo información. Este proceso suele realizarse de forma escalada, iniciando la infección en equipos de usuarios hasta llegar a comprometer los servidores principales.

[11] http://es.wikipedia.org/wiki/Analizador_de_paquetes

- **Sexta Fase, Extracción de información**. Una vez el sistema completo de la víctima está comprometido y el *malware* ha sido capaz de capturar la información requerida, es el momento de iniciar el proceso de extracción de la misma. Este proceso debe llevarse a cabo de manera transparente. Para ello el *malware* hace uso de servicios comunes como peticiones DNS o HTTPS, de modo que, modificando estos paquetes y añadiéndoles fragmentos de información robada, es capaz de enviarla al exterior. Tal y como se confirma en el informe [7], en ataques avanzados se ha detectado la utilización de técnicas de esteganografía, consistente en la ocultación de información en archivos comunes.
- **Séptima Fase, Mantenimiento de la persistencia**. Independientemente de que el modelo la considere como la última fase, ésta representa un proceso continuo que desarrolla el APT para garantizar su permanencia en el sistema infectado, evitando ser detectado. En esta fase se contemplan acciones como la modificación de los canales de comunicación con el exterior, la mutación del *malware* para modificar sus firmas digitales o el proceso continuo de infección de nuevos equipos.

A continuación se muestra de forma gráfica las siete fases del ciclo de vida de un APT.

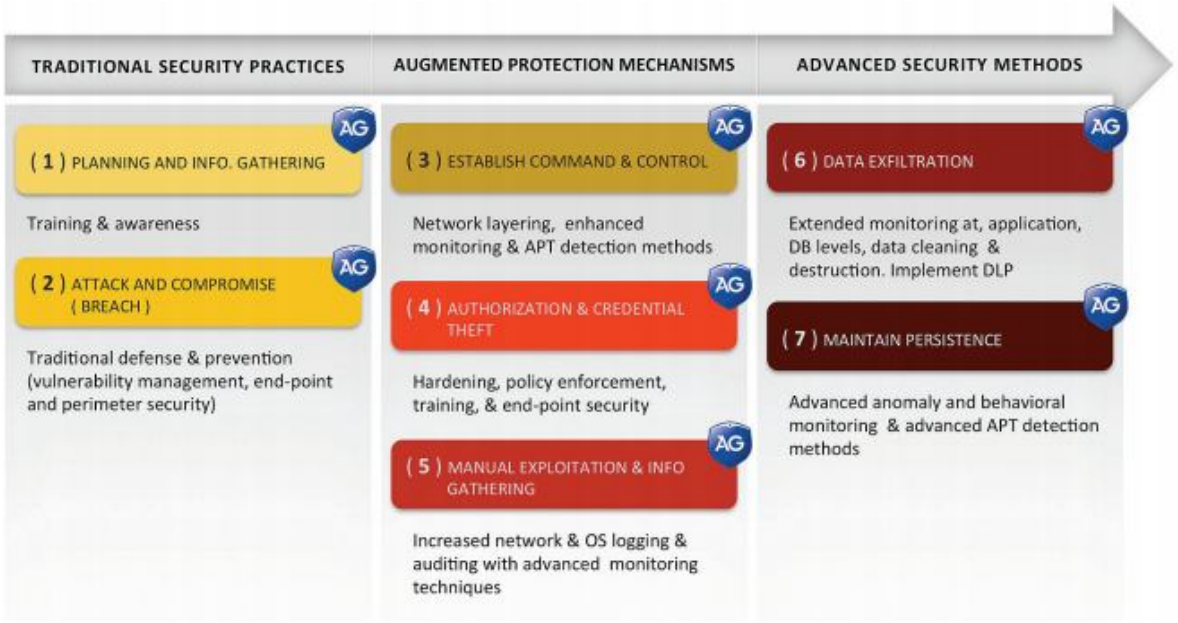


Ilustración 9 Ciclo de vida de un APT [7]

2.3.3 APT: La defensa

Para completar el objetivo de comprender las características y comportamientos de un APT de modo que pueda desarrollarse una lectura comprensiva del presente trabajo, no sólo es necesario mostrar las características de este tipo de *malware* desde el punto de vista del ataque, si no que es imprescindible añadir el punto de vista defensivo.

En este apartado se van a presentar las recomendaciones, más comunes, en materia de defensa contra APTs. Es en este punto donde se vuelve a poner de manifiesto la importancia del desarrollo de metodología de análisis de este tipo de *malware* ya que, como se ha comentado anteriormente, permite la ampliación de conocimiento y su distribución de forma ordenada, garantizando un desarrollo efectivo en los mecanismos de defensa.

Para mantener la coherencia con el apartado anterior, se utiliza como base el informe [7] anteriormente utilizado. En el informe se presentan de forma priorizada y ordenada los controles de ciberseguridad que el Instituto SANS[12] define como críticos en una sección de su página web[13]. Aunque se afirma que cada compañía, en función de la probabilidad de ser víctima de un ataque APT, debe establecer su estrategia particular de seguridad con mayor o menor granularidad en la aplicación de controles, se muestra una línea base que permite definir una estrategia básica de prevención aplicable a la mayoría de las organizaciones.

- **Identificación de activos críticos** (SANS CSC #1 and #2). Para poder definir e implementar una estrategia de defensa es fundamental conocer los activos a proteger, no únicamente los tecnológicos, si no aquellos que también son críticos desde el punto de vista de impacto financiero y otros intangibles como la propia información.
- **Formación en Seguridad** (SANS CSC #20). Tal y como se ha comentado durante los anteriores apartados del presente documento, uno de los vectores de ataque más utilizados es la ingeniería social, cuyo objetivo no es otro si no explotar las vulnerabilidades humanas. Para ello, la formación del personal en este tipo de técnicas, como la utilización del correo electrónico o procedimientos de actuación, es muy importante. Dentro del proceso de formación será necesario clasificar los empleados en función de los privilegios de acceso y autorización que dispongan.

[12] http://www.sans.org/
[13] https://www.sans.org/critical-security-controls/

- **Test continuos de seguridad** (SANS CSC #4, #6, #7). Una actividad proactiva en la revisión de las medidas de seguridad permitirá disponer de los sistemas preparados ante un ataque. Para ello será necesario implementar procesos de auditoría, simulaciones de ataques de penetración o análisis de vulnerabilidades tanto de plataformas software como de las propias aplicaciones. Estos controles permitirán detectar situaciones de riesgo e implantar medidas para su tratamiento previas a la explotación por parte de un atacante.
- **Uso apropiado de los sistemas críticos** (SANS CSC #12). La correcta identificación y clasificación de los sistemas críticos permitirá definir políticas de seguridad con sus correspondientes procedimientos de uso de modo que se evite la exposición innecesaria de dichos sistemas. Desde un exhaustivo control del acceso y autorización de los usuarios, gracias a una definición de roles, como el aislamiento de este tipo de sistemas de redes o servicios vulnerables, garantizará una mayor seguridad global.
- ***Hardening* y actualización de parches** (SANS CSC #3 and #4). En un punto anterior se ha hecho referencia a la utilización de la ingeniería social por parte de los atacantes, sin embargo no hay que olvidar que uno de las bases para el éxito, de un ataque de este tipo, es la explotación de vulnerabilidades en los sistemas. Para disminuir la superficie de ataque del sistema global se recomienda la implantación de una política de actualización de los sistemas, mediante procedimientos continuos y estables, junto con una configuración robusta (bastionado) de aquellos sistemas considerados como críticos.
- **Anti-virus / *Anti-Spyware*** (SANS CSC #5). Aunque previamente se haya mencionado que las técnicas de desarrollo de APT permiten desarrollar código malicioso difícilmente detectable por sistemas antivirus, no debe interpretarse esto como una sugerencia a la no utilización de estas herramientas. La detección continua de *malware* por parte de las principales empresas de desarrollo de soluciones de antivirus y la actualización continua de sus bases de datos, convierte a estas herramientas en un aliado perfectamente válido que, aun no siendo eficaz en los ataques más sofisticados, puede resultar efectivo contra amenazas ya detectadas. Este tipo de herramientas deben ser utilizadas en toda su extensión, cubriendo desde los equipos de usuario, las comunicaciones hacia el exterior y el control de servicios como el correo electrónico.

- **Cuentas de usuarios y control de privilegios** (SANS CSC #12, #15). Esta es una de las medidas más simples y obvias que deben implantarse. Se centra en la definición y clasificación de los tipos de usuarios, la asignación de permisos de accesos y autorización a los recursos y su posterior control y mantenimiento. El proceso de creación de usuarios no suele ser problemático, sin embargo, el proceso de eliminación o bloqueo de una cuenta no suele estar bien definido en todas las situaciones. Este hecho se traduce en la existencia de usuarios con privilegios que no deberían existir y que podrían ser utilizados por el *malware* para ejecutar acciones dentro del sistema.
- **Control y monitorización de logs** (SANS CSC #14). Los sistemas TI son generadores de grandes volúmenes de información, parte de ella de gran interés para las labores de auditoría. Es importante que, desde la fase del diseño del sistema, se determine la información a ser almacena así como su representación. Esta información deberá ser revisada periódicamente, aplicando distintos niveles de auditoría, de modo que sea posible identificar situaciones de riesgo. La implantación de distintos tipos de auditoría en función del tiempo es muy importante ya que, en el caso de las APTs, son ataques basados en la persistencia de modo que un análisis en un periodo corto de tiempo puede no ser efectivo a la hora de detectar la infección.

En el Anexo A se muestra una diagrama obtenido del informe [7], tomado como fuente, en el que se representan las fases anteriormente descritas.

2.4 Evolución y desarrollo de las APTs

El presente apartado tiene como objeto mostrar la evolución del *malware* APT desde su aparición hasta la actualidad. Para ello, se presenta una línea temporal con los principales hitos que demuestran que este tipo de amenaza está en constante evolución y cada vez cuenta con más adeptos. La profesionalidad de los grupos que desarrollan este tipo de código malicioso se confirma con la existencia de una lista, de los más activos, publicada por la empresa Mandiant[14], compañía que lleva a cabo estudios detallados en esta materia.

En primer lugar se va a presentar de forma gráfica una línea temporal con los hitos más recientes presentados en el informe [6] de la compañía Symantec. La siguiente imagen permite generar una idea rápida del desarrollo de las APTs en los últimos tiempos.

[14] https://www.mandiant.com/

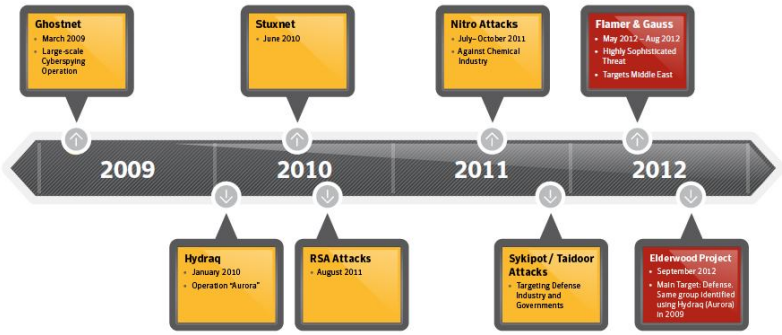


Ilustración 10 Ataques más recientes APT [6]

Observando la continua evolución de este tipo de ataques, cobra sentido establecer una línea temporal que permita comprender el origen y evolución de las APTs. Tomando como referencia los documentos [8] y el [9], elaborado, por la empresa Fortinet[15], junto con la documentación mostrada en la Tesis Doctoral de Don Javier Bermejo [1], se presentan los principales incidentes relacionados con ataques basados en APTs. Es importante destacar que este tipo de información no es fácil recopilarla ya que en muchas ocasiones son las víctimas las que deciden no publicar estos datos para evitar el posible impacto que pueda tener en su negocio así como para evitar facilitar información a los propios atacantes.

- **Años 1998-2000**. "Moonlight Maze", cuyos objetivos fueron el Pentágono, NASA, the US Energy Department, laboratorios de investigación y universidades privadas. El ataque tuvo éxito permitiendo el acceso a decenas de miles de archivos de información sensible. (Arquila, 2003) (Central Intelligence Agency, 2007)
- **Año 2003**. "Titan Rain", conjunto de ataques que lograron la intrusión en las redes de Department of Defense de los EEUU, la NASA y empresas de defensa. Este tipo ataque basado en *malware* de alta complejidad dio origen al nombre "Advanced Persistent Threat (APT)".
- **Año 2006**. Las redes oficiales de dos congresistas fueron comprometidas. Se cree que información sobre disidentes críticos del régimen de Beijing (China) fue sustraída. (The Washington Times, 2008)
- **Año 2007**. "Oak Ridge National Laboratory". La utilización de técnicas de ingeniería social basadas en el uso de correos electrónicos presuntamente legítimo permitieron a los atacantes accedes a información sensible. Aunque la detección se realizó en 2012, la actividad de la campaña "Red October" se inició en 2007. Se ampliará información sobre esta APT en los siguientes capítulos.

[15] http://www.fortinet.com/

- **Año 2008**. "Buckshot Yankee" *malware* del tipo APT cuyo vector de ataque eran las memorias tipo USB. Desde el punto de vista operativo, implicó un proceso de 14 meses de desinfección. Este ataque se tradujo en la prohibición del uso de memorias extraíbles en el Departamento de Defensa de EEUU y aceleró la creación de un centro de mando de ciberdefensa denominado USCYBERCOM[16].

- **Año 2009**. "Operación Aurora". Google denunció y publicó el haber sido víctima de un ataque de alta complejidad durante más de un año. La empresa americana decidió compartir abiertamente información con la National Security Agency (NSA) de los EEUU.

  Se llevó a cabo un ataque que infectó más de 1200 equipo en 103 países incluyendo diversas embajadas del sudeste asiático con el objetivo de obtener información privilegiada. Este ataque se denominó "GhostNet"

  Se detectó una serie de ataques a las empresas de sector petrolero que se denominaron "Night Dragon" y basaban su éxito en la explotación de las vulnerabilidades de sistemas operativos de Microsoft mediante el uso de ingeniería social aplicada en el correo electrónico.

- **Año 2010** "Stuxnet". *Malware* orientado al control de los sistemas de control del tipo SCADA, utilizados en la industria para el control automático de diversos procesos productivos. En este caso se la infección afectó a varias industrias con especial impacto en el sector nuclear en Irán. El *malware* permitía una reprogramación remota de los sistemas de control pudiendo alterar los procesos habituales de funcionamiento.

  Este mismo año el gobierno francés sufrió un ataque mediante, una vez más, métodos de ingeniería social basada en una campaña de correo electrónico. Este ataque infectó alrededor de 150 equipos del Ministerio de Economía. Se confirma que fueron controlados remotamente y que se extrajo documentación durante meses, incluyendo información del G20 y otros asuntos relacionados con la política exterior.

- **Año 2011**. Año de gran actividad desde el punto de vista de los ataques avanzados basados en *malware*. El ataque perpetrado contra el Fondo Monetario Internacional (FMI) permitió el acceso a información sensible de carácter económico y político gracias a la infección de un equipo.

  "RSA". A través de un ataque masivo de ingeniería social, consistente en el envío de correos electrónicos con un anexo que explotaba una vulnerabilidad de día cero de

[16] http://www.arcyber.army.mil/index.html

Adobe Flash, se consiguió el acceso a la red interna de la compañía consiguiendo extraer información relativas a dos factores de autenticación de tokens, conocidos como SecurID, usados para generar on-time password (OTP). Posteriormente se utilizó esta información contra empresas del sector de defensa, como es el caso de L-3 Communications. Este hecho pone de manifiesto la planificación que existe detrás de este tipo de acciones..

Es en 2011 cuando se descubre la existencia de un código malicioso similar, en el uso de técnicas de ofuscación, a "Stuxnet", aunque el payload es diferente. La empresa Dell SecureNetworks[17] describe en una sección de su página web[18] esta APT como de tipo *Remote Access Trojan* (RAT). El objetivo o misión de este *malware* no es el control de sistemas como es el caso de Stuxnet, si no, proporcionar al atacante el control remoto del equipo comprometido y los permisos para ejecutar aplicaciones.

- **Año 2012**. "Flame", también conocido con nombres como "Flamer" o "Skiwiper" fue uno de los *malware* más sofisticados que se habían detectado hasta la fecha. Su objetivo se centro en Oriente Medio. Su principal propósito es la infección de equipos y dispositivos de comunicación de modo que pueda extraerse información.

  "Gauss", *toolkit* de alta complejidad diseñado para la extracción de información sensible de los equipos infectados. Se centra en información sobre claves de cuentas en navegadores con especial atención a cuentas bancarias online. Esta última es la principal diferencia con los *malware* anteriormente mencionados como "Flame", "Duqu" o "Stuxnet". El impacto, según la empresa Kaspersky Lab, alcanza 2500 infecciones, inferior a "stuxnet" pero superior a "Flame" y Duqu". Como objetivos destacables se puede hacer referencia a Citibank y PayPal.

- **Año 2013** "Departamento de Defensa de EEUU" El 27 de mayo de 2013, The Washington Post informó sobre un ataque sufrido por el Departamento de Defensa de los EEUU, supuestamente a manos hackers ubicados en China. El ataque se tradujo en el robo de valioso información armamentística.

  "Gobierno de Pakistan" El 19 de Mayo de 2013, Information Week[19] [11] informó del ataque perpetrado contra el gobierno pakistaní mediante la utilización de APTs con el objetivo de obtener información sensible de los sectores estratégicos como el militar, automoción, minería, y otros sectores tecnológicos. Se estima que el ataque es de procedencia India y que se inició en el año 2010.

[17] http://www.secureworks.com/
[18] http://www.secureworks.com/cyber-threat-intelligence/threats/duqu/
[19] http://www.informationweek.com/

- **Año 2014** "Operación Arachnophobia". El 20 de agosto de 2014, Information Week [10] informó de que el gobierno Pakistaní había iniciado, en 2013, una campaña de ciberataques mediante la utilización de APTs contra la India como reacción a los ataques previamente sufridos. Este hecho pone de manifiesto la importancia que están tomando este tipo de amenazas tanto desde el punto de vista técnico como político.

Hasta este momento se ha presentado información sobre la evolución de la Amenazas Persistentes Avanzadas desde la perspectiva de la víctima. Sin embargo se considera enriquecedor para el presente trabajo, presentar información relacionada con las organizaciones que desarrollan este tipo de amenazas. Para ello se utiliza como fuente el informe "APT1" [12] elaborado por la empresa Mandiant que hace referencia a la unidad de ciberespionaje ubicada en China con el mismo nombre.

A continuación sólo se presenta información de carácter general, recomendando la lectura del informe a todo aquel que esté interesado en profundizar en esta materia. Dicho informe presenta el grupo APT1 como uno de los 20 grupos radicados en China. En particular, APT1 se trata de una única organización cuyo fin es la comisión de campañas de ciberespionaje en un amplio espectro de víctimas. Sus inicios datan de 2006 y de acuerdo al estudio realizado, se posicionan como uno de los grupos más productivos en este campo en relación a la cantidad de información sustraída.

Aunque el informe confirma que disponer de información limitada, afirma que el grupo APT1 cuenta con 150 víctimas confirmadas en los últimos siete años (el informe fue presentado en 2013). De acuerdo a sus niveles de actividad y los resultados obtenidos, Mandiant concluye que el grupo de ciberespionaje APT1 está respaldado por el gobierno ya que, para llevar a cabo campañas tan complejas y persistentes, es necesario contar con recursos que únicamente puede ser ofrecidos por un soporte gubernamental.

3. APT Octubre Rojo – Red October

En este apartado se presenta la Amenaza Persistente Avanzada elegida para desarrollar el piloto experimental, conocida como “Octubre Rojo” o “Red October”. Desde la perspectiva del lector, se ha considerado interesante introducir este capítulo en esta parte del documento aunque toda la información que se muestra es el resultado de la ejecución y forma parte de una las acciones descritas en la metodología a aplicar, la búsqueda en fuentes abiertas de información relacionada con el malware. Esta tarea se contempla en la segunda fase de dicha metodología, denominada clasificación. En el apartado correspondiente al desarrollo de esta fase se hace mención a este hecho, de modo que no debería existir confusión al respecto.

La APT Octubre Rojo o Red October es uno de los ataques más complejos y con mayor alcance que se han dado hasta la fecha. Iniciado en 2007 y detectado en 2012 por la empresa Kaspersky Lab.. En el presente apartado se describe este *malware* y se presentan datos sobre sus principales objetivos, vectores de ataque, extensión de la infección así como otra información que permitirá comprender la magnitud de esta amenaza.

Para desarrollar este capítulo, se han analizado varios documentos que tratan este caso en detalle. En particular se trata de 6 artículos, 3 artículos [13], [14] y [27] publicados por Kaspersky Labs, el cuarto artículo [15] publicado por la empresa Bitdefender[20], el quinto artículo [16] publicado en el blog “Seguinfo”[21] tomando como fuente a INTECO actual Incibe[22] y el sexto y último artículo [17] alojado en la página web “www.mejor-antivirus.es”

3.1 Red October: Qué es

"Red October" se trata de una campaña de ataques, basada en *malware* tipo APT, dirigida contra objetivos específicos, que ha permanecido activa al menos durante 5 años. Durante la campaña los ataques se han distribuido por todo el mundo y los sectores de los que se tiene constancia que hayan sido víctimas se presentan en la siguiente lista. Es importante destacar que Kaspesky Lab no excluye otros sectores, afirmando que, o bien no se ha iniciado el ataque todavía, o no se ha detectado.

[20] http://www.bitdefender.es/
[21] http://seguinfo.blogspot.com.es/
[22] https://www.incibe.es/

- Gobierno
- Embajadas y otros objetivos diplomáticos
- Instituciones de investigación
- Mercado y comercio
- Centros de investigación energética y nuclear
- Compañías del sector Oil&Gas
- Sector Aeroespacial
- Sector Defensa/Militar

El *malware* Octubre Rojo fue desarrollado a medida por los atacantes, que lo denominaron "Rocra" (abreviatura de Red October). Cuenta con una arquitectura modular única, incluyendo módulos especializados en el establecimiento de puertas traseras mediante *troyanos* del tipo *Backdoor* y en el robo de información.

Uno de los parámetros que permite confirmar que se trata de una campaña sostenida en el tiempo es la reutilización de la información de sitios previamente comprometidos, para la generación de bases de conocimiento concentradas que permitiesen acceder a nuevas víctimas (principalmente información relacionada con credenciales). Los atacantes diseñaron un escenario de distribuido de grandes dimensiones, basado en más de 60 dominios diferentes distribuidos en varios países, principalmente Alemania Y Rusia. Esta infraestructura les permitía controlar toda la red de equipos infectados.

El *malware* se centraba en el robo de información contenida en distintos tipos de archivos. Las extensiones de documentos [13] incluidas como objetivos son "txt", "csv", "eml", "doc", "vsd", "sxw", "odt", "docx", "rtf", "pdf", "mdb", "xls", "wab", "rst", "xps", "iau", "cif", "key", "crt", "cer", "hse", "pgp", "gpg", "xia", "xiu", "xis", "xio", "xig", "acidcsa", "acidsca", "aciddsk", "acidpvr", "acidppr", "acidssa". Como se observa, esta larga lista confirma que el alcance es realmente amplio. Los artículos consultados hacen especial hincapié en la extension "acid*", relacionada aparentemente con la herramienta software clasificada de cifrado "Acid Cryptofiler", utilizada por muchas entidades de la UE y OTAN.

En relación a la procedencia, tras los análisis realizados sobre el código como el estudio de los dominios y C&C analizados, inducen a pensar en un origen situado en una zona de habla rusa.

3.2 Red October: Dónde actúa

De acuerdo a las afirmaciones iniciales sobre la amplia extensión del ataque, es importante presentar datos cuantitativos que confirmen la anterior afirmación. Tomando como base la información obtenida por Kaspersky Lab [14] y [27], se identifican 39 países como víctimas, donde la concentración se sitúa en Suiza, Kazajistan y Grecia, tal y como se muestra en la siguiente figura. La manera de identificar los países se basa en la detección y comprobación del origen de las conexiones a los dominios donde residen los C&C. Junto con la información sobre la distribución geográfica, se añade el volumen de conexiones procedentes de estas zonas en relación a los dominios comprometidos y monitorizados por Kaspersky Lab a través del establecimiento de un *sinkhole* o sumidero.

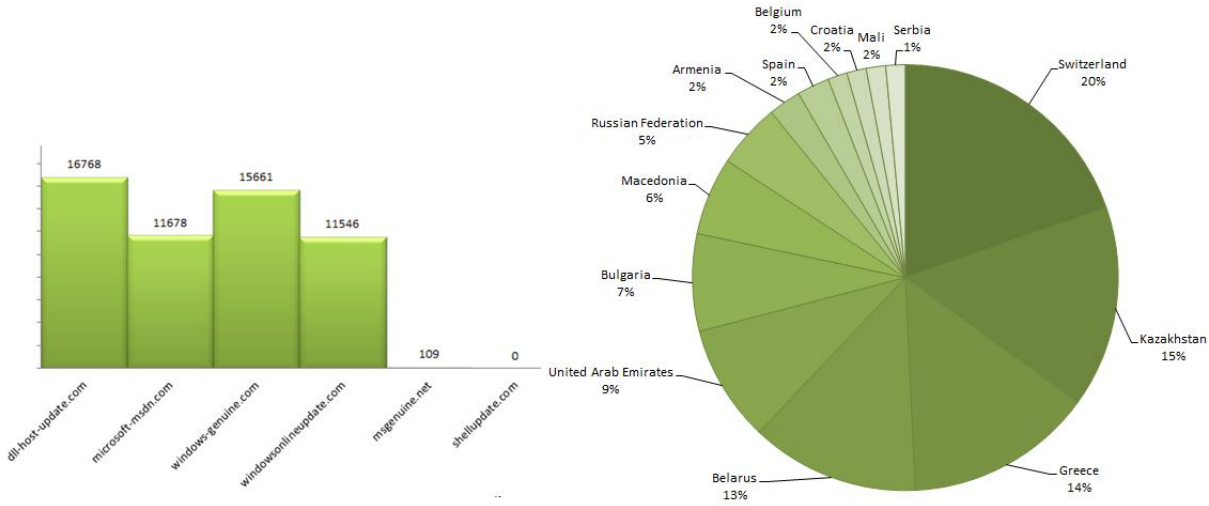


Ilustración 11 Dominios monitorizados y Países afectados por "Octubre Rojo" [14] y [27]

En los informes utilizados como base documental [14] y [27] se muestran datos sobre identificadores de los sistemas comprometidos como el propio identificador, sector al que pertenece e instalación o propietario del mismo, así como la cuantificación de sistemas infectados y detectados, por países. En el Anexo B se presenta un mapa global en el que se indica tanto el alcance geográfico como todos los sectores afectados por "Octubre Rojo".

3.3 Red October: Cómo opera

Una vez situado el *malware* "Red October", es importante conocer, de forma simple, su forma de actuar. En primer lugar se presentan las principales características del *malware* desde las vulnerabilidades que explota y la arquitectura necesaria para su control. Como se verá en los apartados en los que se desarrolla el análisis del *malware*, esta información es complementaria y coherente con los resultados obtenidos.

La Amenaza Avanzada Persistente "Red October" tiene como objetivo víctimas previamente seleccionadas, de modo que los ataques son muy específicos. Esto se confirma con el descubrimiento de que cada módulo está particularizado y cada víctima se identifica con un ID único tal y como demuestra Kasrpersky Lab [14] y [27]. Una vez la víctima ha sido infectada, la interacción entre el C&C y el *malware* es intensa, perfilando la documentación que contiene la víctima para poder adaptar el *malware* y extraer esta información de forma eficaz.

El vector de ataque principal se centra en campañas de *spear phising* o envío de correos electrónicos dirigidos. Una vez más se confirma que la ingeniería social es uno de los mejores métodos para acceder a los sistemas de las víctimas. En este caso, el contenido de los correos electrónicos estaba relacionado con la oferta de vehículos de ocasión, tal y como se muestra en la siguiente figura.

Ilustración 12 Campaña de Spear Phising utilizada en el proceso de infección [14]

Tal y como confirma Kaspersky Lab [14] y [27], el contenido fue reutilizado de otras campañas de ciberataques, modificando únicamente el *malware* adjuntado.

Una vez identificado el vector de ataque principal es necesario analizar el método de infección. La parte principal del *malware* está orientada a establecer un punto de entrada y contacto con el C&C, de modo que posteriormente se vayan descargando y activando módulos que permitan la obtención de la información del sistema víctima, garantizando una adaptación casi perfecta. Los análisis realizados [14], [15], [16] y [27] confirman que los *exploits*[23] utilizados para infectar los equipos se sirven de las vulnerabilidades **CVE-2009-3129**[24] (MS Excel), **CVE-2010-3333**[25] (MS Word) y **CVE-2012-0158**[26] (MS Word).

Tras la infección de la víctima, el *malware* comienza a obtener información sobre la red interna de la víctima, sin iniciar una propagación inmediata. Esta información es compartida con el C&C permitiendo modelar dicha red, identificando los activos/sistemas críticos. Finalmente se inicia el proceso de propagación hacia sistemas previamente seleccionados mediante la explotación de la vulnerabilidad conocida como **MS08-067**[27] [14]. A continuación se muestra una imagen con el diagrama de la primera fase del ataque que incluye la recepción del correo malicioso, ejecución del archivo infectado e infección del equipo.

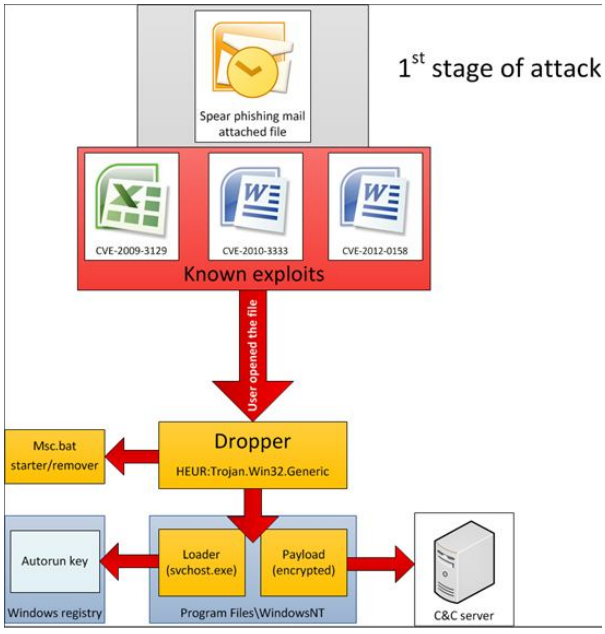


Ilustración 13 Primera fase ataque, Infección de la víctima por "Red October" [14] y [27]

[23] http://es.wikipedia.org/wiki/Exploit
[24] http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2009-3129
[25] http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2010-3333
[26] http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2012-0158
[27] https://technet.microsoft.com/en-us/library/security/ms08-067.aspx

Tras la infección del equipo, se inicia la segunda fase, centrada en establecer y explotar la comunicación con el Centro de Comando y Control. La comunicación con el C&C tiene como objetivo la descarga de módulos funcionales que se pueden clasificar en dos tipos.

- **Módulos Offline** residentes en el equipo local con capacidad de crear claves de registro propias, logs y procesos de comunicación con el C&C.
- **Módulo Online** residentes en la memoria y nunca almacenados en el disco del equipo infectado. Estos módulos no tienen capacidad de crear registros y los *log* son almacenados únicamente en la memoria. La comunicación con el C&C la realizan utilizando su propia codificación.

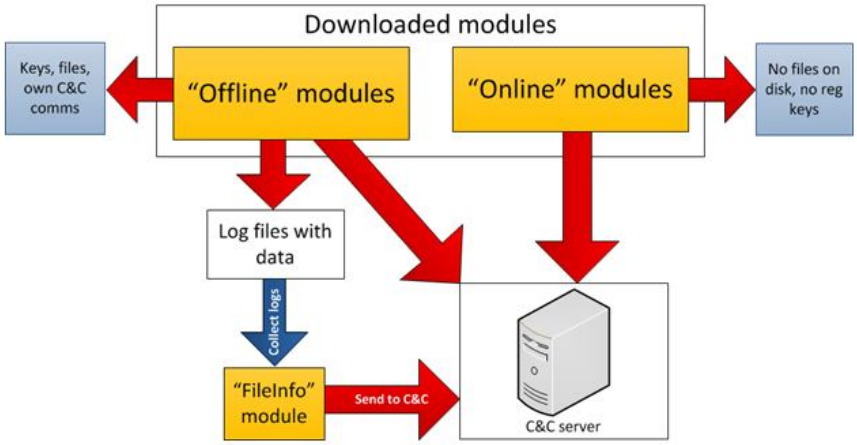


Ilustración 14 Fase 2 de ataque. Utilización de módulos y comunicación con el C&C [14] y [27]

Una vez el *malware* ha infectado la víctima, es importante conocer las acciones para las que está programado. A continuación que se describen las más importantes [16].

- Robo de archivos y ficheros del equipo almacenados en unidades locales así como almacenados en dispositivos extraíbles como memorias USB o discos duros externos. El *malware* tiene la capacidad, mediante el uso de un módulo específico, de capturar archivos previamente eliminados.
- Acceso y robo de información almacenada en terminales móviles, principalmente iPhone y Nokia, por ser los más utilizados entre las víctimas objetivo. Para ello el *malware* es capaz de detectar la sincronización con el equipo infectado.
- Captura de la configuración de determinados dispositivos de red, principalmente del fabricante Cisco[28].
- Interactuar con servidores FTP ubicados en la red de la víctima para sustraer información y enviarla al exterior.

[28] http://www.cisco.com/web/ES/index.html

Hasta el momento se ha presentado información sobre la interactuación del *malware* con el equipo víctima. Sin embargo, es muy importante comprender el funcionamiento con el centro de Comando y Control (C&C) ya que, tal y como se ha mencionado, es el responsable de enviar los módulos específicos que garanticen el éxito del ataque. Como se ha mencionado anteriormente, Kaspersky Lab [13] ha identificado más de 60 dominios concentrados principalmente en Rusia y Alemania. Estos dominios albergan los C&C desde los que los atacantes interactúan con los equipos infectados. A continuación se muestra un diagrama de red que modela la parte descubierta de la infraestructura de control existente. En el Anexo C, se amplía la información sobre la red de control.

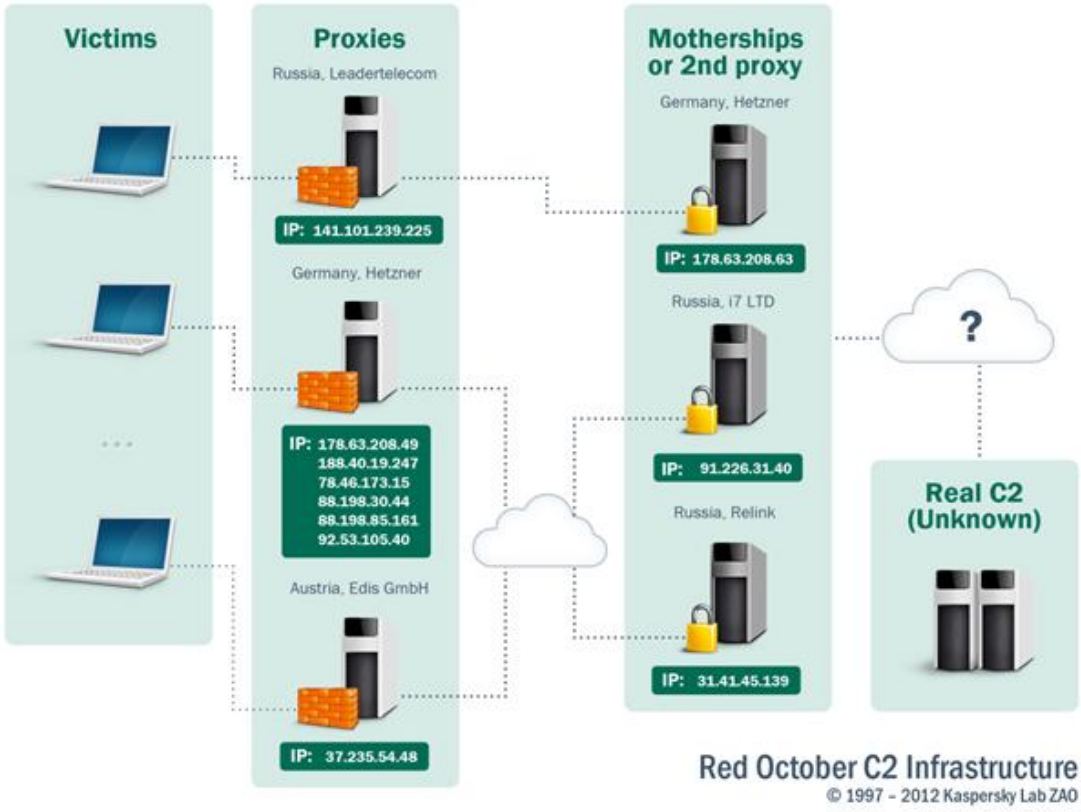


Ilustración 15 Infraestructura de control de la red de equipos infectados "Red October" [27]

Como se observa es una red basada en múltiples saltos gestionado por *proxies* que no permiten alcanzar el servidor principal, demostrando el alto nivel de desarrollo de los atacantes que se encuentran detrás de esta campaña.

3.4 Red October: Modularidad y Variantes

El *malware* "Red October" se compone de distintos módulos, específicos en función de la víctima y el objeto de la infección particular de cada una. La identificación y muestra de los diferentes módulos que componen este *malware*, refuerza la complejidad en su desarrollo y la importancia de su análisis, una de las líneas argumentales principales del presente trabajo.

El análisis realizado por Kaspersky Lab [14], identifica más de 1000 módulos diferentes agrupados en más de 30 categorías diferentes, a continuación se muestran los módulos que han podido ser analizados y clasificados. En el Anexo D se presenta una tabla con los criterios de clasificación y una tabla con los módulos identificados más importantes agrupados.

3.5 Red October: Relación con otras APTs y Evolución

Para concluir con la descripción del APT que se toma como fuente para la aplicación de la metodología, se presenta información sobre la posible relación con otras APTs contemporáneas como pueden ser "Gauss", "Flame" o "Duqu" y campañas como "NightDragon" o "Aurora", todos ellos ya presentados en apartados anteriores, y sobre su evolución en el tiempo.

En primer lugar, el informe [14] confirma que no se han encontrado conexiones con otras APTs como "Flame", "Gauss" o "Duqu". Si bien estas fueron utilizadas en campañas de ciberespionaje automatizado, "Red October" se identifica como una campaña personalizada en función de las víctimas seleccionadas.

Desde el punto de vista de las operaciones de ataques mediante la utilización de APTs, se confirma que la campaña basada en "Octubre Rojo" es más sofisticada que "Aurora" y "Night Dragon". La identificación de más de 1000 módulos agrupados en 30 familias diferentes, posiciona a "Red October" como una campaña de alta complejidad, situando "Aurora" y "NightDragon" como campañas menos complejas basadas en *malware* más simple.

En relación a la evolución de "Red October", el informe [27] consultado presenta una línea temporal en la que se identifican hitos de nuevas creaciones de ficheros, 115, utilizados en campañas de ataques entre los años 2010 y 2015. A continuación se presenta una lista con

estos hitos temporales en los que se llevaron a cabo ataques masivos, confirmando su carácter persistente en el tiempo, característica fundamental en una Amenaza Avanzada Persistente.

2010	2011	2012
19-may	05-ene	05-ene
21-jul	14-mar	
04-sep	05-abr	
	23-jun	
	06-sep	
	21-sep	

Tabla 1 Registro temporal de incidentes "Octubre Rojo" [27]

Para concluir con el capítulo que describe la APT "Octubre Rojo" se hace referencia al artículo "Octubre Rojo ataca de nuevo" [17] que presenta un nuevo descubrimiento de un ataque basado en APT y denominado "Cloud Atlas" por parte de Kaspersky Lab. El análisis de esta APT muestra la reutilización del nombre de un archivo utilizado en la campaña de infección, "Diplomatic Car for sale.doc", que fue utilizado previamente en la campaña de infección basada en el APT "Red October", desmantelada en 2012.

Este hecho refuerza la existencia de grupos organizados altamente cualificados y estables, cuyo objetivo es el desarrollo de código malicioso de alta complejidad y la ejecución de campañas de infección a gran escala con objetivos estratégicos.

Esta última afirmación junto con el resto de características descritas en este capítulo, coincidentes con las principales que tiene una APT, confirman la clasificación de "Red October" como *malware* del tipo APT y lo posiciona como uno de los más complejos y avanzados detectados hasta la fecha.

4. Descripción de la metodología

En este apartado se realiza una breve descripción de los tipos de metodologías y mecanismos de análisis de *malware*, centrándose en la metodología que se va a aplicar y sobre la que se basa el presente trabajo.

Es importante destacar que, actualmente, se disponen de diferentes técnicas de análisis de *malware*, sin embargo no existe gran variedad en las metodologías de su aplicación. Teniendo en cuenta la complejidad del análisis, la rápida evolución de las APTs, así como la necesidad de coordinación entre diferentes agentes, es relevante disponer de herramientas metodológicas que permitan utilizar de forma ordenada y coherente las distintas técnicas de análisis de *malware* disponibles.

La Tesis Doctoral de Don Javier Bermejo [1], fuente principal del presente trabajo, profundiza en la descripción de las técnicas y metodologías. Teniendo en cuenta que no es ese el objetivo principal de este piloto experimental, únicamente se van a introducir para completar la información en este campo.

Desde el punto de vista de las técnicas aplicadas para el análisis de *malware*, se presenta una revisión [18] de las principales técnicas que se resume en los siguientes puntos:

- **Análisis dinámico o de comportamiento**, técnica fundamentada en el análisis del código en tiempo de ejecución, supervisando y monitorizando los cambios en el propio equipo infectado como en el entorno (redes y comunicaciones con el C&C). Estas técnicas permiten al analista obtener información para, posteriormente, llevar a cabo técnicas de ingeniería inversa.
- **Análisis dinámico de código**, técnica basada en el análisis del código mediante el uso de herramientas de depuración que permiten al analista recorrer el código de forma exhaustiva, facilitando un análisis en profundidad de registros y modificaciones en memoria.
- **Análisis estático de código**, técnica que consiste en el análisis del código binario sin ejecutarlo con el objetivo de obtener la máxima información. Para ello se utilizan técnicas de ingeniería inversa y o desensamblado.

Concluyendo con la breve descripción de las técnicas de análisis, es importante mencionar la posibilidad de automatización de este tipo de análisis mediante el uso de sistemas *sandbox*. Estos sistemas no son más que entornos controlados, principalmente basados en sistemas virtuales, que disponen de sistemas de monitorización y permiten la ejecución en tiempo real de código malicioso de forma acotada.

Cada una las anteriores técnicas tiene ventajas y desventajas, por ello, desde la perspectiva metodológica, cobra interés su utilización de forma organizada y en base a procedimientos preestablecidos, permitiendo obtener resultados válidos de forma eficiente. El primer método y más simple de todos los existentes en el análisis de *malware*, es el de la observación, basado en la comprobación de los cambios de estado de un equipo víctima antes y después de ejecutar el código malicioso [19].

Tal y como se ha comentado en la introducción del presente apartado, la necesidad de un proceso sistemático definido por una metodología es necesaria por los siguientes motivos:

- Necesidad de optimizar las técnicas disponibles, aprovechando las ventajas individuales de cada una y solventando las desventajas individuales mediante la complementariedad de las mismas.
- Necesidad de facilitar la utilización de las herramientas disponibles, teniendo en cuenta su creciente evolución tanto en número como en complejidad.

La Tesis Doctoral [1] describe tres metodología diferentes para el análisis de *malware*, resumidas a continuación, que se utilizan como base documental para la elaboración de las conclusiones tras aplicar la metodología seleccionada en este piloto experimental.

- **“Malware Analyst´s Cookbook and DVD. Tools and Techniques for Fighting Malicious Code, 2011”[20]** se presenta una metodología basada únicamente en el análisis dinámico automático utilizando una *sandbox* que engloba las siguientes fases:

Ilustración 16 Fases de la Metodología presentada por Ligh, Adair, Harstein & Richar, 2011

- **"Malware Analysis: An Introduction, Hornat, 2007"[21]** Se presenta una metodología completa de análisis de *malware* que no detalla procedimientos, quedándose a un nivel general. Esta metodología comprende las siguientes fases:

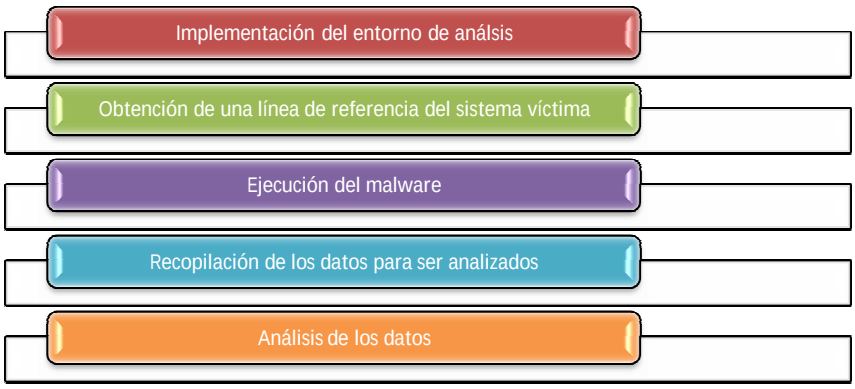


Ilustración 17 Fases de la metodología "Malware Analysis" presentada por Hornat, 2007

- **"Malware Analysis and Reverse Engineering (MARE), Timeline & Goldman, 2011"[22]** Se presenta como la metodología más completa identificada hasta el momento. En la Tesis Doctoral de Don Javier Bermejo[1] se utilizada como referencia principal para la evaluación de las ventajas de la metodología que se aplica en el presente trabajo y que es resultado de la citada Tesis Doctoral. Fases de la metodología se muestran en la siguiente imagen

Ilustración 18 Fases de la metodología "Malware Analysis and Reverse Engineering" (MARE), 2011

4.1 Introducción a la metodología a aplicar

Tras presentar las metodologías que actualmente están disponibles para los analistas de *malware* se procede a la descripción con mayor detalle de la metodología propuesta por Don Javier Bermejo [1]. Teniendo en cuenta que esta metodología es la que se va a utilizar en este piloto experimental, es relevante comprender las fases de las que se compone y las ventajas que ofrece con respecto a las anteriormente presentadas.

Es muy importante destacar que la innovación de la metodología a utilizar se basa en una modificación en el orden de ejecución de las acciones contempladas dentro de la fase de análisis de código, realizando en primer lugar el análisis de estático para, posteriormente, ejecutar el dinámico. Finalmente la metodología utilizada contempla la posibilidad de efectuar ciclos iterativos derivados del comportamiento del *malware* al ser ejecutado, permitiendo volver de la fase de análisis de comportamiento a la de análisis de código.

Desde el punto de vista de su aplicación, esta metodología está enfocada al análisis de *malware*, fundamentalmente archivos ejecutables y librerías, desarrollado contra sistemas Windows, sin embargo, los procesos pueden ser reutilizados para el análisis de otros sistemas operativos como Linux o Android u otros de menor utilización.

En primer lugar se describen las fases que componen esta metodología. Se trata de cuatro fases bien diferenciadas que deben utilizarse como un proceso sistemático para el análisis de *malware*. Su diseño se ha definido teniendo en cuenta la flexibilidad necesaria para poder ser adaptado a los distintos tipos de códigos maliciosos y su evolución. La descripción de las fases es una transcripción del documento de la Tesis Doctoral [1].

- ***Acciones Iniciales**. Consiste principalmente en la realización de una serie de acciones encaminadas a obtener un registro de la configuración de las máquinas que intervienen en el análisis, con el propósito de obtener una referencia que nos permita comparar el estado de las mismas, antes y después de ejecutar el malware bajo estudio.*
- ***Clasificación**. Consiste en examinar el archivo ejecutable del malware, pero sin acceder a su código, con el objetivo de, primero identificar el tipo al que pertenece, y, seguidamente, obtener información sobre su funcionalidad, la cual facilitará la realización de las siguientes fases de análisis, e incluso permitirá, en caso de ser necesario urgentemente por razones de seguridad, la realización de firmas simples de red, para la protección de los sistemas.*
- ***Análisis de Código**. Esta fase consiste básicamente en la realización de un análisis estático y otro dinámico del código ensamblador del malware, navegando a través de él, al objeto de conseguir un mejor entendimiento de su funcionamiento. Se trata de un proceso complejo que necesita de analistas en ingeniería inversa y, que permite obtener información inicial para la realización de la siguiente fase. Durante la misma, se encuentran funcionalidades que han permanecido ocultas durante la fase anterior y que suponen nuevas rutas de ejecución del malware a probar.*
- ***Análisis Dinámico o de Comportamiento**. Consiste básicamente en la realización de un análisis de la actuación del malware en el entorno de ejecución, con el objeto de observar su comportamiento y obtener así conocimiento de las acciones que realiza sobre el sistema objetivo (modificaciones del registro, ficheros, etc.) y del tráfico de red que genera.*

.

4.2 Descripción de la metodología

Tras haber descrito en la sección anterior las fases de la metodología propuesta, es en este apartado se detalla cada una de ellas, indicando los hitos principales que se deben cumplir y que se presentarán de forma práctica en los siguientes capítulos. Al igual que en la descripción de las fases, y no pudiendo ser de otra manera, para su desarrollo se toma como base la Tesis Doctoral [1].

4.2.1 Acciones Iniciales

El principal objetivo de esta fase en la comprobación de la integridad del entorno de análisis, así como la obtención de registros de estado iniciales, principalmente de configuración, que sirvan como línea base de referencia para una posterior comparación con el estado final tras la ejecución del *malware*.

De forma complementaria se garantiza el inicio del análisis en un escenario limpio. Este escenario deberá poder recuperarse de forma natural mediante la generación de imágenes de disco para sistemas físicos o *snapshot* de entornos virtuales, recuperables durante todo el proceso en caso de que sea necesario reiniciar el análisis.

Las actividades a desarrollar en esta fase, tal y como se enumeran en la Tesis Doctoral [1] tomada como referencia, son las siguientes:

- *Realizar una línea de base de la configuración del sistema víctima (foto instantánea) después de instalar todas la herramientas de análisis, con objeto de tener así una referencia con la que comparar el estado obtenido después de haber realizado los procesos de análisis de malware.* Para ello se podrán utilizar las herramientas "Systracer" y "WinMD5" que permite obtener un hash del sistema.
- *Si se trabaja en un entorno virtual, generar una instantánea inicial, snapshot, o bien realizar una imagen si estamos en el entorno físico.*
- *En la etapa inicial, al configurar el sistema operativo, desactivando todos los servicios del sistema que pueden modificar archivos binarios, principalmente los de restauración y actualización del sistema.*
- *Antes de iniciar un nuevo análisis de malware, comprobar la integridad de los ficheros involucrados.* Para ello se utilizará la herramienta "WinMD5", únicamente sobre los archivos binarios, evitando los archivos que son modificados en tiempo de ejecución por el S.O.
- *Tomar otra instantánea y compararla con la obtenida de referencia, para comprobar ver si no se han realizado cambios en las entradas del registro y/ o sistema de ficheros.* Para ello se recomienda el uso de la herramienta "Systracer".
- *Grabar el tráfico entre la máquina host y la virtual, a fin de comprobar la inexistencia de tráfico generado por el malware hacia la máquina host.* Para ello se recomienda la utilización de la herramienta "Vmnetsniffer".

4.2.2 Clasificación

Fase consistente en examinar el archivo ejecutable del *malware* sin acceder al código malicioso, para poder obtener información inicial que garantice mayor efectividad en la ejecución de las siguientes fases de la metodología. Las actividades a desarrollar durante la ejecución de esta fase son las siguientes:

- *Transferir el malware.*
- *Identificación del malware, obteniendo su hash*[29] *(MD5 o SHA).* Para llevar a cabo esta tarea se recomienda la utilización de las herramientas "md5deep" y "WinMD5".
- *Clasificación del malware según el tipo, o si pertenece a una familia anterior por modificación de alguno de sus componentes.* Esta actividad consiste en la búsqueda de la existencia de una identificación del *malware* a analizar mediante la utilización de filtros de antivirus. Es importante utilizar varios filtros con el objeto de contar con el mayor número de firmas de *malware* conocidos. Se recomienda el uso herramientas como "ClamAV", "Panda", "Bitdefender", "AVG" y servicios externos de análisis como "VirusTotal", "Anubis"o "VirScan" entre otros.
- *Búsqueda de información en fuentes abiertas OSSINT*[30]*.* Para ello se utilizará el *hash* MD5 del *malware* como vector de búsqueda en dichas fuentes.
- *Búsqueda de cadenas de texto en el archivo ejecutable.* Se recomienda la herramienta "Strings" que permite la búsqueda de cadenas en formato ASCII, Unicode y ambos al mismo tiempo.
- *Identificación de potenciales técnicas de ofuscación y empaquetamiento.* Para llevar a cabo esta acción, se analizará la información relacionada con las cadenas de texto obtenidas para evaluar la existencia o no de técnicas de ofuscación. Junto con el anterior análisis, se recomienda la utilización de la herramienta "PEiD" para detectar posibles técnicas de ofuscación en códigos que estén empaquetados y comprimidos.
- *Formato y estructura del fichero.* Actividad centrada en el análisis del formato y la estructura del fichero que contiene el *malware*. Las herramientas recomendadas para la ejecución de esta fase son "Dependecy Walker" y "PEBrowse".

[29] http://es.wikipedia.org/wiki/Funci%C3%B3n_hash
[30] http://en.wikipedia.org/wiki/Open-source_intelligence

4.2.3 Análisis de Código

En esta fase los esfuerzos se centran en la realización de un análisis del código del *malware*, en lenguaje ensamblador, con el objetivo de mejorar la comprensión del código malicioso antes de ser ejecutado. El análisis de código comprende dos tipos de análisis, uno estático basado en herramientas de desensamblado y otro dinámico basado en herramientas de depuración. Toda la información obtenida en esta fase deberá servir para mejorar la efectividad en la ejecución de la siguiente fase de la metodología, basada en el análisis de comportamiento del *malware*. Las actividades comprendidas en esta fase son las siguientes

- *Comprobación del funcionamiento general del mismo.*
- *Análisis estático de su código mediante un desensamblador y transformadores de código ensamblador a lenguaje de alto nivel.*
- *Análisis dinámico de su código mediante un depurador.*

Para la ejecución de esta fase se proponen distintas herramientas, todas enfocadas a facilitar las tareas de ingeniería inversa del analista. Algunas de las herramientas recomendadas son “PE Explorer”, “IDA Pro” y “Ollydbg”.

4.2.4 Análisis de Comportamiento

Esta fase se centra en la observación del comportamiento de la víctima y los cambios que sufre durante la ejecución del *malware*. Don Javier Bermejo, en su Tesis Doctoral [1], expone que, si bien el análisis del comportamiento puede ser profundo y relativamente rápido, la información obtenida en la fase de análisis de código es muy importante para adquirir una comprensión completa del código malicioso que se está analizando.

Don Javier Bermejo recomienda una ejecución progresiva de este análisis, mediante la agregación de distintos servicios al entorno de ejecución con el objeto de mejorar la comprensión del comportamiento del *malware*. De la misma manera, refuerza la necesidad de llevar a cabo esta tarea de forma progresiva, ya que la realización de varios cambios simultáneos en el entorno, puede desencadenar un comportamiento del *malware* difícil de trazar debido a la complejidad del mismo, dificultando en gran medida la tarea del analista.

Las actividades a desarrollar en esta etapa tal y como se describen en el documento [1] tomado como fuente son las siguientes:

- *Tareas previas a la ejecución. En este paso se realizarán las tareas necesarias antes de ejecutar el malware.*
- *Ejecutar malware. VMware tiene utilidades de línea de comandos que se pueden utilizar para ejecutar un programa, como el malware, con los privilegios de cualquier usuario de la máquina a la que se va a transferir. Si se está trabajando con las máquinas físicas, se puede hacer lo mismo con "PsExec".*
- *Proporcionar servicios al malware. Realizar el análisis de forma progresiva, agregando servicios en el entorno de ejecución para aprender más sobre la muestra.*
- *Tareas posteriores a la ejecución. Tareas a realizar después de ejecutar el malware, como la ejecución de herramientas en el sistema infectado y la toma de instantáneas para obtener datos por comparación, parar capturas de paquetes activa, tomar capturas de pantalla del escritorio o nuevas ventanas, y así sucesivamente.*
- *Volcado y análisis de la RAM. Si se está trabajando con máquinas virtuales, este paso implica suspender la máquina virtual y acceder a su archivo de la memoria en el sistema de archivos del host. Si se está trabajando con los sistemas físicos, este paso implica el hacer un volcado de la misma en un archivo, o directamente a través de la red a la máquina de análisis. A continuación se utiliza una herramienta como "Volatility" para su análisis.*
- *Analizar el disco duro. Si se está trabajando con máquinas virtuales, este paso implica el montaje del disco de la máquina virtual en el sistema operativo host para proceder a analizar los cambios en los archivos, secciones del registro, registros de eventos, registros de aplicación, y así sucesivamente. Si está trabajando con las máquinas físicas, se debe montar la partición C:\ en otro sistema operativo, es decir transferir la imagen de disco a otra máquina de análisis. El disco y las diferencias de registro deben ser verificados en el modo fuera de línea. Con ello se asegurará contra la pérdida ningún rootkits.*

5. Objetivos

El presente piloto experimental clasifica sus objetivos en dos grupos. El primero y principal directamente ligado con la aplicación de la metodología de análisis de *malware* propuesta por Don Javier Bermejo [1]. El segundo, relacionado con la aportación de información actualizada sobre el desarrollo continuo de las APTs y la reiteración en la necesidad de desarrollar y validar metodologías de análisis de *malware* que permitan mejorar la defensa de los sistemas actuales ante este tipo de amenazas

De manera concreta y resumida, los tres objetivos principales ligados a la aplicación de la metodología son los siguientes:

- Probar la validez, de forma práctica, de la aplicabilidad de la metodología propuesta por Don Javier Bermejo en su Tesis Doctoral [1], mediante el análisis de la APT conocida con el nombre “Octubre Rojo” o “Red October”.
- Identificar futuras áreas de desarrollo y mejora en el campo del análisis de *malware* basado en la utilización de un proceso sistemático y metódico recogido en una metodología reconocida y validada, contribuyendo a futuros trabajos de investigación en esta materia.
- Contribuir al incremento de información y documentación disponible, mediante un caso práctico, en el desarrollo y aplicación de la metodología propuesta enfocada al análisis de *malware*, en particular, del tipo APT.

De manera concreta y resumida, los dos objetivos principales relacionados con la aportación de información son los siguientes:

- Ampliar la documentación e información sobre la evolución del desarrollo de *malware* del tipo APT, complementando, de forma coherente, la presentada en la Tesis Doctoral realizada por Don Javier Bermejo[1], utilizada como base fundamental para el desarrollo del presente trabajo.
- Reiterar la importancia en el desarrollo y evolución de las metodologías que permitan contrarrestar de forma efectiva el desarrollo de *malware* mediante evidencias basadas en sucesos históricos.

6. Definición del escenario

Para llevar a cabo la aplicación práctica de la metodología, anteriormente presentada, es necesario definir tanto el tipo de investigación como el escenario de pruebas sobre el que realizar la investigación. En la Tesis Doctoral [1] tomada como referencia se presenta una clasificación de los tipos de investigación existentes presentados en el documento [23].

Siguiendo esta clasificación podemos confirmar que este trabajo pertenecería al grupo identificado como “Desarrollo experimental” y definido como aquellos trabajos sistemáticos basados en conocimientos ya existentes, es decir, la Tesis Doctoral [1], derivados de la investigación y/o experiencia práctica, dirigidos a la producción de nuevos materiales, productos, dispositivos, procesos, sistemas y servicios, o a la mejora de los ya existentes. El resultado debe permitir contar con un caso de estudio adicional al presentado en la Tesis Doctoral [1], así como ser una fuente de nuevos desarrollos o mejoras.

6.1 Entorno y Herramientas

El desarrollo del experimento debe realizarse en un entorno controlado que permita ejecutar de forma segura y acotada cada una de las fases descritas en la metodología a aplicar. Aun siendo un entorno controlado, el objetivo es contar con el escenario más realista de modo que las pruebas y resultados sean útiles y lo más aproximados a la realidad.

Desde el punto de vista del diseño, será necesario implementar un escenario que comprenda todos los elementos necesarios para simular un entorno real, esto es, máquinas, aplicación, servicios y red de comunicación, así como las herramientas de monitorización que permitan tomar evidencias durante el desarrollo del piloto. Para diseño del laboratorio que se utiliza en el presente trabajo, se toma como base el laboratorio completo presentado en el documento de A. Sanabria [23]. Este laboratorio tiene una dimensión global, siendo válido su diseño para el análisis de todos los tipos de *malware* en distintos escenarios. La arquitectura base se divide en un entorno físico y un entorno virtual.

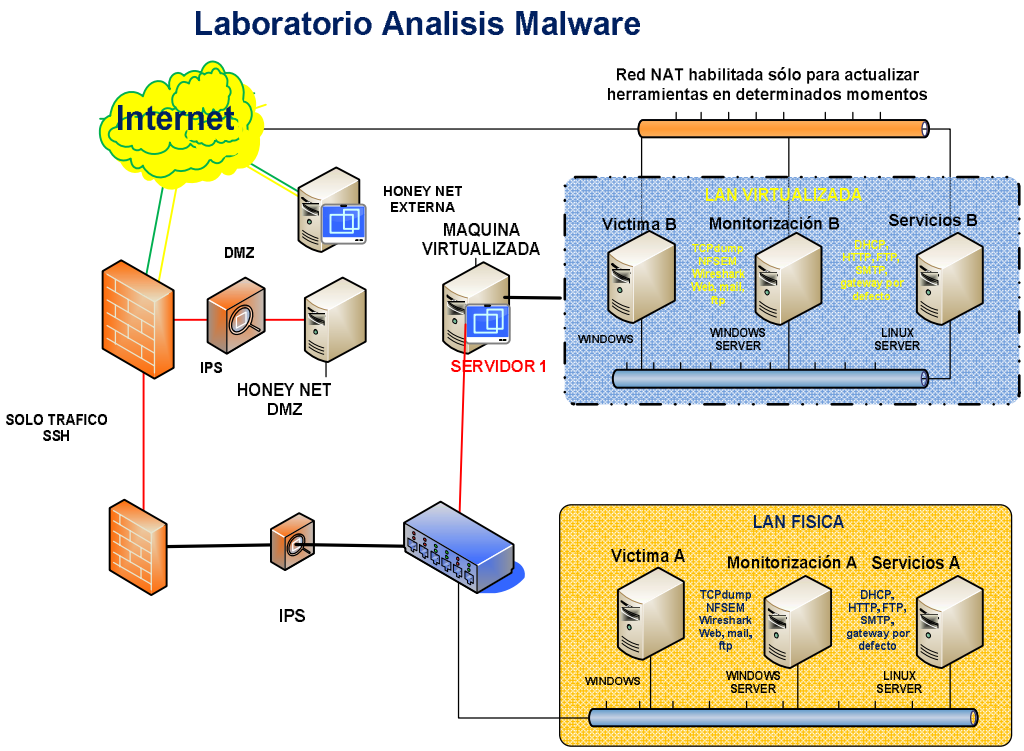


Ilustración 19 Laboratorio análisis *malware* [1]

Como se desprende de la anterior figura, el laboratorio consta de varios escenarios.

- Enlace de conexión con internet y distintos elementos de red que monitorizan y controlan la conectividad entre los distintos escenarios
- *HoneyNet*[31] totalmente expuesta a internet
- *HoneyNet* ubicada en entorno semicontrolado, como es una DMZ
- Segmento de Red de Área Local (LAN)en entorno físico
- Segmento de Red de Área Local (LAN) en entorno virtual

Teniendo en cuenta que el presente piloto experimental se centra en el análisis del APT conocido como "Red October", anteriormente presentado, se confirma que el diseño del entorno se centrará en los dos últimos puntos anteriores, es decir, LAN en entorno físico o virtual. Si el lector desea profundizar en el resto de elementos que componen el laboratorio, se recomienda consultar la Tesis Doctoral de Don Javier Bermejo [1] y el documento [23] utilizados como fuente.

[31] http://es.wikipedia.org/wiki/Honeynet

Analizando la composición de ambos escenarios, se comprueba que es idéntica, con la salvedad de su naturaleza. Ambos se pueden dividir en los siguientes 3 componentes:

- **Víctima**, sistema sobre el que se va a ejecutar y monitorizar el *malware* durante la fase de análisis dinámico o de comportamiento.
- **Monitorización y Servicios Windows**, sistema cuyo objetivo es la grabación y monitorización del tráfico de red producido por el *malware* y proporcionar servicios específicos de sistemas Windows.
- **Servicios**, sistema que proporciona servicios adicionales al *malware* para que interactúe con el entorno, como http, DHCP, Chat, IRC Server y similares. Así como herramientas específicas.

Una vez identificados los principales componentes del laboratorio, es imprescindible identificar las herramientas[32] a utilizar, para ello se toma como base el conjunto de herramientas, propuesto en el documento de base [1], agrupadas por sistema y que se muestra en la siguiente tabla.

Victima	Monitorización y Servicios Windows	Servicios
md5deep	Md5sum	Netcat
WinMD5	md5deep	HTTP Apache
Strings	WinMD5	Aplicación FTP
BinText.	YARA + firmas ClamAV	Fake DNS
PEBrowse	ClamAV	Inetsim
BGInfo	Ssdeep	AVG
Process Explorer	Bitdefender	F-prot
Process Hacker	AntiVir	Radare2
Process Monitor.	Panda	Binwalk
PsFile	Strings	Volatility
RootkitRevealer.	PEiD	Foremost
McAfee Rootkit Remover.	Dependency Walker	
Streams	PEBrowse	
AutoRuns	Windump	
TCPView	Wireshark	
Fport	Snort	
Hfind	WinHex	
Vision.	IDA Pro	
Filewatch	Reverse Engineering Compiler.	
Attacker	ProcDump 32	
Winalysis	Ollydbg	
YARA + firmas ClamAV	PE Explorer	
Ssdeep	Windbg	
Windump	Fake DNS	
Wireshark	Fakenet	
PeStudio	ISS	
CaptureBAT+WinPcap		
VMMap		
Systracer		
Resource Hacker		
PEViewer		
HexView		
DiskPulse		
GMER		

Tabla 2 Herramientas sugeridas para el análisis de *malware* [1]

[32] Todas las herramientas mostradas en el presente documento se detallan en la Tesis Doctoral de Don Javier Bermejo [1]

6.2 Diseño del laboratorio

En el anterior apartado se ha presentado la arquitectura general del laboratorio y las herramientas propuestas para el análisis de *malware*. A continuación se presenta y justifica el laboratorio real que se ha diseñado para desarrollar este piloto experimental.

La arquitectura seleccionada para desarrollar este piloto se centra es un escenario virtual basado en la herramienta de virtualización "VMWare", generando un entorno con 2 sistemas Windows que adoptarán los roles de "Víctima" y "Monitorización y servicios Windows" y un tercer sistema Linux que adoptará el rol de "Servicios".

La decisión de seleccionar un entorno virtualizado en lugar de físico, se debe principalmente a un motivo de recursos y pragmatismo ya que, en la actualidad, los escenarios virtuales permiten optimizar el uso de recursos físicos, permitiendo disponer de sistemas con capacidad de cómputo suficiente para desarrollar las pruebas necesarias.

A continuación se presenta un diagrama del laboratorio y las especificaciones técnicas del mismo.

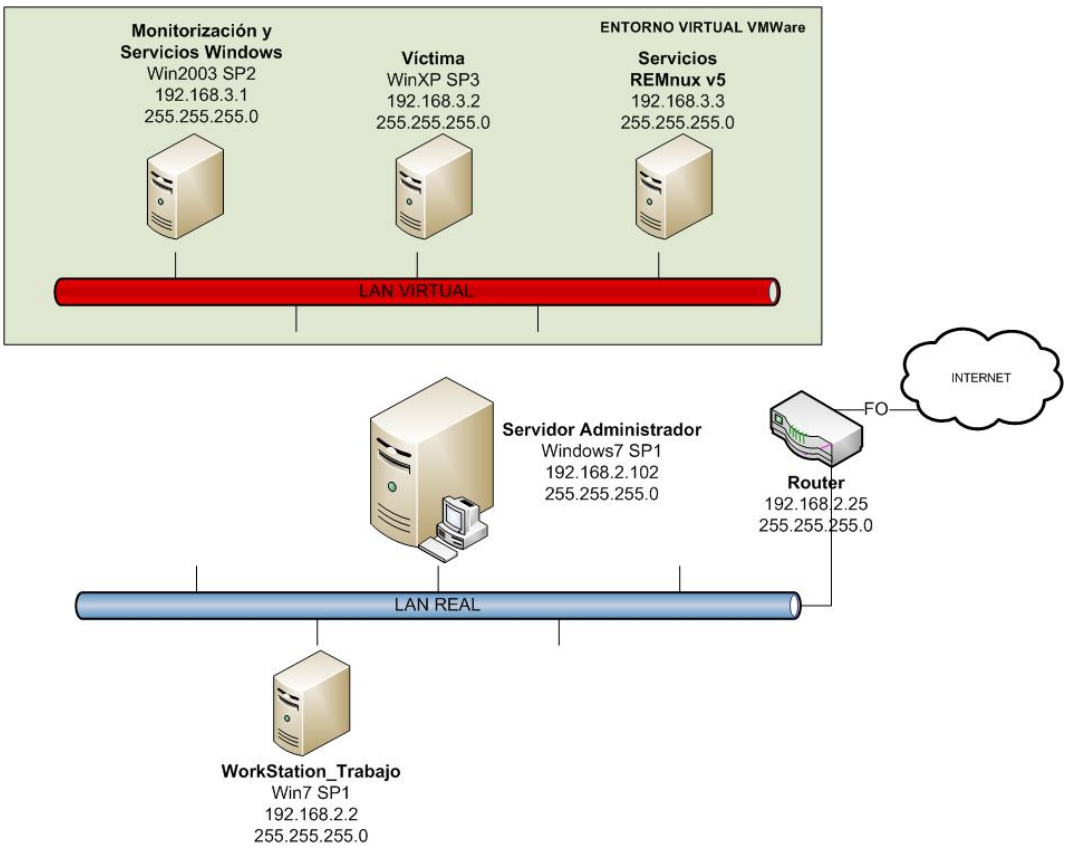


Ilustración 20 Laboratorio diseñado para el piloto experimental

A continuación se presentan las características técnicas principales de cada una de las máquinas utilizadas en el laboratorio

- **Servidor Administrador.**
  - S.O: Windows 7 SP1
  - Arquitectura: 64bits
  - Procesador: Intel Core i3-2120T 2.60GHz
  - Memoria RAM; 4GB

- **Víctima**
  - S.O: Windows XP Profesional SP3
  - Procesador: Intel Core i3-2120T 2.60GHz
  - Arquitectura 32bits
  - Memoria RAM: 1GB

- **Monitorización y Servicios Windows**
  - S.O: Windows Server 2003, SP2
  - Arquitectura 32bits
  - Procesador: Intel Core i3-2120T 2.60GHz
  - Memoria RAM: 1GB

- **Servicios**
  - Distribución REMnux v5
  - Múltiples servicios para llevar a cabo tareas de ingeniería inversa de *malware*. En el Anexo E se presenta un resumen de estos servicios

Por último se presentan las herramientas implementadas en cada una de las máquinas del laboratorio. La selección de estas herramientas se ha hecho en base a los recursos disponibles, principalmente licencias, confirmando que son *freeware* o versiones de evaluación. Del mismo modo se ha tenido en cuenta el análisis realizado en la Tesis Doctoral [1] sobre el *malware* "Flame", con el objetivo de dotar de coherencia el realizado en el presente trabajo, permitiendo evaluar la aplicabilidad de la metodología propuesta, no sólo desde el concepto sino desde la práctica. En el Anexo F se presenta un listado descriptivo de las herramientas utilizadas.

Victima		Monitorización y Servicios Windows	Servicios
WinMD5	PEViewer	Md5sum	REMnux
Md5sum	Process Monitor.	WinMD5	
Strings	RegShot	Strings	
BinText.	PeStudio	PEiD	
Avira	VMMap	Dependency Walker	
Process Explorer	Systracer	PEBrowse	
PEiD	DiskPulse	Ollydbg	
Dependency Walker	GMER	PE Explorer	
Ollydbg	Winpmem	Wireshark	
PE Explorer	Wireshark		
IDA Pro			

Tabla 3 Herramientas utilizadas en el laboratorio "Octubre Rojo"

Junto con las herramientas es importante disponer de servicios que simulen un entorno real de modo que el *malware* pueda desplegarse en toda su extensión. En este caso se ha optado por incluir dos herramientas. La primera se utiliza como base y está basada en los servicios DNS, FTP y Web que permite activa Windows 2003 Server. La segunda se basa en la distribución Linux, REMnux v5, que se utilizará para el análisis de los volcados de memoria.

A continuación se muestra una captura en la que se presenta es aspecto real del laboratorio creado para llevar a cabo este piloto experimental. Se puede confirmar el entorno de virtualización utilizado, los sistemas presentados en el diagrama general y las herramientas disponibles junto con el *malware* a analizar.

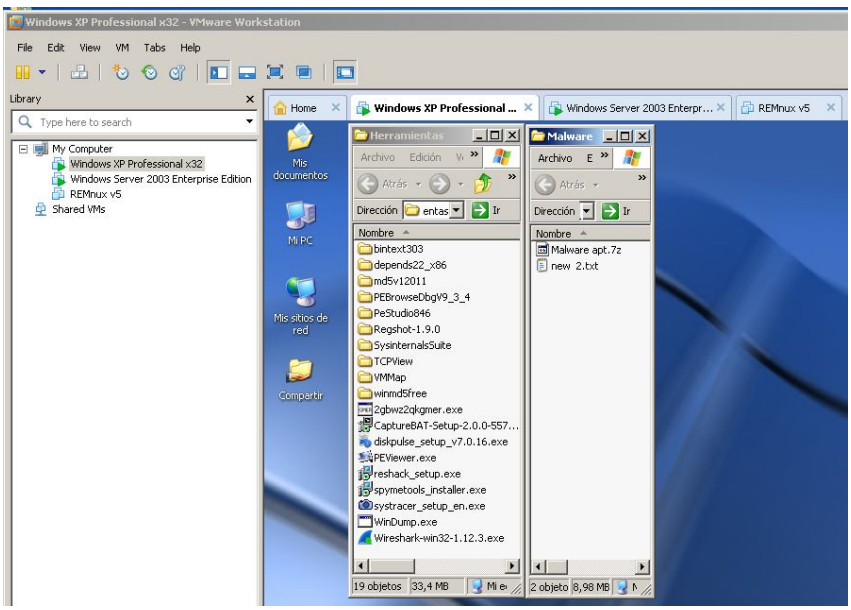


Ilustración 21 Captura de pantalla del laboratorio "Octubre Rojo"

7. Desarrollo de Pruebas y Resultados

En este aparado se muestra la aplicación práctica de la metodología propuesta y previamente presentada. El desarrollo de la metodología seguirá fielmente el orden de fases marcado y tomará como base el caso práctico incluido en el documento [1] de referencia.

7.1 Fase 1. Acciones Iniciales

Esta fase tiene como objetivo definir el entorno inicial, adquirir evidencias que permitan garantizar su integridad y permitir un mecanismo práctico de volver a reproducirlo para continuar desarrollando el análisis de *malware* en caso de ser necesario, sin que ello conlleve ningún impacto en el resultado. A continuación se describen las actividades realizadas.

- **Preparación de los equipos de análisis**. Instalación de todas las herramientas y desactivación de los servicios (actualizaciones automáticas y recuperación del sistema).
- **Toma de línea base de configuración**. Adquisición de evidencias que permitan garantizar la integridad de los sistemas en estado limpio.
    - Al tratarse de un entorno virtual, se toma un *snapshot* de cada una de las máquinas virtuales. Para ello se hace uso de la utilidad nativa de “VMWare” que permite generar un *snapshot* de los tres sistemas, víctima, monitor y servicios.
    - Toma de *snapshot* de las unidades “C:” de las máquinas virtuales con la utilización de la herramienta “Systracer”
    - Toma de hash MD5 de los *snapshot* de las unidades “C:” tomadas previamente utilizando las herramientas “Md5sum” y comprobación de integridad con la herramienta “WinMD5”

Esta acción se realiza tanto sobre el sistema víctima, monitor y servicios, con objeto de tener acotado todo los sistemas con acceso al *malware*. A continuación se presentan una imagen del equipo víctima que evidencia la ejecución de esta tarea.

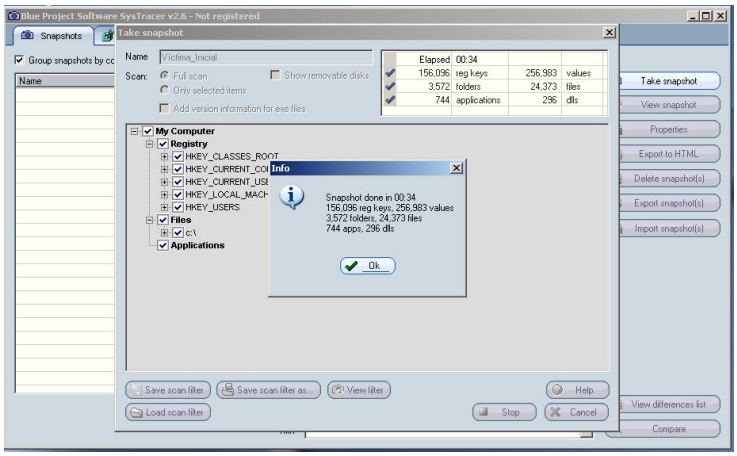


Ilustración 22 Creación *Snapshot* Víctima, "Systracer"

En este punto se da por finalizada la primera fase. Teniendo en cuenta el flujo de acciones iniciales propuesto en la metodología base, las anteriores acciones abarcan la preparación del escenario únicamente. En el caso de que el desarrollo del análisis implique la ejecución del *malware* en más de una ocasión partiendo de la situación inicial, será necesario restaurar los sistemas. En particular, en este caso, tratándose de un laboratorio virtual, se deberán cargar las imágenes tomadas inicialmente con la herramienta "VMWare" y comprobar su integridad.

7.2 Fase 2. Clasificación

Esta fase tiene como objetivo analizar el *malware* sin llegar a ejecutarlo. Se divide en 7 actividades diferentes que se realizan sobre la muestra del *malware* "Red October", tal y como se muestra a continuación.

7.2.1 Transferencia del *malware*

La primera acción consiste en la transferencia del *malware* a los sistemas en los que se va a analizar. Para ello se ha utilizado una unidad de almacenamiento externa (USB) para transportar el código malicioso, previamente comprimido y protegido por contraseña.

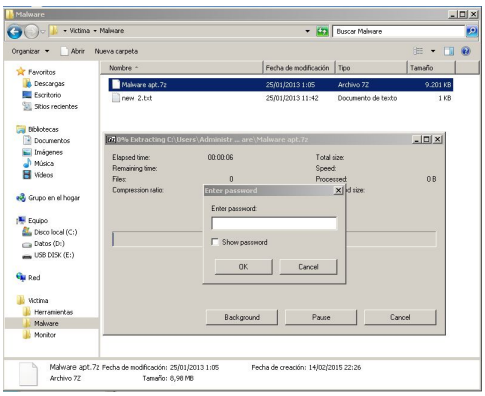


Ilustración 23 Método de transferencia del *malware*

Una vez descomprimida la muestra, se observa el contenido compuesto por los siguientes 6 archivos

- red_oct.document.exploit – archivo que contiene el *exploit*
- red_oct.bin.drop – archivo con función de *dropper*[33]
- ms32trayX.exe.orig – archivo con el modulo destinado al robo de información almacenada en usb o recuperación de archivos eliminados.
- igfxtrayms.exe.orig – archivo con el módulo de robo de documentos.
- imapisync32.dat – archivo de configuración original del archivo "igfxtrayms.exe"
- b7327bfa4a101a21f0cc1b366aa8e107 – archivo con el módulo de robo de credenciales de correo electrónico.
- 76e1d54a890befed31a369ce40b44ee6 – archive con el modulo de robo de información alojada en un iPhone

Cabe destacar que los comentarios proceden de la información suministrada con las muestras. Durante esta fase se comprobará si esta información es coherente con la que se obtenga de las fuentes consultadas.

7.2.2 Identificación del *malware*

Esta acción consiste en obtener la información del *malware* basada en nombre, tamaño y HASH MD5. Esta información se utilizará posteriormente en el análisis. El gran número de muestras disponibles y su análisis excede en extensión, complejidad y objetivo el alcance del presente piloto experimental.

Por esta razón se procede a su presentación, incluyendo toda la información necesaria para llevar a cabo un análisis de cada uno de los archivos. Sin embargo, a partir de este momento, el desarrollo de la metodología se va a aplicar sobre dos archivos encargados de la primera fase de la infección. El archivo que contiene el *exploit* inicial, "red_oct.document.exploit", y el archivo que contiene el *dropper,* "red_oct.bin.drop". La razón principal, aparte de tratarse de los archivos desencadenantes de la infección mediante la explotación de la vulnerabilidad y la distribución del *malware* por la víctima, es la complejidad en la gestión de los módulos por parte del *malware*. Tal y como se ya ha comentado en el capítulo de presentación de "Red October", este *malware* presenta una arquitectura modular muy compleja implicando una gran dificultad en el análisis, objetivo fuera del alcance principal de este piloto experimental..

[33] http://es.wikipedia.org/wiki/Dropper_(malware)

NOMBRE ARCHIVO	TAMAÑO	MD5	COMENTARIOS
red_oct.document.exploit	547KB	bb2f6240402f765a9d0d650b79cd2560	archivo que contiene el exploit
red_oct.bin.drop	521KB	598ce670b5e4be42966dbfae18188792	archivo con función de dropper
ms32trayX.exe.orig	376KB	1a88fb2ce3827706fc8ae98b4ebec69a	archivo con el modulo destinado al robo de información almacenada en usb o recuperación de archivos eliminados
igfxtrayms.exe.orig	482KB	8514afd62efdd9e6f4334e386ef32b23	archivo con el módulo de robo de documentos
imapisync32.dat	2KB	b5c892e79ff20708c38bdab5c5f2e5fb	archivo de configuración original del archivo "igfxtrayms.exe"
b7327bfa4a101a21f0cc1b366aa8e107	232KB	b7327bfa4a101a21f0cc1b366aa8e107	archivo con el módulo de robo de credenciales de correo electrónico
76e1d54a890befed31a369ce40b44ee6	331KB	76e1d54a890befed31a369ce40b44ee6	archive con el modulo de robo de información alojada en un iPhone

Tabla 4 Relación de las muestras de *malware* y su hash MD5

7.2.3 Comprobación del tipo de *malware*

Para desarrollar esta acción se van a utilizar tanto herramientas de detección de *malware* online como aplicaciones instalables. En este caso, sólo se va a analizar el archivo "red_oct.document.exploit", ya que es el que porta el *exploit* y se considera prioritario desde el punto de vista de la comprobación y detección.

Como herramienta base online, se utiliza "Virustotal"[34] que identifica el archivo y ofrece información de gran relevancia, como las herramientas antivirus que reconocen el *malware* y las que no lo hacen, así como las dependencias y relaciones con otros ficheros sospechoso con los nombres "Reed October.zip" y "red_october.7z",confirmando una relación directa con el APT "Octubre Rojo".

Ilustración 24 Identificación del archivo "red_oct.document.exploit", "Virustotal"

[34] https://www.virustotal.com/es/

En relación al informe, realizado por "Virusyotal" sobre la detección del *malware* por parte de las principales herramientas de detección, en la siguiente figura se muestran los principales resultados. Destacar que la herramienta como "Symantec" relaciona el archivo directamente con la APT "Red October", así como la no detección por la herramienta de Panda Security. La información completa se presenta en el Anexo G

ENTIDAD	DETECCIÓN	FECHA REVISIÓN
Avast	XLS:CVE-2009-3129 [Expl]	20140624
Kaspersky	Trojan-Dropper.MSWord.Agent.ga	20140624
TrendMicro	TROJ_OLEXP.B	20140624
Fortinet	MSExcel/CVE_2009_3129.A!exploit	20140624
McAfee-GW-Edition	Heuristic.BehavesLike.Exploit.X97.CodeExec.O	20140623
Microsoft	Exploit:Win32/CVE-2009-3129	20140624
Ad-Aware	Exploit.CVE-2009-3129.Gen	20140624
BitDefender	Exploit.CVE-2009-3129.Gen	20140624
McAfee	Exploit-MSExcel.ac	20140624
AVG	Dropper.Generic_c.NZR	20140624
Symantec	Backdoor.Rocra	20140624
ClamAV	NO	20140624
Panda	NO	20140624

Tabla 5 Detección del malware por principales Herramientas, "Virustotal"

Para completar esta acción utilizando una herramienta de detección no online, se utiliza el antivirus Avira[35], no contemplado en el análisis anterior.

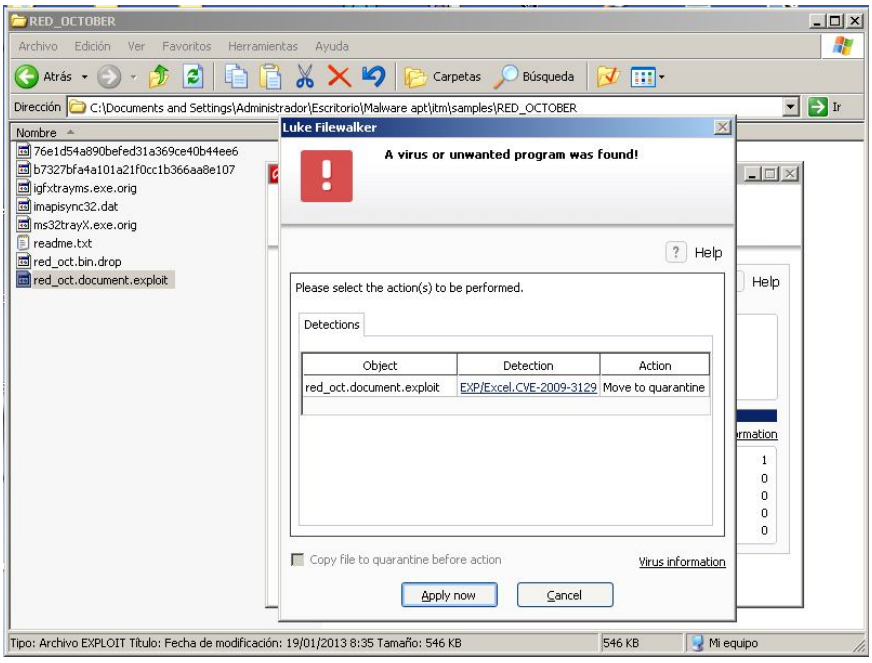


Ilustración 25 Detección malware mediante antivirus, "Avira"

[35] http://www.avira.com/es/avira-free-antivirus

Salvo por las excepciones de que algunas herramientas de detección no han identificado el archivo como *malware*, según el informe de "Virustotal", se confirma que la mayoría sí lo hacen. Esto no debería sorprender teniendo en cuenta que la detección de este código malicioso se llevó a cabo en 2012, de modo que, actualmente se ha contado con el tiempo necesario para la actualización de los sistemas de antivirus.

7.2.4 Información obtenida de fuentes abiertas

Tras realizar los anteriores pasos, se ha identificado que el *malware* que se está analizando está relacionado con la APT "Octubre Rojo". En todo momento se utiliza el concepto de relación con la amenaza "Octubre Rojo", ya que su complejidad no permite hablar únicamente de un archivo *malware*, es decir, que esta APT no se basa en un único archivo o fragmento de código, si no que cuenta con un alto número de módulos específicos tal y como ya se ha comentado.

Con la identificación del código malicioso que se está analizando, es en esta tarea, en la que se consulta toda la información útil de fuentes externas para que permita al analista ampliar su comprensión sobre la amenaza que está analizando. Tal y como se menciona en el capítulo dedicado a la presentación del *malware* del tipo APT "Octubre Rojo", prácticamente la totalidad de la información que se presenta en él, forma parte de esta fase con lo que se da por supuesta su inclusión y por ello no se repite.

Si bien ya se han presentado en un capítulos anterior información relacionados con el modo de operación, distribución y dimensión de la amenaza e impacto de la APT "Octubre Rojo", obtenida durante esta fase del análisis, en este capítulo se va a completar dicha información con otra de mayor rigor técnico. Los diagramas del proceso de infección y desarrollo del *malware* en el equipo víctima mostrados anteriormente se toman como base para la realización de esta tarea.

En primer lugar es necesario precisar las vulnerabilidades que se explotan, si bien, previamente sólo se ha hecho mención a vulnerabilidades explotables desde una campaña de *phising*, es necesario añadir otras vulnerabilidades descubiertas a posteriori de acuerdo al informe [25] elaborado por Kaspersky junto con Alienvault[36] y los informes [14], [26], [27] y [28] elaborados por el Global Research & Analysis Team de Kaspersky Lab, tomados como base para desarrollar esta capítulo.

[36] https://www.alienvault.com/

Actualmente se conocen los siguientes 5 vectores de ataque. En el Anexo H, se muestra una lista con los *hash* MD5 de los archivos utilizados para portar el *malware* e infectar a las víctimas.

- CVE-2009-3129: Ficheros XLS (MS Excel)
- CVE-2010-3333 Ficheros DOC (MS Word)
- CVE-2012-0158 Ficheros DOC (MS Word)
- CVE-2011-3544[37] Vulnerabilidad relacionada con Java
- CVE-2008-4250[38] Vulnerabilidad que permite infectar otros equipos de la red

Se han desarrollado los siguientes parches para solventar las anteriores vulnerabilidades [25]:

- Microsoft Security Bulletin MS09-067 – Important - Vulnerabilities in Microsoft Office Excel Could Allow Remote Code Execution (972652)
- Microsoft Security Bulletin MS10-087 – Critical - Vulnerabilities in Microsoft Office Could Allow Remote Code Execution (2423930)
- Microsoft Security Bulletin MS12-027 – Critical - Vulnerability in Windows Common Controls Could Allow Remote Code Execution (2664258)
- Microsoft Security Bulletin MS08-067 – Critical - Vulnerability in Server Service Could Allow Remote Code Execution (958644)

El informe [26] confirma que el *malware* no hace uso de ninguna vulnerabilidad del tipo 0-day, de modo que se aprovecha de sistemas que no se encuentran correctamente parcheados o actualizados.

Las campañas basadas en el correo electrónico hacen uso de servidores de correo anónimos que operan sobre proveedores de servicio gratuitos o utilizan servidores que previamente se han utilizado para otras campañas de lanzamiento del *malware*.

Una vez el usuario ejecuta el archivo infectado, adjuntado al correo electrónico, en el caso de que este sea el vector de ataque utilizado, el *malware* extrae y ejecuta 3 archivos adicionales [27]:

- %TEMP%MSC.BAT
- %ProgramFiles%WINDOWS NTLHAFD.GCP (Este nombre puede variar)
- %ProgramFiles%WINDOWS NTSVCHOST.EXE

[37] https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2011-3544
[38] http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2008-4250

El informe [27] presenta el código del archivo “MSC.BAT”, que contiene en su primera línea un identificador clave, relacionado con la interpretación del caracteres cirílicos. Se trata del comando utilizado para cambiar la tabla de caracteres de referencia para codificación o “codepage”, en este caso concreto se ejecuta el valor “1251”.

MSC.BAT file has the following contents:
chcp 1251
:Repeat
attrib -a -s -h -r "%DROPPER_FILE%"
del "%DROPPER_FILE%"
if exist "%DROPPER_FILE%" goto Repeat
del "%TEMP%msc.bat"

Para permitir la comunicación con el C&C, el archivo "LHAFD.GCP", cifrado bajo RC4 y comprimido con la librería “Zlib”, es el que funciona como *Backdoor*. Este código es descifrado y cargado en memoria con el módulo de carga, “svchost.exe”, quedando listo para la comunicación con el C&C. A continuación se muestra un diagrama de esta secuencia.

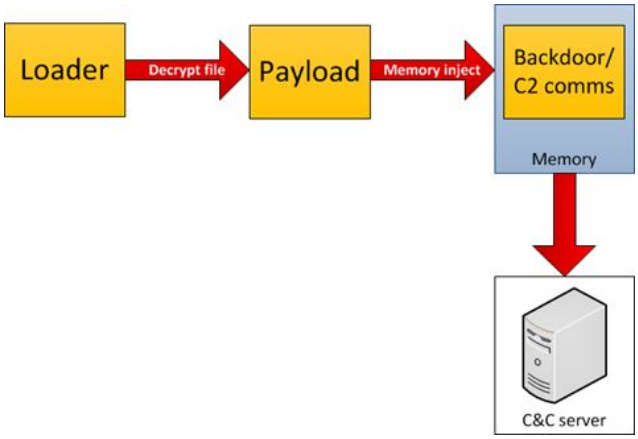


Ilustración 26 Secuencia de carga del *Backdoor* en memoria [27]

Si el *malware* no valida una conexión con el exterior no inicia su actividad. Los dominios con los que establece conexión están relacionados con tres host de Microsoft y se presentan a continuación [26]

- update.microsoft.com
- www.microsoft.com
- support.microsoft.com

Ilustración 27 Dominios de conexión del *malware* [27]

Tras la identificación de una conexión válida, se inicia la comunicación con el C&C tal y como muestra la siguiente captura de tráfico de red.

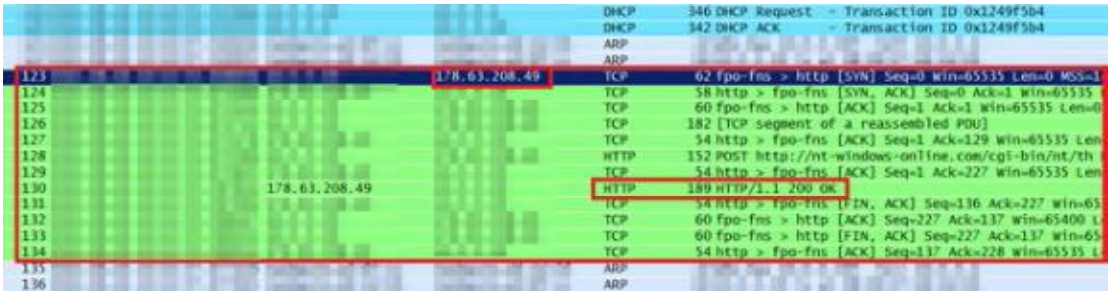


Ilustración 28 Paquete de comunicación con el C&C [27]

La comunicación entre el equipo infectado y el C&C se realiza de forma cifrada. Se utilizan distintos tipos de cifrado. A continuación se muestra una captura de un paquete cifrado entre el *malware* y el C&C.

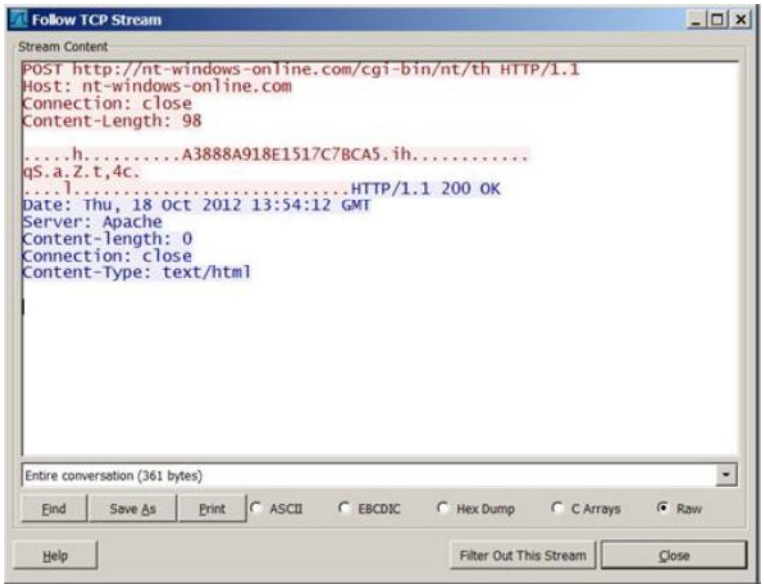


Ilustración 29 Paquete cifrado entre *malware* y C&C [27]

Llegados a este punto, el *malware* comienza la adaptación al sistema mediante la descarga y activación de módulos online y/o offline, previamente descritos, iniciándose la segunda fase de operación.

Durante esta fase el *malware* muestra una característica especial y es que asigna un ID único a cada una de las víctimas. A continuación se muestra una captura de la fase de análisis de código en la que se muestra un ID. Esta particularidad, permite al analista aislar estos IDs y dibujar un mapa de víctimas.

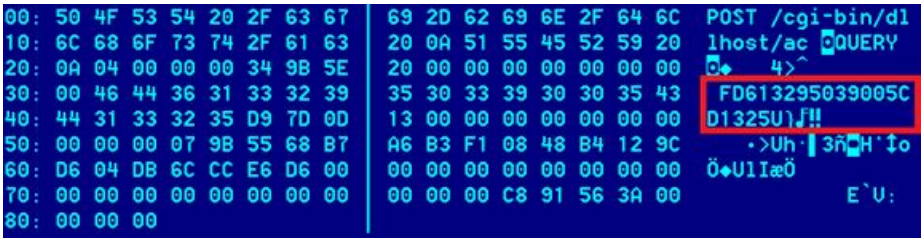


Ilustración 30 Identificador concreto de una víctima [27]

Hasta este momento toda la información obtenida está relacionada con el equipo infectado. Es necesario ampliar el alcance y obtener información sobre la distribución y operación de los C&C. En el apartado dedicado a presentar “Octubre Rojo” y que presenta información obtenida durante esta fase y en particular, esta acción, se muestra una arquitectura muy compleja basada en proxies y dominios que permiten controlar cada una de las víctimas. Siendo esta información relevante, se incluye en el Anexo I un listado de los principales dominios en los que se han registrado comunicaciones relacionadas con este APT, así como las direcciones IP desde las que se han dado dichas comunicaciones. Es importante destacar que durante el análisis, se identificó que la redirección entre proxies se realiza utilizando el puerto 40080 [25].

Para finalizar con la información recabada de fuentes abiertas, se presenta un diagrama de operación en el que se muestra el flujo y los principales archivos activos durante las 2 fases de la operación, indicando su objetivo. Este diagrama se ha realizado tomando como base la imagen presentada en el documento [15].

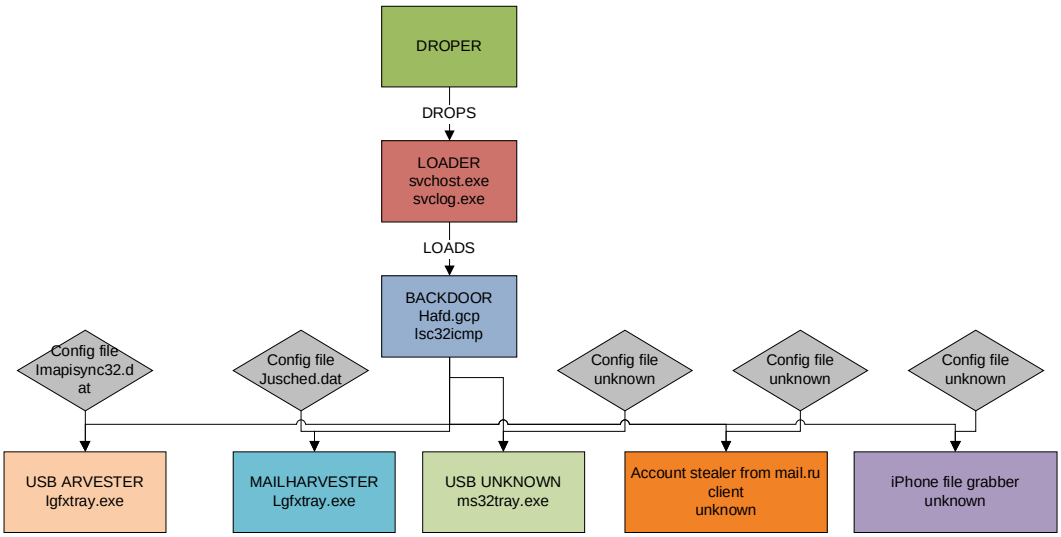


Ilustración 31 Proceso de Infección y Carga de Módulos "Red October"

7.2.5 Búsqueda de cadenas de texto

El objetivo de esta tarea, tal y como se menciona en la Tesis Doctoral [1], es el identificar cadenas que muestren información relevante sobre el *malware*, como pueden ser directorios, direcciones IP o la identificación de palabras específicas relacionadas con paquetes *malware* previamente detectados entre otros. El objetivo es identificar cadenas sospechosas para localizarlas durante el análisis de código y profundizar en su comportamiento y utilización dentro del *malware*. Para ello se va a hacer uso de las dos herramientas propuestas, "Bintext" y "Strings". Los resultados de este tipo de análisis son muy extensos, por ello se presentan a continuación las cadenas más relevantes y en el Anexo J se presentan fragmentos con el resto de cadenas consideradas interesantes.

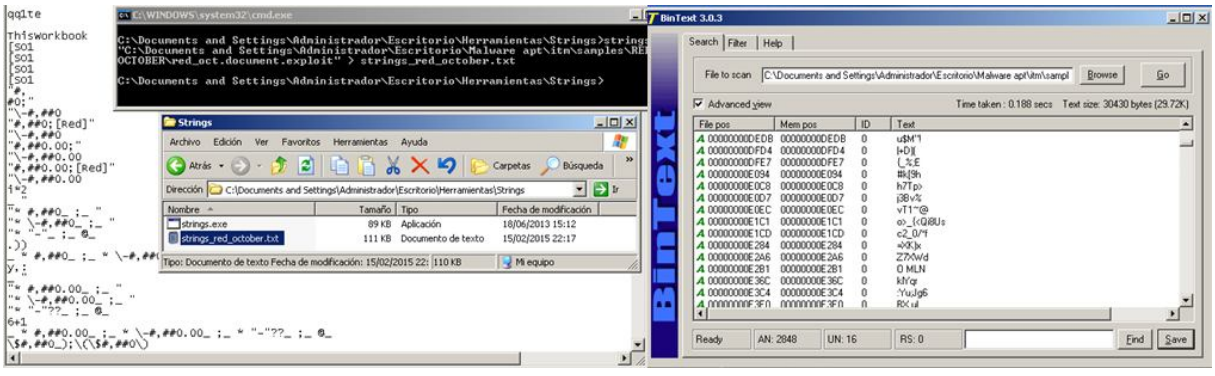


Ilustración 32 Obtención de cadenas de texto, "Strings" y "Bintext"

Teniendo en cuenta que las muestras que se disponen incluyen el *exploit* y el *dropper*, se van a obtener las cadenas de texto de ambos ficheros e identificar aquellas que se consideren más relevantes.

ARCHIVO	STRING	POSICION	COMENTARIO
red_oct.document.exploit	CHINA	000000088590	El exploit se reutilizó de otro malware cuya característica era el documento había sedio editado por una aplicación excel china "simplified Chinese Excel"
	]\[ZYXWVUTSRQPONMLKJIHGFEDCBA@?>=<;:9876543210/.-,+*)('&%$#"!	Multiples y agrupadas	Se repite un conjunto de cadenas que representan todos los caracteres imprimibles ASCII
	~}|{zyxwvutsrqponmlkjihgfedcba	Multiples y agrupadas	
	! %s	00000001AFDE	Puede ser utilizado como carácter fin de linea para construir ataques de desbordamiento y sobreescribir código malicioso en la memoria
	@j[ip	000000032D95	Puede estar relacionado con un direccionamiento IP
	%~}%X9	000000063081	La estructura en la que se utilizan los símbolos % recuerda a la definición de variables
	EXCEL.EXE	000000002E5A	Identificación de la aplicación Excel
	Microsoft Excel	0000000042AC	Idenitificación de la aplicación y fabricante
	SamLab.ws	000000084C2B	Directamente se accede a una web rusa
red_oct.bin.drop	KERNEL32.DLL	00000000172C	Librería DLL sospechosa
	EncodePointer	00000000171C	Cadena relacionado con codificación
	DecodePointer	00000000173C	Cadena relacionda con decodificación
	CreateThread	000000017D5E	Función sospechosa
	GetCurrentProcess	000000017D9C	Función sospechosa
	HeapAlloc	000000017DD6	Función sospechosa
	HeapCreate	000000017DE2	Función sospechosa
	HeapFree	000000017DF0	Función sospechosa
	WriteFile	000000017EA2	Función sospechosa
	TlsGetValue	000000017FB4	Función TLS sospechosa. Puede ser utilizada para evitar depuradores
	TlsAlloc	000000017FC2	Función TLS sospechosa. Puede ser utilizada para evitar depuradores
	TlsSetValue	000000017FCE	Función TLS sospechosa. Puede ser utilizada para evitar depuradores
	TlsFree	000000017FDC	Función TLS sospechosa. Puede ser utilizada para evitar depuradores
	TerminateProcess	0000000180BC	Función sospechosa
	Sleep	000000018150	Cadena sospechosa que puede manifestar comportamiento
	HeapReAlloc	000000018168	Función sospechosa
	<program name unknown>	0000000016E8	Descriptor sospechoso del programa
	IstrcpyW	000000017E10	Función sospechosa
	IstrlenW	000000017E1C	Función sospechosa
	IsDebuggerPresent	0000000180EE	Cadena sospechosa que puede manifestar comportamiento ante depurador

Tabla 6 Cadenas de texto más relevantes

Una información importante que se extrae de las cadenas de texto, es la identificación de la aplicación de Microsoft Excel. Esta información permite centrar el análisis desde el punto de vista del vector de ataque, relacionado con dicha aplicación donde la vulnerabilidad explotada se identifica como **CVE-2009-3129** propia de Ficheros XLS (MS Excel).

7.2.6 Identificación de técnicas de ofuscación

Los archivos que contienen *malware* hacen uso de distintas técnicas de ofuscación como medida de autoprotección ante una posible detección. Las técnicas que se utilizan principalmente se dividen en 4 categorías, empaquetamiento, cifrado, polimorfismo y metamorfismo. El objetivo con este análisis es poder identificar posibles técnicas utilizadas por el desarrollador que permitan al analista conocer la realidad del archivo que dispone.

La herramienta propuesta es “PEiD”, que se utiliza en archivos PE[39] con el objetivo de identificar técnicas de empaquetado y/o cifrado. Durante el análisis de las cadenas de texto

[39] Archivos Portables y Ejecutables son archivos con formato ejecutable por ejemplo *.exe, *.xls.

realizadas en la acción anterior se observó que el archivo “red_oct.document.exploit” no tenía apenas cadenas de texto ASCII. Este hecho suele ser una característica de programas ofuscados, en contrapartida con los programas legítimos. En el segundo archivo, “red_oct.bin.drop” se detectó un comportamiento similar aunque menos acusado. A continuación se muestran los resultados y se confirma la validez de estos supuestos.

En el caso del archivo “red_oct.document.exploit” la herramienta PEiD no es capaz de identificar el tipo de archivo como PE y analizar el tipo de empaquetado y posible cifrado. Aunque el motor de la herramienta PEiD es potente, se han encontrado un informe de Mandiant [29] en el que muestran un caso similar.

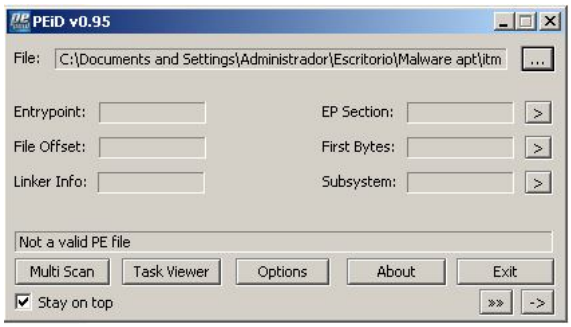


Ilustración 33 Error en análisis de archivo “red_oct.document.exploit”

A continuación se realiza el análisis sobre el archivo “red_oct.bin.drop”. En este caso se obtienen resultados interesantes sobre el tipo de archivo. Es importante destacar que tras este análisis ha sido necesario restaurar el sistema a un punto limpio, aprovechando las acciones llevadas a cabo en la Fase 1, Acciones Iniciales. La restauración se debe a que durante el proceso de intento de desempaquetado del archivo, existía riesgo de ejecución del mismo.

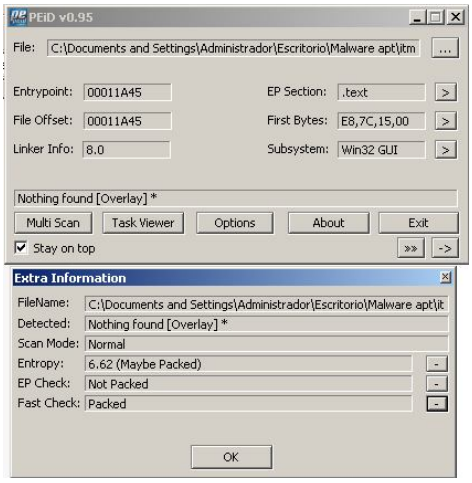


Ilustración 34 Información obtenida del archivo “red_octo.bin.drop”, “PEiD”

En la anterior figura se observa que, si bien la herramienta no ha detectado un método conocido de ofuscación, nos informa de un posible empaquetado de acuerdo al cálculo de la entropía del archivo con un valor de 6,62.

Observando las secciones, no se descubre nada extraño, ya que se detectan las secciones comunes como son la sección de texto (.text) donde se escriben las instrucciones, la sección de datos donde se escriben los datos inicializados (.data) y la sección bss donde se escriben las variables sin inicializar.

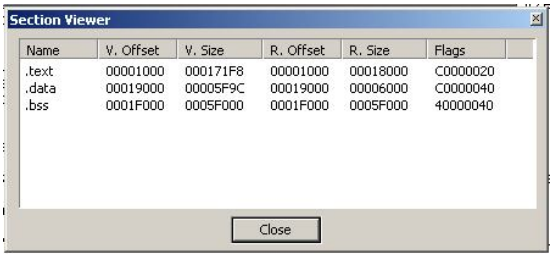


Ilustración 35 Secciones detectadas en el programa (ensamblador), "PEiD"

Es importante analizar las librerías que se importan, en particular, la librería Kernel.DLL con las funciones asociadas e identificadas mediante el análisis de cadenas de texto en el punto anterior.

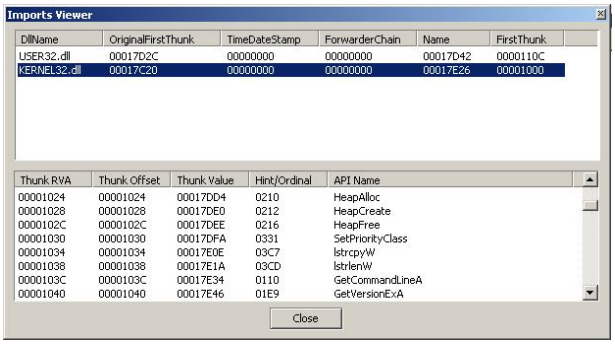


Ilustración 36 Librerías importadas y funciones asociadas, "PEiD"

Por último se detecta la utilización de un algoritmo de cifrado tipo CRC32 en parte del código tal y como se puede ver en la siguiente imagen.

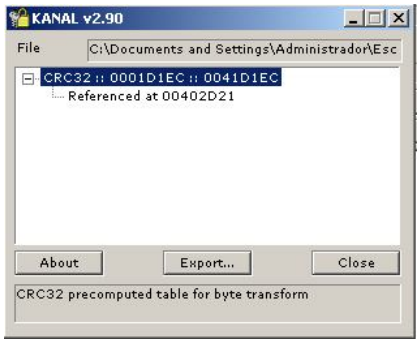


Ilustración 37 Detección de Cifrado CRC32, "PEiD"

Con la información anterior se puede afirmar que el archivo cumple con parte de las principales características de código malicioso, como son el uso de técnicas de ofuscación (empaquetado y cifrado), carencia de cadenas de texto, importación de librerías sospechas y uso de funciones típicas implementadas en *malware*.

7.2.7 Formato y estructura del fichero

En un análisis de *malware* la información del formato y estructura del fichero sospechoso es relevante. Desde el uso y relación de librerías como la inexistencia de versiones del archivo, utilización de fechas inválidas o secciones de entrada sospechosas son parámetros que confirman la existencia de *malware* en el archivo a analizar.

En este caso se va a hacer uso de dos herramientas, "Dependency Walker" que genera una relación de dependencia de todos los módulos existentes y "PEBrowse" que permite la realización de un análisis del archivo por las secciones que lo componen. Teniendo en cuenta que durante la acción anterior el archivo "red_oct.document.exploit" no ha sido reconocido[40] como un archivo PE, este análisis se va a centrar únicamente en el archivo "red_oct.bin.drop".

La información obtenida permite identificar posibles comportamientos sospechosos. La relación de los módulos es compleja ya que se detectan gran cantidad de ellos. Sin embargo, una navegación por los principales módulos detectados nos permite identificar funciones que son utilizadas en programas maliciosos. A continuación se muestran algunas capturas dichas funciones. En el Anexo K se muestra un fragmento del árbol de dependencias generado.

En esta primera captura se observa la utilización de la librería "CRYPT32.DLL" y su función "CryptStringToBinaryW" relacionada con posible código cifrado que es necesario descifrar en tiempo de ejecución de la aplicación que contiene el *malware*.

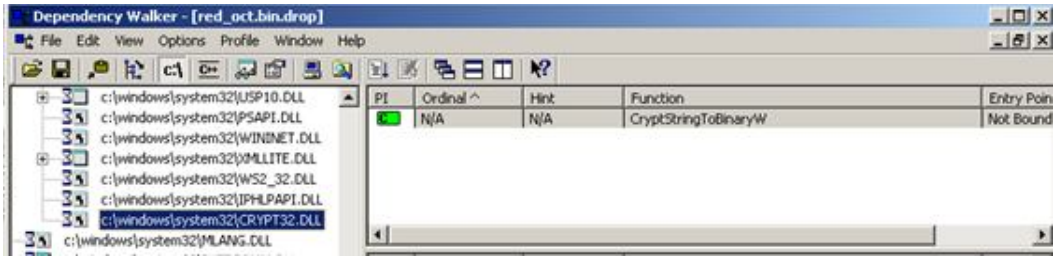


Ilustración 38 Librería CRYPT32.DLL, "Dependency Walker"

[40] Se ha comprobado con la herramienta "Dependency Walker" que se reconoce como PE y es imposible llevar continuar con el análisis.

En esta segunda captura se destaca el módulo “MPR.DLL” (Multi Provider Router), que hace uso de múltiples funciones relacionadas con el establecimiento de distintas conexiones así como la identificación de cambios de clave e inicios de sesión.

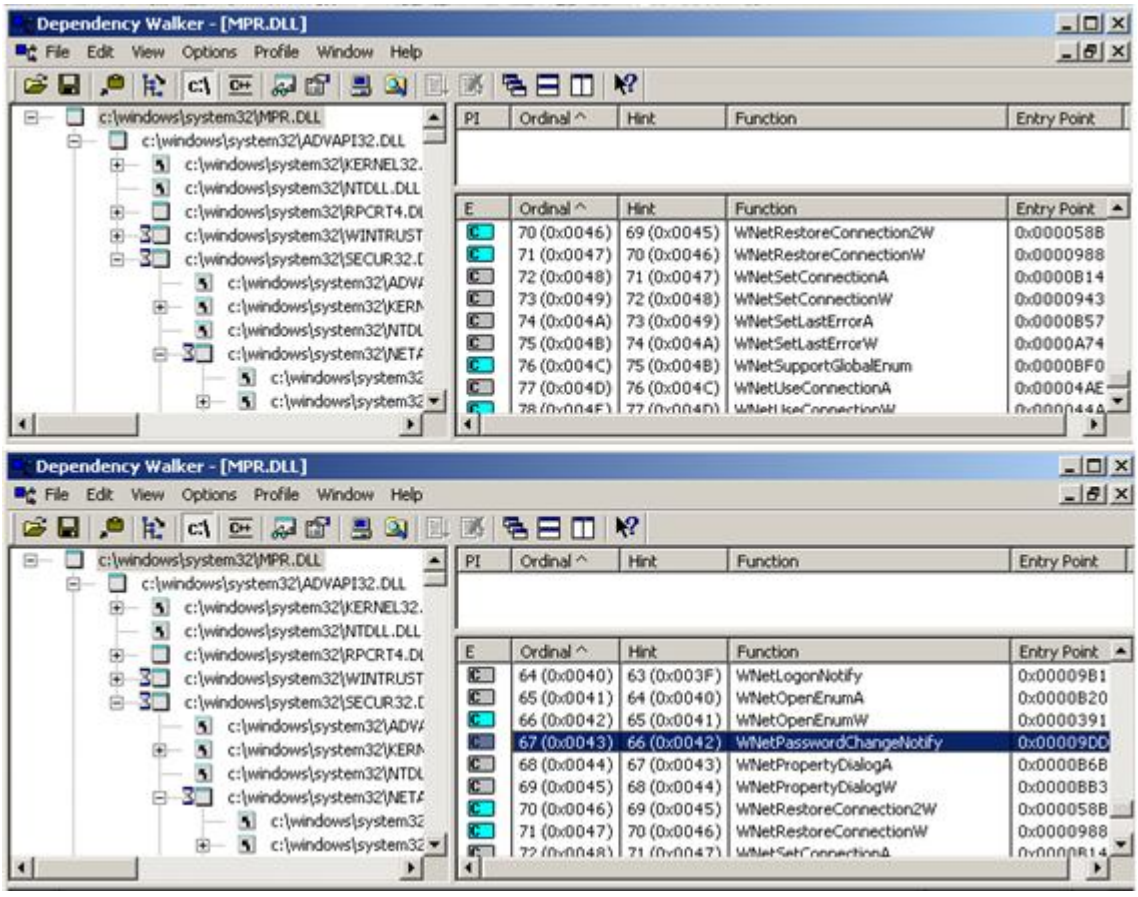


Ilustración 39 Módulo MPR.DLL, “Dependency Walker”

Por último se observa la dependencia de la librería “SHELL32.DLL”, utilizada para abrir distintos componentes de Windows como el explorador y sus controles, archivos y páginas web.

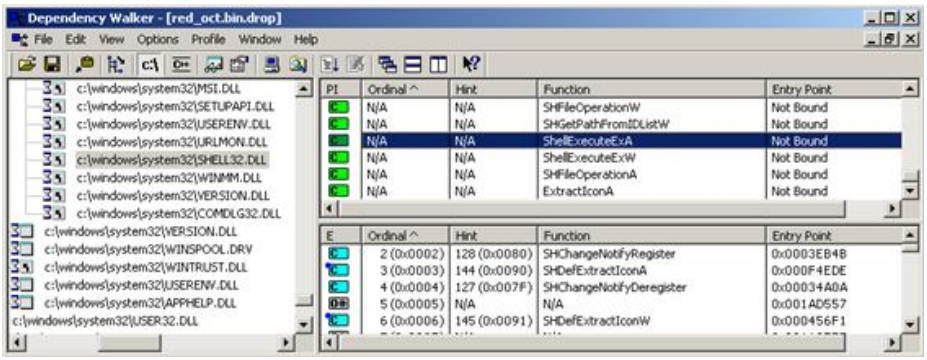


Ilustración 40 Librería SHELL32.DLL, “Dependency Walker”

A continuación se analiza el archivo utilizando la aplicación “PEBrowse”. El objetivo se centra en el análisis de las secciones que componen el archivo. Tras el análisis de todas las secciones correspondientes al archivo analizado la conclusión es que no se puede aportar información relevante, principalmente porque las cadenas de texto no muestran información clara, hecho probablemente relacionado con las técnicas de ofuscación utilizadas.

Se ha realizado también un análisis del archivo utilizando la herramienta “PEViewer” que muestra información de referencia tanto del archivo, como fechas de creación, tipo de archivo y arquitectura como del código contenido. A continuación se muestra una captura con la información general del archivo, confirmando lo que ya se intuía. Teniendo en cuenta que la información obtenida mediante el uso de la “PEBrowse” y parte de la obtenida mediante “PEViewer” no muestra datos significativos, se adjunta en el Anexo L para su consulta.

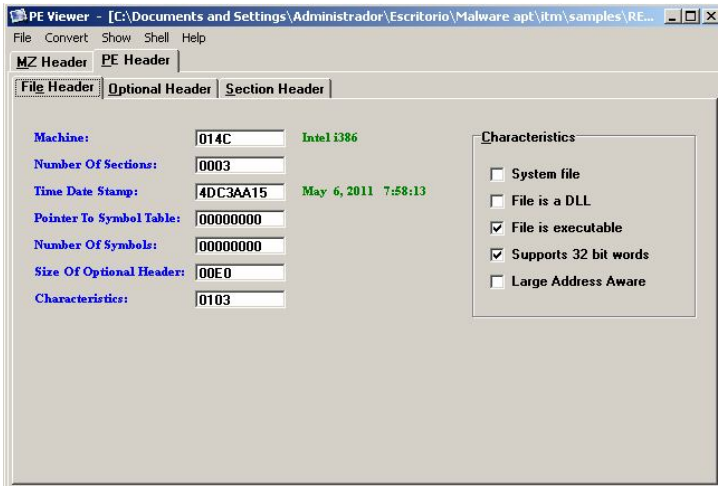
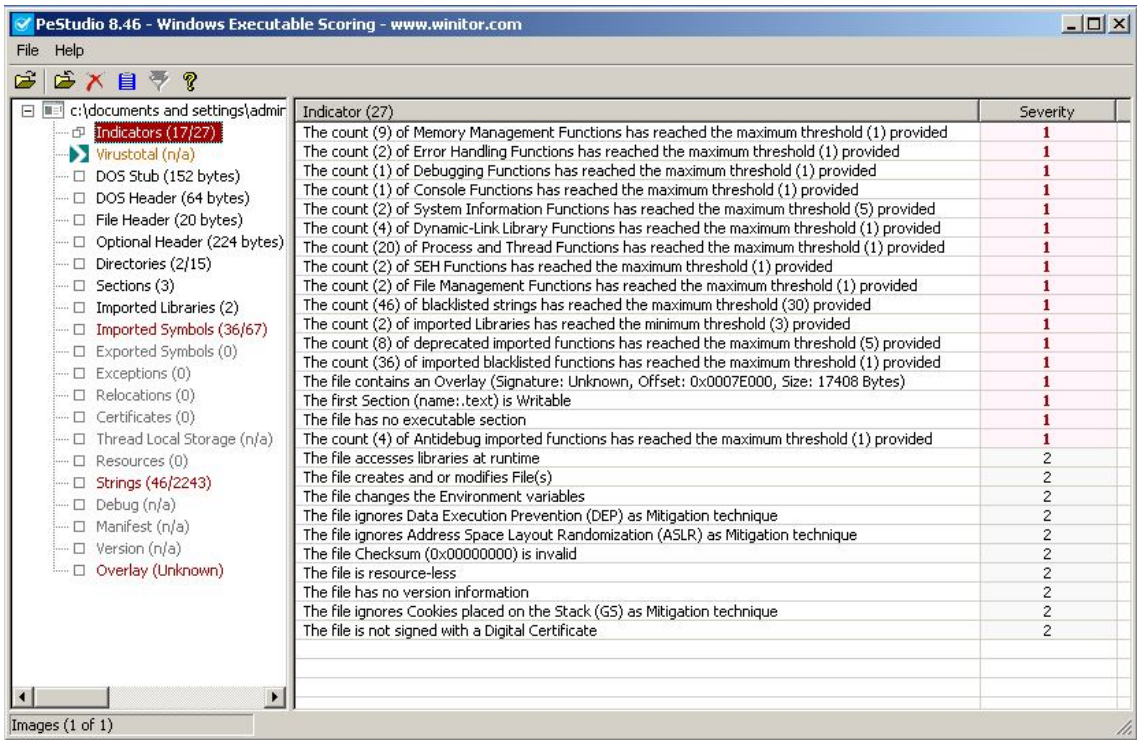


Ilustración 41 Información general de “red_oct.bin.drop”, “PEViewer”

Por último se hace uso de la herramienta “PEStudio” que permite obtener información global del archivo, clasificándolo en función del riesgo. Es cierto que lo indicadores de sospecha no tienen por qué implicar que el código en ejecución tiene asociado un comportamiento no deseado, sin embargo, su conocimiento es de gran valor a la hora de la clasificación del código y su posterior análisis.

En la siguiente captura se presenta lo más parecido a un análisis de riesgo del archivo analizado. Como se puede observar, dicho archivo supera los límites razonables de numerosos criterios de valoración que contempla la herramienta utilizada. Cabe destacar la detección de un parte del código tipo “overlay”, que no es más que una técnica de programación cuyo objetivo es embeber código en bloques diferenciados para que en

tiempo de ejecución se almacene en memoria adicional a la reservada por el propio programa. En el Anexo M se amplía la información obtenida tras este análisis.



Indicator (27)	Severity
The count (9) of Memory Management Functions has reached the maximum threshold (1) provided	1
The count (2) of Error Handling Functions has reached the maximum threshold (1) provided	1
The count (1) of Debugging Functions has reached the maximum threshold (1) provided	1
The count (1) of Console Functions has reached the maximum threshold (1) provided	1
The count (2) of System Information Functions has reached the maximum threshold (5) provided	1
The count (4) of Dynamic-Link Library Functions has reached the maximum threshold (1) provided	1
The count (20) of Process and Thread Functions has reached the maximum threshold (1) provided	1
The count (2) of SEH Functions has reached the maximum threshold (1) provided	1
The count (2) of File Management Functions has reached the maximum threshold (1) provided	1
The count (46) of blacklisted strings has reached the maximum threshold (30) provided	1
The count (2) of imported Libraries has reached the minimum threshold (3) provided	1
The count (8) of deprecated imported functions has reached the maximum threshold (5) provided	1
The count (36) of imported blacklisted functions has reached the maximum threshold (1) provided	1
The file contains an Overlay (Signature: Unknown, Offset: 0x0007E000, Size: 17408 Bytes)	1
The first Section (name:.text) is Writable	1
The file has no executable section	1
The count (4) of Antidebug imported functions has reached the maximum threshold (1) provided	1
The file accesses libraries at runtime	2
The file creates and or modifies File(s)	2
The file changes the Environment variables	2
The file ignores Data Execution Prevention (DEP) as Mitigation technique	2
The file ignores Address Space Layout Randomization (ASLR) as Mitigation technique	2
The file Checksum (0x00000000) is invalid	2
The file is resource-less	2
The file has no version information	2
The file ignores Cookies placed on the Stack (GS) as Mitigation technique	2
The file is not signed with a Digital Certificate	2

Ilustración 42 Información general "red_oct.bin.drop", "PEStudio"

Llegados a este punto se da por finalizada la segunda fase que ha permitido ampliar la información en gran medida sobre la APT "Octubre Rojo". Desde un nivel de funcionamiento y operativa, hasta información de carácter más técnico. Esta fase garantiza un nivel de preparación considerable para el desarrollo de las fases de análisis de código y de comportamiento.

7.3 Fase 3. Análisis de Código

Esta fase tiene como objetivo analizar el código del *malware* utilizando herramientas de depuración para el análisis dinámico y de desensamblado para el análisis estático. De acuerdo a los procedimientos más estandarizados, el análisis de código debe realizarse en 3 pasos diferenciados.

- Comprobación del funcionamiento mediante la lectura de código
- Análisis estático mediante el uso de una herramienta de desensamblado
- Análisis dinámico mediante una herramienta de depuración.

Sin embargo, Don Javier Bermejo en su Tesis Doctoral [1] introduce una innovación basada en la alteración del orden de ejecución de los anteriores pasos, anteponiendo el análisis dinámico al estático. Teniendo en cuenta que el presente piloto experimental tiene como objetivo validar esta propuesta, se procede a realizar los pasos en este último orden. Como avance, el trabajo realizado hasta el momento induce a pensar que esta propuesta es válida, ya que la ofuscación del código, práctica habitual en las APTs, dificulta en gran medida el análisis estático, siendo necesaria la aplicación de técnicas avanzadas de ingeniería inversa para superar los mecanismos de protección del *malware*. Gracias a la metodología propuesta, se ha llegado hasta este punto con un conocimiento relativamente alto de la complejidad del archivo a analizar.

7.3.1 Comprobación del funcionamiento

Tal y como se ha hecho hasta el momento, se va a comprobar el funcionamiento del archivo "red_oct.bin.drop" bajo análisis. Para ello se hace uso de la herramienta "PE Explorer". Se comprueba que el archivo no puede ser desempaquetado por la herramienta tal y como no muestra la consola.

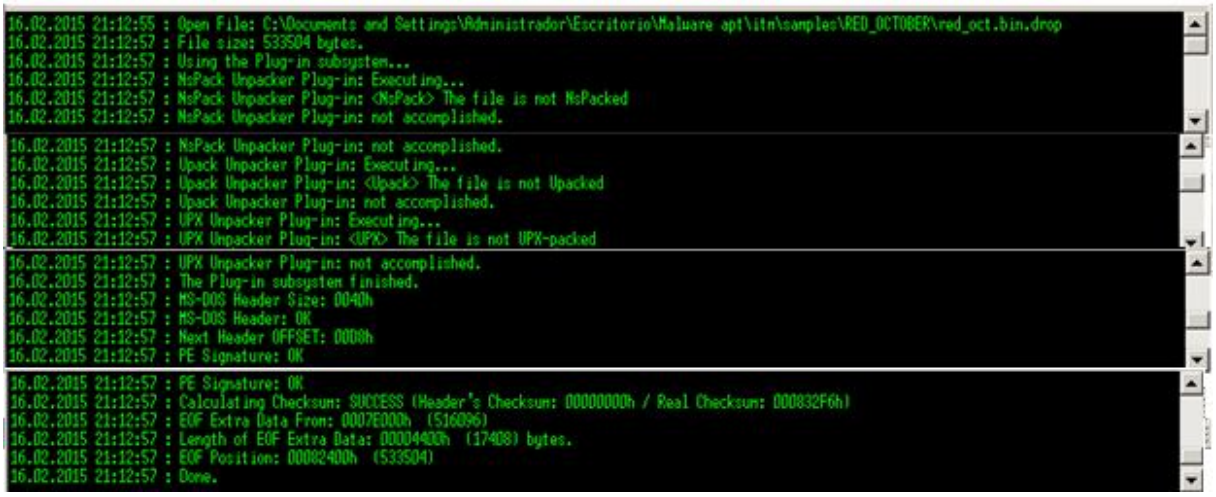


Ilustración 43 Información consola, "PE Explorer"

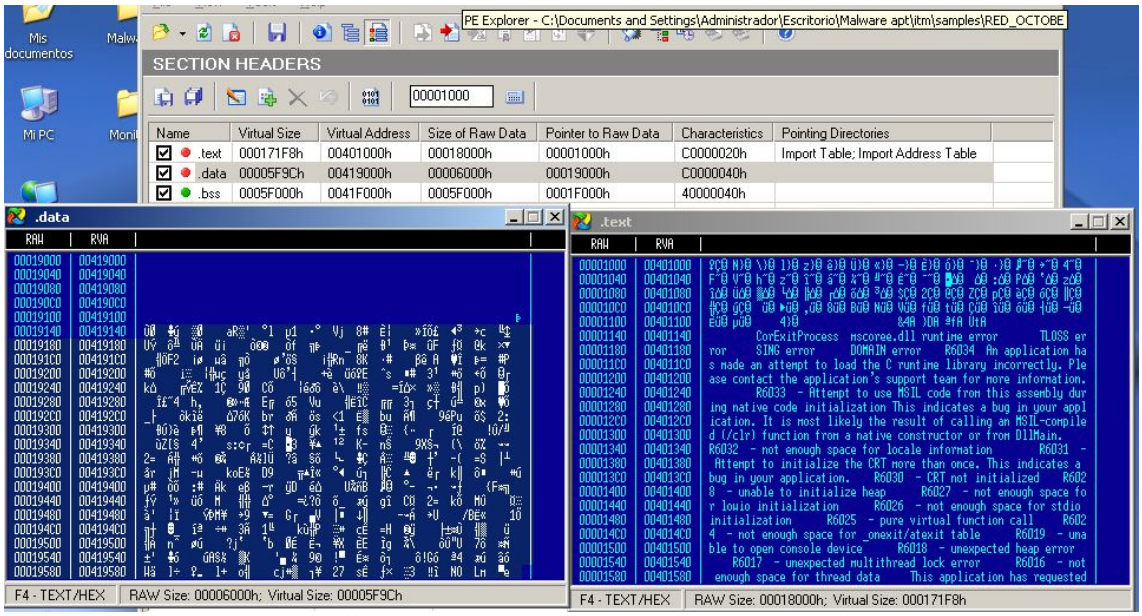


Ilustración 44 Confirmación de que el archivo no ha sido desempaquetado, "PEExplorer"

Teniendo en cuenta este factor, no es posible analizar más de lo que ya se ha realizado, que son las cadenas de texto no codificadas dentro de las tres secciones previamente detectadas. Sin embargo, este análisis ha dado tres informaciones interesantes, por un lado, la utilización de un empaquetado no convencional, ha validado la fecha de creación del mismo (06/05/2011 7:58:13), y ha identificado un punto de entrada distinta (00411A45) al mostrado en la fase de clasificación (00011A45). Este último punto es una característica típica de código malicioso. En este punto se da por finalizado este paso.

7.3.2 Análisis dinámico de código

En este paso se procede a analizar el archivo haciendo uso de la herramienta de depuración "Ollydbg" que permite navegar por el código como si éste se estuviese ejecutando, pero sin necesidad de hacerlo.

En los puntos anteriores se ha confirmado la utilización de técnicas de ofuscación que complican en gran medida un análisis estático. De la misma forma, se han identificado funciones con el objetivo de cifrar y descifrar código en tiempo de ejecución, por ello se espera poder obtener más información sobre el comportamiento del *malware* utilizando este método.

Durante el primer análisis, se ha procedido a navegar por todo el código con la intención de obtener una idea generalizada. El resultado es que se ha detectado una medida de autoprotección adicional y es que el archivo es capaz de detectar que se encuentra en modo depuración y se autoelimina, tal y como se muestra en la siguiente captura cuando se ha lanzado la orden de reinicio de la depuración.

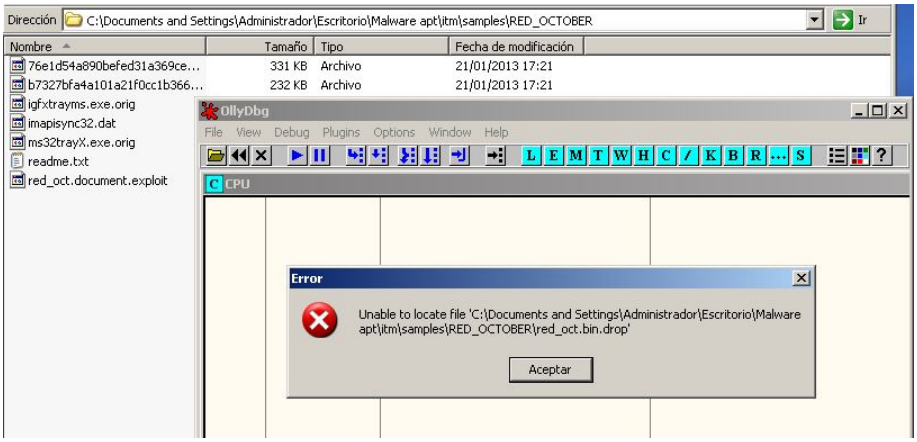


Ilustración 45 Medida de autoprotección del *malware* en modo depuración, "Ollydbg"

Durante la fase de análisis dinámico de código se ha podido constatar la referencia a los tres ficheros característicos del *dropper* "Red October", confirmando el funcionamiento de la primera fase de infección tal y como mencionaban las fuentes abiertas consultadas. A continuación se muestran las capturas de pantalla en las que se han detectados dichas referencias.

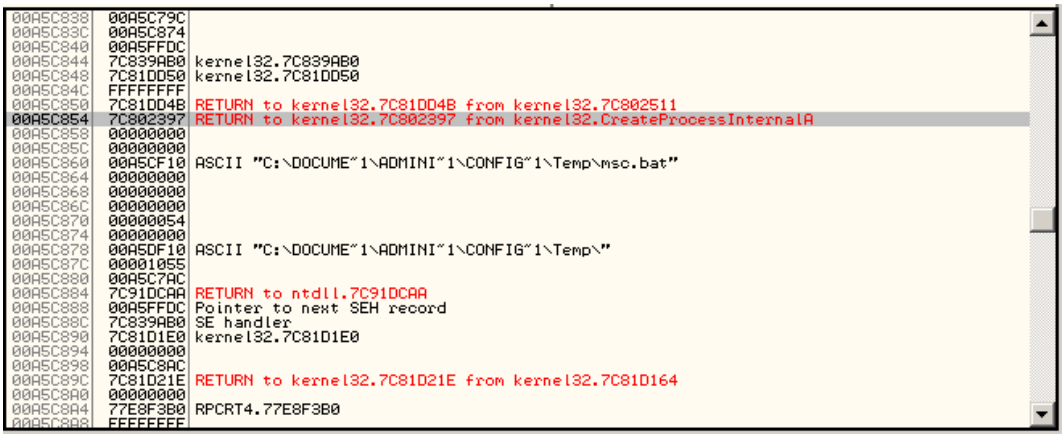


Ilustración 46 Referencia al archivo "msc.bat" característico de "Red October", "Ollydbg"

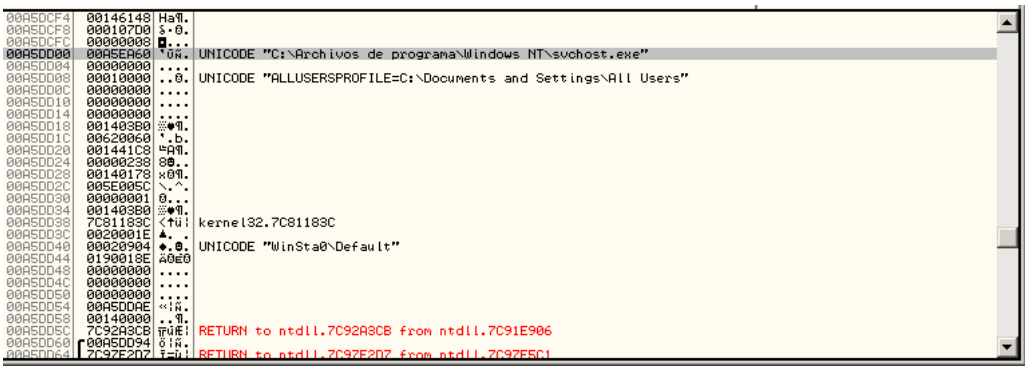


Ilustración 47 Referencia al archivo "scvhost.exe" característico de "Red October", "Ollydbg"

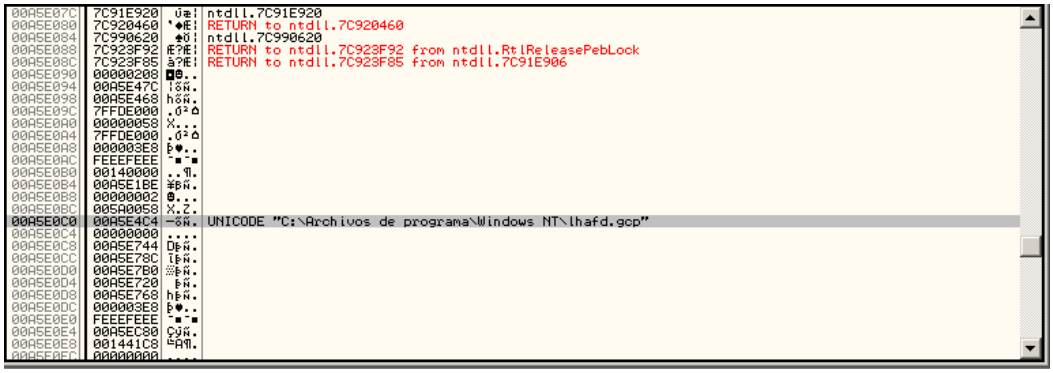


Ilustración 48 Referencia al archivo "lhfd.gcp" característico de "Red October", "Ollydbg"

Junto con estas referencias se han podido navegar por otras partes del código relacionadas con la actividad del *malware* como el acceso a DLL[41]s propensas a ser infectadas por *malware* del tipo *Backdoor*, como es el caso de RPCRT4.DLL, funciones previamente identificadas como sospechosas, cadenas de texto relacionadas con posibles servicios de comunicación ("sshd") o mensajes codificados. Las evidencias de las anteriores afirmaciones se presentan en el Anexo N.

Por el contrario, no se ha podido obtener el código de los archivos anteriormente mencionados y que serían de gran interés para el analista profesional. Debido a la complejidad de la APT que se está analizando, junto con la no especialización en técnicas de ingeniería inversa del autor del presente trabajo, no ha sido posible identificar el método de decodificación del código ofuscado y la identificación correcta de su ubicación. Sin embargo, y aunque la propia incapacidad de llevar a cabo este análisis es un resultado en sí mismo, se considera que queda fuera del objetivo principal del presente piloto experimental, tal y como se han definido previamente. De todos modos, el conocimiento obtenido durante

[41] http://support.microsoft.com/kb/815065/es

esta acción se aprovechará en las siguientes fases con el objetivo de mejorar el análisis de la APT “Octubre Rojo”.

7.3.3 Análisis estático de código

En este paso se procede a analizar el archivo haciendo uso de la herramienta de desensamblado “IDA42 Pro” (versión de evaluación). Como era de esperar teniendo en cuenta los resultados de la fase anterior, no se ha podido obtener información sobre la creación y contenido de los archivos que se distribuyen por el sistema infectado. Sin embargo este análisis ha permitido recabar información sobre la estructura del archivo que se está analizando de forma detallada. Tanto la identificación de las posiciones de memoria de las funciones que más interés han generado, como la relación entre dichas funciones y con otros parámetros, ofrece una información que, utilizada por un analista profesional, puede ayudar en el análisis dinámico del código. Este hecho pone de manifiesto que podría darse un ciclo iterativo entre estos dos análisis con el objetivo de ir acotando el problema y mejorando los resultados. Si bien, se considera un resultado importante, puede estar condicionado por el desequilibrio entre el conocimiento del analista que está llevando a cabo estas pruebas y el nivel de complejidad del código a analizar.

A continuación se muestra información gráfica de los resultados más relevantes obtenidos tras esta fase, ampliándose en el Anexo O. Con la muestra de los resultados considerados como interesantes.

En primer lugar se presenta la estructura del archivo donde se puede confirmar la composición de cada una de las secciones. Se observa que existe código ofuscado en cada una de ellas, principal causante de la dificultad del análisis del *malware*. Así mismo se detectan bloques de memoria vacíos dentro del código. A continuación se muestra información gráfica de dicha composición.

Ilustración 49 Estructura del código analizado, “IDA Pro”

[42] https://www.hex-rays.com/products/ida/

Durante el análisis se ha identificado la parte de código relacionada con una función llamada "IsDebuggerPresent", almacenada en la posición de memoria 004180F0, que a priori, sería la encargada de hacer desaparecer el archivo al detectar que está siendo sometido a un proceso de depuración, tal y como se constató en el anterior apartado. Uno de los objetivos importantes desde el punto de vista de la fase de análisis de código para este *malware* sería el neutralizar la ejecución de esta función detectada y modelada como se muestra a continuación, para poder avanzar en el análisis dinámico de código.

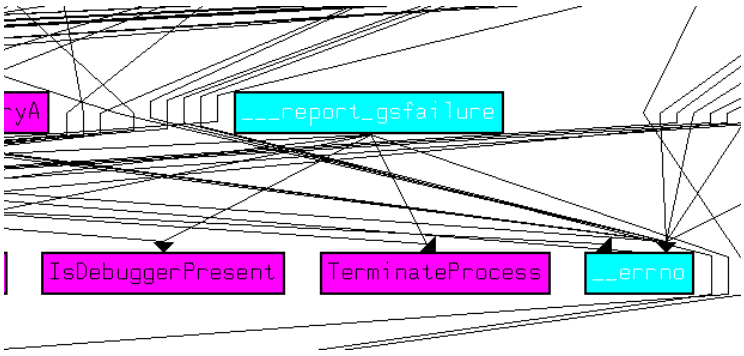


Ilustración 50 Identificación de "IsDebuggerPresent" en el flujo de llamadas de funciones, "IDAPro"

```
.text:004180F0 ; ---------------------------------------------------------------------------
.text:004180F0                 inc     esp
.text:004180F1                 bound   esi, gs:[ebp+67h]
.text:004180F5                 db      67h, 65h
.text:004180F5                 jb      near ptr 8149h
.text:004180F9                 jb      short near ptr aVirtualalloc+8
.text:004180FB                 jnb     short near ptr aVirtualalloc+0Ah
.text:004180FD                 outsb
.text:004180FE                 jz      short $+2
.text:00418100                 push    edx
.text:00418100 ; ---------------------------------------------------------------------------
```

Ilustración 51 Código relacionado con la función "IsDebuggerPresent" (004180F0), "IDA Pro"

Durante la fase de consulta de fuentes externas abiertas se han localizado dos documentos [30] y [31] que presentan los hallazgos más importantes tras el análisis de código dinámico y estático de un archivo *dropper* (msmx21.exe) perteneciente al APT "Octubre Rojo". El archivo utilizado en el análisis [30] y [31] no es el mismo que el analizado en este trabajo. Dicho *malware* explota las vulnerabilidades CVE20103333 y CVE20120158, previamente descritas, y ejecuta un proceso idéntico de infección que el archivo que sí se analiza en este trabajo. Este hecho permite mostrarlo como información de interés para el lector ya que se muestran resultados prácticos tras aplicar ingeniería inversa de código al *malware*. El proceso debería ser similar en el caso que compete en este trabajo de fin de máster. Por último es importante destacar que las herramientas utilizadas en los informes [30] y [31] son las mismas que las que se han aplicado en el desarrollo de esta acción.

7.4 Fase 4. Análisis de comportamiento

Esta fase tiene como objetivo analizar el *malware* en tiempo de ejecución, monitorizando los cambios que se realizan mediante comparación con un estado previo. El análisis se centra en monitorizar los cambios producidos por el *malware* en:

- El sistema víctima, mediante el análisis de creación y modificación de archivos y variables del registro
- En la red de comunicaciones, mediante el análisis del tráfico de red y uso de servicios (Web, DNS, FTP…)

Tal y como se ha mostrado en el apartado anterior, el análisis estático no ha permitido obtener información detallada sobre el código malicioso contenido en el archivo "red_oct.bor.drop", por ello, se utilizará este análisis para intentar obtener esta información. Para ello y previo a la ejecución del *malware* se va a tener en cuenta la información obtenida durante las fases anteriores resumida en los siguientes tres puntos:

- Comprobación de la conectividad con el exterior
- Localización de los 3 archivos creados por el *malware* y previamente identificados, "msc.bat" que activa el *malware*, "lhafd.gcp" que realiza las funciones de *backdoor* y "svchost.exe" que carga el código y lo ejecuta durante la operación normal.
- Comprobación de que el *malware* elimina el fichero de inicio de la infección, tal y se observa en el código del archivo "msc.bat"

A continuación se describen las acciones y resultados obtenidos en cada una de las tareas definidas en la metodología para la correcta ejecución de esta fase de análisis de comportamiento.

7.4.1 Tareas Iniciales

Este paso comprende la preparación del escenario previo a la ejecución del análisis. El análisis de comportamiento se ha dividido en 2 partes tal y como se ha comentado al inicio del apartado, siendo necesaria la ejecución y monitorización del *malware* en múltiples ocasiones. No es obligatorio desarrollar el análisis de comportamiento de este modo, pero se ha considerado más efectivo desde el punto de vista del propio análisis y del uso de herramientas.

En primer lugar se ha preparado el entorno de la red, imprescindible para la ejecución correcta del *malware*. Para ello se ha modificado el archivo "host" del equipo víctima para que las consultas a los dominios, que consulta el *malware,* sean resueltos por el sistema de monitorización. En este último sistema se ha tenido que modificar la configuración durante las pruebas, adaptándose a los requerimientos del *malware*, basados en la necesidad de comunicar con un Servidor DNS[43] principal o con autoridad sobre el dominio. Finalmente la herramienta utilizada para monitorizar la comunicación ha sido "Whireshark", tanto en el equipo víctima como en el sistema monitor.

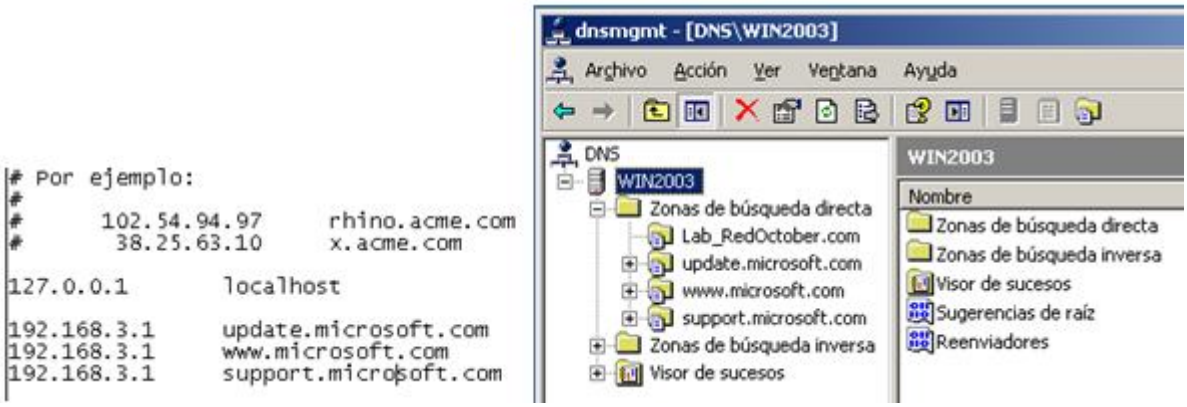


Ilustración 52 Archivo "hosts" modificado en la víctima y configuración servicio DNS

En segundo lugar se prepara el entorno de análisis en la máquina víctima. Estas tareas se han centrado en los siguientes puntos.

- Inicialización a un punto de estado limpio, aprovechando las acciones realizadas en la primera fase de la metodología, para eliminar los cambios que hayan podido darse en las fases anteriores
- Captura del estado inicial del sistema mediante la herramienta "Systracer" y "RegShot", con el objetivo de, tras un tiempo de ejecución del *malware*, monitorizar los cambios.
- Localizar las rutas en las que el *malware* crea los archivos para comprobarlo visualmente.
- Durante la ejecución, utilizar las herramientas "DiskPulse", "ProcessMonitor" y "VMMap" para monitorizar los cambios en tiempo real sobre el sistema.

[43] http://es.wikipedia.org/wiki/Domain_Name_System

7.4.2 Ejecución del *malware*

Este paso comprende la preparación del archivo que contiene el código malicioso y llevar a cabo su ejecución. En el caso del archivo "red_oct.bin.drop" se observa que la extensión no es común, aunque en las fases previas se ha confirmado que es un archivo portable y ejecutable. Se realiza un cambio de extensión de "*.drop" a "*.exe" y se comprueba que se archivo se ejecuta correctamente. Tras, aproximadamente, 15 segundos de ejecución el archivo "red_oct.bin.exe" desaparece del escritorio y del equipo tal y como se esperaba.

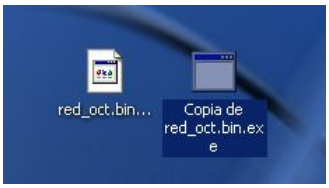


Ilustración 53 Archivo ejecutable con el *malware*

7.4.3 Activación de servicios para el *malware*

Este paso comprende el incremento, gradual, de servicios que puedan ser utilizados por el *malware*. En el caso de la APT "Octubre Rojo", su operación se basa en dos fases, la primera de infección y establecimiento del Backdoor y la segunda de carga de módulos específicos desde el C&C. Teniendo en cuenta que este análisis únicamente se centra en la primera fase, el servicio que se ha ido modificando ha sido el de DNS. Se han realizado tres modificaciones, su desactivación, activación como servidor secundario y activación como servidor principal.

7.4.4 Tareas posteriores a la ejecución

Este paso comprende el análisis de los resultados tras el periodo de ejecución del *malware*. De acuerdo a la metodología, los análisis realizados sobre el *malware* en ejecución han durar alrededor de 20 minutos de monitorización. A continuación se van a mostrar los resultados obtenidos en las distintas fases, manteniendo la clasificación definida al inicio del capítulo.

Análisis de cambios en la víctima

La realización de este análisis se basa en la repetición múltiple del proceso de infección de la víctima. De acuerdo a los objetivos iniciales planteados, tras la realización de varias pruebas se confirma que se han obtenido resultados satisfactorios.

En primer lugar, se ha monitorizado el proceso de infección conocido como fase 1 descrito de manera secuencial en las siguientes líneas.

- **12:04:25**, se ejecuta el *malware* en el equipo. Se crea un proceso con nombre "red_oct-bin.exe" e identificador PID 2340 que inicia el acceso a variables de registro relacionadas con procesos de comunicación. Posiblemente compruebe la existencia de conexión con el exterior antes de continuar con la infección.
- **12:04:30**, comienza la carga de datos en memoria y la ejecución de funciones. El proceso "csrss.exe" con identificador PID 728 inicia un comportamiento extraño no observado antes de la ejecución del *malware*. Se crean los ficheros "svchost.exe", "msc.bat" y "hafd.gcp".
- **12:04:36**, El archivo "svchost.exe" previamente creado se activa como proceso con el identificador PID 2464 y comienza a cargar el código malicioso en memoria.
- **12:04:40**, Se accede al archivo "msc.bat".
- **12:04:41**, Se cierra el proceso PEID 2340 y se elimina el archivo "red_oct.bin.exe". Se crea el proceso con nombre "cmd.exe" y PID 2640, que se encarga de ejecutar múltiples acciones accediendo a las librerías identificadas durante la Fase 3 de la metodología aplicada. Se elimina el archivo "msc.bat"
- **12:19:44**, Hasta este momento no se observa ninguna otra actividad relacionada con el proceso sospechoso. A partir de este momento el proceso "svchost.exe" accede a variables de registro relacionadas con procesos de comunicación. Esta fase coincide con la detección de tráfico en red relacionado con el *malware*, concluyendo que, es en este momento donde se inicia en intento de comunicación con el C&C.
- **12:22:30**, Desde este momento en adelante, el proceso "svchost.exe" inicia un comportamiento cíclico, cada 3 minutos aproximadamente, mediante el cual intenta establecer la conexión con el exterior. Cobra interés la detección del intento de acceso a rutas relacionadas con aplicaciones de navegación web como Opera[44] o Firefox[45], lo que podría significar que la muestra estaba orientada a una víctima que utilizase este tipo de navegador.

[44] http://www.opera.com/es-es
[45] https://www.mozilla.org/es-ES/firefox/new/

Tras la identificación de la secuencia de infección, se procede a analizar los archivos creados y su importancia en la infección.

- El archivo "**msc.bat**" permanece de forma temporal en el sistema. La identificación del proceso ha permitido aislarlo y comprobar su contenido, que coincide por el mostrado en las fuentes abiertas consultadas en las fases anteriores [30] y [31]. Este archivo es el encargado de activar la infección dentro del sistema.

Ilustración 54 Contenido del archivo "msc.bat"

- El archivo "**svchost.exe**" es un archivo que no se puede eliminar del sistema. Se ejecuta de forma automática creando un proceso que contiene código cifrado. Este hecho, junto con el acceso a variables de registro relacionadas con el cifrado, induce a pensar que la decodificación del *malware* se realiza en tiempo real, no siendo posible la obtención de información de este proceso de formar sencilla mediante este análisis. Será necesario recurrir al análisis de código.
- El archivo ""**hafd.gcp**" es un archivo oculto y residente. Se accede a él únicamente durante la fase de infección. Si se elimina, el *malware* continua comportándose de la misma manera, incluso después del reinicio. Es un archivo codificado.

Del análisis anterior, se obtienen las siguientes conclusiones:

- El proceso de infección de la APT "Octubre Rojo" se lleva a cabo en unos 15 segundos, a continuación el *malware* permanece inactivo para iniciar la comunicación con el C&C 15 minutos más tarde.
- Durante el proceso de infección se crean 3, "svchost.exe", "msc.bat" y "hafd.gcp".
- En el proceso de infección se elimina el archivo que la inicia, "red_oct.bin.exe" y el que la activa "msc.bat". Sin embargo se mantienen ocultos los archivos "svchost.exe", "msc.bat" y hafd.gcp".

Durante esta parte del análisis de comportamiento se ha hecho uso de la herramienta "DiskPulse", de gran utilidad para identificar los cambios en los archivos y relacionarlos, mediante las marcas de tiempo, con la herramienta "ProcessMonitorr".

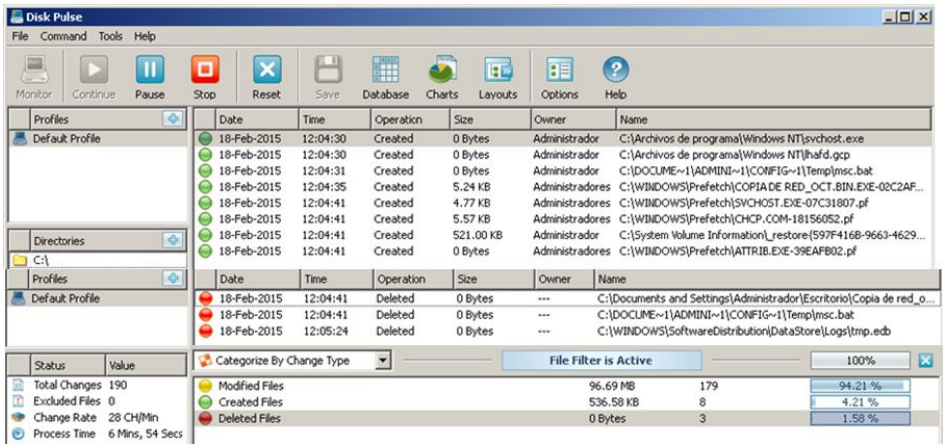


Ilustración 55 Monitorización de modificación archivos, "DiskPulse"

La secuencia completa de infección basada en la herramienta "ProcessMonitor" se incluye en el Anexo P. La información relativa a los archivos creados por el *malware* se incluye en el Anexo Q

Durante el análisis en tiempo de ejecución, se ha utilizado la herramienta "VMMap" para monitorizar el proceso malicioso "svchost.exe" con PID 2464. Mediante esta herramienta se ha accedido a los *strings* del proceso que ha aportado información valiosa, tal y como se puede observar en la siguiente imagen, donde se presentan cadenas de texto almacenadas en la memoria del proceso, sección "private data" (00400000-00421FF) directamente relacionadas con el comportamiento del *malware*. Como *strings* a destacar, los relacionados con el método "Get http" utilizado para descargar módulos desde el C&C o los nombres DNS de los servidores externos que sirven como *proxy* e intermedian la comunicación entre la víctima y el C&C. Este hecho confirma que el proceso creado durante la secuencia de infección contiene código malicioso orientado a la creación de un *Backdoor* para comunicar con el C&C. En el Anexo R se incluye información extendida sobre las cadenas de texto capturadas.

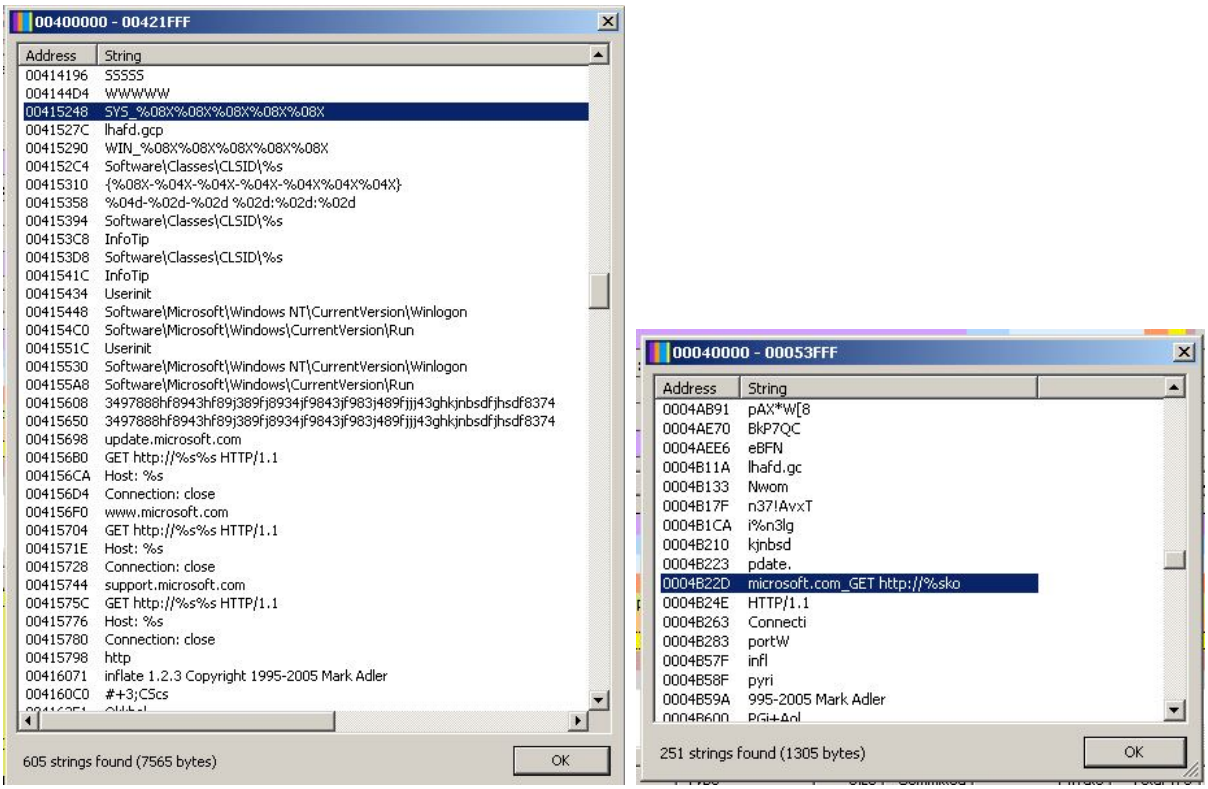


Ilustración 56 Monitorización del proceso PID 2464, "VMMap"

Dentro de este análisis se han comprobado las modificaciones que se han realizado en el sistema mediante la comparación del estado previo y posterior a la infección. Para ello se ha hecho uso de dos herramientas, "Systracer" y "RegShot". Durante esta fase del análisis, se han confirmado los resultados comentados anteriormente y se añaden hallazgos relacionados con las modificaciones de las variables de registro.

Se observa que el *malware* mayoritariamente modifica variables de registro de configuración de usuario (HKU) y del equipo local (HKLM). Estas variables contienen tanto la configuración del usuario como de las aplicaciones que se albergan en el sistema. Se concluye que el *malware* modifica dichas variables para permitir las ejecución de todas las tareas asociadas al proceso de infección mediante la utilización de los permisos del usuario A continuación se muestra una selección de las variables modificadas relacionadas directamente con los archivos de *malware* creados.

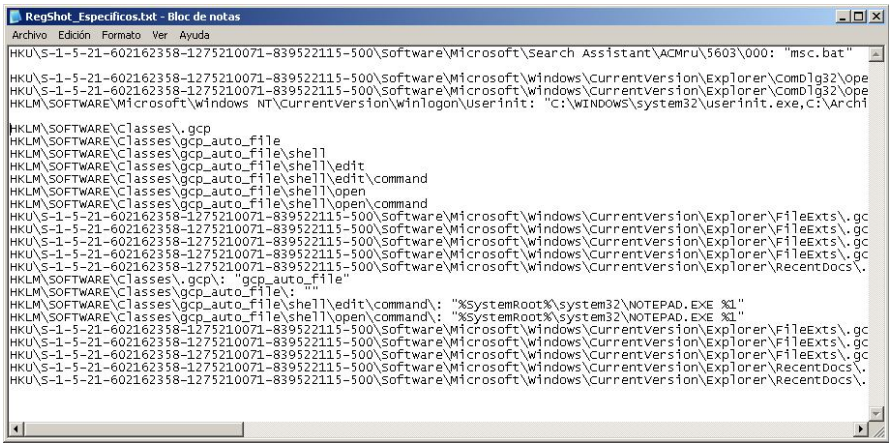


Ilustración 57 Selección de modificaciones en registro por el *malware*, "Regshot"

Gracias a la herramienta "Systracer" es posible detectar las siguientes modificaciones relevantes como se muestran en las siguientes capturas de pantalla.

- Detección de modificaciones del registro con valores sospechosos, relacionados con el código malicioso bajo análisis.

Application not registered			add
(Default)	REG_SZ	"C:\Archivos de programa\Microsoft Office\OFFICE11\EXCEL.EXE" /e /n	add
command	REG_MULTI_SZ	1ugAVn-}f(ZXfeAR6.jiEXCELFiles>!De@]Vz(r=f`1lfq`?R& /e /n	add
Application not registered			add
(Default)	REG_SZ	[new("%1")]	add
Application not registered			add
(Default)	REG_SZ	Excel	add
Application not registered			add
(Default)	REG_SZ	system	add
Application not registered			add
(Default)	REG_SZ	&Abrir	add
Application not registered			add
(Default)	REG_SZ	"C:\Archivos de programa\Microsoft Office\OFFICE11\EXCEL.EXE" /e	add
command	REG_MULTI_SZ	1ugAVn-}f(ZXfeAR6.jiEXCELFiles>!De@]Vz(r=f`1lfq`?R& /e	add

Ilustración 58 Modificaciones sospechosas en el registro, "Systracer"

- Detección de la nueva aplicación relacionada con la creación del proceso malicioso "svchost.exe"

Name	Version	Company	Status	Startup	File name	Modified	File size	Info
Application not registered								mod
svchost.exe					C:\Archivos de programa\Windows NT\svchost.exe	2012-10-03 05:58.02	204,800	add

Ilustración 59 Nueva aplicación detectada, "Systracer"

- Detección de la creación de dos nuevos archivos. Sólo se identifican los archivos que residen en el sistema ya que, como se ha confirmado previamente, el archivo "msc.bat" es eliminado durante el proceso de infección.

Application not registered				mod
hafd.gcp	2012-10-03 05:58.02	74,554	RHS--	add
svchost.exe	2012-10-03 05:58.02	204,800	RHS--	add

Ilustración 60 Archivos nuevos detectados, "Systracer"

Llegados a este punto se finaliza el análisis de comportamiento en la víctima con buenos resultados. Estos resultados confirman la veracidad de la información recogida de fuentes abiertas e incluso sirve para aportar mayor información sobre el *dropper* analizado. En los informes [30] y [31] se analizada un archivo con código malicioso relacionado con la APT "Octubre Rojo", pero desarrollado para aprovechar otra vulnerabilidad. En este caso se ha comprobado que, aun siendo el vector de ataque diferente, el contenido del *malware* es el mismo como demuestra el proceso de infección, la tipología de los archivos creados y el contenido del archivo "msc.bat".

Análisis del tráfico de red

Durante este análisis se ha obtenido información sobre las peticiones, por parte del *malware*, de conexión con los servidores previamente identificados. Es importante destacar que el *malware* ha iniciado, en las distintas repeticiones del análisis, los intentos de comunicación aproximadamente 19 minutos después de la ejecución del *malware*. Aunque el dato en sí es totalmente dependiente de los recursos de la máquina víctima, este comportamiento induce a pensar en que el proceso de inicio de comunicación está programado, dejando un tiempo de guarda entre la infección del equipo y el inicio de intento de comunicación con el C&C. Este análisis se ha dividido en 3 partes.

La primera se corresponde con la desactivación del servicio previa a la ejecución del archivo infectado. En esta fase se ha detectado que el *malware* realiza consultas de forma aleatoria al Gateway intentando conectar con los servidores DNS que operan como proxy.

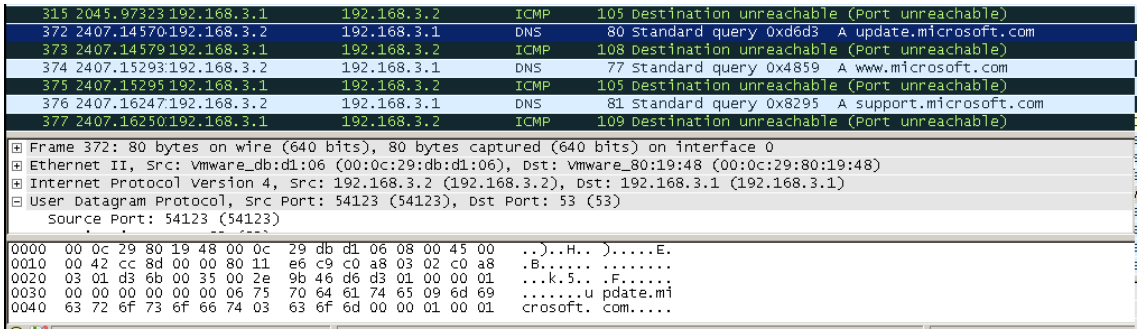


Ilustración 61 Intento de comunicación *malware* con C&C

En la segunda prueba se activa el servicio y se observa que el *malware* interactúa con el servidor DNS. Requiere que éste se presente como servidor principal del dominio tal y como se puede ver en la siguiente captura. Esto muestra una medida de autoprotección ante posibles servidores de monitorización que puedan utilizarse para prevenir de esta amenaza.

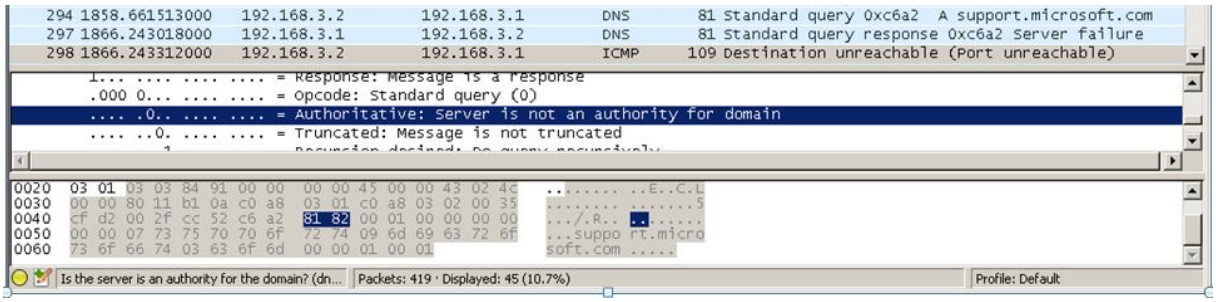


Ilustración 62 Petición de confirmación de conexión con servidor principal

Por último, se muestra la comunicación entre el equipo víctima infectado y el servidor DNS cuando este se presenta como servidor autorizado. Tras esta confirmación no se han conseguido reproducir más intentos de comunicación. Teniendo en cuenta que la arquitectura de comunicación con el C&C se basa en una red de *proxies* compleja, posiblemente el *malware* esté esperando algún tipo de confirmación remota para poder enviar el ID único de la víctima y comenzar la descarga de módulos. Se detecta que en este caso, los intentos de comunicación se hacen de forma periódica con un intervalo de tiempo aproximado de 1752segundos.

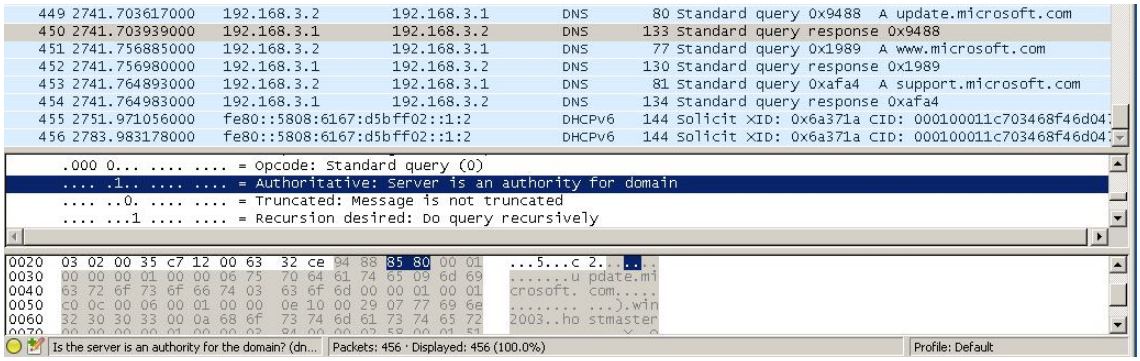


Ilustración 63 Comunicación válida con el servidor DNS.

Por último, se ha obtenido otro resultado como consecuencia de la prueba realizada sobre el equipo víctima, que consiste en la eliminación del archivo “lhafd.gcp“y monitorización del comportamiento del *malware*. Tras la eliminación y teniendo en cuenta que el proceso “svchosts.exe” infectado permanecía activo, no se observó ninguna alteración del comportamiento. Tras el reinicio de la máquina víctima, se monitorizó la red para ver si se daba alguna alteración, sin embargo, transcurridos los aproximados 15 minutos tras el arranque, las peticiones DNS volvieron a recogerse, concluyendo que eliminación del archivo anteriormente mencionado no afecta al comportamiento del *malware* en lo que a comunicación se refiere. Información extendida sobre este análisis se incluye en el Anexo S.

7.4.5 Volcado y análisis de memoria

Este es el último paso del análisis de comportamiento y se basa en el volcado y análisis de la memoria capturada tras un proceso completo de infección nunca inferior a 20 minutos. Para llevar a cabo esta tarea se ha utilizado la herramienta "Winpmem", que permite el volcado de memoria, junto con la herramienta "Volatility", instalada en el sistema Servicios, para su análisis.

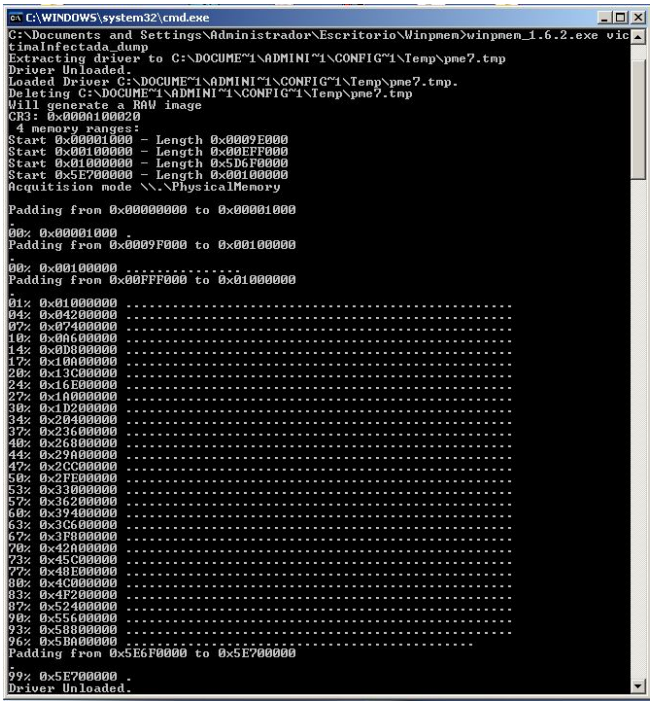


Ilustración 64 Volcado de memoria de la máquina infectada, "Winpmem"

El análisis de la memoria se ha centrado en la identificación de contenidos sospechosos, aprovechando el plugin "malfind[46]" de la herramienta "Volatility", desarrollado para el análisis de *malware*. Durante este análisis, el proceso infectado "svchost.exe" tenía el identificador PID 1540. Se han obtenido resultados con la ejecución de búsquedas basadas en la identificación de *hooks*[47] tanto en "*user* como en *kernel mode*[48]" e identificación de código o DLLs que hayan sido inyectados o escondidos de forma sospechosa en el proceso. Esto confirma una vez más la existencia de código malicioso en el proceso infectado. A continuación se muestra parte de la información obtenida, si se desea consultar la información completa, ésta se encuentra en el Anexo T.

[46] https://code.google.com/p/volatility/wiki/CommandReference#malfind
[47] https://msdn.microsoft.com/en-us/library/windows/desktop/ms632589(v=vs.85).aspx
[48] https://msdn.microsoft.com/en-us/library/windows/hardware/ff554836(v=vs.85).aspx

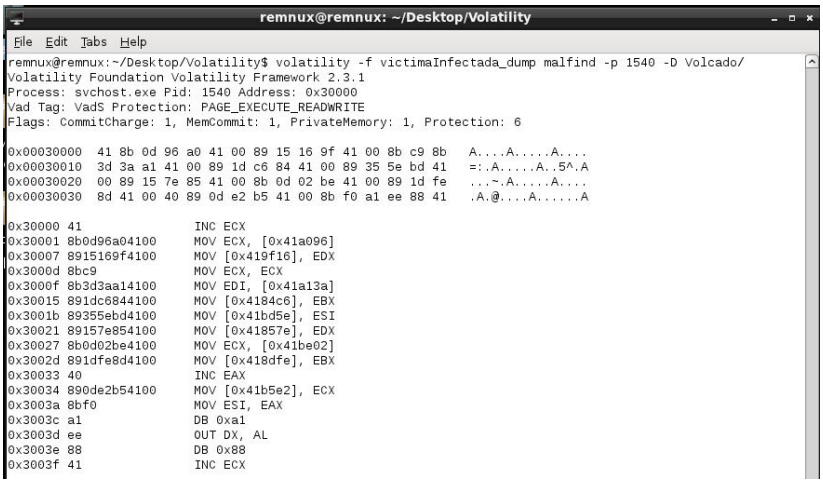


Ilustración 65 Identificación de código sospechoso en memoria, "Volatility"

Comentar que, aunque en la metodología utilizada se contempla el análisis del disco duro, llegados a este punto se ha decidido no llevarlo a cabo por considerar que este análisis no aportaría información relevante. Teniendo en cuenta que el estudio se ha centrado en la fase 1 de operación de la APT "Octubre Rojo", y que ya se dispone información similar a la que aportaría esta prueba, se ha omitido esta tarea. En el caso de analizar la segunda fase de operación de este *malware*, sería de gran interés llevarla a cabo, teniendo en cuenta la instalación de módulos complejos y su acceso a información almacenada en el disco.

Tras la realización de todos los pasos del análisis de comportamiento se puede confirmar que se trata de un análisis totalmente necesario, que ofrece información muy valiosa permitiendo acceder a procesos en ejecución, monitorizar el comportamiento del sistema y las comunicaciones, así como simular distintos escenarios de ejecución del *malware* para obtener un modelo lo más completo posible de su operación. Sin embargo, es necesario destacar que sin la información de las fases anteriores, el análisis dinámico no sería tan efectivo, ya que sería necesario establecer importantes hipótesis de partida traduciéndose en un incremento sustancial de las repeticiones necesarias para obtener resultados válidos.

8. Conclusiones

La realización del presente piloto experimental confirma la complejidad de las Amenazas Persistentes Avanzadas, validando la necesidad de disponer de mecanismos sistemáticos y metodológicos que permitan adquirir mayor conocimiento de su comportamiento con el objetivo de definir sistemas de protección efectivo.

La Amenaza Avanzada Persistente “Octubre Rojo”, tomada como muestra para el desarrollo del presente piloto experimental, es un *malware* altamente complejo. El análisis se ha centrado únicamente en el *exploit* y el *dropper*, quedando pendientes todos los módulos asociados a su arquitectura. Este hecho representa el alto grado de complejidad de esta APT y la necesidad de recursos especializados, tanto técnicos como humanos, para el desarrollo de un análisis completo

La complejidad en el desarrollo del *malware* caracterizado como APT requiere, no sólo de recursos organizativos como la metodología propuesta o técnicos como las herramientas o un buen laboratorio, si no que se hace imprescindible disponer de un conocimiento en ingeniería inversa, por parte del analista, muy elevado. El implantador de sistemas de ciberdefensa orientados al análisis de *malware*, debe valorar este hecho.

La validez de la aplicabilidad de la metodología tomada como referencia, se confirma de forma rotunda. La ejecución del proceso completo de acuerdo al orden propuesto ha garantizado un avance ordenado y coherente en el análisis del *malware* “Octubre Rojo” cuyos resultados, acordes con los obtenidos por fuentes expertas, evidencian su éxito.

La primera fase relacionada con las acciones iniciales y fundamentada en la obtención de una línea base ha sido de gran utilidad. La generación de procedimientos en su ejecución, que permitan una automatización de este proceso, sería de gran utilidad así como tener en cuenta la protección del sistema administrador del propio entorno virtual.

La fase de clasificación se ha identificado como crítica para la realización del presente piloto. La información recabada ha permitido orientar de forma efectiva las pruebas realizadas, optimizando el análisis. Este hecho pone de manifiesto que esta fase es vital en el análisis de aquellas APTs que hayan sido previamente detectadas por terceros. En el caso de ser un *malware* no identificado, la relevancia de esta fase disminuiría drásticamente haciendo el análisis más complejo.

Durante la fase de análisis de código se confirma que la innovación propuesta de intercambiar el orden, situando primero el análisis de código dinámico al estático, facilita el desarrollo del estudio del *malware* si este es de alta complejidad y utiliza avanzadas técnicas de ofuscación. Sin embargo se ha detectado, posiblemente condicionada por el nivel del analista, la necesidad de incluir un proceso cíclico e iterativo entre ambos análisis con el objetivo de ir acotando dicho proceso conforme aumenta el conocimiento sobre el *malware*. Este hecho podría suponer una innovación derivada de la ya expuesta.

La necesidad de la fase de análisis de comportamiento del *malware* queda confirmada ya que permite aumentar en gran medida el conocimiento sobre el comportamiento de la APT que se está analizando. Sin embargo, la efectividad de esta fase está condicionada por la calidad de la información obtenida en las fases previas, reduciendo de forma drástica un posible interés en ejecutar esta fase antes de las anteriores. La metodología recoge un mecanismo iterativo coherente con la posibilidad de tener que analizar nuevas muestras debido a la complejidad del *malware*, materializada durante la ejecución, situación que hubiera ocurrido si se hubiese iniciado la fase 2 de la APT “Octubre Rojo” y su descarga de módulos específicos desde el C&C.

Durante la ejecución de las pruebas, y teniendo en cuenta su alcance, se ha puesto de manifiesto que tanto las pruebas en la fase de análisis de código, estático y de comportamiento, pueden centrarse en el sistema víctima, simplificando el número de herramientas a instalar en el resto de sistemas. Esto permite centrar el objetivo del sistema monitor y del sistema servicios como piezas clave en la constitución de un escenario realista para que el *malware* muestre todo su comportamiento.

El éxito del análisis se fundamente en el uso de múltiples herramientas. Su descontinuación, como es el caso detectado de la herramienta “PEiD”, puede resultar en un problema de evolución y dificultad de homogeneización de las prácticas de uso.

9. Futuras Líneas de Desarrollo

- **Adaptar la metodología a entornos relacionados con la movilidad**

Tal y como confirma Don Javier Bermejo, la metodología utilizada está centrada en sistemas que operan bajo plataforma Windows. Teniendo en cuenta que una metodología crece en interés cuanto mayor es su aplicabilidad, el crecimiento exponencial en el uso de sistemas móviles basados en plataformas diferentes, como el caso del sistema operativo Android[49], directamente proporcional al crecimiento de las amenazas a las que se expone, genera una ventana de oportunidad clara para el desarrollo de la metodología aquí probada

- **Explotación del entorno de pruebas**

Se ha confirmado la importancia de disponer de los recursos técnicos necesarios para generar un escenario de prueba y análisis lo más parecido posible a un entorno real. Aunque en el desarrollo de la metodología se presenta como un recurso, su relevancia invita al desarrollo de una línea especialista que, desde el punto de vista del laboratorio y siendo coherente a las fases descritas en la metodología, garantice la preparación de un entorno óptimo.

- **Adaptación de la metodología a distintos perfiles**

La metodología trabajada se presenta desde un punto de vista neutral, sin embargo, debido a la importancia que está tomando la ciberdefensa como disciplina complementaria a la ciberseguridad, sería interesante tener en cuenta los distintos perfiles de analistas (administración, empresas desarrollo de aplicaciones, empresas de desarrollo de herramientas de defensa,…) que se pueden encontrar en función de sus objetivos e interesas. De este modo se podrían generar líneas de desarrollo fundamentadas en la profundización u orientación de las fases y prácticas, descritas en la metodología, en función de estos objetivos e intereses buscados.

- **Gestión y Clasificación de herramientas**

La aplicación de la metodología se fundamenta en la utilización de múltiples herramientas, muchas de ellas de libre uso y otras previa adquisición de licencia. Teniendo en cuenta que el segundo tipo de herramientas ya dispone de condicionantes legales, el primero es más difícil de controlar (soporte, continuidad,…). Una línea de desarrollo enfocada a la gestión y mantenimiento efectivo de dichas herramientas permitiría mejorar la gestión de los recursos técnicos y humanos pudiendo dar pie a posibles estandarizaciones en sectores específicos.

[49] https://www.android.com/

Referencias y Bibliografía

[1] Don Javier Bermejo, (2015). *Desarrollo de un sistema de análisis e ingeniería inversa de código malicioso*

[2] INTECO, (2012). *Desmontando Malware*

[3] Kirillov, Beck, Chase, & Martin, (2011). *Malware Attribute Enumeration and Characterization*. MITRE Corporation.

[4]Microsoft, (2012). *Microsoft Security Intelligence Report Special Edition*

[5] Panda Security, (2013). *Informe anual PandaLabs Resumen 2013*

[6] Symantec Inc., (2013) *Internet Security Threat Report 2013*

[7] Solutionary, (2012) *Defending Against Advanced Persistent Threats*

[8] Command Five Pty Ltd, (2011). *A Decade in Review Command Five Pty LtdJune 2011*

[9] Fortinet, (2013) *Threats on the Horizon: The Rise of the Advanced Persistent Threat*

[10] Kelly Jackson Higgins, Dark Reading (Aug 20, 2014). *Arachnophobia - Pakistan's cyberespionage campaign against India*
(http://www.informationweek.in/informationweek/news-analysis/297595/arachnophobia-pakistans-cyberespionage-campaign-india)

[11] Kelly Jackson Higgins, DarkReading (May 19, 2013). *Pakistan hit by targeted cyber attack out of India*
(http://www.informationweek.in/informationweek/news-analysis/176860/pakistan-hit-targeted-cyber-attack-india)

[12] Mandiant, (2013). *APT1: Exposing One of China's Cyber Espionage Units*

[13] Kaspersky Lab, (2013).Kaspersky Lab Identifies Operation “Red October,” an Advanced Cyber-Espionage Campaign Targeting Diplomatic and Government Institutions Worldwide” (http://www.kaspersky.com/about/news/virus/2013/Kaspersky_Lab_Identifies_Operation_Red_October_an_Advanced_Cyber_Espionage_Campaign_Targeting_Diplomatic_and_Government_Institutions_Worldwide)

[14] Kaspersky Lab, (2013).The "Red October" Campaign - An Advanced Cyber Espionage Network Targeting Diplomatic and Government Agencies.
(https://securelist.com/blog/incidents/57647/the-red-october-campaign/)

[15] Marius Tivadar, Bitdefender (2013). *Red October*

[16] INTECO, Seguinfo.blogspot.com (2013). *Análisis de Octubre Rojo, el Malware Diplomático*
(http://seguinfo.blogspot.com.es/2013/02/analisis-de-octubre-rojo-el-malware.html)

[17] www.Mejor-Antivirus.es, (2014). *Octubre Rojo ataca de nuevo*
(http://www.mejor-antivirus.es/noticias/octubre-rojo-ataca-de-nuevo.html)

[18] Theerthagiri, (2009). *Reversing: A detection intelligence with in-depth security analisys*

[19] Jiang, X. (2006) *Enabling Internet Worms and Malware Investigation and Defense using Virtualization*. PhD Thesis, Purdue University, Center for Education and Research in Information Assurance and Security.

[20] Ligh, M. H., Adair, S., Hartstein, B., & Richard, M. (2011). *Malware Analyst’s Cookbook and DVD. Tools and Techniques for Fighting Malicious Code*. Wiley Publishing, Inc.

[21] Hornat, C. (2007). *Malware Analysis: An Introduction*. SANS Institute InfoSec Reading Room

[22] Timeline, M. &., & Goldman, J. E. (2011). *Malware Analysis Reverse Engineering* (MARE) Methodology & Malware Defense (M.D.) Timeline.

[23] Coryn S., (2006).*The fundamenteal characteristics of research*

[24] Sanabria A., (2007). *Malware Analysis. Enviroment Designs and Architecture*. SANS Institute.

[25] Kaspersky Lab y AlienVault, (2013). Operation *"Red October": Indicators of Compromise and Mitigation Data*

[26] Kaspersky Labs' Global Research & Analysis Team, (2013). *Red October – Java Exploit Delivery Vector Analysis*

[27] Kaspersky Labs' Global Research & Analysis Team, (2013). *Red October" Diplomatic Cyber Attacks Investigation"*
(https://securelist.com/analysis/36740/red-october-diplomatic-cyber-attacks-investigation/)

[28] Kaspersky Lab, Securelist*, (2013). "Red October". Detailed Malware Description 2. Second Stage of Attack*
(https://securelist.com/analysis/36842/red-october-detailed-malware-description-2-second-stage-of-attack/)

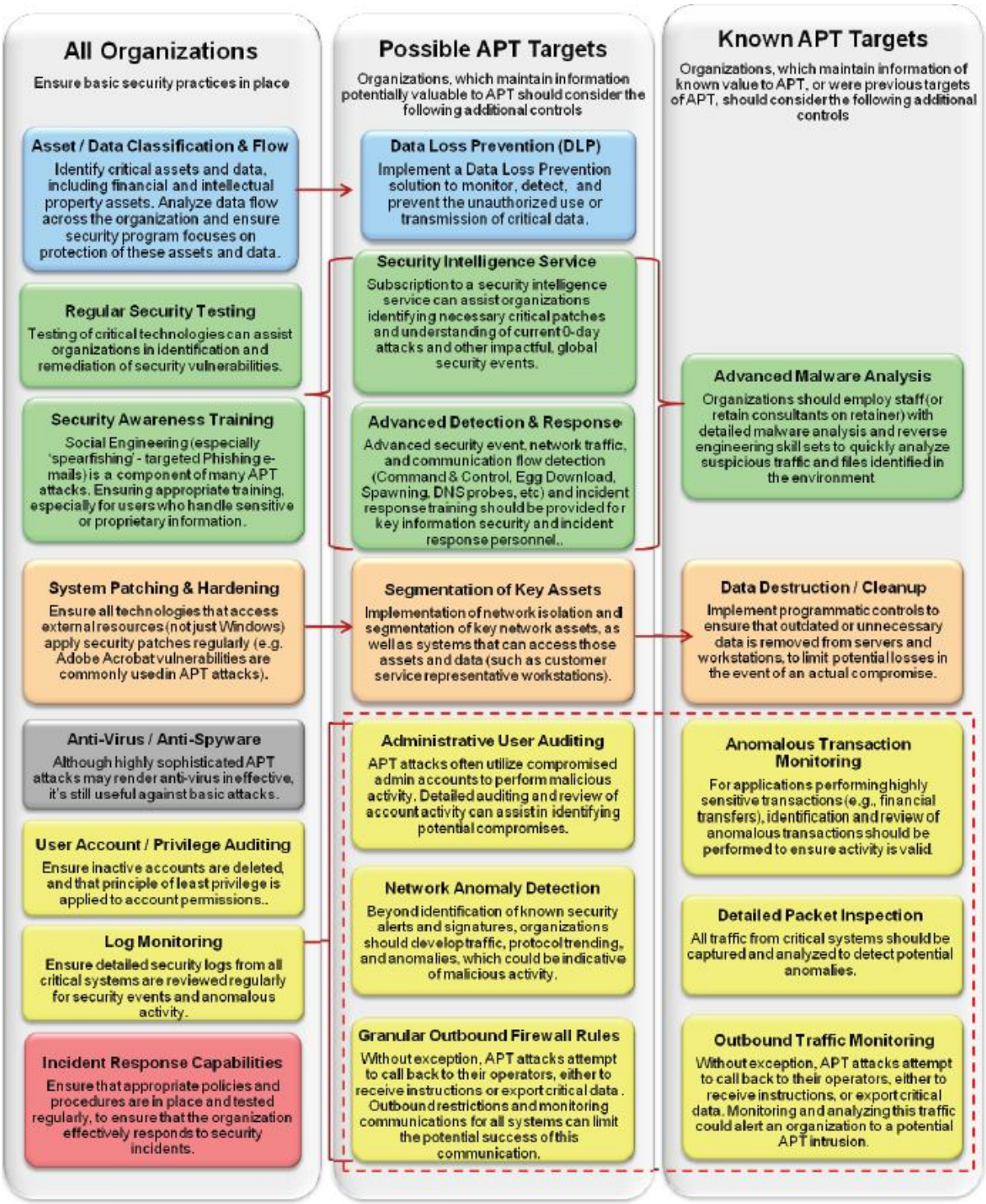
[29] Kendall Kris, (2007). *Practical Malware Analysis*
(https://www.blackhat.com/presentations/bh-dc-07/Kendall_McMillan/Paper/bh-dc-07-Kendall_McMillan-WP.pdf)

[30] CERT Malware.lu, (2013). *Analysis of the sample "Red October" - Part 1*
(http://www.malware.lu/articles/2013/01/15/analysis-of-the-sample-red-october-part-1.html)

[31] CERT Malware.lu, (2013). *Analysis of the sample "Red October" - Part 2*
(http://www.malware.lu/articles/2013/01/15/analysis-of-the-sample-red-october-part-2.html)

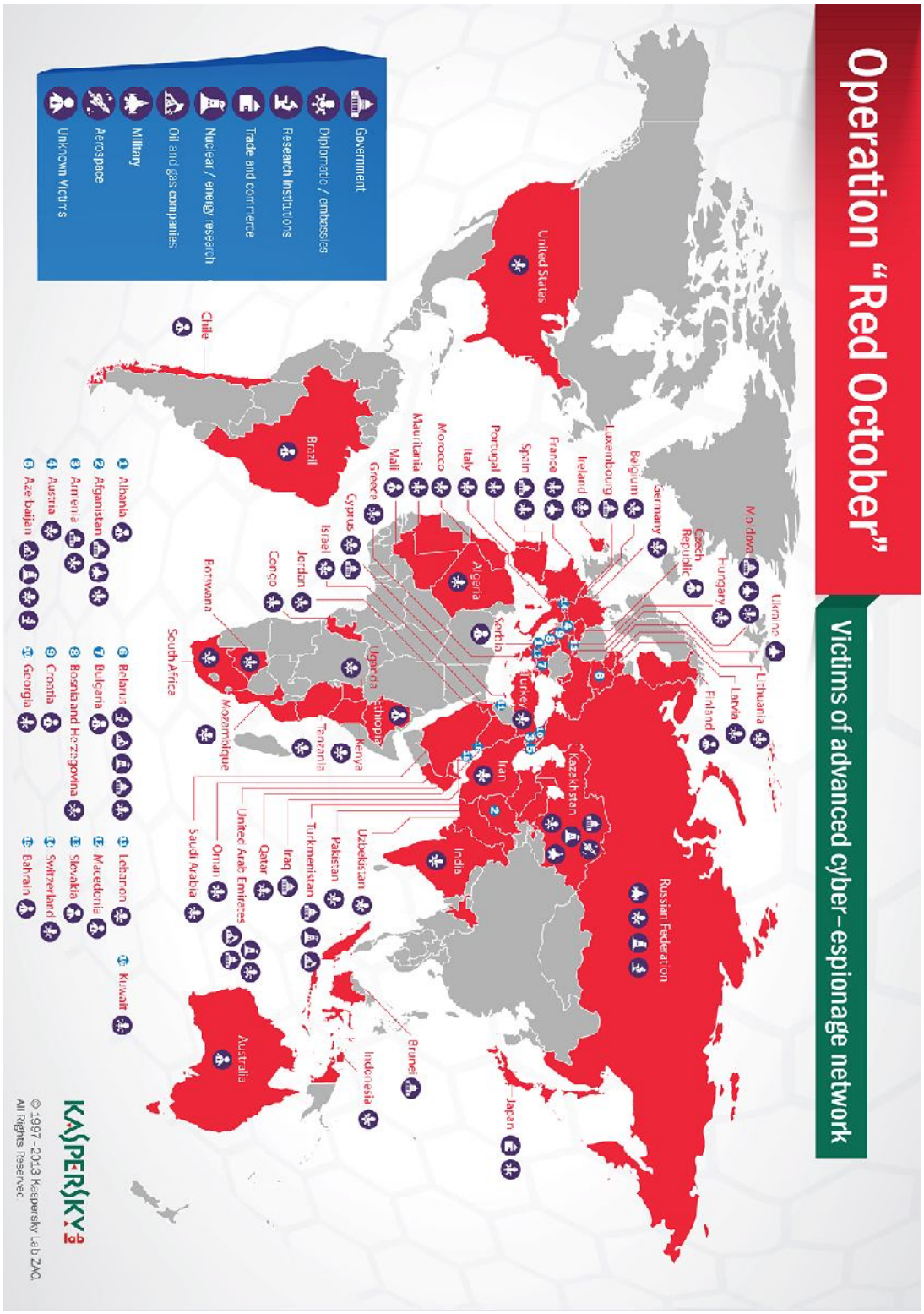
Anexo A Estrategias de defensa contra APTs

- **Diagrama relación estrategias en defensa contra APTs [7]**



Anexo B Mapa Alcance Campaña "Octubre Rojo"

- **Mapa global de la campaña "Octubre Rojo" [13]**



Anexo C Estructura de control de la red "Octubre Rojo"

- **Principales dominios utilizados para controlar los equipos infectados [24]**

Domain	Registered by	Registrant e-mail	Created
nt-windows-online.com	Ustuygov Denis Egorovich	ustuygov_d@mail.ru	1-Apr-11
genuine-check.com	Privacy Protect	?	18-Jun-10
genuineupdate.com	Igor Shaven / Sellsgroup LLC	shaven@mail.ru	21-Jun-10
nt-windows-update.com	Privacy Protect	?	1-Apr-11
nt-windows-check.com	Privacy Protect	?	4-Apr-11
genuineservicecheck.com	Igor S Zorin	zorin_24@mail.ru	24-Jun-10
svchost-check.com	Denis Kajan	dkajan@list.ru	7-Jun-11
svchost-online.com	Privacy Protect	?	7-Jun-11
microsoftosupdate.com	Simmy Tujk	simmutijjk@rambler.ru	5-Dec-08
microsoft-msdn.com	SINKHOLED Simmu Ivanovich / Suur-Karja	stijk@yandex.ru	6-Dec-08
microsoftcheck.com	Privacy Protect	?	5-Dec-08
msinfoonline.org	Yuriy Poletaev	kleyton107@rambler.ru	12-Nov-07
win-check-update.com	Privacy Protect	?	22-Dec-09
mobile-update.com	Privacy Protect	?	14-Jan-11
ms-software-check.com	Denis gartovanov	gartovanov@bk.ru	26-Sep-11
ms-software-update.com	Valdis Nevelskij	nevelskij@bk.ru	26-Sep-11
ms-software-genuine.com	Sergej Kalinin	kaliniserg@rambler.ru	26-Sep-11
windowscheckupdate.com	Privacy Protect	?	27-Oct-09
windows-genuine.com	SINKHOLED Pushkareva Sofya Sergeevna	pyshkareva_76@mail.ru	27-Oct-09
windowsonlineupdate.com	SINKHOLED Jan kilkys	kilkys@yandex.ru	27-Oct-09
csrss-check-new.com	Privacy Protect	?	27-Apr-12
csrss-update-new.com	Leonid Kluev	kluev.leonid@rambler.ru	27-Apr-12
csrss-upgrade-new.com	Aleksandr Lavrov	aleksandrlavrov@lenta.ru	3-May-12
dll-host-update.com	SINKHOLED NEVER REGISTERED	SINKHOLED	2-Nov-12
dll-host-check.com	Volin Sergej	volinsergej@yandex.ru	4-Oct-10
dll-host.com	Sergey I Orlov	orlov.orloffsergej@yandex.ru	1-Oct-10
win-driver-upgrade.com	Lykash V.D.	lykashvadim@rambler.ru	11-Apr-12
update-genuine.com	Valdas Palajtis / thinks Sells	valdas-palajtis@yandex.ru	22-Apr-09
svchost-update.com	Sergej Dumkovskij	dumkovskij@rambler.ru	7-Jun-11
os-microsoft-check.com	Contact Privacy Inc. Customer 0123124787	?	23-Feb-10
xponlineupdate.com	Eherik Kristi	eherik-kirsti@rambler.ru	5-Nov-08
dll-host-udate.com	Privacy Protect	?	4-Oct-10
new-driver-upgrade.com	Dima Grivnev	dgriven@mail.ru	21-Mar-12
dllupdate.info	Privacy Protect	?	1-Oct-08
os-microsoft-update.com	Syhar Denis Ivavovich	den-syhar@rambler.ru	23-Feb-10
wingenuine.com	Privacy Protect	?	31-Mar-09
drivers-update-online.com	Ivan Iystenko	Iystenko@inbox.ru	1-Feb-10
wins-update.com	Igor Proskyren	praskyren@mail.ru	11-Feb-08
wins-driver-update.com	Privacy Protect	?	11-04-2012
msonlineupdate.com	Denis Dumkov	denis-dumkov@rambler.ru	14-04-2010
wins-driver-check.com	anton Zinin	zinin-ant@bk.ru	11-04-2012
drivers-check.com	Mihail Stupin	stypin_86@mail.ru	11-Oct-12
drivers-get.com	Igor Sidorenko	sidorenko_81@list.ru	5-Feb-12
osgenuine.com	Vidmans Semenov	vidmans-semenov@yandex.ru	22-Apr-09
msgenuine.net	SINKHOLED Vacheslav Bulanov	BULANOV24@YAHOO.COM	22-May-07
msonlinecheck.com	Denis Butov	botov_denis@mail.ru	14-Apr-10
msonlineget.com	Demidkov Denis	denis_demidkov@mail.ru	18-May-10

- **Identificación de los servidores principales de la red de control [24]**

IP	Active	Confirmed Malicious	Location	Hosting
141.101.239.225	Oct-12	Yes	Russia	Leadertelecom Ltd.
178.63.208.49	Oct-12	Yes	Germany	Nuremberg Hetzner Online Ag
188.40.19.247	Oct-12	Yes	Germany	Nuremberg Hetzner Online Ag
37.235.54.48	Oct-12	Yes	-unclear- ? Austria / UK /Spain	Edis Gmbh
78.46.173.15	Oct-12	Yes	Germany	Nuremberg Hetzner Online Ag
88.198.30.44	Oct-12	Yes	Germany	Nuremberg Hetzner Online Ag
88.198.85.161	Oct-12	Yes	Germany	Nuremberg Hetzner Online Ag
92.53.105.40	Oct-12	Yes	Russia	Ooo Lira-s
31.41.45.119	Nov-12	Yes	Russia	Relink Ltd
176.9.241.254	Nov-12	Yes	Germany	Nuremberg Hetzner Online Ag

IP	Date	Confirmed malicious	Country	ISP
31.41.45.139	Oct-12	Yes, mini-mothership	Russia	Relink Ltd.
91.226.31.40	Oct-12	Yes, mini-mothership	Russia	i7 Ltd
178.63.208.63	Oct-12	Yes, mini-mothership	Germany	Nuremberg Hetzner Online Ag

Anexo D Clasificación de los módulos de "Octubre Rojo"

- **Criterios de agrupación de módulos [28]**

Group name	Descrition
Recon	Modules of this group designed to be used during first stage of cyberattack right after initial infiltration. Their main purpose is to collect general information about target system which helps locate and identify the infected machine, estimate potential value of current computer data and define which other modules should be pushed next. Also, these modules collect initial easy-to-get type of information such as browser history, browser cached credentials and FTP client settings.
Password	This group of modules is designed to steal credentials from various applications and resources, from Mail.ru Agent (popupal free app from mail.ru) to MS Outlook credentials and Windows account hashes (including cached Windows Domain account hashes). Capable of using low-level and direct disk access to copy protected files.
Email	This group serves stealing emails from local MS Outlook storage or remote POP3/IMAP mail server. It's capable of dumping full email bodies with headers, saving attachments with predefined file extensions.
USB drive	This group is used to steal files from attached USB devices. It monitors USB device events and starts every time new device is attached. It can copy files from predefined extension list, size and age. This group capable of recognition, restoration and copying already deleted files of MS Office document formats by using own FAT-based filesystem parser.
Keyboard	This group is dedicated to recording keystrokes, grabbing text from password input fields and making screencaptures.
Persistence	Current group contains installer and payload code to plant a plugin in popular applications such as MS Office or Adobe Reader. The backdoor code is activated when specially crafted document is opened on target machine. This is used to regain lost access on a machine in case of unexpected loss of control (C&C server takedown or local malware cleaning).
Spreading	Modules of this group are used to scan for other hosts on the network, fingerprint them and then infect via MS08-067 or a list of stolen admin credentials. A module from this group is capable of dumping Cisco network router configuration via SNMP commands and embedded TFTP server.
Mobile	Mobile group is used to dump all valuable information about locally attached mobile device. It is capable of copying contact information, calendars, SMS and Emails databases and many other private data. These modules are capable of checking if a device was jailbroken.
Exfiltration	While some of other modules work in "offline" mode, collect and store data locally, this group of modules transfers all collected data to the C&C server. Modules of this group are capable of reaching FTP servers, remote network shares as well as local disk drives and copy files from these resources. Unlike Recon data collection modules these modules are designed to run repeatedly and bring only new valuable data.

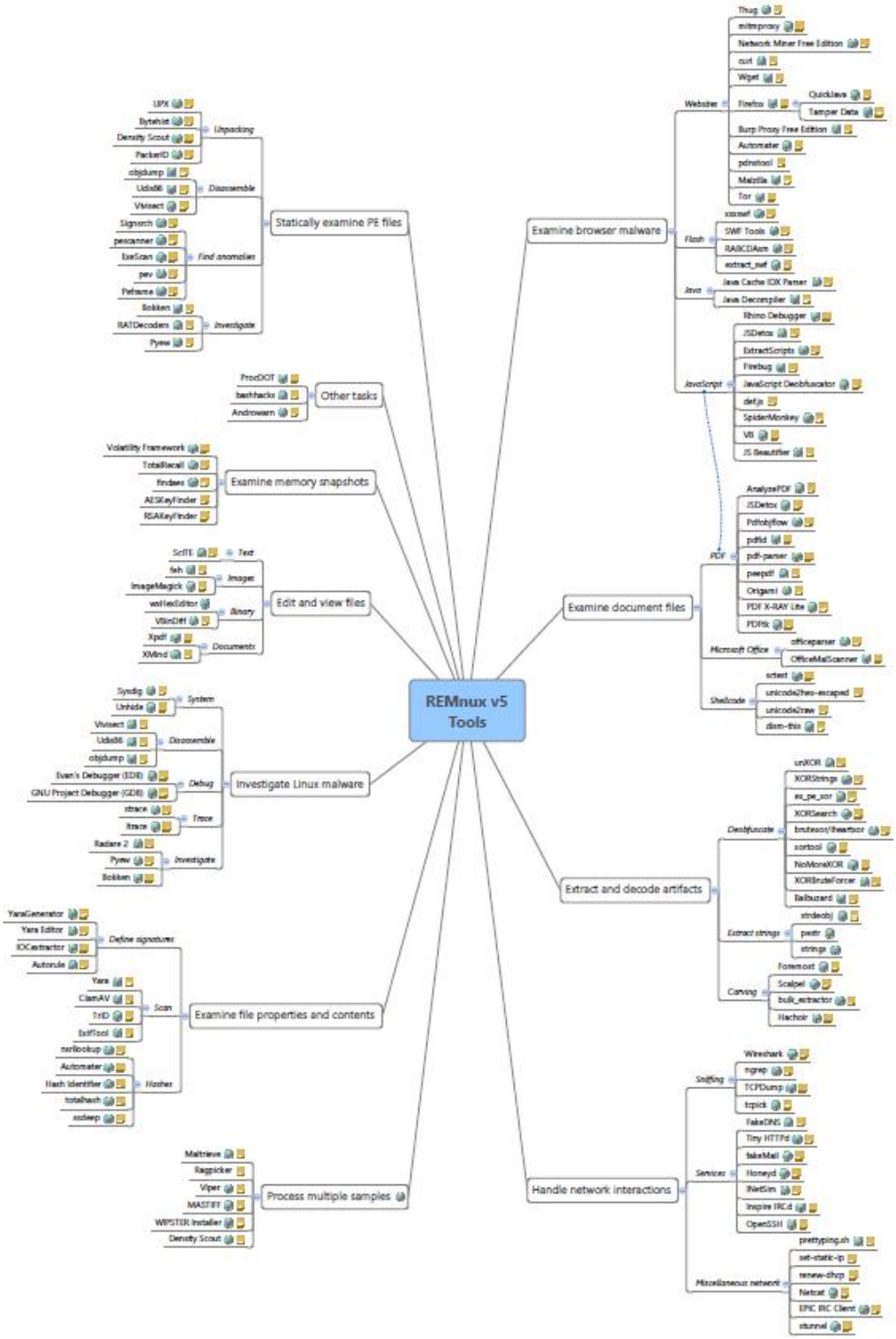
- **Clasificación de los principales módulos de infección [14]**

№	Name	Group			Size (Kb)	Summary
1	RegConn	Recon			~160	Query system software environment
2	WnHttp	Recon			~142	Get external IP and send to the C&C
3	SysInfo	Recon			~503	Get browser history,usb drives,processes,disks,...
4	GetWebFtp	Recon			~157	Get browser history,http/ftp credentials
5	AuthInfo	Recon			~660	Get file manager,browser,ftp,mail client credentials
6	Logic	Recon			~160	Get general information about current Windows machine and available remote network shares
7	ILogic	Recon			~150	Grab Internet Explorer URL history from the local system
8	Repeat2	Recon			~150	Get listing from remote shares available in Windows network neighborhood
9	Reference	Recon			~150	Grab directory/file listings of all drives attached to the local system
10	PswSuperMailru	Password			230-260	Steal Mail.ru account info and Outlook attachments
11	PswOutlook	Password			~31	Steal Outlook account info
12	MSHash	Password			400-550	Steal Windows account hashes
13	MAPIClient	Email			418-440	Steal e-mail data using local MAPI
14	POP3Client	Email			1100-1200	Steal e-mail data from POP3 server
15	USBContainer	USB drive			649-690	Loads and runs embedded USBStealer
16	USBRestore	USB drive			372-376	Recover and steal deleted files on USB drives
17	USBStealer	USB drive			448-504	Steal interesting files from USB drives
18	Keylogger	Keyboard			300-312	Makes screenshots, records keystrokes
19	Scheduler	Persistence			~620	Run various tasks from spec folders
20	DocBackdoor	Persistence			75-88	Runs an embedded module from MSOffice/PDF doc
21	OfficeBDInstaller	Persistence			~286	Installs DocBackdoor plugin in MS Office
22	AdobeBDInstaller	Persistence			~218	Installs DocBackdoor plugin in Adobe Reader
23	FilePutExec	Spreading			~305	Extract and run an embedded file locally or remotely
24	Netscan	Spreading			~315	Port scanner, vuln. scanner, Cisco cfg dumper
25	MSExploit	Spreading			~1200	Infect target host using MS08-067 exploit
26	DASvcInstall	Spreading			~276	Infect target host using admin credentials
27	Frog	Spreading			~102	Initial backdoor, used in MSExploit/DASvcInstall
28	iPhone	Mobile			329-331	Steals data from locally attached iPhone
29	Nokia	Mobile			~337	Steals data from locally attached Nokia phone
30	Winmobile	Mobile			~400-700	Infect locally attached Windows Mobile phones with a native backdoor/updater modules
31	Winmobile	Mobile			~7-100	Native mobile backdoor/utilites
32	WnFtpScan	Exfiltration			~209	Steals files from local FTP server
33	GetFileReg	Exfiltration			~340	Steals files from local/network disks
34	FileInfo	Exfiltration			339-340	Uploads various collected files to the C&C

- "online" module: all data is sent to the C&C; no local files created;
- "offline" module; no network communication; all data is stored locally;
- module with embedded script/config in resource named "AAA";
- module with all values hardcoded.

Anexo E Mapa Servicios ofrecidos por REMnux V5

- **Mapa de herramientas disponible en REMnux V5 (remnux.org)**



Anexo F Herramientas utilizadas en el análisis de “Octubre Rojo”

Avira: herramienta antivirus gratuita.

https://www.avira.com/es/index

Bintext: herramienta para la identificación y extraer strings ASCII y Unicode.

http://www.mcafee.com/es/downloads/free-tools/bintext.aspx

Dependency Walker: herramienta para escanear archivos ejecutables y generar árboles de dependencia entre sus módulos.

http://www.dependencywalker.com/

Diskpulse: herramienta para monitorizar los cambios del disco en tiempo real.

http://www.diskpulse.com/downloads.html

GMR: herramienta para la detección de rootkits.

http://www.gmer.net/

IDA Pro: herramienta de desensamblado de código.

https://www.hex-rays.com/products/ida/support/download_freeware.shtml

MD5Sum: herramienta para crear hash MD5.

http://www.etree.org/md5com.html

Ollydbg: herramienta de depuración utilizada durante la fase de análisis dinámico de código

http://www.ollydbg.de/

PEid: herramienta para detectar archivos empaquetados y cifrados.

http://www.softpedia.com/get/Programming/Packers-Crypters-Protectors/PEiD-updated.shtml

PE Explorer: herramienta que permite analizar, a nivel de posiciones en memoria, el contenido de un proceso.

http://www.heaventools.com/download-pe-explorer.htm

PEBrowse: herramienta que permite desensamblar archivos PE.

http://www.smidgeonsoft.prohosting.com/pebrowse-pro-interactive-debugger.html

PE Studio: herramienta para el análisis estático de archivos ejecutables.

http://www.winitor.com/

PEViewer: herramienta para el análisis de archivos PE.

http://download.cnet.com/PE-Viewer/3000-2352_4-10966763.html

ProcessExplorer (SysInternals): herramienta para comprobar los accesos al sistema del un proceso

https://technet.microsoft.com/en-us/sysinternals/bb842062.aspx

ProcessMonitor (SysInternals): herramienta para la supervisión de procesos en tiempo real.

https://technet.microsoft.com/en-us/sysinternals/bb842062.aspx

Regshot: herramienta para comparar el registro.

http://sourceforge.net/projects/regshot/

REMnux: distribución Linux desarrollada para el análisis inverso de código. Cuenta con múltiples herramientas, entre ellas Volatility.

https://remnux.org/

Strings (SysInternals): herramienta para la identificación y extraer strings ASCII y Unicode.

https://technet.microsoft.com/en-us/sysinternals/bb842062.aspx

Systracer: herramienta para la captura de estados del sistema y análisis de cambios

http://www.blueproject.ro/systracer

VMMAP: herramienta para analizar la memoria del sistema en tiempo real.

https://technet.microsoft.com/en-us/library/dd535533.aspx

WinMD5: herramienta para comprobar la integridad de ficheros mediante hash MD5.

http://www.winmd5.com/

Winpmem: aplicación que permite realizar volcados de memoria en sistemas Windows para poder ser analizados porsteriormente por la aplicación Volatility.

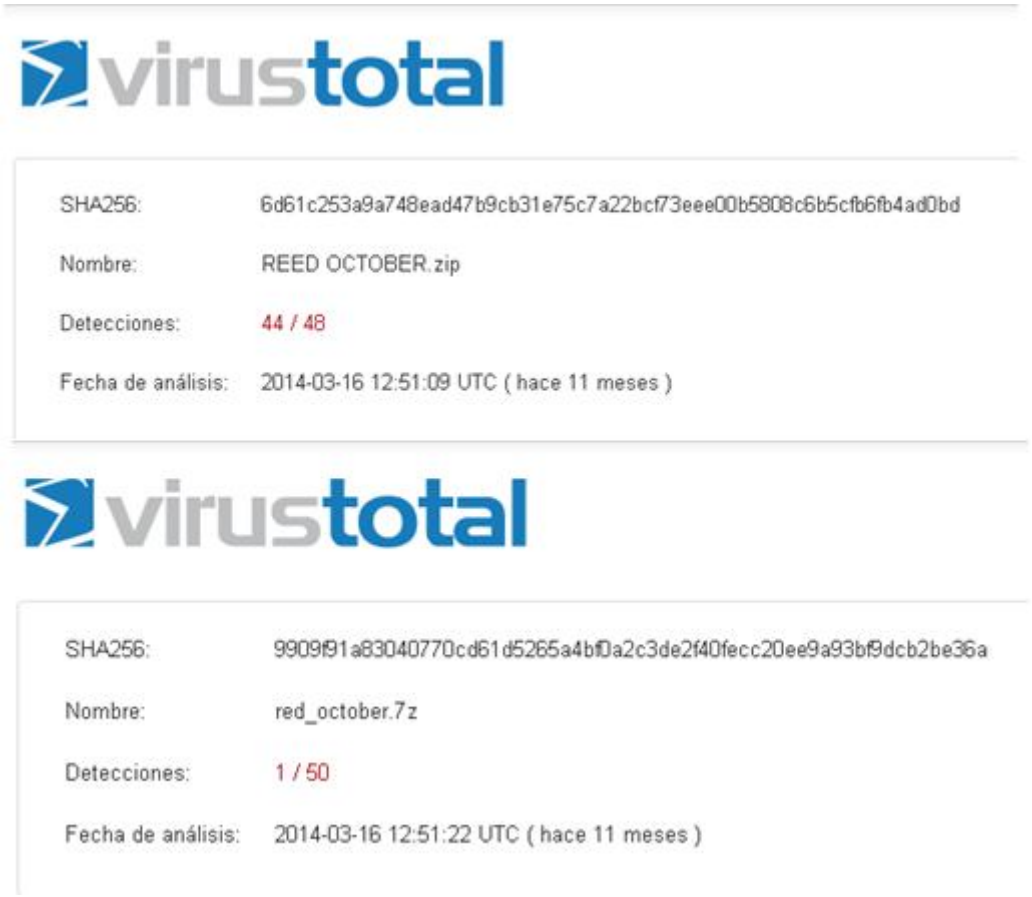
https://github.com/google/rekall/releases

Wireshark: herramienta para el análisis de paquetes de red.

https://www.wireshark.org/download.html

Anexo G Clasificación – Identificación "Virustotal"

- **Detección de relación con otros archivos sospechosos**



SHA256: 6d61c253a9a748ead47b9cb31e75c7a22bcf73eee00b5808c6b5cfb6fb4ad0bd

Nombre: REED OCTOBER.zip

Detecciones: 44 / 48

Fecha de análisis: 2014-03-16 12:51:09 UTC (hace 11 meses)

SHA256: 9909f91a83040770cd61d5265a4bf0a2c3de2f40fecc20ee9a93bf9dcb2be36a

Nombre: red_october.7z

Detecciones: 1 / 50

Fecha de análisis: 2014-03-16 12:51:22 UTC (hace 11 meses)

- **Informe herramientas antivirus que detectan el archivo como amenaza**

Antivirus	Resultado	Actualización
Avast	XLS:CVE-2009-3129 [Expl]	20140624
AhnLab-V3	XLS/Cve-2009-3129	20140624
ViRobot	XLS.S.CVE-2009-3129.559616	20140624
Tencent	Word.Trojan-dropper.Agent.Pfj	20140624
ESET-NOD32	Win32/Exploit.CVE-2009-3129.N	20140624
Qihoo-360	virus.exp.ole.17	20140624
Comodo	UnclassifiedMalware	20140624
Kaspersky	Trojan-Dropper.MSWord.Agent.ga	20140624
Emsisoft	Trojan-Dropper.MSWord.Agent (A)	20140624

Ikarus	Trojan-Dropper.MSWord.Agent	20140624
K7AntiVirus	Trojan (0040f0511)	20140623
K7GW	Trojan (0040f0511)	20140623
TrendMicro	TROJ_OLEXP.B	20140624
TrendMicro-HouseCall	TROJ_OLEXP.B	20140624
Sophos	Troj/DocDrop-S	20140624
Norman	Shellcode.B	20140624
CAT-QuickHeal	Shell.Gen.AI	20140624
Rising	NORMAL:Hack.Exploit.Macro.CVE-2009-3129.a!1611213	20140623
Fortinet	MSExcel/CVE_2009_3129.A!exploit	20140624
McAfee-GW-Edition	Heuristic.BehavesLike.Exploit.X97.CodeExec.O	20140623
Microsoft	Exploit:Win32/CVE-2009-3129	20140624
Commtouch	Exploit/XLS.gen	20140624
F-Prot	Exploit/XLS.gen	20140624
VBA32	Exploit.Win32.OLE.77	20140624
NANO-Antivirus	Exploit.MSExcel.CVE-2009-3129.ccxskf	20140624
VIPRE	Exploit.Excel.CVE-2009-3129 (v)	20140624
DrWeb	Exploit.Excel.18	20140624
Bkav	Exploit.CVE-2009-3129.Heur	20140623
Ad-Aware	Exploit.CVE-2009-3129.Gen	20140624
BitDefender	Exploit.CVE-2009-3129.Gen	20140624
F-Secure	Exploit.CVE-2009-3129.Gen	20140624
GData	Exploit.CVE-2009-3129.Gen	20140624
MicroWorld-eScan	Exploit.CVE-2009-3129.Gen	20140624
nProtect	Exploit.CVE-2009-3129.Gen	20140624
McAfee	Exploit-MSExcel.ac	20140624
AntiVir	EXP/Excel.CVE-2009-3129	20140624
AVG	Dropper.Generic_c.NZR	20140624
Symantec	Backdoor.Rocra	20140624

- **Informe herramientas antivirus que no detectan el archivo como amenaza**

AegisLab	✔	20140624
Agnitum	✔	20140623
Antiy-AVL	✔	20140624
Baidu-International	✔	20140624
ByteHero	✔	20140624
CMC	✔	20140624
ClamAV	✔	20140624
Jiangmin	✔	20140624
Kingsoft	✔	20140624
Malwarebytes	✔	20140624
Panda	✔	20140624
SUPERAntiSpyware	✔	20140624
TheHacker	✔	20140622
TotalDefense	✔	20140624
Zillya	✔	20140624
Zoner	✔	20140616

Anexo H Clasificación - Hash MD5 Documentos empleados en ataques

- **Relación de archivo potenciales de infección [14]**

114ed0e5298149fc69f6e41566e3717a	93d0222c8c7b57d38931cfd712523c67
1f86299628bed519718478739b0e4b0c	9950a027191c4930909ca23608d464cc
2672fbba23bf4f5e139b10cacc837e9f	9b55887b3e0c7f1e41d1abdc32667a93
350c170870e42dce1715a188ca20d73b	9f470a4b0f9827d0d3ae463f44b227db
396d9e339c1fd2e787d885a688d5c646	a7330ce1b0f89ac157e335da825b22c7
3ded9a0dd566215f04e05340ccf20e0c	b9238737d22a059ff8da903fbc69c352
44e70bce66cdac5dc06d5c0d6780ba45	c78253aefcb35f94acc63585d7bfb176
4bfa449f1a351210d3c5b03ac2bd18b1	fc3c874bdaedf731439bbe28fc2e6bbe
4ce5fd18b1d3f551a098bb26d8347ffb	bb2f6240402f765a9d0d650b79cd2560
4daa2e7d3ac1a5c6b81a92f4a9ac21f1	bd05475a538c996cd6cafe72f3a98fae
50bd553568422cf547539dd1f49dd80d	c42627a677e0a6244b84aa977fbea15d
51edea56c1e83bcbc9f873168e2370af	cb51ef3e541e060f0c56ac10adef37c3
5d1121eac9021b5b01570fb58e7d4622	ceac9d75b8920323477e8a4acdae2803
5ecec03853616e13475ac20a0ef987b6	cee7bd726bc57e601c85203c5767293c
5f9b7a70ca665a54f8879a6a16f6adde	d71a9d26d4bb3b0ed189c79cd24d179a
639760784b3e26c1fe619e5df7d0f674	d98378db4016404ac558f9733e906b2b
65d277af039004146061ff01bb757a8f	dc4a977eaa2b62ad7785b46b40c61281
6b23732895daaad4bd6eae1d0b0fef08	dc8f0d4ecda437c3f870cd17d010a3f6
731c68d2335e60107df2f5af18b9f4c9	de56229f497bf51274280ef84277ea54
7e5d9b496306b558ba04e5a4c5638f9f	ec98640c401e296a76ab7f213164ef8c
82e518fb3a6749903c8dc17287cebbf8	f0357f969fbaf798095b43c9e7a0cfa7
85baebed3d22fa63ce91ffafcd7cc991	f16785fc3650490604ab635303e61de2
91ebc2b587a14ec914dd74f4cfb8dd0f	

Anexo I Clasificación - Dominios C&C y Direcciones IP de Ataque

- **Principales Dominios relacionados con APT "Octubre Rojo" [25]**

bb-apps-world.com	ms-software-genuine.com
blackberry-apps-world.com	ms-software-update.com
blackberry-update.com	new-driver-upgrade.com
csrss-check-new.com	nt-windows-check.com
csrss-update-new.com	nt-windows-online.com
csrss-upgrade-new.com	nt-windows-update.com
dailyinfonews.net	osgenuine.com
dll-host.com	os-microsoft-check.com
dll-host-check.com	os-microsoft-update.com
dll-host-udate.com	security-mobile.com
dll-host-update.com	shellupdate.com
dllupdate.info	svchost-check.com
drivers-check.com	svchost-online.com
drivers-get.com	svchost-update.com
drivers-update-online.com	update-genuine.com
genuine-check.com	win-check-update.com
genuineservicecheck.com	windowscheckupdate.com
genuineupdate.com	windows-genuine.com
hotinfonews.com	windowsonlineupdate.com
microsoftcheck.com	win-driver-upgrade.com
microsoft-msdn.com	wingenuine.com
microsoftosupdate.com	wins-driver-check.com
mobile-update.com	wins-driver-update.com
msgenuine.net	wins-update.com
msinfoonline.org	winupdateonline.com
msonlinecheck.com	winupdateos.com
msonlineget.com	world-mobile-congress.com

msonlineupdate.com	xponlineupdate.com

- **Direcciones IP identificadas con APT "Octubre Rojo" [25]**

141.101.239.225	88.198.85.162
178.162.129.237	92.53.105.40
178.162.182.42	95.168.172.69
178.63.208.49	31.41.45.139
188.40.19.247	91.226.31.40
31.184.234.18	178.63.208.63
31.41.45.9	31.41.45.119
37.235.54.48	176.9.241.254
46.4.202.86	31.41.45.179
77.72.133.161	176.9.189.36
78.46.173.15	92.53.105.214
88.198.30.44	188.40.19.244
88.198.85.161	85.25.104.57

Anexo J Clasificación - Cadenas De Texto “Bintext”

- **Fragmento del registro de *strings* del archivo “red_oct.document.exploit”**

```
File pos        Mem pos         ID   Text
========        =======         ==   ====

00000000022B    00000000022B     0   qq1te
000000000235    000000000235     0
B
0000000002B5    0000000002B5     0   ThisWorkbook
0000000004F5    0000000004F5     0   _ * #,##0_ ;_ * \-#,##0_ ;_ * "-"_ ;_ @_
0000000005A4    0000000005A4     0   _ * #,##0.00_ ;_ * \-#,##0.00_ ;_ * "-"??_ ;_ @_
0000000005DE    0000000005DE     0   \$#,##0_);\(\$#,##0\)
0000000005FC    0000000005FC     0   \$#,##0_);[Red]\(\$#,##0\)
00000000061F    00000000061F     0   \$#,##0.00_);\(\$#,##0.00\)
000000000643    000000000643     0   \$#,##0.00_);[Red]\(\$#,##0.00\)
00000000089D    00000000089D     0   Sheet1
0000000009C4    0000000009C4     0   DINU"
000000000C50    000000000C50     0   InputBin
0000000013A2    0000000013A2     0   jiiii
0000000013B4    0000000013B4     0   ibii8
00000000140C    00000000140C     0   yoii:
000000001419    000000001419     0   icii:
0000000014BB    0000000014BB     0   nKiii
000000002952    000000002952     0   MbP?_
000000002E5A    000000002E5A     0   EXCEL.EXE
00000000332A    00000000332A     0   IPACDRWDSGATCAPI
00000000336A    00000000336A     0   PPBHAERP
0000000041D0    0000000041D0     0   Sheet1g
0000000042AC    0000000042AC     0   Microsoft Excel
0000000052C8    0000000052C8     0   Sheet1
000000006681    000000006681     0   Ua"E
0000000066D0    0000000066D0     0   0/.-,+*)<v&%7c
000000006782    000000006782     0   ~}|{zyxwvutsrqponmlkjihgfedcba
0000000067A3    0000000067A3     0   ]\[ZYXWVUTSRQPKNMI
0000000067B6    0000000067B6     0   JIHGFEDCBA@?>=<;:987654321
0000000067D1    0000000067D1     0   2w31+*)
0000000067D9    0000000067D9     0   {f%$'"! ?
000000006882    000000006882     0   ~}|{zyxwvutsrqponmlkjihgfedcba
0000000068A3    0000000068A3     0
]\[ZYXWVUTSRQPONMLKJIHGFEDCBA@?>=<;:9876543210/.-,+*)('&%$#"!
000000006982    000000006982     0   ~}|{zyxwvutsrqponmlkjihgfedcba
0000000069A3    0000000069A3     0
]\[ZYXWVUTSRQPONMLKJIHGFEDCBA@?>=<;:9876543210/.-,+*)('&%$#"!
000000006A82    000000006A82     0   ~}|{zyxwvutsrqponmlkjihgfedcba
000000006AA3    000000006AA3     0
]\[ZYXWVUTSRQPONMLKJIHGFEDCBA@?>=<;:9876543210/.-,+*)('&%$#"!
000000006B82    000000006B82     0   ~}|{zyxwvutsrqponmlkjihgfedcba
```

- **Fragmento cadenas de texto archivo "red_oct.bin.drop"**

```
File pos      Mem pos        ID  Text
========      =======        ==  ====

000000017DFC  000000417DFC    0  SetPriorityClass
000000017E10  000000417E10    0  lstrcpyW
000000017E1C  000000417E1C    0  lstrlenW
000000017E26  000000417E26    0  KERNEL32.dll
000000017E36  000000417E36    0  GetCommandLineA
000000017E48  000000417E48    0  GetVersionExA
000000017E58  000000417E58    0  GetProcessHeap
000000017E6A  000000417E6A    0  GetStartupInfoA
000000017E7C  000000417E7C    0  GetProcAddress
000000017E8E  000000417E8E    0  GetModuleHandleA
000000017EA2  000000417EA2    0  WriteFile
000000017EAE  000000417EAE    0  GetStdHandle
000000017EBE  000000417EBE    0  GetModuleFileNameA
000000017F3C  000000417F3C    0  WideCharToMultiByte
000000017F52  000000417F52    0  GetLastError
000000017F62  000000417F62    0  GetEnvironmentStringsW
000000017F7C  000000417F7C    0  SetHandleCount
000000017F8E  000000417F8E    0  GetFileType
000000017F9C  000000417F9C    0  DeleteCriticalSection
000000017FB4  000000417FB4    0  TlsGetValue
000000017FC2  000000417FC2    0  TlsAlloc
000000017FCE  000000417FCE    0  TlsSetValue
000000017FDC  000000417FDC    0  TlsFree
000000017FE6  000000417FE6    0  InterlockedIncrement
000000018026  000000418026    0  HeapDestroy
000000018034  000000418034    0  VirtualFree
000000018042  000000418042    0  QueryPerformanceCounter
00000001805C  00000041805C    0  GetCurrentProcessId
000000018072  000000418072    0  GetSystemTimeAsFileTime
00000001808C  00000041808C    0  LeaveCriticalSection
0000000180A4  0000004180A4    0  EnterCriticalSection
0000000180BC  0000004180BC    0  TerminateProcess
0000000180D0  0000004180D0    0  SetUnhandledExceptionFilter
0000000180EE  0000004180EE    0  IsDebuggerPresent
000000018102  000000418102    0  LoadLibraryA
000000018112  000000418112    0  InitializeCriticalSection
00000001812E  00000041812E    0  GetCPInfo
00000001813A  00000041813A    0  GetACP
000000018144  000000418144    0  GetOEMCP
000000018150  000000418150    0  Sleep
000000018158  000000418158    0  VirtualAlloc
000000018168  000000418168    0  HeapReAlloc
000000018182  000000418182    0  HeapSize
0000000181A4  0000004181A4    0  GetLocaleInfoA
0000000181B6  0000004181B6    0  LCMapStringA
0000000181C6  0000004181C6    0  LCMapStringW
0000000181D6  0000004181D6    0  GetStringTypeA
0000000181E8  0000004181E8    0  GetStringTypeW
```

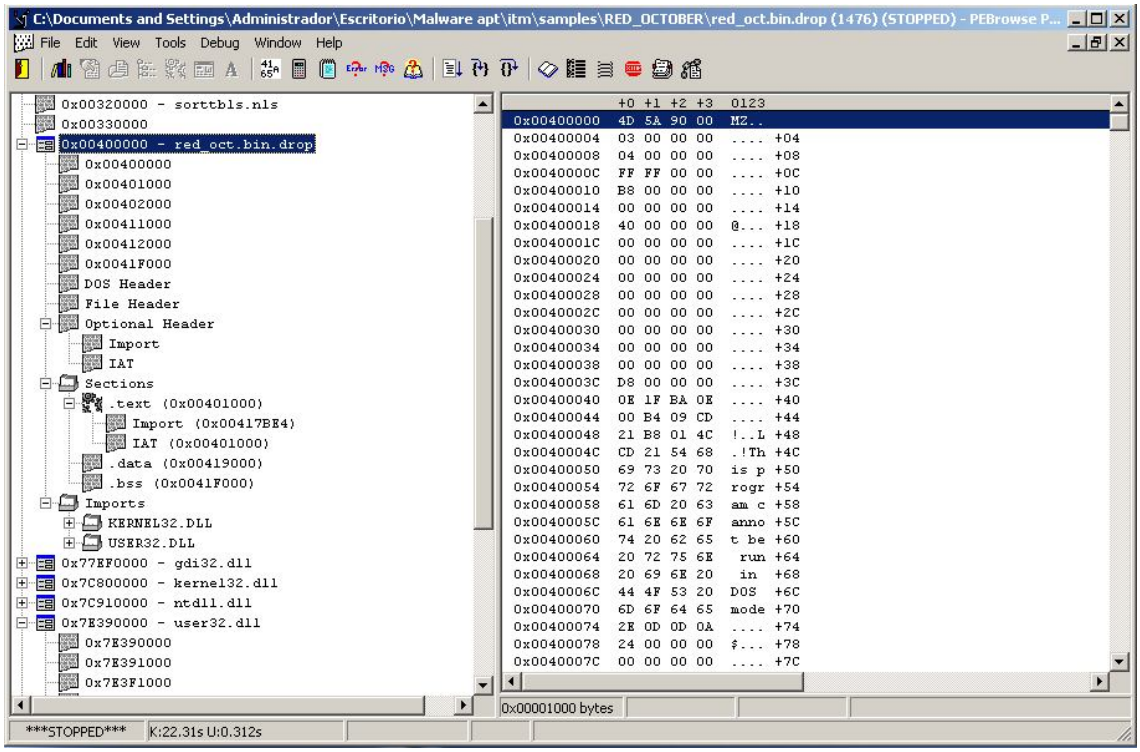
Anexo K Clasificación – Dependencias "DependencyWalker"

- **Fragmento informe de dependencias en el archivo "red_oct.bin.drop"**

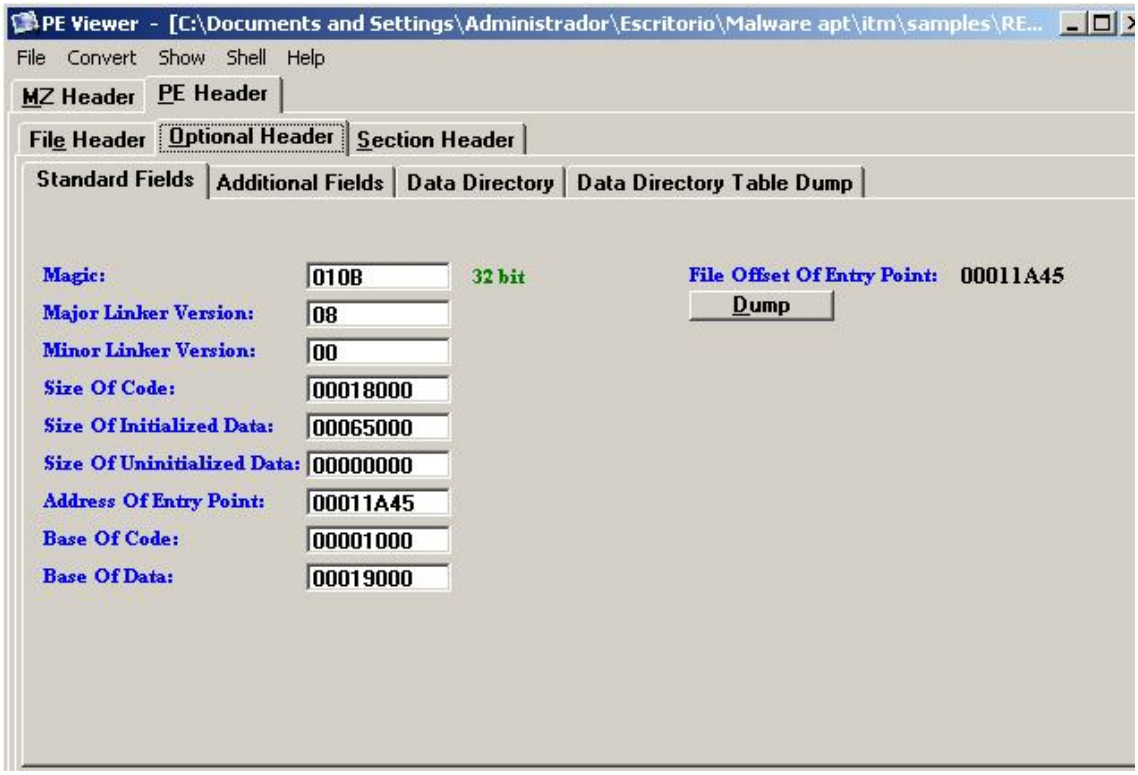
```
**************************| Module Dependency Tree |**************************
*                                                                            *
* Legend: F  Forwarded Module   ?  Missing Module       6  64-bit Module     *
*         D  Delay Load Module  !  Invalid Module                             *
*         *  Dynamic Module     E  Import/Export Mismatch or Load Failure     *
*                               ^  Duplicate Module                          *
*                                                                            *
******************************************************************************
[   ] RED_OCT.BIN.DROP
    [   ] USER32.DLL
        [   ] GDI32.DLL
            [ ^ ] KERNEL32.DLL
                [F^ ] NTDLL.DLL
            [ ^ ] NTDLL.DLL
            [ ^ ] USER32.DLL
        [ ^ ] KERNEL32.DLL
            [F^ ] NTDLL.DLL
        [   ] NTDLL.DLL
        [D  ] ADVAPI32.DLL
            [ ^ ] KERNEL32.DLL
                [F^ ] NTDLL.DLL
            [ ^ ] NTDLL.DLL
            [   ] RPCRT4.DLL
                [ ^ ] ADVAPI32.DLL
                [ ^ ] KERNEL32.DLL
                    [F^ ] NTDLL.DLL
                [ ^ ] NTDLL.DLL
                [ ^ ] SECUR32.DLL
            [D  ] WINTRUST.DLL
                [ ^ ] ADVAPI32.DLL
                [ ^ ] CRYPT32.DLL
                [ ^ ] IMAGEHLP.DLL
                [ ^ ] KERNEL32.DLL
                    [F^ ] NTDLL.DLL
                [ ^ ] MSASN1.DLL
                [ ^ ] MSVCRT.DLL
                [ ^ ] RPCRT4.DLL
                [ ^ ] USER32.DLL
            [D  ] SECUR32.DLL
                [ ^ ] ADVAPI32.DLL
                [ ^ ] KERNEL32.DLL
                    [F^ ] NTDLL.DLL
                [ ^ ] NTDLL.DLL
                [D  ] NETAPI32.DLL
                    [ ^ ] ADVAPI32.DLL
                    [ ^ ] KERNEL32.DLL
                        [F^ ] NTDLL.DLL
                    [   ] MSVCRT.DLL
                        [ ^ ] KERNEL32.DLL
                            [F^ ] NTDLL.DLL
                        [ ^ ] NTDLL.DLL
                    [ ^ ] NTDLL.DLL
                    [ ^ ] RPCRT4.DLL
                    [D^ ] SECUR32.DLL
```

Anexo L Clasificación - Información complementaria "PEBrowse" y "PEViewer"

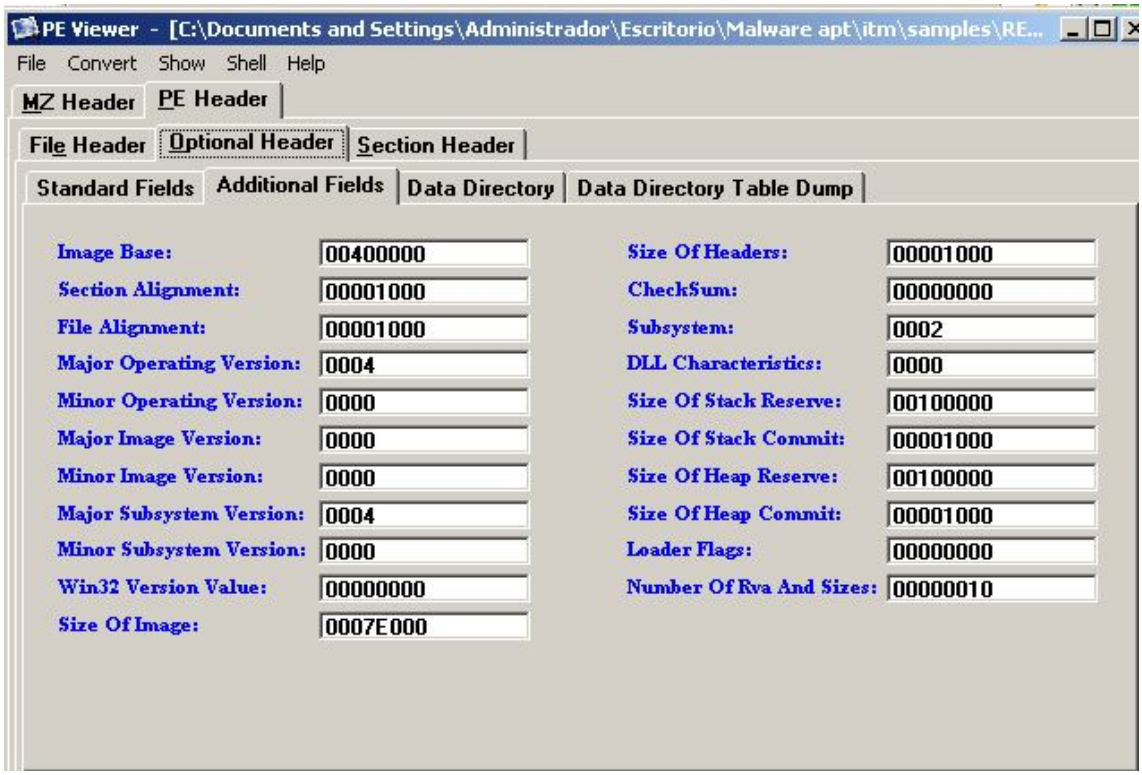
- **Estructura del archivo "red_oct.bin.drop"**



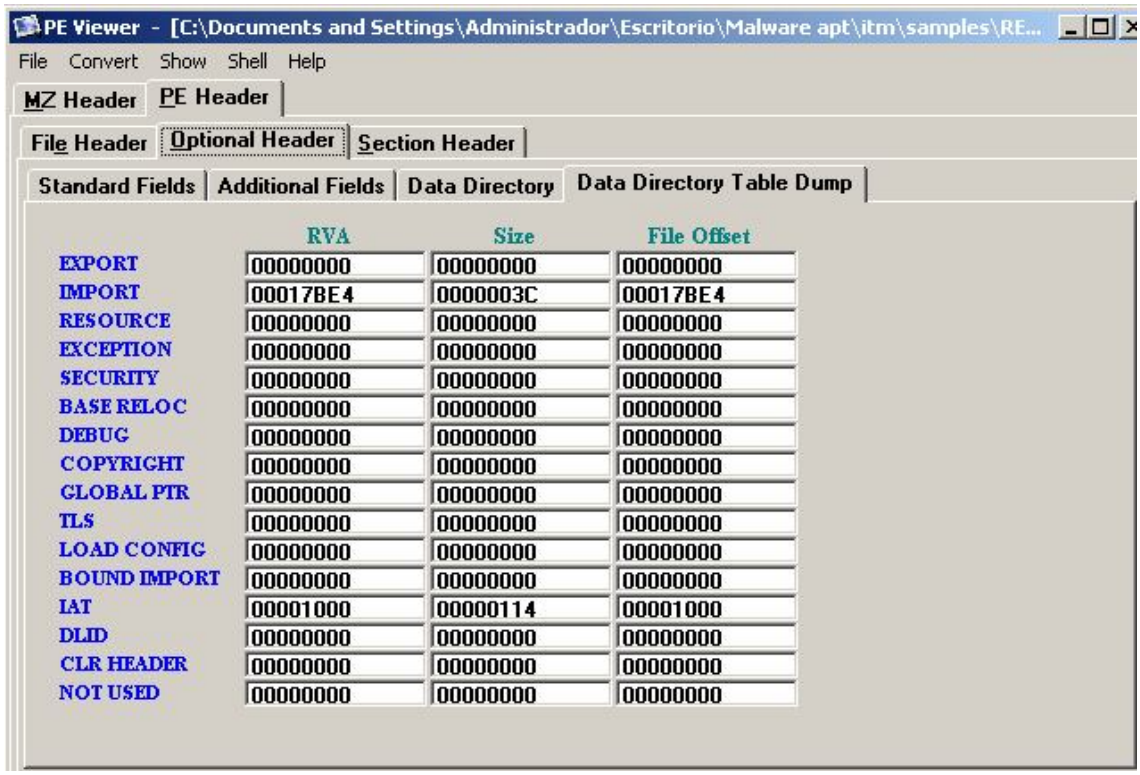
- **Información opcional de cabecera "red_oct.bin.drop" (*standard fields*)**



- **Información opcional de cabecera "red_oct.bin.drop" (*additional fields*)**

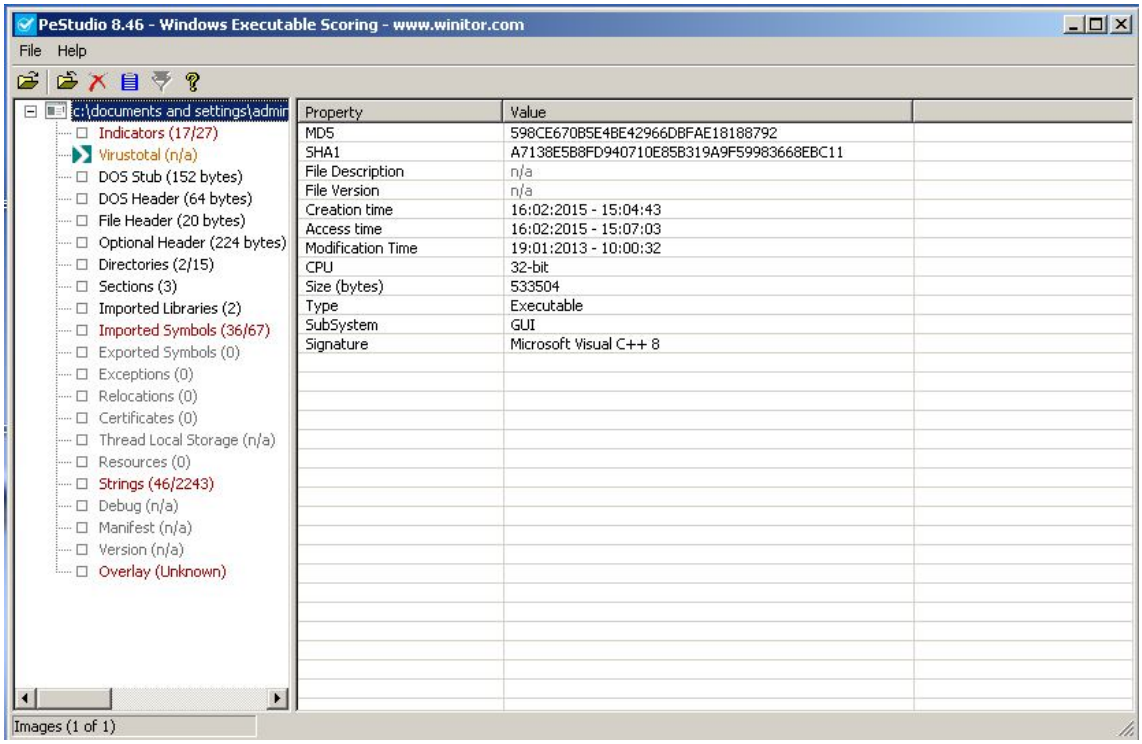


- **Información *Data Directory Table Dump* "red_oct.bin.drop"**

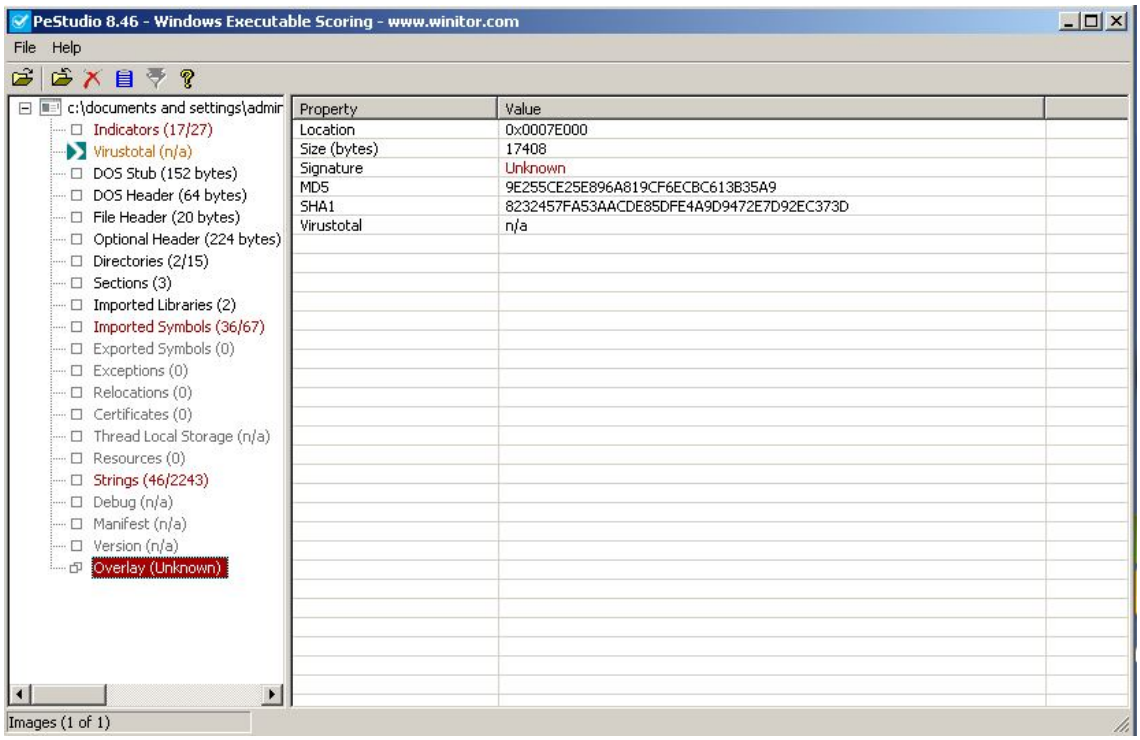


Anexo M Clasificación - Información complementaria "PeStudio"

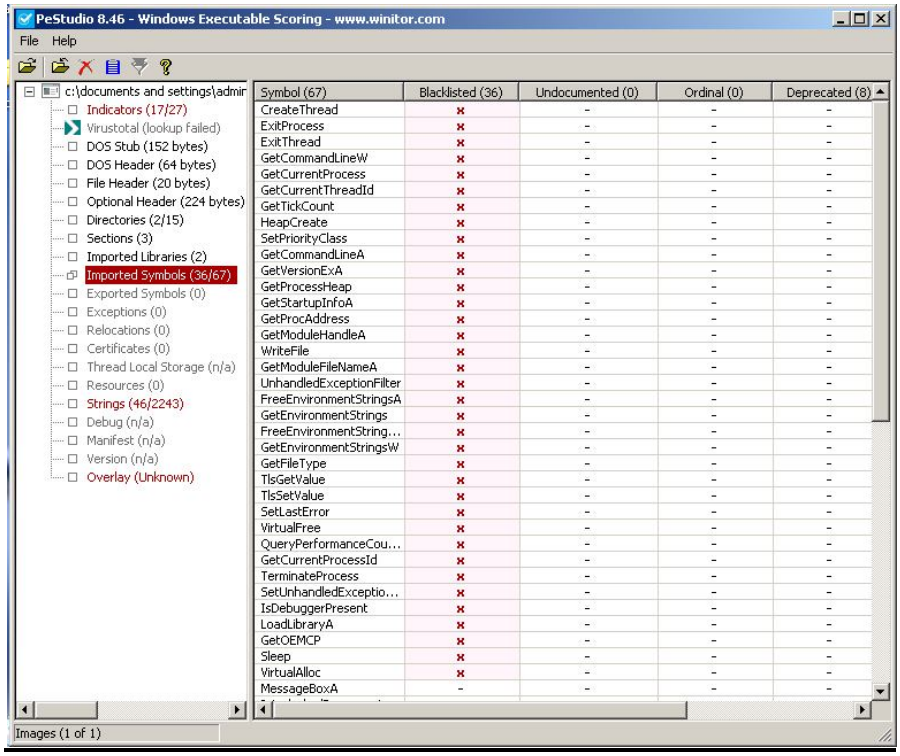
- **Información general archivo "red_oct.bin.drop"**



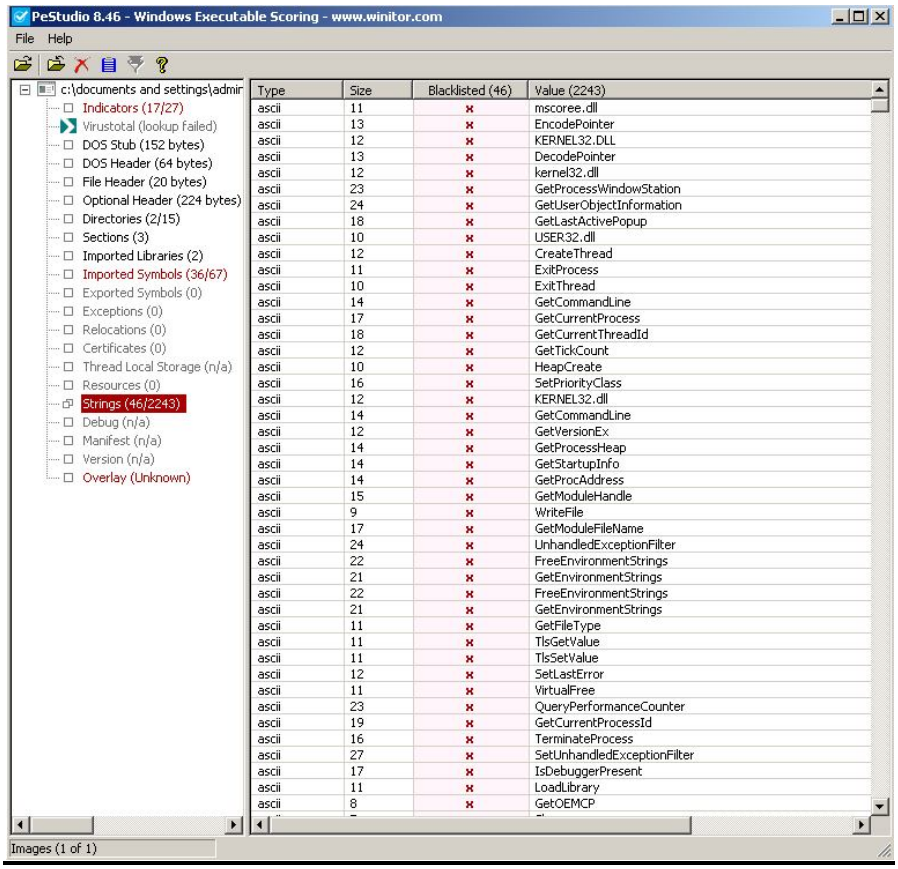
- **Información de *Overlay* en el archivo "red_oct.bin.drop"**



- **Símbolos importados sospechosos en el archivo "red_oct.bin.drop"**

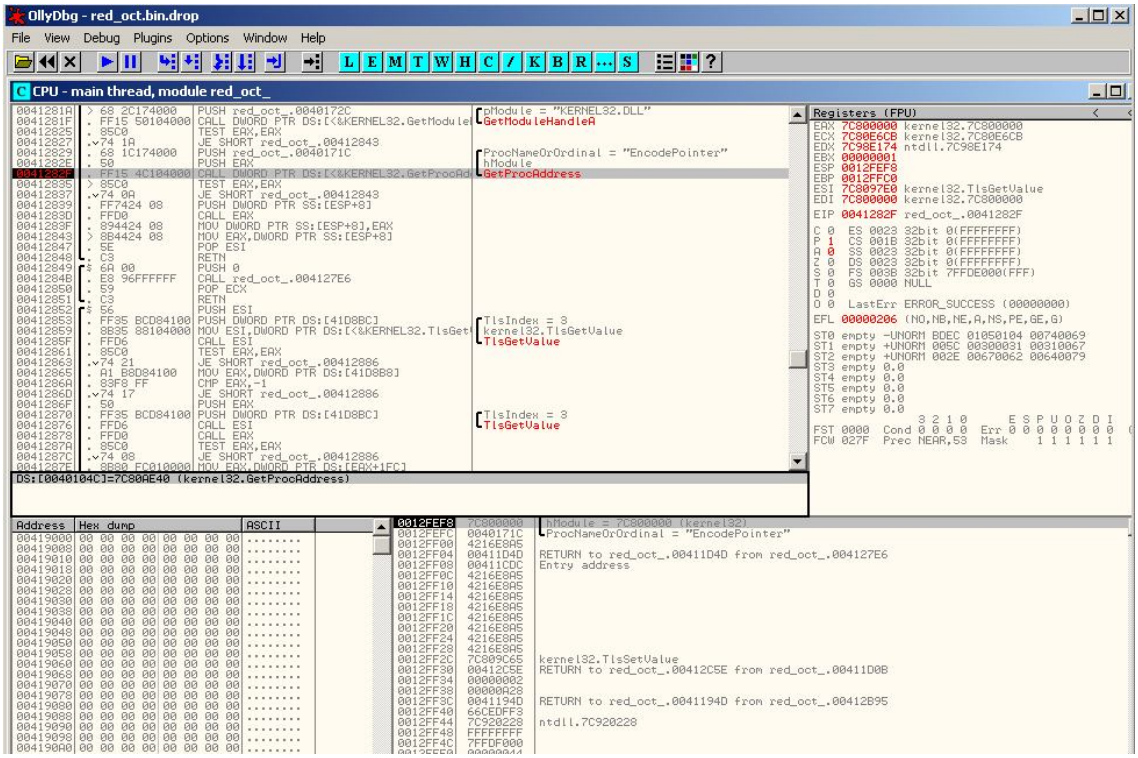


- **Cadenas de texto sospechosas en el archivo "red_oct.bin.drop"**

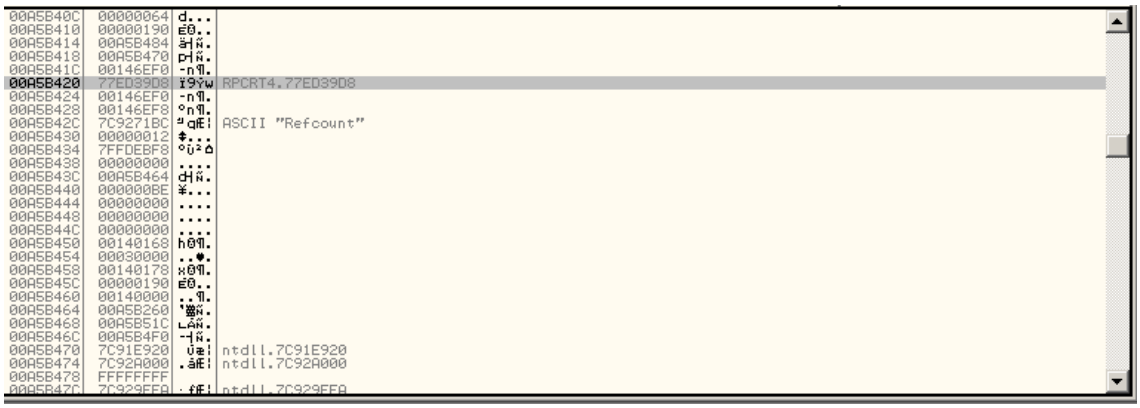


Anexo N Análisis Dinámico - Información complementaria "Ollydbg"

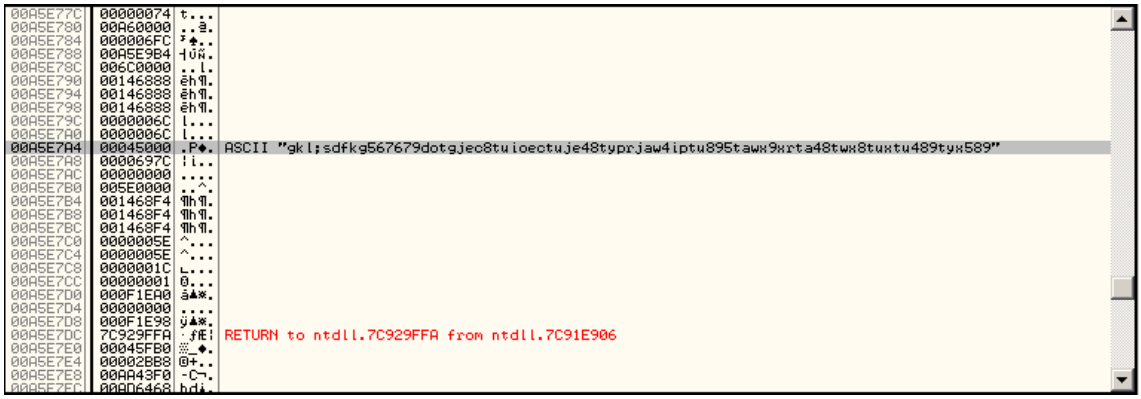
- **Detección de punto de entrada a la función "encoder"**

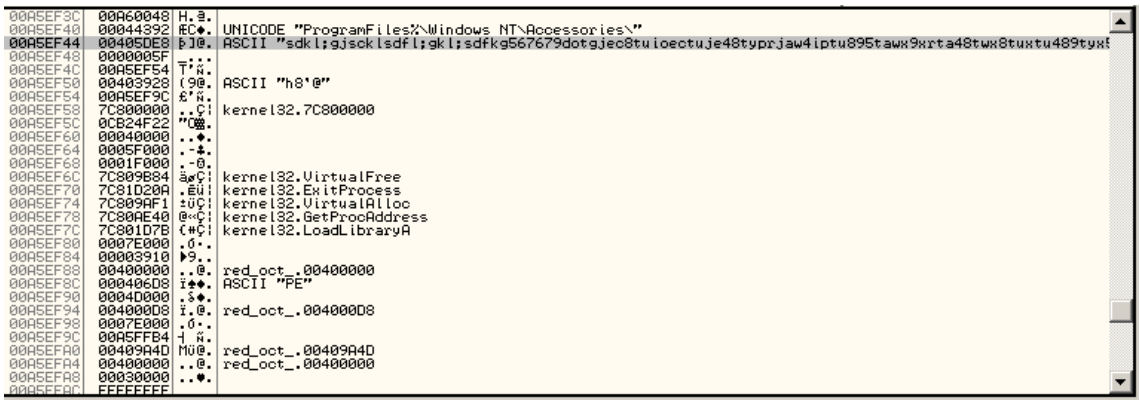


- **Posible infección de DLL para implementar el *Backdoor***

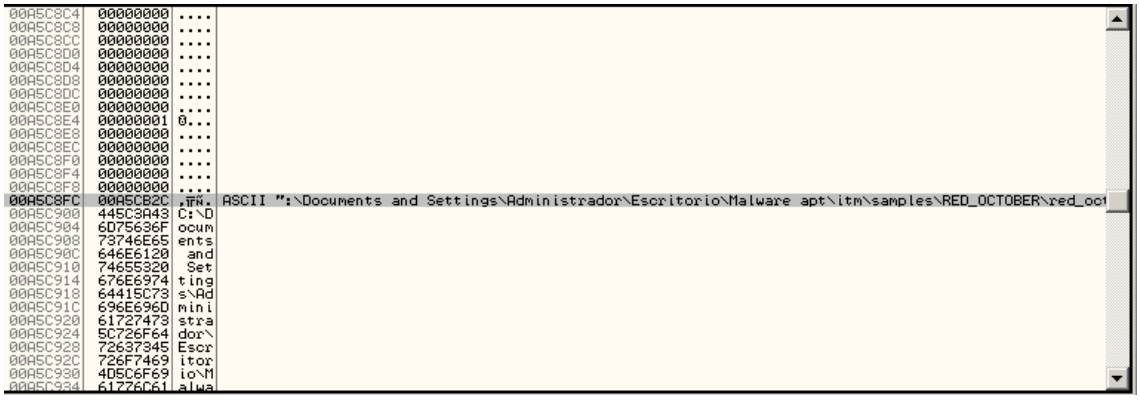


- **Cadena sospechosa**





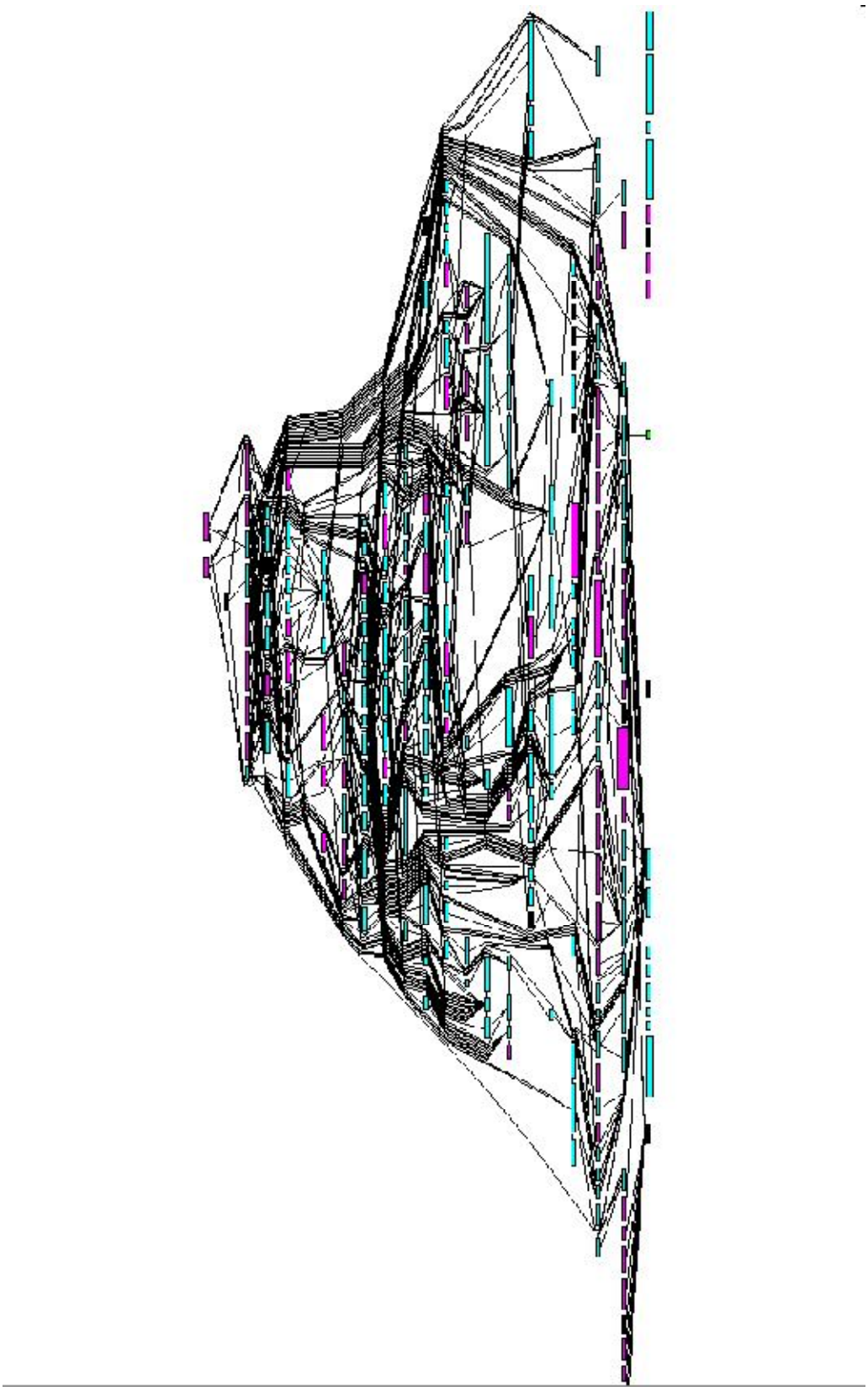
- **Cadena sospechosa de eliminación de archivo**



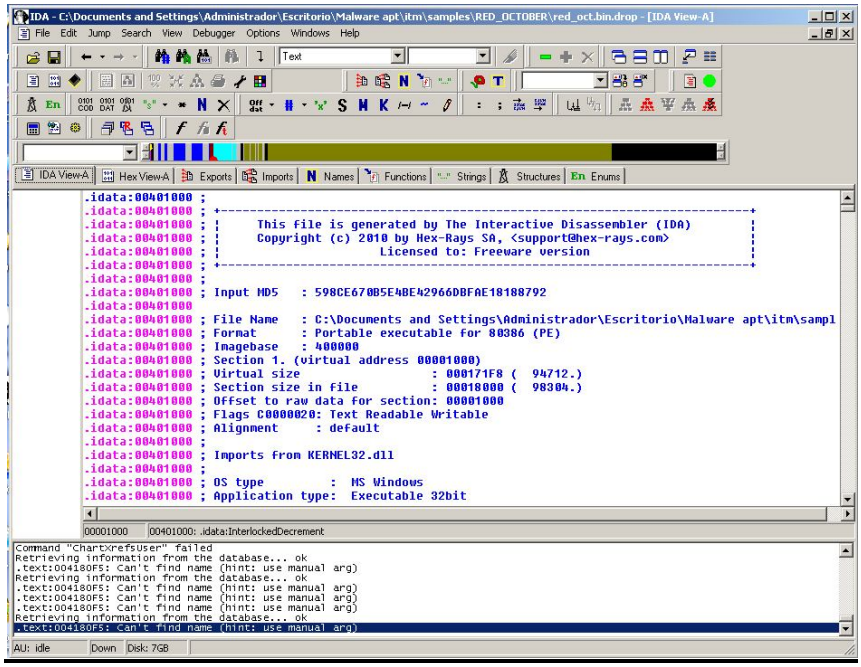
ASCII ":\Documents and Settings \ Administrador \ Escritorio \ Malware apt \ itm \ samples \ RED_OCTOBER \ red_oct.bin.drop" goto Repeat del "C:\ DOCUME~1 \ ADMINI~1 \ CONFIG~1 \ Temp \ msc.bat"

Anexo O Análisis Estático Información complementaria "IDA Pro"

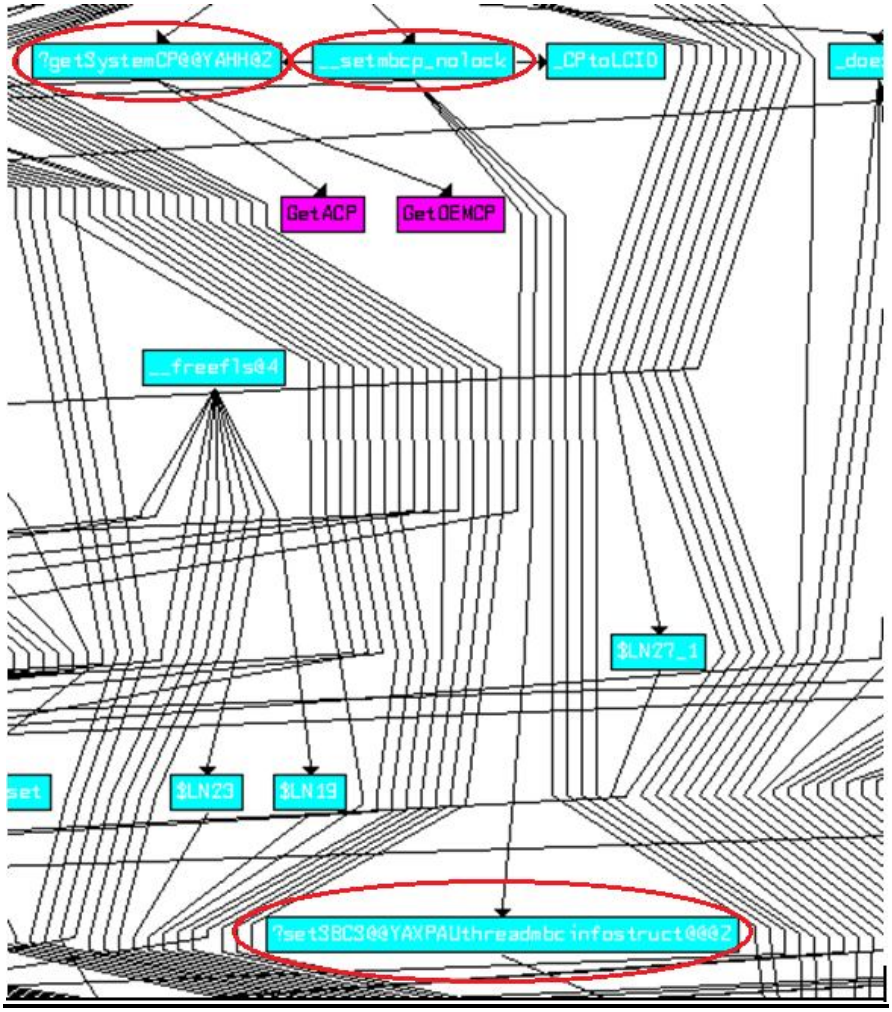
- **Flujo de llamadas entre funciones del malware**



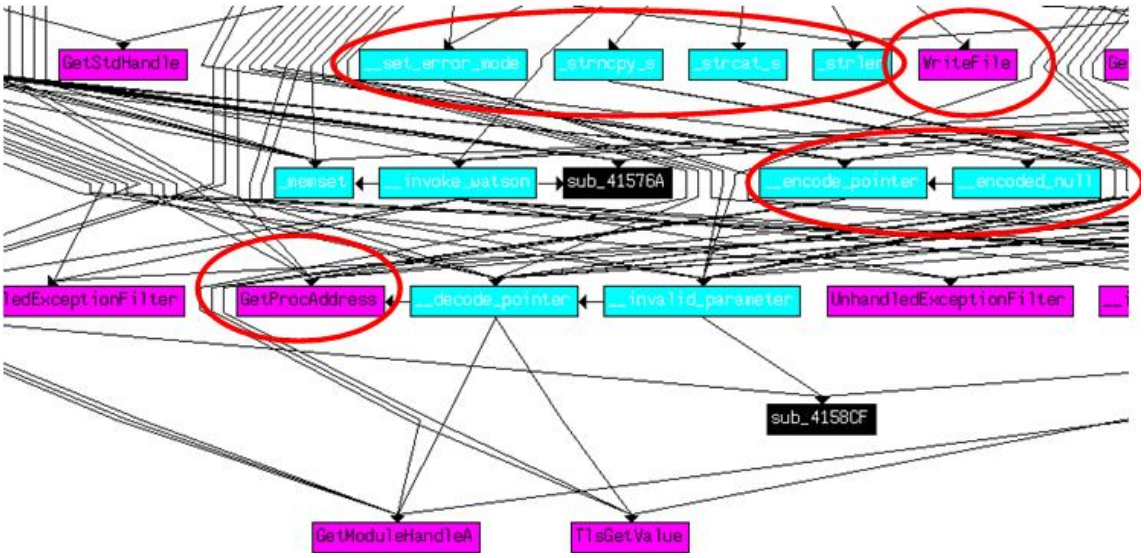
- **Proceso de desensamblado mediante IDA Pro**



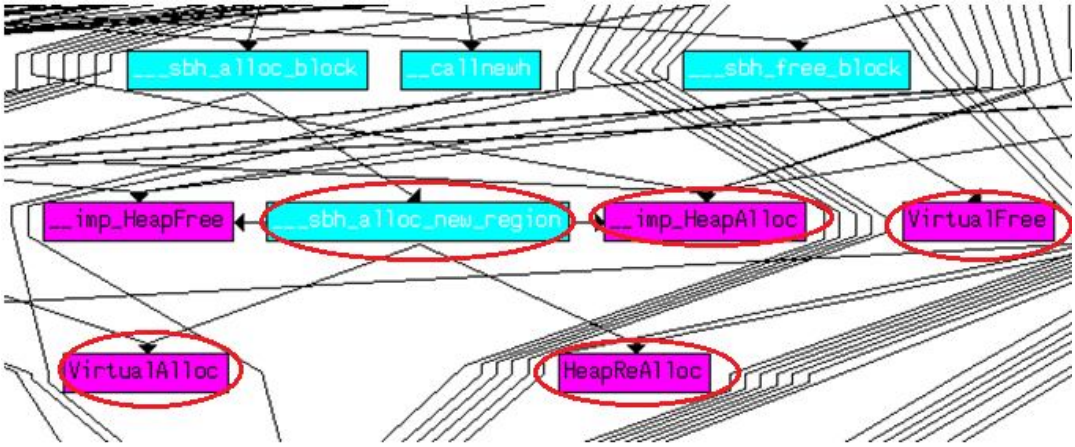
- **Funciones sospechosas (Codificación)**



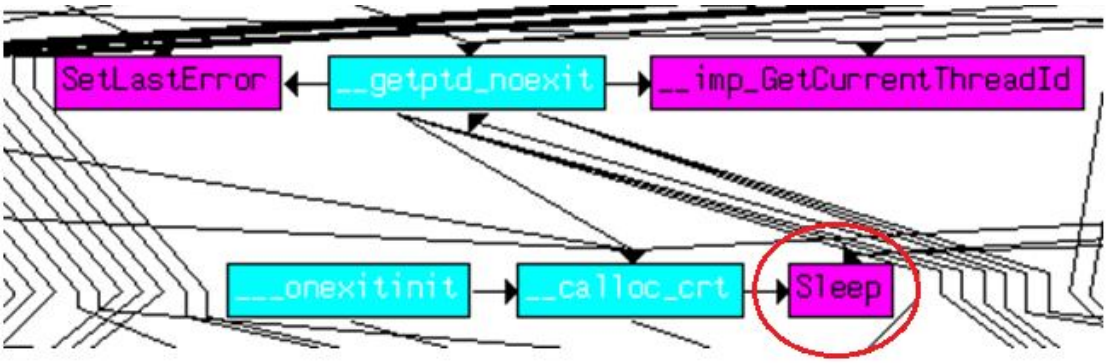
- **Funciones sospechosas (Modificación Procesos y Archivos)**



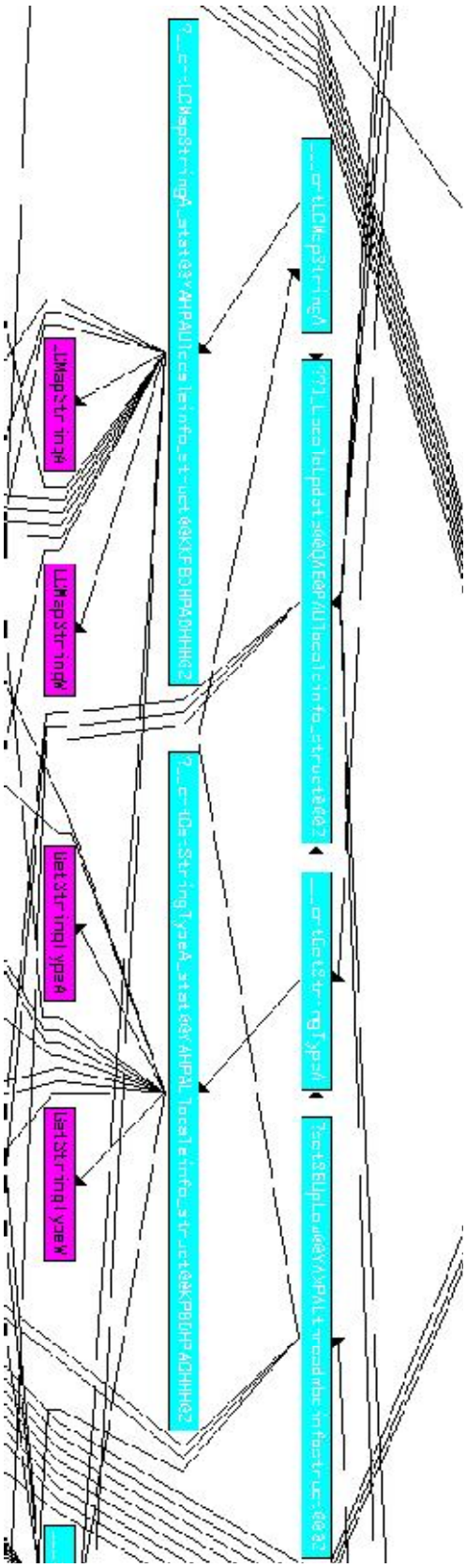
- **Funciones sospechosas (uso memoria)**



- **Funciones sospechosas (sleep)**

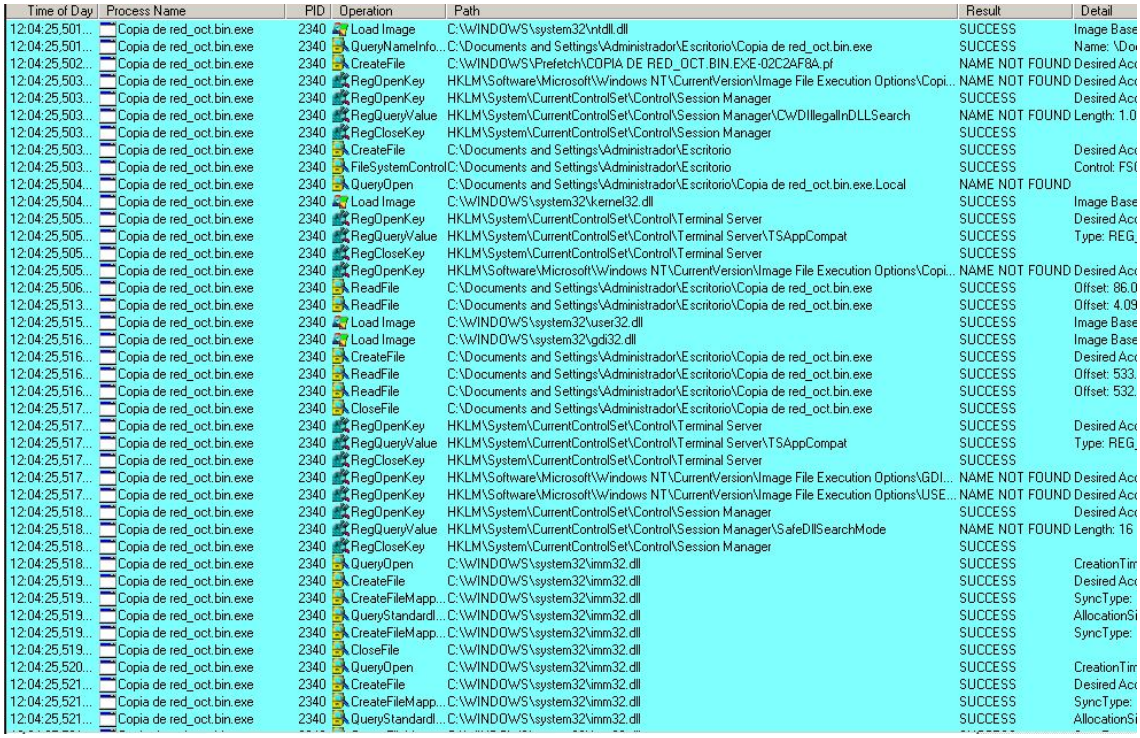


- **Parámetros codificados**

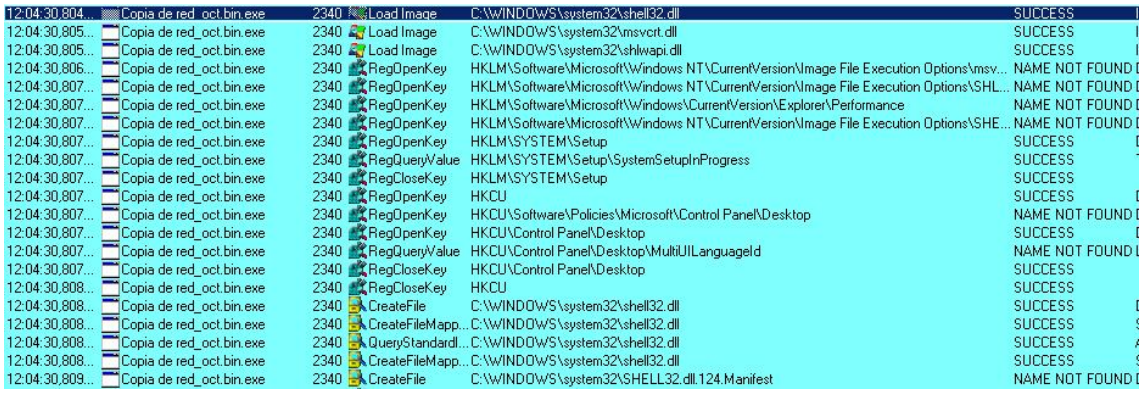


Anexo P Análisis Comportamiento – Proceso de Infección "Process Monitor"

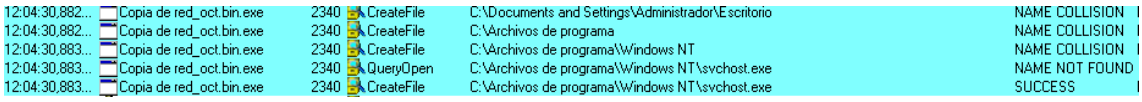
- **Secuencia del proceso de infección**



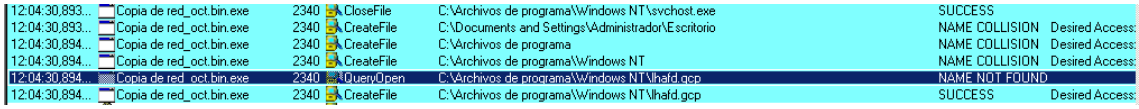
Time of Day	Process Name	PID	Operation	Path	Result	Detail
12:04:25,501...	Copia de red_oct.bin.exe	2340	Load Image	C:\WINDOWS\system32\ntdll.dll	SUCCESS	Image Base
12:04:25,501...	Copia de red_oct.bin.exe	2340	QueryNameInfo...	C:\Documents and Settings\Administrador\Escritorio\Copia de red_oct.bin.exe	SUCCESS	Name: \Do
12:04:25,502...	Copia de red_oct.bin.exe	2340	CreateFile	C:\WINDOWS\Prefetch\COPIA DE RED_OCT.BIN.EXE-02C2AF8A.pf	NAME NOT FOUND	Desired Acc
12:04:25,503...	Copia de red_oct.bin.exe	2340	RegOpenKey	HKLM\Software\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\Copi...	NAME NOT FOUND	Desired Acc
12:04:25,503...	Copia de red_oct.bin.exe	2340	RegOpenKey	HKLM\System\CurrentControlSet\Control\Session Manager	SUCCESS	Desired Acc
12:04:25,503...	Copia de red_oct.bin.exe	2340	RegQueryValue	HKLM\System\CurrentControlSet\Control\Session Manager\CWDIllegalInDLLSearch	NAME NOT FOUND	Length: 1.0
12:04:25,503...	Copia de red_oct.bin.exe	2340	RegCloseKey	HKLM\System\CurrentControlSet\Control\Session Manager	SUCCESS	
12:04:25,503...	Copia de red_oct.bin.exe	2340	CreateFile	C:\Documents and Settings\Administrador\Escritorio	SUCCESS	Desired Acc
12:04:25,503...	Copia de red_oct.bin.exe	2340	FileSystemControl	C:\Documents and Settings\Administrador\Escritorio	SUCCESS	Control: FS
12:04:25,504...	Copia de red_oct.bin.exe	2340	QueryOpen	C:\Documents and Settings\Administrador\Escritorio\Copia de red_oct.bin.exe.Local	NAME NOT FOUND	
12:04:25,504...	Copia de red_oct.bin.exe	2340	Load Image	C:\WINDOWS\system32\kernel32.dll	SUCCESS	Image Base
12:04:25,505...	Copia de red_oct.bin.exe	2340	RegOpenKey	HKLM\System\CurrentControlSet\Control\Terminal Server	SUCCESS	Desired Acc
12:04:25,505...	Copia de red_oct.bin.exe	2340	RegQueryValue	HKLM\System\CurrentControlSet\Control\Terminal Server\TSAppCompat	SUCCESS	Type: REG_
12:04:25,505...	Copia de red_oct.bin.exe	2340	RegCloseKey	HKLM\System\CurrentControlSet\Control\Terminal Server	SUCCESS	
12:04:25,505...	Copia de red_oct.bin.exe	2340	RegOpenKey	HKLM\Software\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\Copi...	NAME NOT FOUND	Desired Acc
12:04:25,506...	Copia de red_oct.bin.exe	2340	ReadFile	C:\Documents and Settings\Administrador\Escritorio\Copia de red_oct.bin.exe	SUCCESS	Offset: 86.0
12:04:25,513...	Copia de red_oct.bin.exe	2340	ReadFile	C:\Documents and Settings\Administrador\Escritorio\Copia de red_oct.bin.exe	SUCCESS	Offset: 4.09
12:04:25,515...	Copia de red_oct.bin.exe	2340	Load Image	C:\WINDOWS\system32\user32.dll	SUCCESS	Image Base
12:04:25,516...	Copia de red_oct.bin.exe	2340	Load Image	C:\WINDOWS\system32\gdi32.dll	SUCCESS	Image Base
12:04:25,516...	Copia de red_oct.bin.exe	2340	CreateFile	C:\Documents and Settings\Administrador\Escritorio\Copia de red_oct.bin.exe	SUCCESS	Desired Acc
12:04:25,516...	Copia de red_oct.bin.exe	2340	ReadFile	C:\Documents and Settings\Administrador\Escritorio\Copia de red_oct.bin.exe	SUCCESS	Offset: 533
12:04:25,516...	Copia de red_oct.bin.exe	2340	ReadFile	C:\Documents and Settings\Administrador\Escritorio\Copia de red_oct.bin.exe	SUCCESS	Offset: 532
12:04:25,517...	Copia de red_oct.bin.exe	2340	CloseFile	C:\Documents and Settings\Administrador\Escritorio\Copia de red_oct.bin.exe	SUCCESS	
12:04:25,517...	Copia de red_oct.bin.exe	2340	RegOpenKey	HKLM\System\CurrentControlSet\Control\Terminal Server	SUCCESS	Desired Acc
12:04:25,517...	Copia de red_oct.bin.exe	2340	RegQueryValue	HKLM\System\CurrentControlSet\Control\Terminal Server\TSAppCompat	SUCCESS	Type: REG_
12:04:25,517...	Copia de red_oct.bin.exe	2340	RegCloseKey	HKLM\System\CurrentControlSet\Control\Terminal Server	SUCCESS	
12:04:25,517...	Copia de red_oct.bin.exe	2340	RegOpenKey	HKLM\Software\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\GDI...	NAME NOT FOUND	Desired Acc
12:04:25,517...	Copia de red_oct.bin.exe	2340	RegOpenKey	HKLM\Software\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\USE...	NAME NOT FOUND	Desired Acc
12:04:25,518...	Copia de red_oct.bin.exe	2340	RegOpenKey	HKLM\System\CurrentControlSet\Control\Session Manager	SUCCESS	Desired Acc
12:04:25,518...	Copia de red_oct.bin.exe	2340	RegQueryValue	HKLM\System\CurrentControlSet\Control\Session Manager\SafeDllSearchMode	NAME NOT FOUND	Length: 16
12:04:25,518...	Copia de red_oct.bin.exe	2340	RegCloseKey	HKLM\System\CurrentControlSet\Control\Session Manager	SUCCESS	
12:04:25,518...	Copia de red_oct.bin.exe	2340	QueryOpen	C:\WINDOWS\system32\imm32.dll	SUCCESS	CreationTim
12:04:25,519...	Copia de red_oct.bin.exe	2340	CreateFile	C:\WINDOWS\system32\imm32.dll	SUCCESS	Desired Acc
12:04:25,519...	Copia de red_oct.bin.exe	2340	CreateFileMapp...	C:\WINDOWS\system32\imm32.dll	SUCCESS	SyncType:
12:04:25,519...	Copia de red_oct.bin.exe	2340	QueryStandardI...	C:\WINDOWS\system32\imm32.dll	SUCCESS	AllocationSi
12:04:25,519...	Copia de red_oct.bin.exe	2340	CreateFileMapp...	C:\WINDOWS\system32\imm32.dll	SUCCESS	SyncType:
12:04:25,519...	Copia de red_oct.bin.exe	2340	CloseFile	C:\WINDOWS\system32\imm32.dll	SUCCESS	
12:04:25,520...	Copia de red_oct.bin.exe	2340	QueryOpen	C:\WINDOWS\system32\imm32.dll	SUCCESS	CreationTim
12:04:25,521...	Copia de red_oct.bin.exe	2340	CreateFile	C:\WINDOWS\system32\imm32.dll	SUCCESS	Desired Acc
12:04:25,521...	Copia de red_oct.bin.exe	2340	CreateFileMapp...	C:\WINDOWS\system32\imm32.dll	SUCCESS	SyncType:
12:04:25,521...	Copia de red_oct.bin.exe	2340	QueryStandardI...	C:\WINDOWS\system32\imm32.dll	SUCCESS	AllocationSi



12:04:30,804...	Copia de red_oct.bin.exe	2340	Load Image	C:\WINDOWS\system32\shell32.dll	SUCCESS	I
12:04:30,805...	Copia de red_oct.bin.exe	2340	Load Image	C:\WINDOWS\system32\msvcrt.dll	SUCCESS	I
12:04:30,805...	Copia de red_oct.bin.exe	2340	Load Image	C:\WINDOWS\system32\shlwapi.dll	SUCCESS	I
12:04:30,806...	Copia de red_oct.bin.exe	2340	RegOpenKey	HKLM\Software\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\msv...	NAME NOT FOUND	I
12:04:30,807...	Copia de red_oct.bin.exe	2340	RegOpenKey	HKLM\Software\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\SHL...	NAME NOT FOUND	I
12:04:30,807...	Copia de red_oct.bin.exe	2340	RegOpenKey	HKLM\Software\Microsoft\Windows\CurrentVersion\Explorer\Performance	NAME NOT FOUND	I
12:04:30,807...	Copia de red_oct.bin.exe	2340	RegOpenKey	HKLM\Software\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\SHE...	NAME NOT FOUND	I
12:04:30,807...	Copia de red_oct.bin.exe	2340	RegOpenKey	HKLM\SYSTEM\Setup	SUCCESS	I
12:04:30,807...	Copia de red_oct.bin.exe	2340	RegQueryValue	HKLM\SYSTEM\Setup\SystemSetupInProgress	SUCCESS	
12:04:30,807...	Copia de red_oct.bin.exe	2340	RegCloseKey	HKLM\SYSTEM\Setup	SUCCESS	
12:04:30,807...	Copia de red_oct.bin.exe	2340	RegOpenKey	HKCU	SUCCESS	I
12:04:30,807...	Copia de red_oct.bin.exe	2340	RegOpenKey	HKCU\Software\Policies\Microsoft\Control Panel\Desktop	NAME NOT FOUND	I
12:04:30,807...	Copia de red_oct.bin.exe	2340	RegOpenKey	HKCU\Control Panel\Desktop	SUCCESS	I
12:04:30,807...	Copia de red_oct.bin.exe	2340	RegQueryValue	HKCU\Control Panel\Desktop\MultiUILanguageId	NAME NOT FOUND	I
12:04:30,807...	Copia de red_oct.bin.exe	2340	RegCloseKey	HKCU\Control Panel\Desktop	SUCCESS	
12:04:30,808...	Copia de red_oct.bin.exe	2340	RegCloseKey	HKCU	SUCCESS	
12:04:30,808...	Copia de red_oct.bin.exe	2340	CreateFile	C:\WINDOWS\system32\shell32.dll	SUCCESS	I
12:04:30,808...	Copia de red_oct.bin.exe	2340	CreateFileMapp...	C:\WINDOWS\system32\shell32.dll	SUCCESS	S
12:04:30,808...	Copia de red_oct.bin.exe	2340	QueryStandardI...	C:\WINDOWS\system32\shell32.dll	SUCCESS	A
12:04:30,808...	Copia de red_oct.bin.exe	2340	CreateFileMapp...	C:\WINDOWS\system32\shell32.dll	SUCCESS	S
12:04:30,809...	Copia de red_oct.bin.exe	2340	CreateFile	C:\WINDOWS\system32\SHELL32.dll.124.Manifest	NAME NOT FOUND	I



12:04:30,882...	Copia de red_oct.bin.exe	2340	CreateFile	C:\Documents and Settings\Administrador\Escritorio	NAME COLLISION	I
12:04:30,882...	Copia de red_oct.bin.exe	2340	CreateFile	C:\Archivos de programa	NAME COLLISION	I
12:04:30,883...	Copia de red_oct.bin.exe	2340	CreateFile	C:\Archivos de programa\Windows NT	NAME COLLISION	I
12:04:30,883...	Copia de red_oct.bin.exe	2340	QueryOpen	C:\Archivos de programa\Windows NT\svchost.exe	NAME NOT FOUND	
12:04:30,883...	Copia de red_oct.bin.exe	2340	CreateFile	C:\Archivos de programa\Windows NT\svchost.exe	SUCCESS	I



12:04:30,893...	Copia de red_oct.bin.exe	2340	CloseFile	C:\Archivos de programa\Windows NT\svchost.exe	SUCCESS	
12:04:30,893...	Copia de red_oct.bin.exe	2340	CreateFile	C:\Documents and Settings\Administrador\Escritorio	NAME COLLISION	Desired Access:
12:04:30,894...	Copia de red_oct.bin.exe	2340	CreateFile	C:\Archivos de programa	NAME COLLISION	Desired Access:
12:04:30,894...	Copia de red_oct.bin.exe	2340	CreateFile	C:\Archivos de programa\Windows NT	NAME COLLISION	Desired Access:
12:04:30,894...	Copia de red_oct.bin.exe	2340	QueryOpen	C:\Archivos de programa\Windows NT\lhafd.gcp	NAME NOT FOUND	
12:04:30,894...	Copia de red_oct.bin.exe	2340	CreateFile	C:\Archivos de programa\Windows NT\lhafd.gcp	SUCCESS	Desired Access:

12:04:30,906...	Copia de red_oct.bin.exe	2340	CreateFile	C:\Archivos de programa\Windows NT\lhafd.gcp	SUCCESS	Desired A
12:04:30,906...	Copia de red_oct.bin.exe	2340	SetBasicInform...	C:\Archivos de programa\Windows NT\lhafd.gcp	SUCCESS	CreationT
12:04:30,906...	lsass.exe	808	RegOpenKey	HKLM\SAM\SAM\DOMAINS\Builtin\Groups\00000220	NAME NOT FOUND	Desired A
12:04:30,906...	lsass.exe	808	RegOpenKey	HKLM\SAM\SAM\DOMAINS\Builtin\Aliases\00000220	SUCCESS	Desired A
12:04:30,906...	lsass.exe	808	RegQueryValue	HKLM\SAM\SAM\Domains\Builtin\Aliases\00000220\C	SUCCESS	Type: RE
12:04:30,907...	lsass.exe	808	RegCloseKey	HKLM\SAM\SAM\Domains\Builtin\Aliases\00000220	SUCCESS	
12:04:30,907...	winlogon.exe	752	NotifyChangeDi...	C:\Archivos de programa\Windows NT		Filter: FILE
12:04:30,907...	Copia de red_oct.bin.exe	2340	CloseFile	C:\Archivos de programa\Windows NT\lhafd.gcp	SUCCESS	
12:04:30,907...	lsass.exe	808	RegCloseKey	HKLM\SECURITY\Policy	SUCCESS	
12:04:30,907...	diskpl.exe	484	NotifyChangeDi...	C:\	SUCCESS	Filter: FILE
12:04:30,907...	Copia de red_oct.bin.exe	2340	CreateFile	C:\Archivos de programa\Windows NT\lhafd.gcp	SUCCESS	Desired A
12:04:30,907...	diskpl.exe	484	CreateFile	C:\Archivos de programa\Windows NT	SUCCESS	Desired A
12:04:30,907...	Copia de red_oct.bin.exe	2340	SetBasicInform...	C:\Archivos de programa\Windows NT\lhafd.gcp	SUCCESS	CreationT
12:04:30,907...	diskpl.exe	484	QueryDirectory	C:\Archivos de programa\Windows NT\lhafd.gcp	SUCCESS	Filter: lhaf
12:04:30,907...	diskpl.exe	484	CloseFile	C:\Archivos de programa\Windows NT	SUCCESS	
12:04:30,907...	Copia de red_oct.bin.exe	2340	CloseFile	C:\Archivos de programa\Windows NT\lhafd.gcp	SUCCESS	
12:04:30,908...	diskpl.exe	484	CreateFile	C:\Archivos de programa\Windows NT\lhafd.gcp	SUCCESS	Desired A
12:04:30,908...	Copia de red_oct.bin.exe	2340	CreateFile	C:\Archivos de programa\Windows NT\svchost.exe	SUCCESS	Desired A

12:04:30,950...	svchost.exe	2464	QueryNameInfo...	C:\Archivos de programa\Windows NT\svchost.exe	SUCCESS	Name: \Archivo
12:04:30,950...	svchost.exe	2464	ReadFile	C:\Archivos de programa\Windows NT\svchost.exe	SUCCESS	Offset: 188.416
12:04:30,951...	Copia de red_oct.bin.exe	2340	CreateFile	C:\Documents and Settings\Administrador\Configuración local\Temp\msc.bat	SUCCESS	Desired Access
12:04:30,952...	Copia de red_oct.bin.exe	2340	CreateFile	C:\Documents and Settings\Administrador\Configuración local\Temp	SUCCESS	Desired Access
12:04:30,952...	Copia de red_oct.bin.exe	2340	CloseFile	C:\Documents and Settings\Administrador\Configuración local\Temp	SUCCESS	

Process Monitor - Sysinternals: www.sysinternals.com

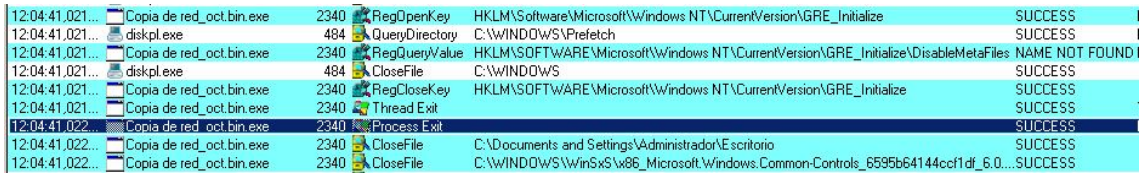
File Edit Event Filter Tools Options Help

Time of Day	Process Name	PID	Operation	Path	Result	Detail
12:04:36,810...	svchost.exe	2464	ReadFile	C:\Archivos de programa\Windows NT\svchost.exe	SUCCESS	Offset: 20
12:04:36,837...	svchost.exe	2464	ReadFile	C:\Archivos de programa\Windows NT\svchost.exe	SUCCESS	Offset: 12
12:04:36,838...	svchost.exe	2464	ReadFile	C:\Archivos de programa\Windows NT\svchost.exe	SUCCESS	Offset: 13
12:04:36,839...	svchost.exe	2464	ReadFile	C:\Archivos de programa\Windows NT\svchost.exe	SUCCESS	Offset: 15
12:04:36,840...	svchost.exe	2464	ReadFile	C:\Archivos de programa\Windows NT\svchost.exe	SUCCESS	Offset: 17
12:04:36,865...	svchost.exe	2464	Load Image	C:\WINDOWS\system32\shlwapi.dll	SUCCESS	Image Ba
12:04:36,866...	svchost.exe	2464	Load Image	C:\WINDOWS\system32\msvcrt.dll	SUCCESS	Image Ba
12:04:36,867...	svchost.exe	2464	RegOpenKey	HKLM\Software\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\msv...	NAME NOT FOUND	Desired A
12:04:36,867...	svchost.exe	2464	RegOpenKey	HKLM\Software\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\SHL...	NAME NOT FOUND	Desired A
12:04:36,867...	svchost.exe	2464	RegOpenKey	HKLM\Software\Microsoft\Windows\CurrentVersion\Explorer\Performance	NAME NOT FOUND	Desired A
12:04:36,868...	svchost.exe	2464	QueryOpen	C:\Archivos de programa\Windows NT\WS2_32.dll	NAME NOT FOUND	
12:04:36,869...	svchost.exe	2464	QueryOpen	C:\WINDOWS\system32\ws2_32.dll	SUCCESS	CreationT
12:04:36,869...	svchost.exe	2464	CreateFile	C:\WINDOWS\system32\ws2_32.dll	SUCCESS	Desired A
12:04:36,869...	svchost.exe	2464	CreateFileMapp...	C:\WINDOWS\system32\ws2_32.dll	SUCCESS	SyncType
12:04:36,869...	svchost.exe	2464	CreateFileMapp...	C:\WINDOWS\system32\ws2_32.dll	SUCCESS	SyncType
12:04:36,870...	svchost.exe	2464	CloseFile	C:\WINDOWS\system32\ws2_32.dll	SUCCESS	
12:04:36,870...	svchost.exe	2464	Load Image	C:\WINDOWS\system32\ws2_32.dll	SUCCESS	Image Ba
12:04:36,871...	svchost.exe	2464	QueryOpen	C:\Archivos de programa\Windows NT\WS2HELP.dll	NAME NOT FOUND	

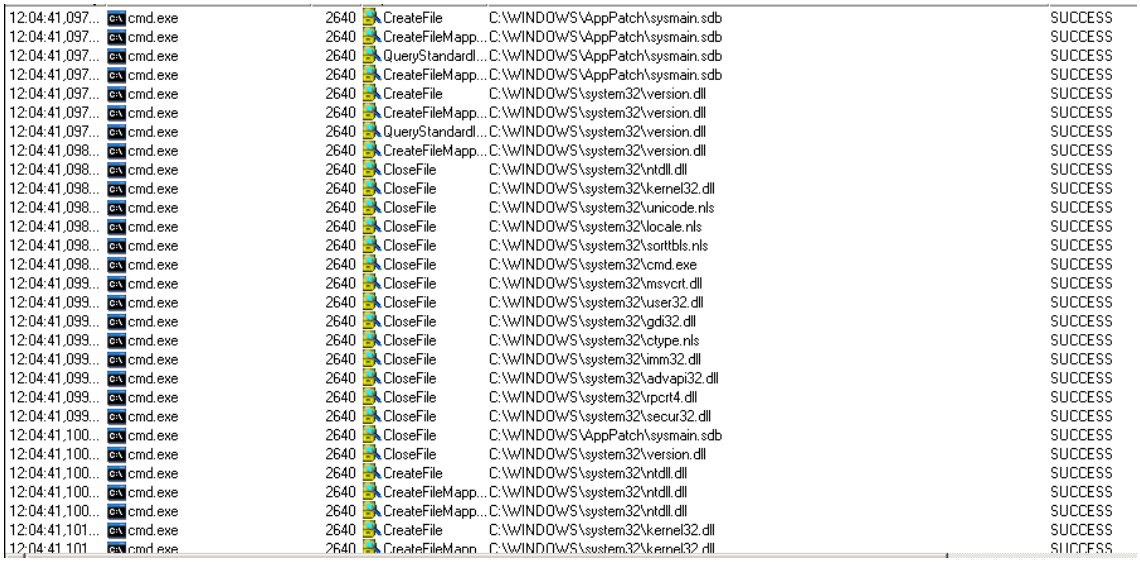
12:04:36,898...	svchost.exe	2464	CloseFile	C:\Archivos de programa\Windows NT\svchost.exe	SUCCESS	
12:04:36,898...	svchost.exe	2464	CreateFile	C:\	SUCCESS	Desired Acc
12:04:36,898...	svchost.exe	2464	QueryNameInfo...	C:\	SUCCESS	Name: \
12:04:36,898...	svchost.exe	2464	QueryInformatio...	C:\	SUCCESS	VolumeCrea
12:04:36,898...	svchost.exe	2464	CloseFile	C:\	SUCCESS	
12:04:36,898...	svchost.exe	2464	RegOpenKey	HKCR\CLSID\{00000000-5C69-A827-83B8-EC2FE0CFC75B}	NAME NOT FOUND	Desired Acc
12:04:36,898...	svchost.exe	2464	RegOpenKey	HKCR\CLSID\{00000000-5C69-A827-83B8-EC2FE0CFC75B}	NAME NOT FOUND	Desired Acc
12:04:36,898...	svchost.exe	2464	RegCreateKey	HKCR\CLSID\{00000000-5C69-A827-83B8-EC2FE0CFC75B}	SUCCESS	Desired Acc
12:04:36,898...	svchost.exe	2464	SetEndOfFileInf...	C:\WINDOWS\system32\config\software.LOG	SUCCESS	EndOfFile: 1
12:04:36,899...	diskpl.exe	484	NotifyChangeDi...	C:\	SUCCESS	Filter: FILE_
12:04:36,899...	svchost.exe	2464	SetEndOfFileInf...	C:\WINDOWS\system32\config\software.LOG	SUCCESS	EndOfFile: 1
12:04:36,899...	diskpl.exe	484	NotifyChangeDi...	C:\	SUCCESS	Filter: FILE_
12:04:36,899...	svchost.exe	2464	SetEndOfFileInf...	C:\WINDOWS\system32\config\software.LOG	SUCCESS	EndOfFile: 2
12:04:36,899...	diskpl.exe	484	NotifyChangeDi...	C:\	SUCCESS	Filter: FILE_
12:04:36,899...	svchost.exe	2464	SetEndOfFileInf...	C:\WINDOWS\system32\config\software.LOG	SUCCESS	EndOfFile: 2
12:04:36,900...	diskpl.exe	484	NotifyChangeDi...	C:\	SUCCESS	Filter: FILE_
12:04:36,900...	svchost.exe	2464	SetEndOfFileInf...	C:\WINDOWS\system32\config\software.LOG	SUCCESS	EndOfFile: 2
12:04:36,900...	diskpl.exe	484	NotifyChangeDi...	C:\	SUCCESS	Filter: FILE_
12:04:36,900...	svchost.exe	2464	SetEndOfFileInf...	C:\WINDOWS\system32\config\software.LOG	SUCCESS	EndOfFile: 4
12:04:36,901...	diskpl.exe	484	NotifyChangeDi...	C:\	SUCCESS	Filter: FILE_
12:04:36,901...	svchost.exe	2464	RegSetValue	HKCR\CLSID\{00000000-5C69-A827-83B8-EC2FE0CFC75B}\(Default)	SUCCESS	Type: REG_
12:04:36,901...	svchost.exe	2464	RegQueryValue	HKCR\CLSID\{00000000-5C69-A827-83B8-EC2FE0CFC75B}\InfoTip	NAME NOT FOUND	Length: 144
12:04:36,901...	vmtoolsd.exe	372	RegQueryKey	HKCU\Software\Classes	SUCCESS	Query: Nam
12:04:36,901...	svchost.exe	2464	RegCloseKey	HKCR\CLSID\{00000000-5C69-A827-83B8-EC2FE0CFC75B}	SUCCESS	

12:04:40,954...	Copia de red_oct.bin.exe	2340	RegOpenKey	HKLM\System\CurrentControlSet\Control\Session Manager	SUCCESS	
12:04:40,954...	Copia de red_oct.bin.exe	2340	RegQueryValue	HKLM\System\CurrentControlSet\Control\Session Manager\SafeProcessSearchMode	NAME NOT FOUND	
12:04:40,954...	Copia de red_oct.bin.exe	2340	RegCloseKey	HKLM\System\CurrentControlSet\Control\Session Manager	SUCCESS	
12:04:40,955...	svchost.exe	1184	CreateFile	C:\WINDOWS\Prefetch\SVCHOST.EXE-07C31807.pf	NAME NOT FOUND	
12:04:40,955...	svchost.exe	1184	QueryOpen	C:\Archivos de programa\Windows NT\svchost.exe	SUCCESS	
12:04:40,956...	Copia de red_oct.bin.exe	2340	QueryOpen	C:\Documents and Settings\Administrador\Configuración local\Temp\msc.bat	SUCCESS	
12:04:40,956...	svchost.exe	1184	CreateFile	C:\Archivos de programa\Windows NT\svchost.exe	SUCCESS	
12:04:40,956...	svchost.exe	1184	QueryFileIntern...	C:\Archivos de programa\Windows NT\svchost.exe	SUCCESS	
12:04:40,956...	svchost.exe	1184	CloseFile	C:\Archivos de programa\Windows NT\svchost.exe	SUCCESS	
12:04:40,957...	svchost.exe	1184	QueryOpen	C:\WINDOWS\system32\advapi32.dll	SUCCESS	
12:04:40,957...	Copia de red_oct.bin.exe	2340	QueryOpen	C:\Documents and Settings\Administrador\Configuración local\Temp\msc.bat	SUCCESS	
12:04:40,958...	svchost.exe	1184	CreateFile	C:\WINDOWS\system32\advapi32.dll	SUCCESS	
12:04:40,958...	svchost.exe	1184	QueryFileIntern...	C:\WINDOWS\system32\advapi32.dll	SUCCESS	
12:04:40,958...	svchost.exe	1184	CloseFile	C:\WINDOWS\system32\advapi32.dll	SUCCESS	
12:04:40,959...	Copia de red_oct.bin.exe	2340	CreateFile	C:\Documents and Settings\Administrador\Configuración local\Temp\msc.bat	SUCCESS	
12:04:40,959...	Copia de red_oct.bin.exe	2340	CreateFileMapp...	C:\Documents and Settings\Administrador\Configuración local\Temp\msc.bat	SUCCESS	
12:04:40,959...	Copia de red_oct.bin.exe	2340	QueryStandardI...	C:\Documents and Settings\Administrador\Configuración local\Temp\msc.bat	SUCCESS	
12:04:40,959...	Copia de red_oct.bin.exe	2340	CreateFileMapp...	C:\Documents and Settings\Administrador\Configuración local\Temp\msc.bat	SUCCESS	
12:04:40,959...	Copia de red_oct.bin.exe	2340	ReadFile	C:\WINDOWS\system32\kernel32.dll	SUCCESS	

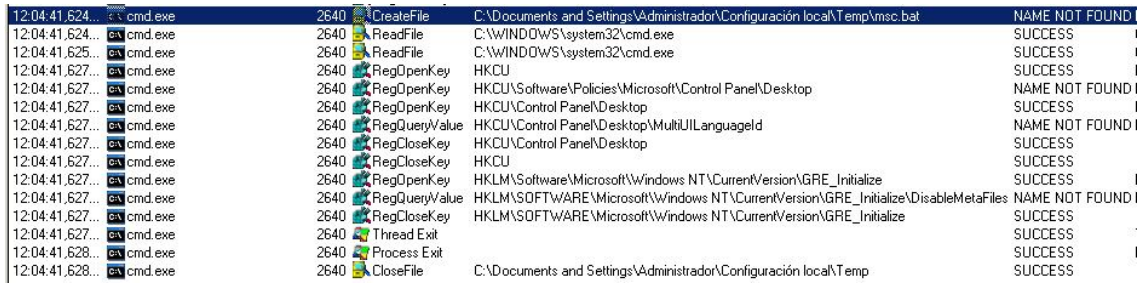
12:04:41,007...	Copia de red_oct.bin.exe	2340	CreateFileMapp...	C:\Documents and Settings\Administrador\Configuración local\Temp\msc.bat	SUCCESS	
12:04:41,008...	Copia de red_oct.bin.exe	2340	RegOpenKey	HKLM\Software\Policies\Microsoft\Windows\Safer\CodeIdentifiers	SUCCESS	
12:04:41,008...	Copia de red_oct.bin.exe	2340	RegQueryValue	HKLM\SOFTWARE\Policies\Microsoft\Windows\Safer\CodeIdentifiers\LogFileName	NAME NOT FOUND	
12:04:41,008...	Copia de red_oct.bin.exe	2340	RegCloseKey	HKLM\SOFTWARE\Policies\Microsoft\Windows\Safer\CodeIdentifiers	SUCCESS	
12:04:41,008...	Copia de red_oct.bin.exe	2340	CloseFile	C:\Documents and Settings\Administrador\Configuración local\Temp\msc.bat	SUCCESS	



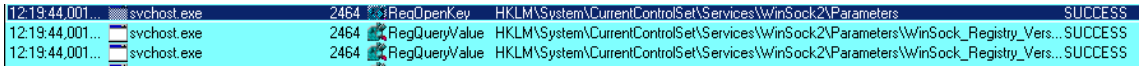
12:04:41,021...	Copia de red_oct.bin.exe	2340	RegOpenKey	HKLM\Software\Microsoft\Windows NT\CurrentVersion\GRE_Initialize	SUCCESS
12:04:41,021...	diskpl.exe	484	QueryDirectory	C:\WINDOWS\Prefetch	SUCCESS
12:04:41,021...	Copia de red_oct.bin.exe	2340	RegQueryValue	HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\GRE_Initialize\DisableMetaFiles	NAME NOT FOUND
12:04:41,021...	diskpl.exe	484	CloseFile	C:\WINDOWS	SUCCESS
12:04:41,021...	Copia de red_oct.bin.exe	2340	RegCloseKey	HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\GRE_Initialize	SUCCESS
12:04:41,021...	Copia de red_oct.bin.exe	2340	Thread Exit		SUCCESS
12:04:41,022...	Copia de red_oct.bin.exe	2340	Process Exit		SUCCESS
12:04:41,022...	Copia de red_oct.bin.exe	2340	CloseFile	C:\Documents and Settings\Administrador\Escritorio	SUCCESS
12:04:41,022...	Copia de red_oct.bin.exe	2340	CloseFile	C:\WINDOWS\WinSxS\x86_Microsoft.Windows.Common-Controls_6595b64144ccf1df_6.0....	SUCCESS



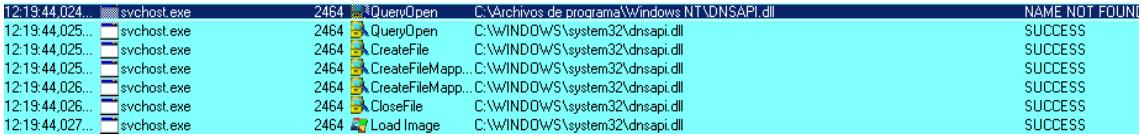
12:04:41,097...	cmd.exe	2640	CreateFile	C:\WINDOWS\AppPatch\sysmain.sdb	SUCCESS
12:04:41,097...	cmd.exe	2640	CreateFileMapp...	C:\WINDOWS\AppPatch\sysmain.sdb	SUCCESS
12:04:41,097...	cmd.exe	2640	QueryStandardI...	C:\WINDOWS\AppPatch\sysmain.sdb	SUCCESS
12:04:41,097...	cmd.exe	2640	CreateFileMapp...	C:\WINDOWS\AppPatch\sysmain.sdb	SUCCESS
12:04:41,097...	cmd.exe	2640	CreateFile	C:\WINDOWS\system32\version.dll	SUCCESS
12:04:41,097...	cmd.exe	2640	CreateFileMapp...	C:\WINDOWS\system32\version.dll	SUCCESS
12:04:41,097...	cmd.exe	2640	QueryStandardI...	C:\WINDOWS\system32\version.dll	SUCCESS
12:04:41,098...	cmd.exe	2640	CreateFileMapp...	C:\WINDOWS\system32\version.dll	SUCCESS
12:04:41,098...	cmd.exe	2640	CloseFile	C:\WINDOWS\system32\ntdll.dll	SUCCESS
12:04:41,098...	cmd.exe	2640	CloseFile	C:\WINDOWS\system32\kernel32.dll	SUCCESS
12:04:41,098...	cmd.exe	2640	CloseFile	C:\WINDOWS\system32\unicode.nls	SUCCESS
12:04:41,098...	cmd.exe	2640	CloseFile	C:\WINDOWS\system32\locale.nls	SUCCESS
12:04:41,098...	cmd.exe	2640	CloseFile	C:\WINDOWS\system32\sorttbls.nls	SUCCESS
12:04:41,098...	cmd.exe	2640	CloseFile	C:\WINDOWS\system32\cmd.exe	SUCCESS
12:04:41,099...	cmd.exe	2640	CloseFile	C:\WINDOWS\system32\msvcrt.dll	SUCCESS
12:04:41,099...	cmd.exe	2640	CloseFile	C:\WINDOWS\system32\user32.dll	SUCCESS
12:04:41,099...	cmd.exe	2640	CloseFile	C:\WINDOWS\system32\gdi32.dll	SUCCESS
12:04:41,099...	cmd.exe	2640	CloseFile	C:\WINDOWS\system32\ctype.nls	SUCCESS
12:04:41,099...	cmd.exe	2640	CloseFile	C:\WINDOWS\system32\imm32.dll	SUCCESS
12:04:41,099...	cmd.exe	2640	CloseFile	C:\WINDOWS\system32\advapi32.dll	SUCCESS
12:04:41,099...	cmd.exe	2640	CloseFile	C:\WINDOWS\system32\rpcrt4.dll	SUCCESS
12:04:41,099...	cmd.exe	2640	CloseFile	C:\WINDOWS\system32\secur32.dll	SUCCESS
12:04:41,100...	cmd.exe	2640	CloseFile	C:\WINDOWS\AppPatch\sysmain.sdb	SUCCESS
12:04:41,100...	cmd.exe	2640	CloseFile	C:\WINDOWS\system32\version.dll	SUCCESS
12:04:41,100...	cmd.exe	2640	CreateFile	C:\WINDOWS\system32\ntdll.dll	SUCCESS
12:04:41,100...	cmd.exe	2640	CreateFileMapp...	C:\WINDOWS\system32\ntdll.dll	SUCCESS
12:04:41,100...	cmd.exe	2640	CreateFileMapp...	C:\WINDOWS\system32\ntdll.dll	SUCCESS
12:04:41,101...	cmd.exe	2640	CreateFile	C:\WINDOWS\system32\kernel32.dll	SUCCESS
12:04:41,101...	cmd.exe	2640	CreateFileMapp...	C:\WINDOWS\system32\kernel32.dll	SUCCESS



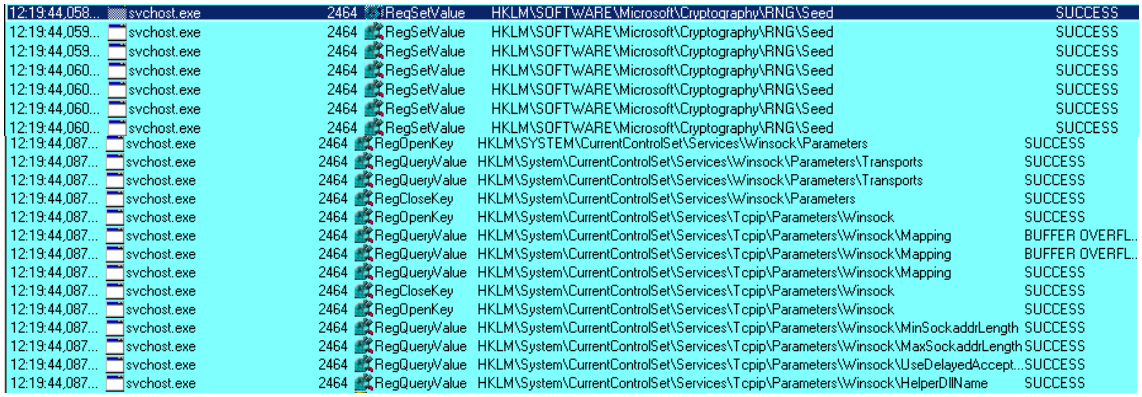
12:04:41,624...	cmd.exe	2640	CreateFile	C:\Documents and Settings\Administrador\Configuración local\Temp\msc.bat	NAME NOT FOUND
12:04:41,624...	cmd.exe	2640	ReadFile	C:\WINDOWS\system32\cmd.exe	SUCCESS
12:04:41,625...	cmd.exe	2640	ReadFile	C:\WINDOWS\system32\cmd.exe	SUCCESS
12:04:41,627...	cmd.exe	2640	RegOpenKey	HKCU	SUCCESS
12:04:41,627...	cmd.exe	2640	RegOpenKey	HKCU\Software\Policies\Microsoft\Control Panel\Desktop	NAME NOT FOUND
12:04:41,627...	cmd.exe	2640	RegOpenKey	HKCU\Control Panel\Desktop	SUCCESS
12:04:41,627...	cmd.exe	2640	RegQueryValue	HKCU\Control Panel\Desktop\MultiUILanguageId	NAME NOT FOUND
12:04:41,627...	cmd.exe	2640	RegCloseKey	HKCU\Control Panel\Desktop	SUCCESS
12:04:41,627...	cmd.exe	2640	RegCloseKey	HKCU	SUCCESS
12:04:41,627...	cmd.exe	2640	RegOpenKey	HKLM\Software\Microsoft\Windows NT\CurrentVersion\GRE_Initialize	SUCCESS
12:04:41,627...	cmd.exe	2640	RegQueryValue	HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\GRE_Initialize\DisableMetaFiles	NAME NOT FOUND
12:04:41,627...	cmd.exe	2640	RegCloseKey	HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\GRE_Initialize	SUCCESS
12:04:41,627...	cmd.exe	2640	Thread Exit		SUCCESS
12:04:41,628...	cmd.exe	2640	Process Exit		SUCCESS
12:04:41,628...	cmd.exe	2640	CloseFile	C:\Documents and Settings\Administrador\Configuración local\Temp	SUCCESS



12:19:44,001...	svchost.exe	2464	RegOpenKey	HKLM\System\CurrentControlSet\Services\WinSock2\Parameters	SUCCESS
12:19:44,001...	svchost.exe	2464	RegQueryValue	HKLM\System\CurrentControlSet\Services\WinSock2\Parameters\WinSock_Registry_Vers...	SUCCESS
12:19:44,001...	svchost.exe	2464	RegQueryValue	HKLM\System\CurrentControlSet\Services\WinSock2\Parameters\WinSock_Registry_Vers...	SUCCESS



12:19:44,024...	svchost.exe	2464	QueryOpen	C:\Archivos de programa\Windows NT\DNSAPI.dll	NAME NOT FOUND
12:19:44,025...	svchost.exe	2464	QueryOpen	C:\WINDOWS\system32\dnsapi.dll	SUCCESS
12:19:44,025...	svchost.exe	2464	CreateFile	C:\WINDOWS\system32\dnsapi.dll	SUCCESS
12:19:44,025...	svchost.exe	2464	CreateFileMapp...	C:\WINDOWS\system32\dnsapi.dll	SUCCESS
12:19:44,026...	svchost.exe	2464	CreateFileMapp...	C:\WINDOWS\system32\dnsapi.dll	SUCCESS
12:19:44,026...	svchost.exe	2464	CloseFile	C:\WINDOWS\system32\dnsapi.dll	SUCCESS
12:19:44,027...	svchost.exe	2464	Load Image	C:\WINDOWS\system32\dnsapi.dll	SUCCESS

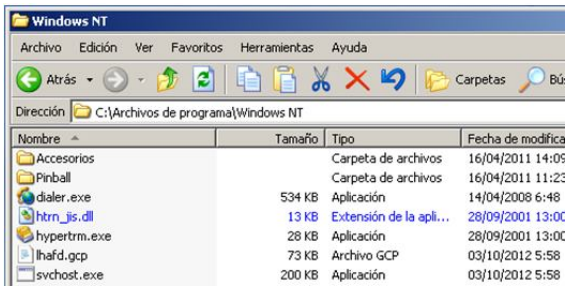


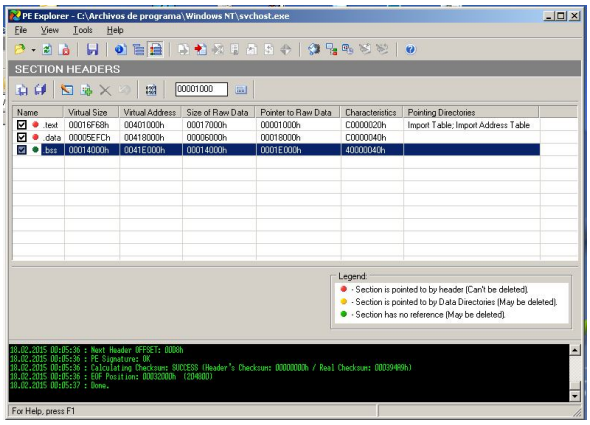
12:19:44,058...	svchost.exe	2464	RegSetValue	HKLM\SOFTWARE\Microsoft\Cryptography\RNG\Seed	SUCCESS
12:19:44,059...	svchost.exe	2464	RegSetValue	HKLM\SOFTWARE\Microsoft\Cryptography\RNG\Seed	SUCCESS
12:19:44,059...	svchost.exe	2464	RegSetValue	HKLM\SOFTWARE\Microsoft\Cryptography\RNG\Seed	SUCCESS
12:19:44,060...	svchost.exe	2464	RegSetValue	HKLM\SOFTWARE\Microsoft\Cryptography\RNG\Seed	SUCCESS
12:19:44,060...	svchost.exe	2464	RegSetValue	HKLM\SOFTWARE\Microsoft\Cryptography\RNG\Seed	SUCCESS
12:19:44,060...	svchost.exe	2464	RegSetValue	HKLM\SOFTWARE\Microsoft\Cryptography\RNG\Seed	SUCCESS
12:19:44,060...	svchost.exe	2464	RegSetValue	HKLM\SOFTWARE\Microsoft\Cryptography\RNG\Seed	SUCCESS
12:19:44,087...	svchost.exe	2464	RegOpenKey	HKLM\SYSTEM\CurrentControlSet\Services\Winsock\Parameters	SUCCESS
12:19:44,087...	svchost.exe	2464	RegQueryValue	HKLM\System\CurrentControlSet\Services\Winsock\Parameters\Transports	SUCCESS
12:19:44,087...	svchost.exe	2464	RegQueryValue	HKLM\System\CurrentControlSet\Services\Winsock\Parameters\Transports	SUCCESS
12:19:44,087...	svchost.exe	2464	RegCloseKey	HKLM\System\CurrentControlSet\Services\Winsock\Parameters	SUCCESS
12:19:44,087...	svchost.exe	2464	RegOpenKey	HKLM\System\CurrentControlSet\Services\Tcpip\Parameters\Winsock	SUCCESS
12:19:44,087...	svchost.exe	2464	RegQueryValue	HKLM\System\CurrentControlSet\Services\Tcpip\Parameters\Winsock\Mapping	BUFFER OVERFL...
12:19:44,087...	svchost.exe	2464	RegQueryValue	HKLM\System\CurrentControlSet\Services\Tcpip\Parameters\Winsock\Mapping	BUFFER OVERFL...
12:19:44,087...	svchost.exe	2464	RegQueryValue	HKLM\System\CurrentControlSet\Services\Tcpip\Parameters\Winsock\Mapping	SUCCESS
12:19:44,087...	svchost.exe	2464	RegCloseKey	HKLM\System\CurrentControlSet\Services\Tcpip\Parameters\Winsock	SUCCESS
12:19:44,087...	svchost.exe	2464	RegOpenKey	HKLM\System\CurrentControlSet\Services\Tcpip\Parameters\Winsock	SUCCESS
12:19:44,087...	svchost.exe	2464	RegQueryValue	HKLM\System\CurrentControlSet\Services\Tcpip\Parameters\Winsock\MinSockaddrLength	SUCCESS
12:19:44,087...	svchost.exe	2464	RegQueryValue	HKLM\System\CurrentControlSet\Services\Tcpip\Parameters\Winsock\MaxSockaddrLength	SUCCESS
12:19:44,087...	svchost.exe	2464	RegQueryValue	HKLM\System\CurrentControlSet\Services\Tcpip\Parameters\Winsock\UseDelayedAccept...	SUCCESS
12:19:44,087...	svchost.exe	2464	RegQueryValue	HKLM\System\CurrentControlSet\Services\Tcpip\Parameters\Winsock\HelperDllName	SUCCESS

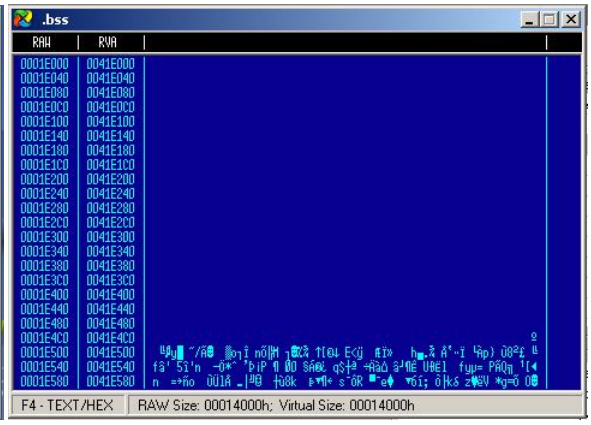
13:41:24,344...	svchost.exe	2464	RegOpenKey	HKLM\Software\Microsoft\Windows NT\CurrentVersion\Winlogon	SUCCESS
13:41:24,344...	svchost.exe	2464	RegQueryValue	HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon\Userinit	BUFFER OVERFL...
13:41:24,344...	svchost.exe	2464	RegQueryValue	HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon\Userinit	SUCCESS
13:41:24,344...	svchost.exe	2464	RegCloseKey	HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon	SUCCESS
13:41:24,345...	svchost.exe	2464	RegOpenKey	HKCU\Software\Microsoft\Windows\CurrentVersion\Run	SUCCESS
13:41:24,345...	svchost.exe	2464	RegQueryValue	HKCU\Software\Microsoft\Windows\CurrentVersion\Run\Lha	NAME NOT FOUND
13:41:24,345...	svchost.exe	2464	RegCloseKey	HKCU\Software\Microsoft\Windows\CurrentVersion\Run	SUCCESS
13:43:04,361...	svchost.exe	2464	RegOpenKey	HKLM\Software\Microsoft\Windows NT\CurrentVersion\Winlogon	SUCCESS
13:43:04,361...	svchost.exe	2464	RegQueryValue	HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon\Userinit	BUFFER OVERFL...
13:43:04,361...	svchost.exe	2464	RegQueryValue	HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon\Userinit	SUCCESS
13:43:04,361...	svchost.exe	2464	RegCloseKey	HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon	SUCCESS
13:43:04,361...	svchost.exe	2464	RegOpenKey	HKCU\Software\Microsoft\Windows\CurrentVersion\Run	SUCCESS
13:43:04,361...	svchost.exe	2464	RegQueryValue	HKCU\Software\Microsoft\Windows\CurrentVersion\Run\Lha	NAME NOT FOUND
13:43:04,361...	svchost.exe	2464	RegCloseKey	HKCU\Software\Microsoft\Windows\CurrentVersion\Run	SUCCESS

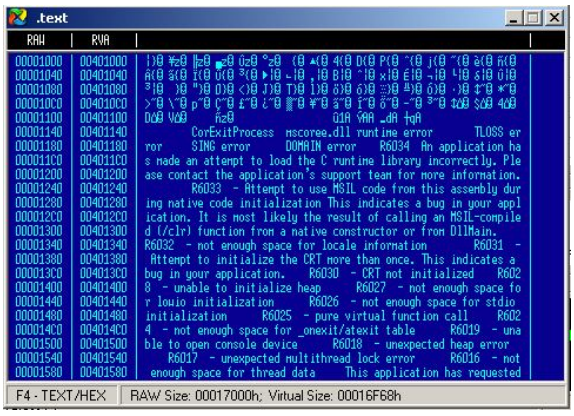
Anexo Q Análisis Comportamiento – Archivos creados por el *malware*

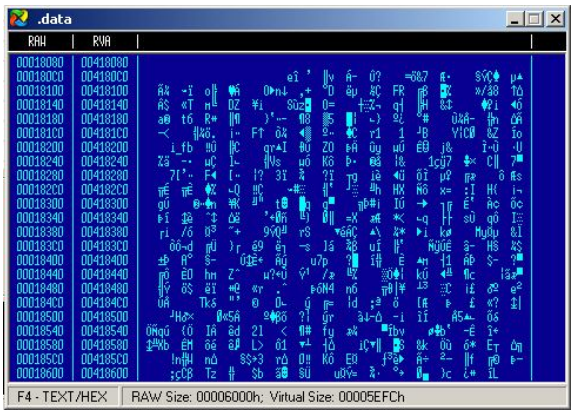
- **"svchost.exe": ubicación, contenido y propiedades.**

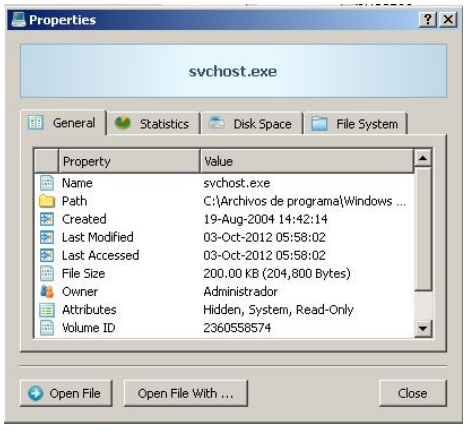




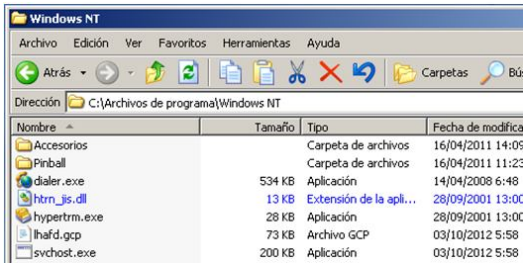


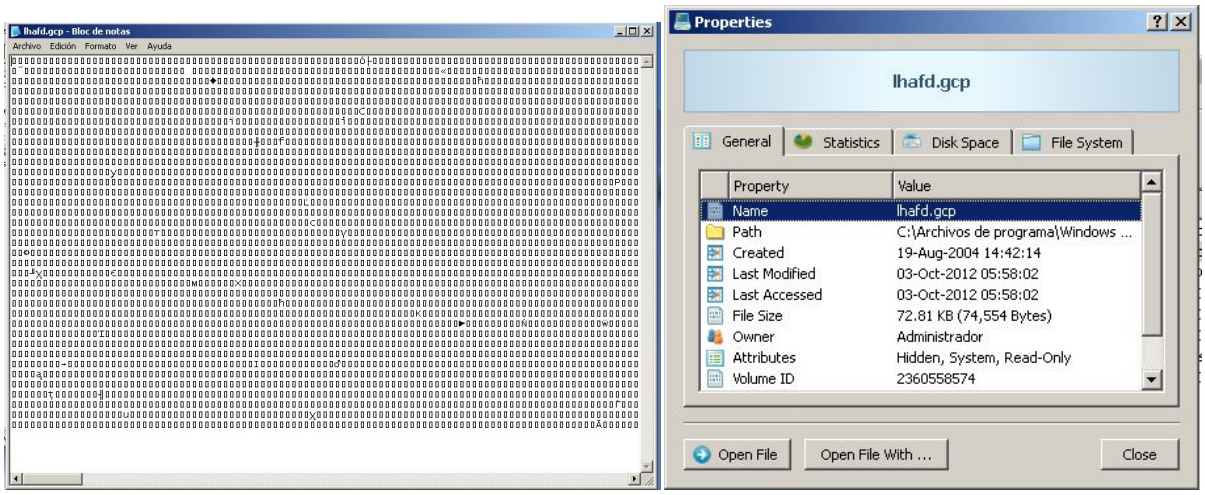




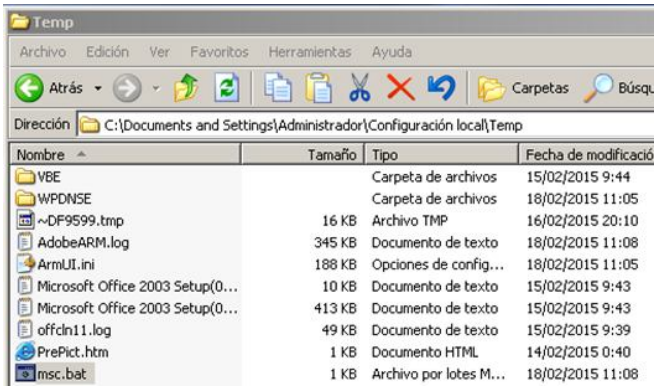


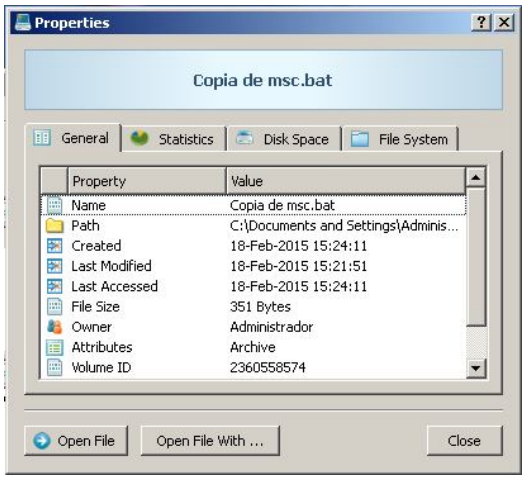
- **"lhafd.gcp.exe": ubicación, contenido y propiedades**

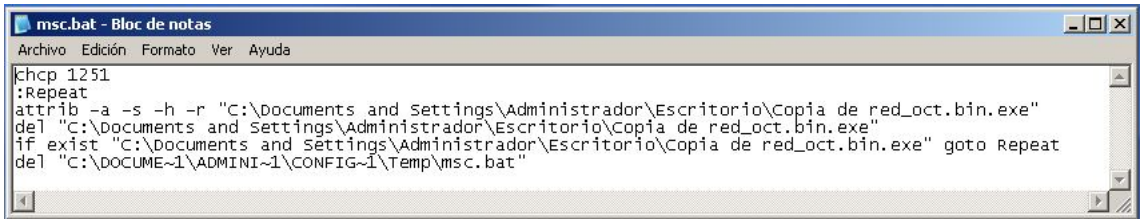




- **"lhafd.gcp.exe": ubicación, contenido y propiedades**

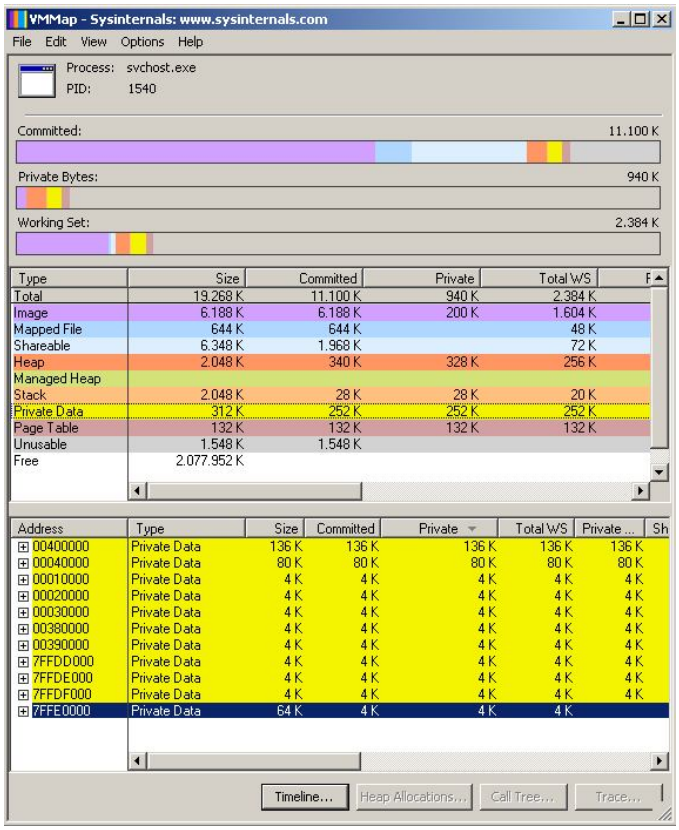


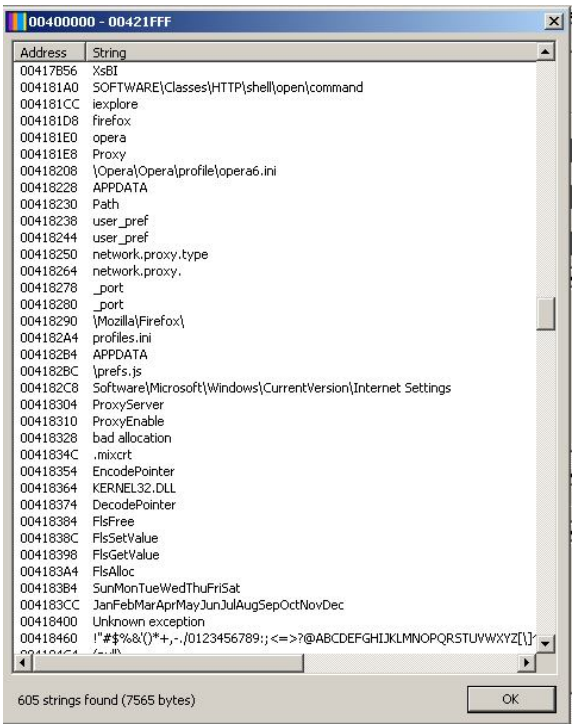




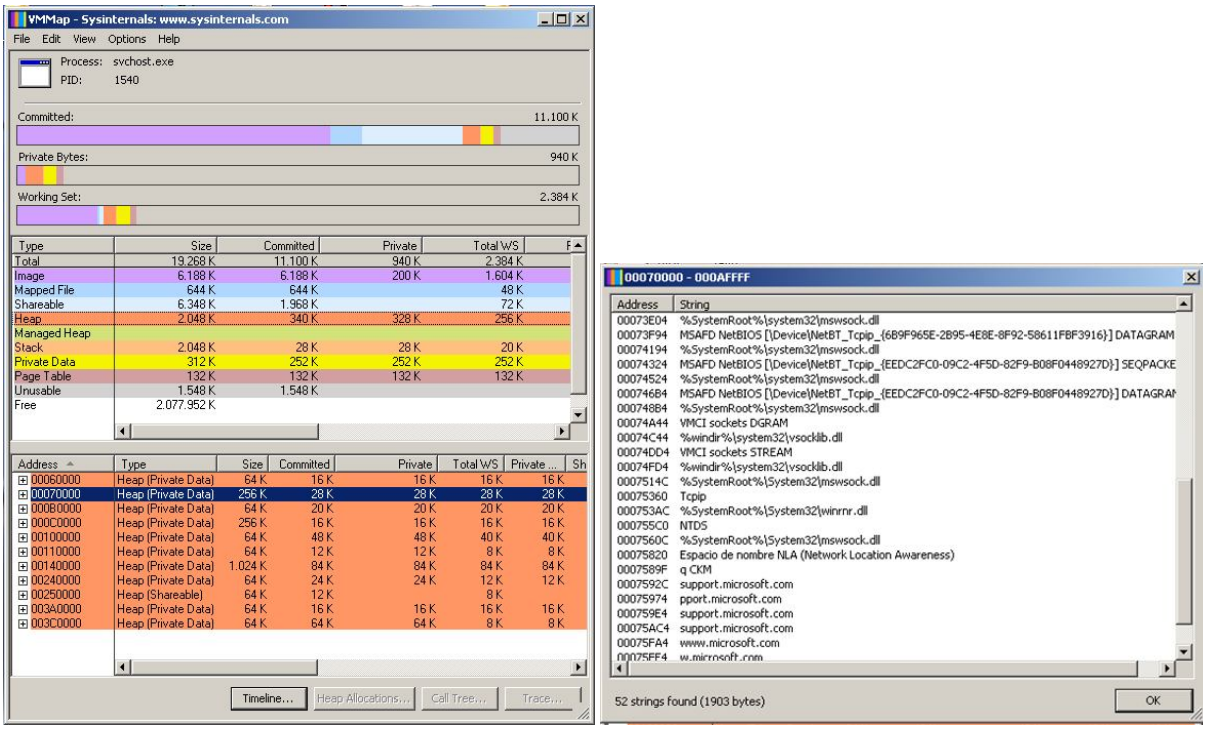
Anexo R Análisis Comportamiento – *Strings* "VMMap"

- ***Strings* sospechosos en proceso "svchost.exe" infectado, área "Private Data"**

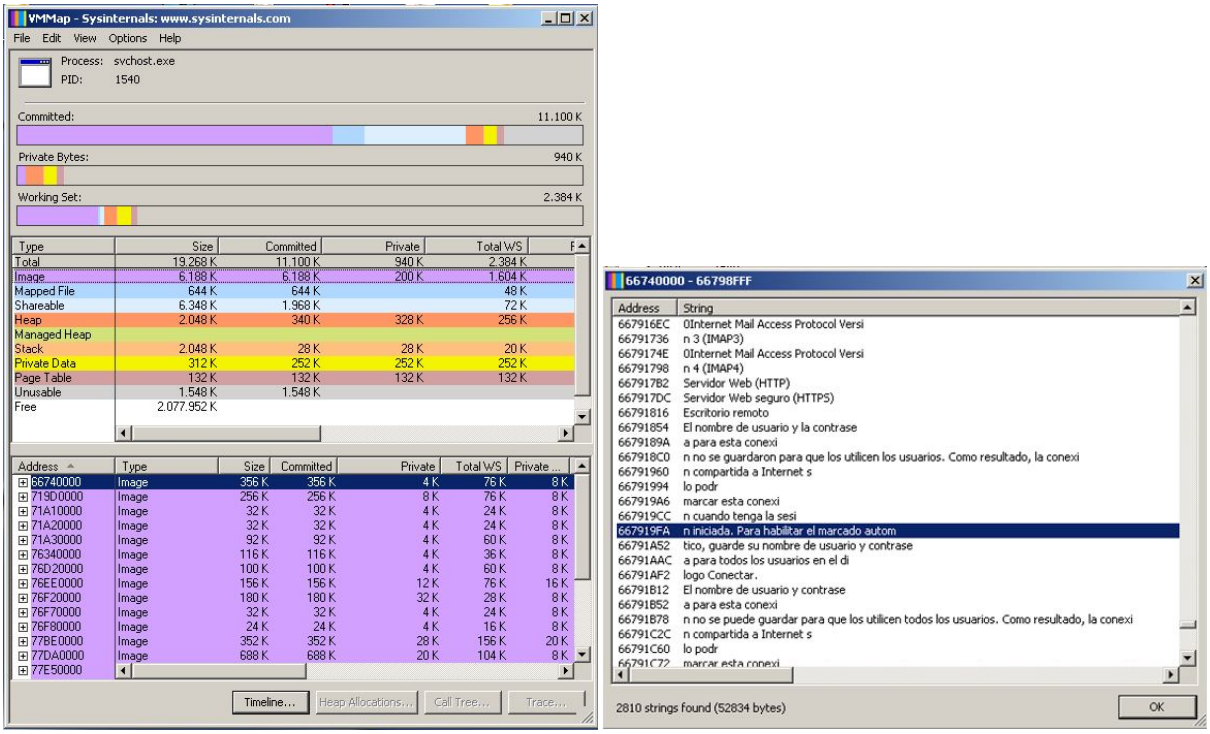




- ***Strings* sospechosos en proceso "svchost.exe" infectado, área "Heap (Private Data)"**

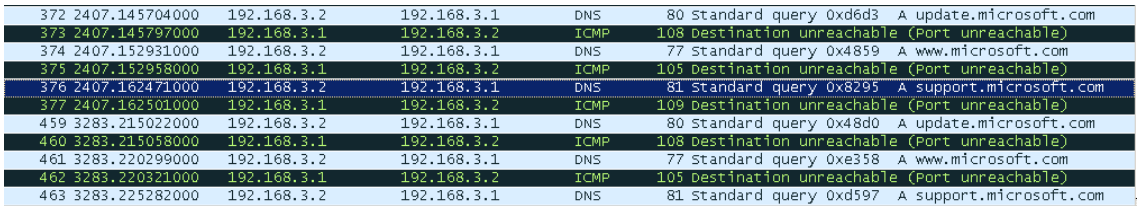


- ***Strings* sospechosos en proceso "svchost.exe" infectado, área "image"**



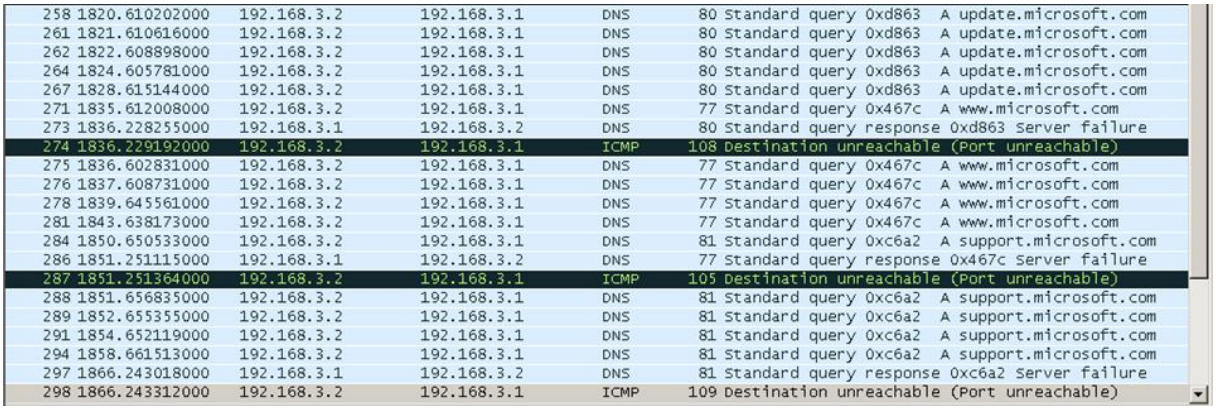
Anexo S Análisis Comportamiento – Comunicaciones "WireShark"

- **Secuencia de comunicación con servicio DNS desactivado**



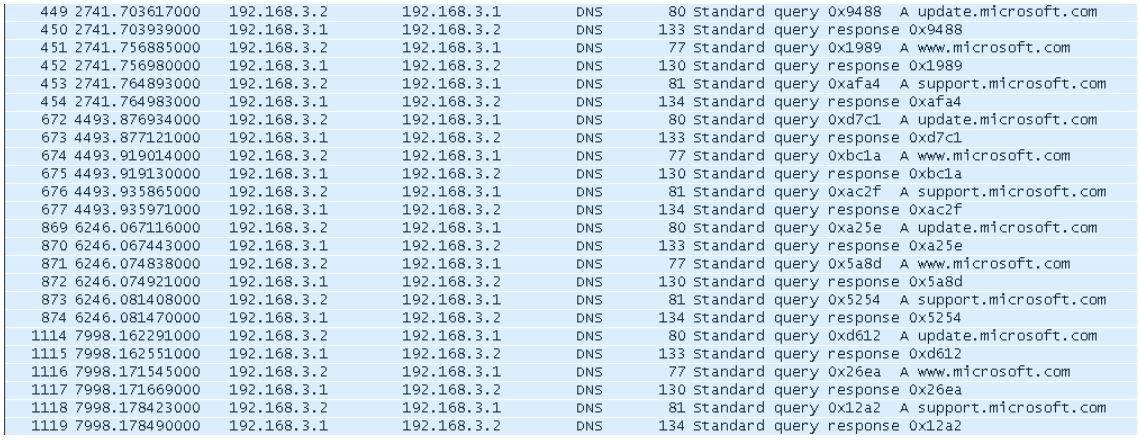
372	2407.145704000	192.168.3.2	192.168.3.1	DNS	80 Standard query 0xd6d3 A update.microsoft.com
373	2407.145797000	192.168.3.1	192.168.3.2	ICMP	108 Destination unreachable (Port unreachable)
374	2407.152931000	192.168.3.2	192.168.3.1	DNS	77 Standard query 0x4859 A www.microsoft.com
375	2407.152958000	192.168.3.1	192.168.3.2	ICMP	105 Destination unreachable (Port unreachable)
376	2407.162471000	192.168.3.2	192.168.3.1	DNS	81 Standard query 0x8295 A support.microsoft.com
377	2407.162501000	192.168.3.1	192.168.3.2	ICMP	109 Destination unreachable (Port unreachable)
459	3283.215022000	192.168.3.2	192.168.3.1	DNS	80 Standard query 0x48d0 A update.microsoft.com
460	3283.215058000	192.168.3.1	192.168.3.2	ICMP	108 Destination unreachable (Port unreachable)
461	3283.220299000	192.168.3.2	192.168.3.1	DNS	77 Standard query 0xe358 A www.microsoft.com
462	3283.220321000	192.168.3.1	192.168.3.2	ICMP	105 Destination unreachable (Port unreachable)
463	3283.225282000	192.168.3.2	192.168.3.1	DNS	81 Standard query 0xd597 A support.microsoft.com

- **Secuencia de comunicación con servicio DNS activado y servidor secundario**



258	1820.610202000	192.168.3.2	192.168.3.1	DNS	80 Standard query 0xd863 A update.microsoft.com
261	1821.610616000	192.168.3.2	192.168.3.1	DNS	80 Standard query 0xd863 A update.microsoft.com
262	1822.608898000	192.168.3.2	192.168.3.1	DNS	80 Standard query 0xd863 A update.microsoft.com
264	1824.605781000	192.168.3.2	192.168.3.1	DNS	80 Standard query 0xd863 A update.microsoft.com
267	1828.615144000	192.168.3.2	192.168.3.1	DNS	80 Standard query 0xd863 A update.microsoft.com
271	1835.612008000	192.168.3.2	192.168.3.1	DNS	77 Standard query 0x467c A www.microsoft.com
273	1836.228255000	192.168.3.1	192.168.3.2	DNS	80 Standard query response 0xd863 Server failure
274	1836.229192000	192.168.3.2	192.168.3.1	ICMP	108 Destination unreachable (Port unreachable)
275	1836.602831000	192.168.3.2	192.168.3.1	DNS	77 Standard query 0x467c A www.microsoft.com
276	1837.608731000	192.168.3.2	192.168.3.1	DNS	77 Standard query 0x467c A www.microsoft.com
278	1839.645561000	192.168.3.2	192.168.3.1	DNS	77 Standard query 0x467c A www.microsoft.com
281	1843.638173000	192.168.3.2	192.168.3.1	DNS	77 Standard query 0x467c A www.microsoft.com
284	1850.650533000	192.168.3.2	192.168.3.1	DNS	81 Standard query 0xc6a2 A support.microsoft.com
286	1851.251115000	192.168.3.1	192.168.3.2	DNS	77 Standard query response 0x467c Server failure
287	1851.251364000	192.168.3.2	192.168.3.1	ICMP	105 Destination unreachable (Port unreachable)
288	1851.656835000	192.168.3.2	192.168.3.1	DNS	81 Standard query 0xc6a2 A support.microsoft.com
289	1852.655355000	192.168.3.2	192.168.3.1	DNS	81 Standard query 0xc6a2 A support.microsoft.com
291	1854.652119000	192.168.3.2	192.168.3.1	DNS	81 Standard query 0xc6a2 A support.microsoft.com
294	1858.661513000	192.168.3.2	192.168.3.1	DNS	81 Standard query 0xc6a2 A support.microsoft.com
297	1866.243018000	192.168.3.1	192.168.3.2	DNS	81 Standard query response 0xc6a2 Server failure
298	1866.243312000	192.168.3.2	192.168.3.1	ICMP	109 Destination unreachable (Port unreachable)

- **Secuencia de comunicación con servicio DNS activado y servidor autorizado**



449	2741.703617000	192.168.3.2	192.168.3.1	DNS	80 Standard query 0x9488 A update.microsoft.com
450	2741.703939000	192.168.3.1	192.168.3.2	DNS	133 Standard query response 0x9488
451	2741.756885000	192.168.3.2	192.168.3.1	DNS	77 Standard query 0x1989 A www.microsoft.com
452	2741.756980000	192.168.3.1	192.168.3.2	DNS	130 Standard query response 0x1989
453	2741.764893000	192.168.3.2	192.168.3.1	DNS	81 Standard query 0xafa4 A support.microsoft.com
454	2741.764983000	192.168.3.1	192.168.3.2	DNS	134 Standard query response 0xafa4
672	4493.876934000	192.168.3.2	192.168.3.1	DNS	80 Standard query 0xd7c1 A update.microsoft.com
673	4493.877121000	192.168.3.1	192.168.3.2	DNS	133 Standard query response 0xd7c1
674	4493.919014000	192.168.3.2	192.168.3.1	DNS	77 Standard query 0xbc1a A www.microsoft.com
675	4493.919130000	192.168.3.1	192.168.3.2	DNS	130 Standard query response 0xbc1a
676	4493.935865000	192.168.3.2	192.168.3.1	DNS	81 Standard query 0xac2f A support.microsoft.com
677	4493.935971000	192.168.3.1	192.168.3.2	DNS	134 Standard query response 0xac2f
869	6246.067116000	192.168.3.2	192.168.3.1	DNS	80 Standard query 0xa25e A update.microsoft.com
870	6246.067443000	192.168.3.1	192.168.3.2	DNS	133 Standard query response 0xa25e
871	6246.074838000	192.168.3.2	192.168.3.1	DNS	77 Standard query 0x5a8d A www.microsoft.com
872	6246.074921000	192.168.3.1	192.168.3.2	DNS	130 Standard query response 0x5a8d
873	6246.081408000	192.168.3.2	192.168.3.1	DNS	81 Standard query 0x5254 A support.microsoft.com
874	6246.081470000	192.168.3.1	192.168.3.2	DNS	134 Standard query response 0x5254
1114	7998.162291000	192.168.3.2	192.168.3.1	DNS	80 Standard query 0xd612 A update.microsoft.com
1115	7998.162551000	192.168.3.1	192.168.3.2	DNS	133 Standard query response 0xd612
1116	7998.171545000	192.168.3.2	192.168.3.1	DNS	77 Standard query 0x26ea A www.microsoft.com
1117	7998.171669000	192.168.3.1	192.168.3.2	DNS	130 Standard query response 0x26ea
1118	7998.178423000	192.168.3.2	192.168.3.1	DNS	81 Standard query 0x12a2 A support.microsoft.com
1119	7998.178490000	192.168.3.1	192.168.3.2	DNS	134 Standard query response 0x12a2

- **Contenido de paquetes capturados en secuencia correcta**

No. Time Source Destination Protocol Length Info
449 2741.703617000 192.168.3.2 192.168.3.1 DNS 80 Standard query 0x9488 A update.microsoft.com

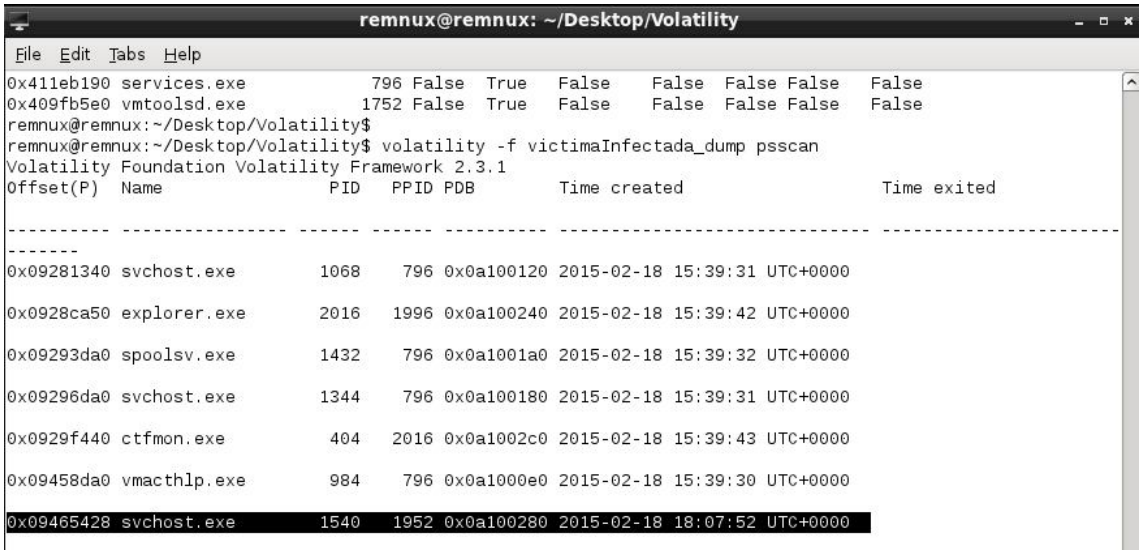
Frame 449: 80 bytes on wire (640 bits), 80 bytes captured (640 bits) on interface 0
Ethernet II, Src: Vmware_db:d1:06 (00:0c:29:db:d1:06), Dst: Vmware_80:19:48 (00:0c:29:80:19:48)
  Destination: Vmware_80:19:48 (00:0c:29:80:19:48)
    Address: Vmware_80:19:48 (00:0c:29:80:19:48)
    .... ..0. .... = LG bit: Globally unique address (factory default)
    ....0 .... = IG bit: Individual address (unicast)
  Source: Vmware_db:d1:06 (00:0c:29:db:d1:06)
    Address: Vmware_db:d1:06 (00:0c:29:db:d1:06)
    .... ..0. .... = LG bit: Globally unique address (factory default)
    ....0 .... = IG bit: Individual address (unicast)
  Type: IP (0x0800)
Internet Protocol Version 4, Src: 192.168.3.2 (192.168.3.2), Dst: 192.168.3.1 (192.168.3.1)
  Version: 4
  Header Length: 20 bytes
  Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00: Not-ECT (Not ECN-Capable Transport))
  Total Length: 66
  Identification: 0xcd6f (52591)
  Flags: 0x00
  Fragment offset: 0
  Time to live: 128
  Protocol: UDP (17)
  Header checksum: 0xe5e7 [validation disabled]
    [Good: False]
    [Bad: False]
  Source: 192.168.3.2 (192.168.3.2)
  Destination: 192.168.3.1 (192.168.3.1)
  [Source GeoIP: Unknown]
  [Destination GeoIP: Unknown]
User Datagram Protocol, Src Port: 50962 (50962), Dst Port: 53 (53)
  Source Port: 50962 (50962)
  Destination Port: 53 (53)
  Length: 46
  Checksum: 0xe9ea [validation disabled]
    [Good Checksum: False]
    [Bad Checksum: False]
  [Stream index: 27]
Domain Name System (query)
  [Response In: 450]
  Transaction ID: 0x9488
  Flags: 0x0100 Standard query
    0... .... = Response: Message is a query
    .000 0... .... = Opcode: Standard query (0)
    .... ..0. .... = Truncated: Message is not truncated
    ....1 .... = Recursion desired: Do query recursively
    .... .0.. .... = Z: reserved (0)
    ....0 .... = Non-authenticated data: Unacceptable
  Questions: 1
  Answer RRs: 0
  Authority RRs: 0
  Additional RRs: 0
  Queries
    update.microsoft.com: type A, class IN
      Name: update.microsoft.com

```
        [Name Length: 20]
        [Label Count: 3]
        Type: A (Host Address) (1)
        Class: IN (0x0001)

No.     Time        Source                Destination           Protocol Length Info
    450 2741.703939000 192.168.3.1        192.168.3.2           DNS      133    Standard query response
0x9488

Frame 450: 133 bytes on wire (1064 bits), 133 bytes captured (1064 bits) on interface 0
Ethernet II, Src: Vmware_80:19:48 (00:0c:29:80:19:48), Dst: Vmware_db:d1:06 (00:0c:29:db:d1:06)
    Destination: Vmware_db:d1:06 (00:0c:29:db:d1:06)
        Address: Vmware_db:d1:06 (00:0c:29:db:d1:06)
        .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
        .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
    Source: Vmware_80:19:48 (00:0c:29:80:19:48)
        Address: Vmware_80:19:48 (00:0c:29:80:19:48)
        .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
        .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
    Type: IP (0x0800)
Internet Protocol Version 4, Src: 192.168.3.1 (192.168.3.1), Dst: 192.168.3.2 (192.168.3.2)
    Version: 4
    Header Length: 20 bytes
    Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00: Not-ECT (Not ECN-Capable
Transport))
    Total Length: 119
    Identification: 0x0280 (640)
    Flags: 0x00
    Fragment offset: 0
    Time to live: 128
    Protocol: UDP (17)
    Header checksum: 0xb0a2 [validation disabled]
        [Good: False]
        [Bad: False]
    Source: 192.168.3.1 (192.168.3.1)
    Destination: 192.168.3.2 (192.168.3.2)
    [Source GeoIP: Unknown]
    [Destination GeoIP: Unknown]
User Datagram Protocol, Src Port: 53 (53), Dst Port: 50962 (50962)
    Source Port: 53 (53)
    Destination Port: 50962 (50962)
    Length: 99
    Checksum: 0x32ce [validation disabled]
        [Good Checksum: False]
        [Bad Checksum: False]
    [Stream index: 27]
Domain Name System (response)
    [Request In: 449]
    [Time: 0.000322000 seconds]
    Transaction ID: 0x9488
    Flags: 0x8580 Standard query response, No error
        1... .... .... .... = Response: Message is a response
        .000 0... .... .... = Opcode: Standard query (0)
        .... .1.. .... .... = Authoritative: Server is an authority for domain
        .... ..0. .... .... = Truncated: Message is not truncated
        .... ...1 .... .... = Recursion desired: Do query recursively
        .... .... 1... .... = Recursion available: Server can do recursive queries
        .... .... .0.. .... = Z: reserved (0)
        .... .... ..0. .... = Answer authenticated: Answer/authority portion was not authenticated by the
server
```

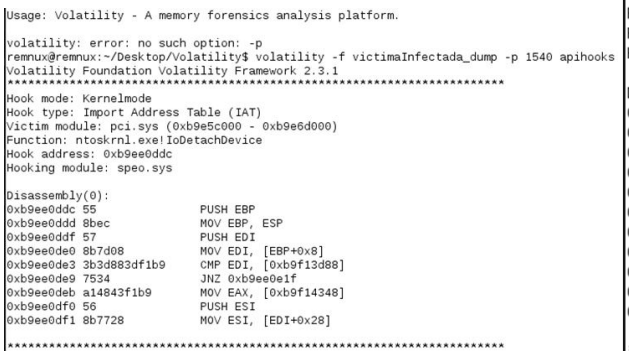
```
    .... .... ...0 .... = Non-authenticated data: Unacceptable
    .... .... .... 0000 = Reply code: No error (0)
  Questions: 1
  Answer RRs: 0
  Authority RRs: 1
  Additional RRs: 0
  Queries
    update.microsoft.com: type A, class IN
      Name: update.microsoft.com
      [Name Length: 20]
      [Label Count: 3]
      Type: A (Host Address) (1)
      Class: IN (0x0001)
  Authoritative nameservers
    update.microsoft.com: type SOA, class IN, mname win2003
      Name: update.microsoft.com
      Type: SOA (Start Of a zone of Authority) (6)
      Class: IN (0x0001)
      Time to live: 3600
      Data length: 41
      Primary name server: win2003
      Responsible authority's mailbox: hostmaster
      Serial Number: 1
      Refresh Interval: 900 (15 minutes)
      Retry Interval: 600 (10 minutes)
      Expire limit: 86400 (1 day)
      Minimum TTL: 3600 (1 hour)
```

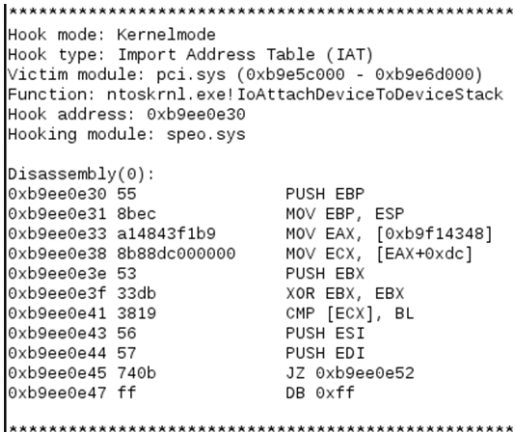
Anexo T Análisis Comportamiento – Volcado Memoria "Volatility"

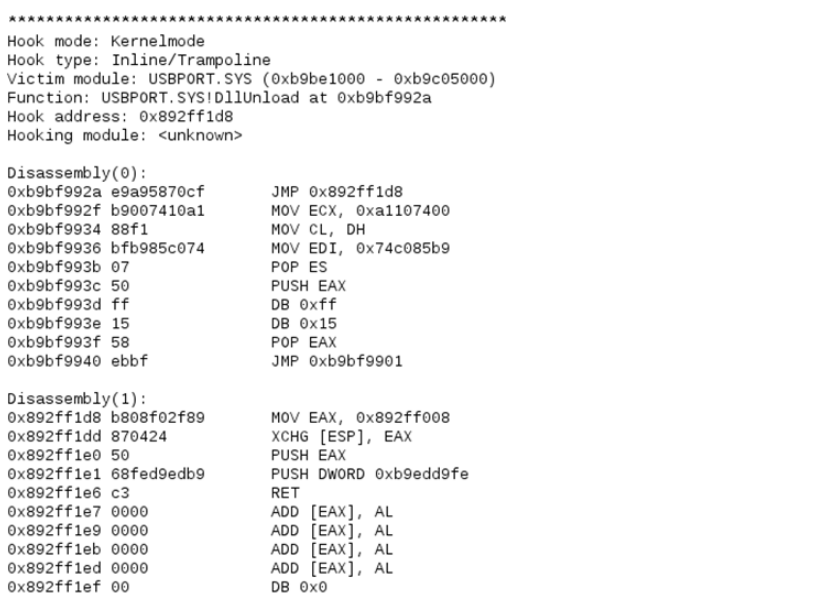
- **Identificación de proceso infectado "svchost.exe" PID 1540**



- **Identificación de *hooks* sospechosos**







- **Identificación de código inyectado sospechoso**