

UNIVERSIDAD
INTERNACIONAL
DE LA RIOJA

unir

**Universidad Internacional de La Rioja
Máster Universitario en Seguridad
Informática**

**Aplicación de Metodología
de Malware para el
Análisis de la amenaza
avanzada persistente
(APT) “Poison Ivy”**

Trabajo de Fin de Máster

Presentado por: Gaviria, Pablo

Director/a: Bermejo, Javier

Ciudad: San Juan de Pasto

Fecha: 22/09/2016

Nota de Aceptación

Firma del Presidente del Jurado

Firma del Jurado

Firma del Jurado

Bogotá D.C., 22 de Septiembre de 2016

A nuestro Creador y arquitecto perfecto por cada uno de los Dones recibidos y la oportunidad de materializarlos en las metas alcanzadas.

A mi madre, quien con prudencia y buenos deseos sigue siendo fiel seguidora y cómplice de mis planes y proyectos.

Con Cariño y Aprecio: Luz María Gaviria.

Agradecimientos

A la Fiscalía General de la Nación de Colombia como apoyo absoluto de mi formación profesional y académica al permitirme ser parte activa en el cumplimiento de su Misión Constitucional, desarrollando y aplicando el conocimiento adquirido durante estos años.

A mi Cuerpo Técnico de Investigación CTI Subdirección Seccional de Policía Judicial Nariño, sus directivos y compañeros, quienes día a día con el trasegar en el campo de la investigación, han sido un estímulo como elemento clave a la hora de consolidar y llevar a feliz término este documento.

A Don Javier Bermejo por darme la oportunidad de conocer su trabajo de investigación, la cual como fuente indiscutible del saber me ha permitido descubrir este fascinante y complejo tema del análisis del malware, que sin duda alguna ha enriquecido mi formación profesional.

Resumen

El avance tecnológico ha permitido crear nuevos escenarios de ataque en los cuales convergen una serie de elementos redundando en sofisticación de las amenazas. Esto requiere implementar nuevos mecanismos articulados sobre una nueva disciplina llamada Ciberdefensa, que permiten reaccionar de manera objetiva ante estos ataques. El presente documento tiene como propósito desarrollar un piloto experimental, con el fin de establecer la validez en la aplicación de la metodología de análisis de malware presentada por Don Javier Bermejo, en el contexto de la Amenaza Persistente Avanzada Poison Ivy. Para ello, fue necesario desarrollar la temática relacionada con los diferentes tipos de malware y las técnicas actuales de análisis, conocer e identificar la amenaza persistente avanzada (APT) “Poison Ivy”, en relación al desarrollo metódico de cada una de las fases que componen la metodología, la aplicación de cada una de las herramientas sugeridas sobre una muestra de malware obtenida en un ambiente controlado el cual se asemejó a un escenario real. Esto permitió demostrar la importancia y funcionalidad de la metodología en particular respecto al análisis de malware como una herramienta efectiva y eficaz, alcanzando una serie de resultados obtenidos, debidamente organizados y documentados. El presente piloto experimental se justificó por cuanto posee valor teórico y una utilidad práctica, basados en el campo del conocimiento mediante la aplicación de la información y documentación existente relacionada con la metodología propuesta enfocada al análisis de malware en particular del tipo APT.

Palabras clave: Metodología, amenaza persistente avanzada (APT), análisis de malware, Poison Ivy.

Abstract

Technological progress has created new attack scenarios where a group of elements converge and there is more sophistication of threats. This situation requires implementing new mechanisms articulated about a new discipline called Cyber Defense, which allow react to these attacks. This document permits to develop an experimental pilot, in order to establish validity in application of the methodology for malware analysis by Don Javier Bermejo, in the context of the Advanced Persistent Threat, Poison Ivy. In this case, it was necessary to develop subjects relate to different types of malware and current analysis techniques, know and identify advanced persistent threat (APT) "Poison Ivy", concerning to the systematic development of each of phases that constitute the methodology, implementation of each of tools suggested on a sample of malware obtained in a controlled environment was similar to a real scenario. This survey allowed demonstrating importance and functioning of the methodology about the malware analysis as an effective and efficient tool, reaching a series of organized and documented results. This experimental pilot is justified because it has theoretical and practical value, based on scope of knowledge through utilization of existing information and documentation relate to the methodology proposed which is focused on the malware analysis type APT.

Keywords: Methodology, advanced persistent threat (APT), malware analysis, Poison Ivy.

Índice de contenidos

1. Introducción	ix
2. Objetivo General.....	xiii
2.1. Objetivos Específicos	xiii
3. Contexto Y Estado Del Arte	xiv
3.1. Sistemas de detección de intrusos	xv
3.2. Definición de Malware	xvi
3.3. Tipos de Malware	xvii
4. Advanced Persistent Thread APT	xxiv
4.1. Principales características de las APT	xxix
4.2. Fases de ataque en un APT	xxxI
4.3. Indicadores de Ataque de un APT	xxxv
4.4. Defensas ante las APTs	xxxvi
4.5. Advanced Persistent Threat (APT) POISON IVY	xxxviii
4.5.1. Antecedentes	xxxix
4.5.2. Estructura de ataque de Poison IVY.....	xli
4.5.3. Protocolo de Comando y Control (C2) Poison IVY	xliii
5. Metodología Aplicable al Piloto Experimental	xlvi
5.1. Introducción a la metodología	xlviii
5.2. Descripción de la metodología.....	I
5.2.1. Acciones Iniciales	I
5.2.2. Clasificación	li
5.2.3. Análisis de Código.....	liii
5.2.4. Análisis Dinámico o de Comportamiento	liii
6. Escenario de Aplicación.....	lvi
6.1. Definición del Entorno o Escenario	lvi
6.2. Definición de herramientas específicas del laboratorio	lx
7. Aplicación de la metodología propuesta al malware Poison IVY.....	v
7.1. Fase 1: Acciones Iniciales:	v
7.2. Fase 2: Clasificación.....	vii
7.2.1. Transferencia de Malware	vii

7.2.2.	Identificación del malware.....	viii
7.2.3.	Comprobación del tipo del malware.....	xi
7.2.4.	Información obtenida por fuentes abiertas	xv
7.2.5.	Búsqueda de cadenas de texto	xix
7.2.6.	Identificación de técnicas de ofuscación	xxi
7.2.7.	Formato y estructura del fichero	xxvii
7.3.	Fase 3: Análisis de Código	xxxii
7.3.1.	Análisis dinámico del código.....	xxxii
7.3.2.	Análisis estático del código.....	xxxiv
7.4.	Fase 4: Análisis de comportamiento	xxxviii
7.4.1.	Ejecución del malware.....	xxxviii
7.4.2.	Análisis del volcado de Memoria.....	xlvi
7.5.	Resumen del análisis desarrollado.....	lv
ANEXO A: CLASIFICACION – RESULTADOS VIRUS TOTAL.....		lxiii
ANEXO B: CLASIFICACION – RESULTADOS VIRUS TOTAL SOBRE “Hack Facebook.exe”		lxvi
ANEXO C: RESULTADOS Bintex SOBRE “Hack Facebook.exe”		lxix
ANEXO D: RESULTADOS – STRINGS SOBRE “Hack Facebook.exe”		lxxii
ANEXO E: RESULTADOS VOLATILITY		lxxv
Referencias y Bibliografía		C
Webgrafía		CV

Índice de figuras

Figura 1. Países y su Índice de infección. (Panda Security, 2015. p. 9).....	xx
Figura 2 Infecciones por tipo de malware (2015). (Panda Security, 2015. p. 8).....	xx
Figura 3. Preocupaciones en materia de seguridad en las empresas de Latinoamérica (Eset, 2015, p. 4)	xxi
Figura 4. Incidentes de Seguridad de la Información en las empresas de Latinoamérica (ESET, 2016, p. 8)	xxii
Figura 5. Estructura de una campaña Buhtrap. (Eset, 2016, p. 21)	xxvi
Figura 6. 20 Critical Security Controls. (blog.segu-info.com.ar, 2012)	xxxvii
Figura 7. Puertos TCP variantes de PIVY en ataques mediante APT. (FireEye, 2013 p. 14).....	xlii
Figura 8. Mutex de proceso variante de PIVY asociadas a ataques mediante APT. (FireEye, 2013 p. 14)	xlii
Figura 9. Protocolo de comunicación inicial de PIVY. (FireEye, 2013 p. 8)	xliv
Figura 10. “Sistema de Análisis e Ingeniería Inversa de Código Malicioso” (Bermejo, 2015).....	xlix
Figura 11. Arquitectura laboratorio de Análisis malware ([1] p. 162).	lvii
Figura 12. Escenario y Subsistemas del laboratorio.....	lviii
Figura 13. Screenshot sobre cada Subsistema.	lix
Figura 14. Herramientas análisis de malware relacionadas con la máquinas del laboratorio. ([2], 2015, p. 169)	lx
Figura 15. Definición de Herramientas para Laboratorio.	iii
Figura 16. Aplicación de SysTracer y generación de snapshot sobre el sistema víctima.....	vi
Figura 17. Aplicación de MD5summ, sobre snapshot del sistema víctima.....	vi
Figura 18. Sitio web MediaFire - fuente de descarga "Poison Ivy".	vii
Figura 19. Transferencia del malware a un dispositivo USB.	viii
Figura 20. Resultado MD5sums sobre la muestra del malware obtenida.	viii
Figura 21. Ejecución Poison Ivy, pantalla principal.....	ix
Figura 22. Identificación muestra obtenida del malware archivo “Poison Ivy 2.3.2.exe”, “Virus Total”.	xi
Figura 23. Identificación muestra obtenida del malware archivo “Hack Facebook”, “Virus Total”.....	xii
Figura 24. Resultado de Avira, sobre la muestra “Poison Ivy 2.3.2.exe”	xiv
Figura 25. Resultado Avira sobre “Hack Facebook.exe”.	xiv
Figura 26. Poison Ivy RAT release timeline, ([23], página 11).....	xv
Figura 27. SHA1 Hashes for Historic Poison Ivy Builders, ([23], página 8).	xvii
Figura 28. Escenario de Poison Ivy, ([23],, página 5)	xviii

Figura 29. Resultado análisis de Bintex, sobre el malware generado archivo "Hack Facebook.exe".	xx
Figura 30. Resultado análisis de Strings, sobre el malware generado archivo "Hack Facebook.exe".	xxi
Figura 31. Resultado análisis PEiD, sobre el malware generado archivo "Hack Facebook.exe".	xxii
Figura 32. PEiD - "Task Viewer", compilador y entropía de "Hack Facebook.exe".	xxiii
Figura 33. PEiD - Secciones de "Hack Facebook.exe".	xxiii
Figura 34. PEiD Task viewer "Hack Facebook.exe".	xxiv
Figura 35. PEiD - Kripto ANALizer "Hack Facebook.exe".	xxiv
Figura 36. Detect It Easy - Compiler "Hack Facebook.exe".	xxvi
Figura 37. Detect It Easy - Entropy "Hack Facebook.exe".	xxvi
Figura 38. PE Explorer y sus resultados sobre "Hack Facebook.exe".	xxvii
Figura 39. Dependency Walker y sus resultados sobre "Hack Facebook.exe".	xxviii
Figura 40. PEBrowse y sus resultados sobre "Hack Facebook.exe".	xxix
Figura 41. PEStudio y sus resultados sobre "Hack Facebook.exe".	xxx
Figura 42. . PEStudio BlackListed DLLs sobre "Hack Facebook.exe".	xxxi
Figura 43. OllyDbg - En ejecución.	xxxii
Figura 44. OllyDbg - Llamada a Hack_Face en el proceso explorer.exe	xxxiii
Figura 45. OllyDbg. Datos y funciones.	xxxiii
Figura 46. Referencia a la librería ntdll.dll.	xxxiv
Figura 47. IDA PRO en ejecución sobre "Hack Facebook.exe".	xxxv
Figura 48. IDA PRO Cadenas de caracteres sobre "Hack Facebook.exe".	xxxvi
Figura 49. Estructura del archivo "Hack Facebook.exe".	xxxvi
Figura 50. IDA PRO Llamada de funciones sobre "Hack Facebook.exe".	xxxvi
Figura 51. IDA PRO - Fragmento subrutina 400400 – "Hack Facebook.exe".	xxxvii
Figura 52. Ejecución del malware "Hack Facebook.exe" sobre la máquina víctima.	xxxviii
Figura 53. Ejecución Poison Ivy desde la máquina de control.	xxxix
Figura 54. Poison Ivy – Listado de Puertos sobre la víctima.	xxxix
Figura 55. Systracer – Listado de Ficheros creados y modificados.	xl
Figura 56. Systracer – Listado de Ficheros creados y modificados.	xl
Figura 57. Listado de Ficheros creados y modificados.	xli
Figura 58. CaptureBAT en ejecución sobre la víctima.	xlii
Figura 59. Contenido archivo "reporte.txt".	xlii
Figura 60. Disk Pulse - Resultados.	xliv
Figura 61. Process Explorer sobre la máquina víctima.	xliv
Figura 62. Process Monitor en ejecución.	xliv

Figura 63. Autoruns sobre la máquina víctima..... xlvi
Figura 64. Volcado de memoria Winpmem..... xlvii
Figura 65. Comparativo inyección de código Poison Ivy Vs otros Malware (Hale, 2016). xlix

Índice de tablas

Tabla 1. Estructura de PI_chunk_header.....	xliv
Tabla 2. Poison Ivy - configuración.	x
Tabla 3. Resultado herramientas de detección según “Virus Total” sobre la muestra.	xii
Tabla 4. Resultado herramientas de detección según “Virus Total” - “Hack Facebook.exe”	xiii
Tabla 5. Extracto resultado del análisis de Bintex sobre el archivo del malware generado.	xxi
Tabla 6. Archivos creados por el malware sobre máquina de la víctima.....	xli
Tabla 7. Fragmento archivo "reporte.txt".	xlili
Tabla 8. Fragmento archivo "reporte.txt" (continuación).....	xlili
Tabla 9. Resultado imageinfo – volatility sobre el volcado de memoria.	xlvii
Tabla 10. Fragmento resultado pslist – volatility sobre el volcado de memoria.	xlviii
Tabla 11. Fragmento resultado pstree - volatility sobre volcado de memoria.	xliv
Tabla 12. Fragmento resultado maldfind - volatility sobre volcado de memoria.....	lii
Tabla 13. Resultado strings sobre process.0x84fe2478.0x60000.dmp.....	liv

1. Introducción

Tras el continuo crecimiento de las tecnologías de la información, nuestro presente y futuro han sido enmarcados por el clickeo del mouse y el brillo de nuestras pantallas refleja una dependencia absoluta sobre nuevos paradigmas que han cambiado de manera significativa la forma de ver el mundo. Mientras muchos trabajan y construyen, otros simplemente se distraen y se relajan mientras sueñan con infinidad de páginas y servicios que ahí se ofrecen y sin importar las clases sociales, filiaciones políticas, rasas o culturas, hoy en día acompañan entre sus objetos personales toda clase de dispositivos, que con sus sonidos particulares anuncian su entrada y permanencia en este nuevo mundo paralelo que ha generado la Internet.

Es el despertar a una nueva era marcada radicalmente por la comunicación digital, que nos hace vulnerables y nos pone muchas veces en desventaja ante usuarios y personas que tienen un nivel de conocimiento superior que un usuario promedio frente al manejo de los dispositivos, y aún más cuando sus intenciones van mucho más allá que una simple sonrisa al otro lado de las pantallas.

En la actualidad existen muchos factores que contribuyen con la inmensa actividad criminal en la red o “Cibercriminalidad” y aún más, cuando la amplia gama y disponibilidad de herramientas de ataque, manuales y videos gratuitos con instrucciones precisas ofrecidas a través de internet, han hecho que los usuarios más principiantes se conviertan en expertos en cuestión de horas. Este es un reflejo de una realidad en donde es razonable pensar que la era del hacker ha quedado atrás, Gobiernos preparándose cada día para afrontar todo tipo de amenazas cibernéticas, acuñando términos como la ciberguerra, personas extrayendo toda información personal, apoderándose de cuentas, perfiles, contraseñas, poniendo en jaque la tranquilidad y los sistemas socialmente tradicionales, hacen que la inseguridad sea latente.

La necesidad de estar cada vez más conectados ha traído consigo todo tipo de soluciones al mercado, con el fin de administrar de manera remota los sistemas de información, herramientas que mal utilizadas o con fines delictivos, pueden de manera irrestricta dar acceso a los sistemas poniendo en riesgo la seguridad.

Dentro de esa infinidad y clasificación de herramientas están las conocidas como Remote Access tools (RAT), las cuales presentan un escenario que requiere conocimientos técnicos mínimos para su uso, generando desde una simpleza engañosa, la base de ataques coordinados o denominados Advanced Persistent Threads (APT), una categoría mucho más avanzada y como uno de los desafíos más importantes y trascendentes dentro del campo de la seguridad informática, y aún más, cuando estos elementos se combinan con otros métodos, como vulnerabilidades de software de día cero hasta sofisticados métodos de ingeniería social.

Por otro lado, el uso del Malware como medio de ataque es y seguirá siendo una de las técnicas más empleadas en la actualidad, en la cual la ciberdefensa como una nueva disciplina en el campo de la ciberseguridad debe centrar sus esfuerzos. Desde esta perspectiva el conocimiento y análisis del malware permite comprender su comportamiento, su actividad sobre un sistema comprometido y su ciclo de vida, y aún más cuando la diversidad de código malicioso existente y herramientas diseñadas para tal fin son tan amplias.

En su evolución se ha roto el paradigma contemporáneo de creer que para su control basta solo con la aplicación de AV, y se han diseñado técnicas y estrategias utilizadas para camuflarse y saltar estos controles, incluidos dispositivos empleados para su detección como firewalls, IDS e IPS¹, los cuales fueron diseñados para descubrir este tipo de aplicaciones a través de firmas y reglas que comparten algunos códigos maliciosos cuando atacan un sistema. Este es el caso de las amenazas avanzadas persistentes, las cuales requieren un análisis más profundo y la aplicación de herramientas especializadas que permitan aplicar medidas y técnicas de defensa estableciendo rastros específicos, origen, vectores de ataques y descubrir las posibles capacidades del atacante sobre un sistema cuando se hayan saltado las barreras tradicionales de protección.

¹ <http://searchsecurity.techtarget.com/Do-you-need-an-IDS-or-IPS-or-both>

Toda esta actualización requiere una necesidad marcada no solo de crear nuevas herramientas y metodológicas si no tras su aplicación, contribuir con los nuevos mecanismos frente al fortalecimiento de la seguridad informática o “Ciberdefensa”.

Estas son las razones de motivación para el desarrollo del presente piloto experimental, que tiene por objeto particularmente la aplicación de la metodología desarrollada y propuesta por Don Javier Bermejo referente al Análisis e ingeniería inversa de código malicioso en su Tesis Doctoral (Bermejo, 2015).

Desde una perspectiva general, esta metodología se centra en la aplicación de cuatro fases o procedimientos, para lo cual refiere: la creación de un escenario de prueba, la clasificación respecto a la información de la APT a analizar, un análisis estático y dinámico del código y finalmente el análisis del comportamiento del código sobre un ambiente real.

Como propósito, es interesante desde el plano de la experimentación, aplicar este piloto experimental sobre la amenaza persistente avanzada (APT) conocida como lo es Poison IVY, la cual como herramienta de acceso remoto gratuita, sigue ofreciendo una serie de funcionalidades que la hacen apetecible por los atacantes, debido a su relativa facilidad de configuración y anonimato que desde su aparición la hacen aún vigente.

Su alcance desde un plano académico con los resultados obtenidos, puede contribuir y ampliar desde el campo del conocimiento mediante la aplicación de un caso práctico, la información y documentación relacionada con la aplicación de la metodología propuesta enfocada al análisis de malware, en particular del tipo APT.

De esta forma el presente trabajo se relaciona mediante la siguiente estructura:

En desarrollo a un **Capítulo 1**, se presenta el estado del arte a fin de contextualizar desde su descripción, referencias a definiciones, conceptos relativos al malware, su clasificación y los escenarios de ataque.

Seguido de un **Capítulo 2**, en el cual se desarrolla conceptos respecto a los APT, que permiten conocer su estructura y comprender la importancia en el tema de la seguridad

informática, para luego continuar y presentar en relación del APT “Poison IVY” como base de aplicación del piloto experimental, la información y datos generales respecto a su aparición, estructura, alcances y escenarios de infección, que permitan conocer este tipo de amenaza.

Un **Capítulo 3**, en el cual se presenta de forma clara y detallada la metodología aplicable en el piloto experimental con el fin de conocer su estructura y alcance.

Se presenta un **Capítulo 4** con el fin de definir y desarrollar un escenario controlado de aplicación en pruebas relacionadas con un laboratorio práctico frente al uso de la herramienta.

Un **Capítulo 5**, con el fin de exponer los resultados en relación a la aplicación de pruebas planificadas desde el campo de aplicación práctico, desarrollando fases y pasos documentados.

Y finalmente un **Capítulo 6**, mediante el cual se presentan las conclusiones al desarrollo del piloto experimental referente a los resultados obtenidos, seguido de las referencias bibliográficas y webgrafía utilizada en su desarrollo.

2. Objetivo General

Aplicar la metodología desarrollada por Don Javier Bermejo en su Tesis Doctoral (Bermejo, 2015), con el fin de evaluar las técnicas y procedimientos frente a la amenaza persistente avanzada (APT) "Poison Ivy", y en sus resultados contribuir con la documentación existente como un mecanismo de defensa y control.

2.1. Objetivos Específicos

- Presentar los diferentes tipos de malware y las técnicas actuales frente a su análisis.
- Conocer e Identificar la amenaza persistente avanzada (APT) "Poison Ivy".
- Aplicar el piloto experimental sobre un ambiente controlado, demostrando la funcionalidad de la metodología sobre la amenaza persistente avanzada (APT) "Poison Ivy".
- Demostrar la importancia de las metodologías de análisis de malware, como herramientas que permitan contrarrestar los ataques de forma efectiva, mediante evidencias documentales basadas en un escenario real.

3. Contexto Y Estado Del Arte

En los últimos años la información dentro de las empresas, ha tomado un valor que en muchos escenarios puede ser incalculable y como activo necesita de estrategias, políticas, controles, documentos y herramientas que la salvaguarden de amenazas que pongan en riesgo su confidencialidad, integridad y disponibilidad enmarcadas dentro de una estructura de seguridad informática. Este escenario se torna un poco más complejo cuando las actividades misionales convergen en el ciberespacio, compartiendo sus servicios y recursos con todo tipo de personas, quienes ocultos tras el anonimato han encontrado como un aliado casi perfecto para cumplir sus objetivos impulsados por todo tipo de motivaciones sociales y económicos, prácticas que se están especializando cada vez más.

Desde el plano de la Seguridad Informática, la cual ha experimentado un profundo cambio, se han realizado todo tipo de inversiones con el fin de fortalecer elementos concretos desde la seguridad, no solo de la información sino a todos los procesos de la empresa, entre ellos: Sistemas IDS (Intrusion Detection System), Sistemas IPS (Intrusion Prevention System), Honey pot, SIEM (Security Information en Event Management), elementos que deben ir acordes con la estructura general de la empresa. En este sentido David López Analista-Consultor, refiere en su publicación que el concepto de seguridad Informática haya evolucionado hacia el concepto de Seguridad de la Información, cuyo objetivo principal consiste en alinear las inversiones en seguridad con los objetivos generales de la empresa y sus estrategias de negocio. La Seguridad de la Información se basa en el diseño de Políticas de Seguridad integrada en los planes estratégicos de la empresa. (López, 2016)

En relación a las características propias de cada herramienta utilizada para cometer los ataques existe una clasificación, y como una particularidad especial encontramos los códigos maliciosos. Sin embargo, existe aún algo de confusión al utilizar términos como “gusanos” o “virus” indistintamente, cuando en realidad podemos agrupar todos los códigos maliciosos sobre el concepto del malware.

Tradicionalmente existen en el mercado todo tipo de programas denominados “antivirus”, que han sido diseñados para detectar y eliminar los tipos de códigos maliciosos antes

mencionados, sin embargo, en la realidad estos programas NO pueden eliminar todos los virus, debido a que las técnicas de desinfección utilizadas.

3.1. Sistemas de detección de intrusos

Para Durán Lara y Jorge Christian en su Tesis: Desarrollo de un laboratorio para el análisis automatizado de códigos maliciosos, define a un sistema de detección de intrusos o IDS por sus siglas en inglés Intrusion Detection System, básicamente en un programa o aplicativo destinado a detectar accesos desautorizados sobre una maquina o sobre una red (Durán, 2010).

Estos IDS suele tener sensores virtuales (por ejemplo, un sniffer de red) con los que el núcleo del IDS puede obtener datos externos o internos (generalmente sobre el tráfico de red o archivos). El IDS detecta, gracias a dichos sensores, anomalías que pueden ser indicio de la presencia de ataques o falsas alarmas, entre ellos, cabe mencionar dos tipos de sistemas de detección de intrusos los cuales son:

- **HIDS (Host IDS).** Un Host IDS vigila una única computadora y por tanto su interfaz corre en modo no promiscuo. La ventaja es que la carga de procesador es mucho menor. Puede resultar efectivo para las siguientes tareas:
 - ✓ Analizar el tráfico sobre un único servidor o PC.
 - ✓ Detectar intentos fallidos de acceso.
 - ✓ Detectar modificaciones en el sistema de archivos, ya sean críticos o no.
- **NIDS (Network IDS).** Un Network IDS, está basado en red y detecta ataques a todo el segmento de ésta. Su interfaz debe funcionar en modo promiscuo capturando así todo el tráfico de la red o bien, recibiendo la información a través de un puerto espejo configurado en el conmutador.

El funcionamiento de estas herramientas se basa en el análisis pormenorizado del tráfico de red, el cual al entrar al analizador es comparado con firmas de ataques

conocidos, o comportamientos sospechosos, como puede ser el escaneo de puertos, paquetes malformados, etc. El NIDS no sólo analiza qué tipo de tráfico es, sino que también revisa el contenido y su comportamiento. Luego entonces, es efectivo para las siguientes labores:

- ✓ Analiza el tráfico de toda la red.
- ✓ Examina paquetes en búsqueda de opciones no permitidas y diseñadas para no ser neutralizadas por los firewalls.
- ✓ Produce alertas cuando se intenta explotar alguna vulnerabilidad sobre algún programa de un servidor.

Debemos explorar la aplicación de otro tipo de métodos de control frente a las amenazas informáticas que se están especializando cada día y aplican técnicas de vanguardia para su propagación, infección, y acción, siendo necesario utilizar otro tipo de herramientas entre ellas el Análisis del malware, campo o área que servirá de base para desarrollo del presente piloto experimental; tema enmarcado en los siguientes conceptos y definiciones que nos permitirán vislumbrar su escenario de aplicación.

3.2. Definición de Malware

La palabra malware proviene del término en inglés Malicious software, y partiendo desde su concepto, puede referirse como cualquier archivo (programa, código, documento, mensaje, imagen) creado con el objetivo de causar perjuicios sobre la información, reflejado en los usuarios de sistemas informáticos.

Según Kaspersky Lab, los códigos maliciosos son uno de los tipos de aplicación con archivos ejecutables que se puede activar por si solos y tomar distintas formas, como applets de Java, controles de ActiveX, contenido insertado, plug-ins, lenguajes de scripts u otros lenguajes de programación que están diseñados para mejorar páginas web y correo electrónico (kaspersky Lab, 2016).

Adentrándonos un poco más en los conceptos, dentro del malware se encuentran los llamados generalmente “virus”, los cuales como uno de los tipos de código malicioso más abundante se subdividen en tres subgrupos: virus, gusanos y troyanos, además de otros grupos.

3.3. Tipos de Malware

- **Los Virus.** Es un tipo de programa informático que se introduce a un ordenador de múltiples formas, en muchas ocasiones se mantienen inocuos por periodos de tiempo y cuando se activan son capaces de multiplicarse mediante la infección de otros programas mayores, teniendo efectos de molestias e incomodidad, perturbando a los usuarios, hasta destruir e irreparar todo un sistema.

A la vez existen diferentes tipos de virus, clasificados por la forma o modo en que efectúan un ataque entre ellos: los que atacan inmediatamente han ingresado al sistema llamados de acción directa, o los que se encuentran en el equipo activos trabajando en paralelo con el sistema llamados residentes.

- **Los gusanos.** Son un tipo de programa que comparte las mismas características de un virus, pero a diferencia realiza copias de sí mismo o fragmentos de él sin necesidad de infectar otros archivos para poder multiplicarse en poco tiempo utilizando los canales de comunicación.
- **Los Troyanos.** Inspirados en la historia del “El Caballo de Troya”, son introducidos al ordenador camuflados en otros programas utilizando cualquier tipo de medio, se instalan y pueden realizar acciones, las cuales van desde afectar el funcionamiento del sistema, recolectar todo tipo de datos y enviarlas a un determinado destinatario, hasta tener el control total del sistema sin que el usuario se percate.

Con la evolución de los códigos maliciosos se han desarrollado amenazas de mayor complejidad; en este sentido, se han registrado subcategorías de troyanos como es el caso de downloaders (que permiten descargar otras amenazas desde Internet para instalarlas posteriormente), droppers (que instalan otros programas maliciosos

incluidos en su código fuente), clickers (para generar tráfico hacia sitios o avisos publicitarios que generan ganancias a sus desarrolladores) o los que se incluyen en la categoría de bancarios, creados especialmente para obtener datos relacionados con entidades financieras (Eset, 2016).

- **Rootkits.** Como otro tipo de programa malicioso diseñado para realizar ataques al sistema basados generalmente en las vulnerabilidades del mismo, permiten el acceso del atacante y la utilización de las funciones y recursos del sistema, ocultan su presencia, procesos, archivos o registros, características que dificultan el proceso de detección.
- **Adware.** Se define como Adware al software que despliega publicidad de distintos productos o servicios. Estas aplicaciones incluyen código adicional que muestra la publicidad en ventanas emergentes, o a través de una barra que aparece en la pantalla simulando ofrecer distintos servicios útiles para el usuario. Generalmente, estas aplicaciones agregan iconos gráficos en las barras de herramientas de los navegadores de Internet o en los clientes de correo. Estas barras de tareas personalizadas tienen palabras claves predefinidas para que el usuario llegue a sitios con publicidad, sea lo que sea que el mismo esté buscando (Segu.info, 2009).
- **Spyware.** Su función está orientada a recolectar información propia del usuario y enviarla a un tercero pasando desapercibido por el mismo.
- **Riskware.** Esta clase de código malicioso incluye todas las aplicaciones que incrementan los riesgos de seguridad. Del mismo modo que la instalación de spyware y adware, la instalación del riskware puede ser confirmada por un acuerdo de licencia. Los "dialers" como programas que desvían la conexión a un número pago preestablecido, son un ejemplo común de riskware. Estos programas pueden ser empleados para realizar el pago de servicios a través de Internet pero frecuentemente son mal utilizados y el desvío de información se produce sin el conocimiento del usuario (Eset, 2015).

Alan Marshall profesor de Seguridad de Redes en la universidad de Liverpool, comentó sobre una amenaza llamada “Chameleon”, diseñada por unos estudiantes de este mismo claustro, que este código malicioso que puede propagarse por el aire a través de redes Wi-Fi en áreas muy concurridas busca puntos de acceso desprotegidos, aunque su intención no es dañar o desactivar las redes, por el contrario, se infiltra sin ser detectada para recolectar datos y credenciales de sesión de todos los usuarios que estén conectados a la red y se encuentren navegando en sitios sin protocolos de cifrado. Asimismo, busca otras conexiones Wi-Fi para propagarse (Marshall, 2016)

En estos días se está desarrollando un tipo de ataque creciente denominado Ransomware, una técnica que impide el acceso a la información aplicando métodos de cifrado o bloqueo de la misma, no solo ataca a equipos de escritorio, sino a equipos que cuenten con un sistema operativo como el caso de Android. Una vez en el sistema, el atacante exige una cantidad de dinero con el fin de entregar una clave que permita acceder y desbloquear información.

En contraste, existen en el mercado otro tipo de programas diseñados para detectar y eliminar los códigos maliciosos antes mencionados tradicionalmente denominados “antivirus”, sin embargo, la realidad es que estos programas NO pueden eliminar todos los códigos maliciosos y sus derivados debido a que las técnicas de infección se están actualizando, además la creciente demanda de servicios en la red y las nuevas tendencias como el internet de las cosas (IoT, por sus siglas en inglés de Internet of Things), hacen que se busquen nuevas fuentes relacionadas con la información y se afecten otro tipo de dispositivos, entre ellos: SmartTV, Smartphone, smartwatch, entre otros.

La utilización o el empleo de malware por parte de los cibercriminales está enmarcado por las tendencias tecnológicas, la cual se ve reflejada en su evolución, así como también a motivaciones económicas.

Panda Security en su Informe Anual PandaLabs 2015, refiere un panorama al respecto y asegura que el 2015 ha sido, otra vez, el año de la historia que más

cantidad de malware se ha creado. A nivel geográfico, los países más infectados del mundo están liderados por China, con un 57,24% de infecciones, seguida de Taiwán, con un índice de infección del 49,15%, y Turquía con un 42,52%. Asia y Latinoamérica son las regiones con mayores infecciones. El resto de países, con un porcentaje mayor a la media mundial, son Colombia (33,17%), Uruguay (32,98%), Chile (32,54%) y España (32,15%), (Panda Labs, 2016).



Figura 1. Países y su Índice de infección. (Panda Security, 2015. p. 9)

En este mismo reporte relaciona como los troyanos lideran la cantidad de infecciones durante el 2105, seguido de otros o PUPs (Potentially Unwanted Programs, Programas Potencialmente No Deseados) con un 28.98% con casi un tercio de las infecciones, muy por delante del Adware/Spyware (5,19%), gusanos (2,98%) y virus (2,55%). Las técnicas agresivas de distribución junto a programas de software legítimos utilizadas por los PUPs hacen que consigan un alto ratio de instalación en los ordenadores de los usuarios.

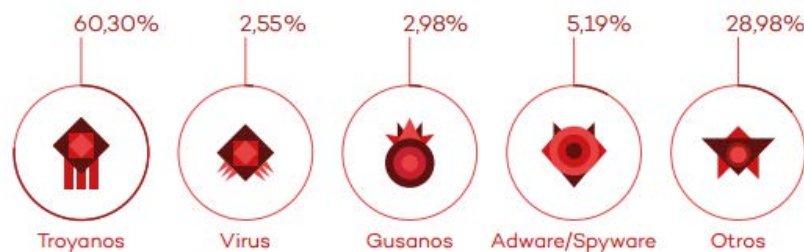


Figura 2 Infecciones por tipo de malware (2015). (Panda Security, 2015. p. 8)

Para el caso de Latinoamérica se muestra la siguiente panorámica:

De acuerdo con los resultados presentados el ESET Security Report 2016, refiere que la principal preocupación son las “Vulnerabilidades de software y sistemas” con el 58% de las respuestas afirmativas, seguido por el “Malware” (54%) y, en el tercer puesto, el “Acceso indebido la información” (46%), resultados que al ser comparados con los presentados en el ESET Security Report 2015 permiten ver que dos de las preocupaciones se mantuvieron en el mismo lugar.

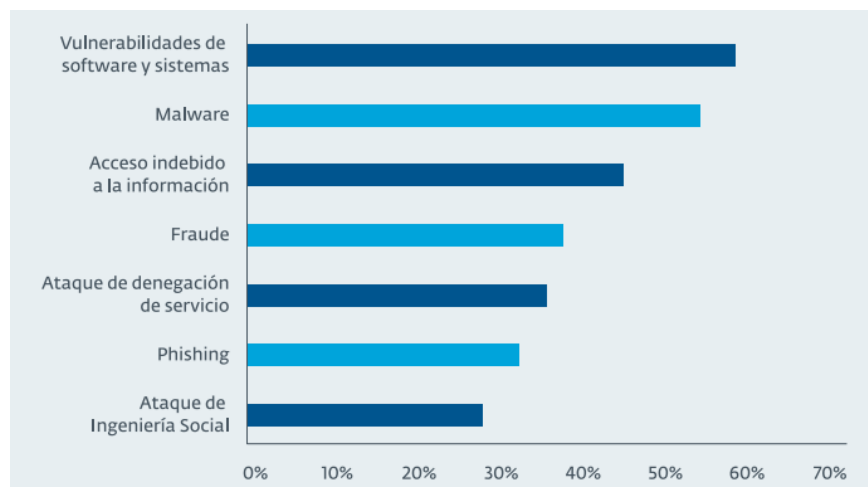


Figura 3. Preocupaciones en materia de seguridad en las empresas de Latinoamérica (Eset, 2015, p. 4)

Este mismo reporte presenta una segunda sección la cual se enfoca en conocer los principales incidentes de seguridad que afectaron a las empresas, cuya probabilidad significativa puede comprometer sus operaciones y atentar contra la confidencialidad, integridad y disponibilidad de la información:

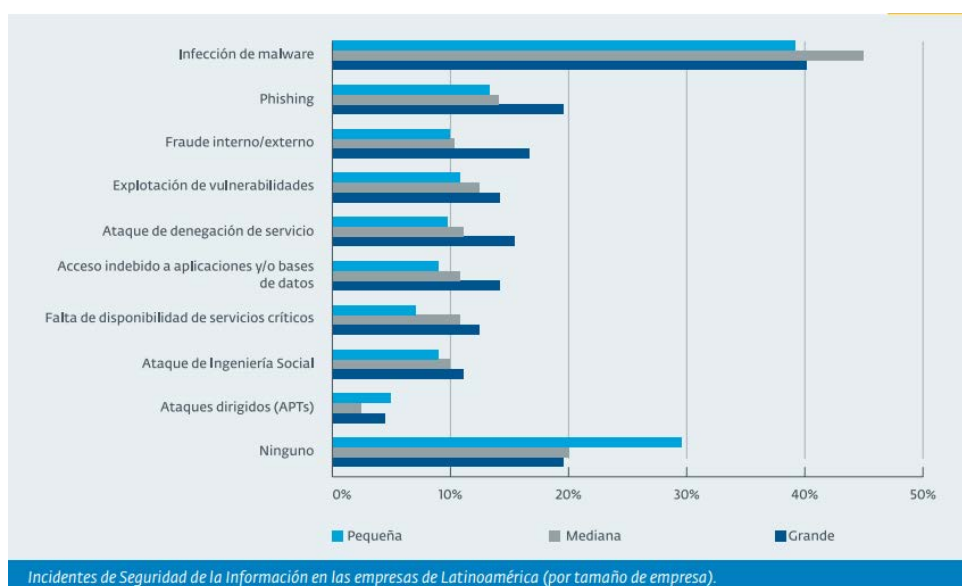


Figura 4. Incidentes de Seguridad de la Información en las empresas de Latinoamérica (ESET, 2016, p. 8)

La grafica anterior representa la cantidad de incidentes de seguridad de la información y sus porcentajes de ocurrencia. Dichos resultados permiten observar que la infección de malware se encuentra en el primer lugar, seguida en su segunda posición por el Phising y como dato adicional aparece cerrando dichos resultados los ataques dirigidos (APTS).

La firma Panda Labs en su informe anual refiere que habrá un aumento de ataques dirigidos. El uso de técnicas de rootkit, que permiten al atacante ocultarse de la vista del sistema operativo y de las soluciones de seguridad se intensificará en este tipo de intrusiones. Las empresas se van a ver obligadas a tomar medidas de seguridad extra para estar protegidas ante esta amenaza real que puede dañar seriamente la imagen y las finanzas de las víctimas, teniendo en cuenta que puede tener como objetivo tanto información confidencial de la empresa (datos financieros, planes estratégicos, etc.) como datos de clientes de la misma (Panda Labs, 2016).

Este panorama o escenario analizado y presentado por las empresas anteriormente mencionadas, sus resultados y posiciones, nos permite ampliar la visión respecto a la importancia para consolidar el presente documento como soporte esencial del desarrollo de este piloto experimental.

4. Advanced Persistent Thread APT

Como se mencionó en el capítulo anterior, la base en el desarrollo del presente piloto experimental está enfocado a la aplicación de la metodología de “Análisis e ingeniería inversa de código malicioso” desarrollado por Don Javier Bermejo en su Tesis Doctoral (Bermejo, 2015), la cual según su autor, se justifica mediante el desarrollo de un proceso de experimentación, aplicando el método Desarrollo de Conceptos y Experimentación (CD&E) sobre el sobre la amenaza persistente avanzada (APT) conocida como lo es Poison IVY, creemos imperativo desarrollar una serie de conceptos que permitan comprender el origen, características y demás elementos propios de este tipo de software malicioso.

Según Don Javier Bermejo en su metodología, refiere respecto a las APTs que constituyen el tipo de arma cibernética más complejo desarrollado en la actualidad. Fabricadas normalmente por organizaciones criminales de gran poder económico o por los servicios de inteligencia de diversos estados.

El estado actual de las APTs respecto a su tecnología, se debe gracias a la evolución propia del malware, la cual ha permitido generar diferentes métodos y formas de propagación, ocultamiento, accesos, sofisticando los ataques, modificando sus vectores, orientando todos los esfuerzos a un objetivo en particular que depende de quien se encuentre detrás de cada incidente, sobre periodos de tiempo pre establecidos; una serie de características que las convierte en una herramienta predilecta por las organizaciones cibercriminales. Y como lo presenta en su Tesis Doctoral (Bermejo, 2015), las APTs presentan los siguientes objetivos:

- ✓ Políticos. Enfocados al el mantenimiento de la estabilidad interna de un país.
- ✓ Económicos. Basados en el robo de propiedad intelectual, diseños, innovaciones, etc.
- ✓ Técnicos. Los cuales incluyen el acceso al código fuente de los desarrollos de seguridad para aprovechar o aprender cómo funcionan las defensas con el fin de evadirlas o interrumpirlas.
- ✓ Militares. Que Incluyen la identificación de puntos débiles de la víctima.

Sin lugar a dudas desde hace algún tiempo atrás se han reportado una serie de ataques definidos como APT, el cual a diferencia de otros, permiten concluir que no se trata de campañas comunes y corrientes respecto a la aplicación de código malicioso sobre un plano general de ataque, sino que se han definido puntos y objetivos de ataques. El informe de “Tendencias 2015: el mundo corporativo en la mira”, presentado por ESET Lab resalta el impacto que las APTs tienen para la seguridad de las empresas y cómo se convirtió en uno de los mayores desafíos a los que una organización se puede enfrentar en términos de Seguridad de la Información.

Por otra parte, el repositorio APTNotes (GitHub) creado y mantenido por la comunidad GitHub, el cual recopila todos los informes de ataques dirigidos publicados desde el año 2008 hasta la fecha, reportó un total de setenta y un (71) informes en el año 2015 de las cuales resaltan campañas de ataques dirigidos contra una empresa en particular, instituciones o gobiernos, especialmente de países como China, Israel, Taiwán y Japón.

Como podemos observar este panorama es presentado desde el punto de vista de la compañía ESET, y aunque un plano general, consolidado y real de los ataques ocurridos se torna algo complejo, debido a que muchas empresas deciden no reportar que han sido víctimas con el fin de proteger su imagen comercial y social disminuyendo así su impacto, tomando como base la documentación presentada en la Tesis Doctoral de Don Javier Bermejo (Bermejo, 2015), las cuales combinadas con otras fuentes, nos permiten tener una visión respecto a los principales incidentes y sus escenarios relacionados con ataques basados en APTs, en los últimos tres años:

- Desde un análisis presentado por ESET LABs en el informe TENDENCIAS 2016 (IN) SECURITY EVERYWHERE (Eset, 2016), los sucesos más importantes durante el año 2015, fueron son las campañas denominadas como: Potao Express, Animal Farm, Terracota VPN, Mumblehard y Carbanak entre otras, de la cual resalta Potao Express, como una campaña de malware específica con múltiples herramientas para el ciberespionaje, los cuales centraron como objetivos de ataque diferentes secciones del gobierno ucraniano, entidades militares de ese país e incluso agencias de noticias, así como también los países de Rusia, Georgia y Bielorrusia.

Este código malicioso referido en la familia conocida como win32/Potao, permite a través de una estructura modular, elegir diferentes herramientas de acuerdo a la acción que el atacante quiera desarrollar respecto a un objetivo, el cual utiliza como vector de ataque relacionado con infecciones a través de unidades USB.

Por otro lado, se encuentra el ataque al grupo denominado Animal Farm, a quienes se le atribuye la creación de familias de malware referidas como Dino, Casper, Bunny y Babar, que en su gran mayoría su aparición se remonta a años anteriores, demostrando claramente la persistencia de este tipo de ataques en compañías dirigidas, cuyos objetivos según los analistas de ESET fueron objetivos situados en Irán y Siria.

Ampliando el campo de ataque, una investigación de impacto se refirió bajo la operación Buhtrap la cual orientó sus objetivos a diferentes bancos de Rusia; campaña descubierta a finales del 2014. Este ataque se originaba cuando los atacantes instalaban amenazas en aquellos sistemas que tuvieran definido como lenguaje determinado al RUSO, utilizando como vector de infección una vulnerabilidad detectada en Microsoft Word la cual había sido descubierta tres (3) años antes.

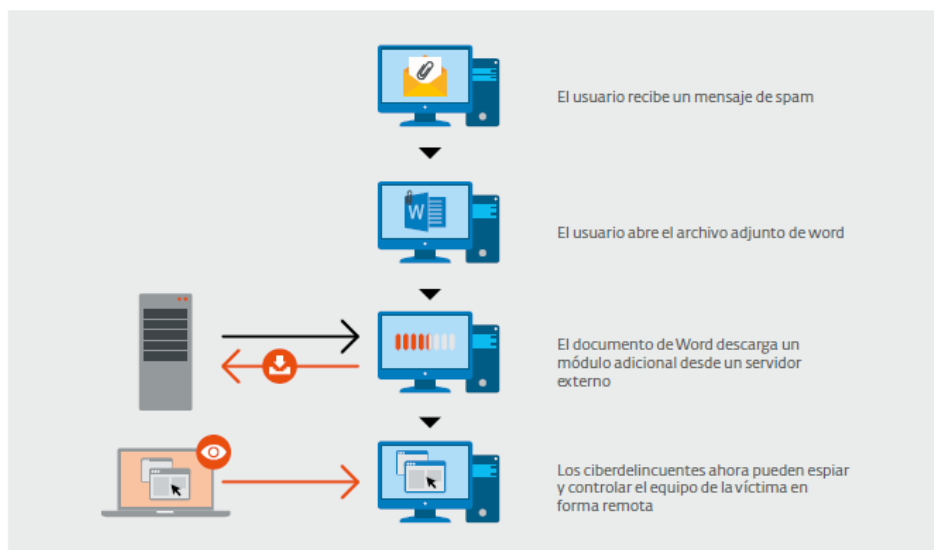


Figura 5. Estructura de una campaña Buhtrap. (Eset, 2016, p. 21)

En el año 2015 se detectó persistencia sobre esta campaña respecto a su propagación a través de un sitio web conocido, utilizando técnicas spear phishing, materializando casos como el downloaders Lurk, Corebot, Rnayus y el RAT Netwire, como ataques dirigidos a sistemas Windows con la finalidad de robar información confidencial del usuario, espiar sus actividades y así, recolectar información.

Si bien es cierto los sistemas pertenecientes a la familia Microsoft son los más atacados, en el año 2015 se detectaron campañas con más de cinco años de antigüedad con objetivo sobre servidores basados en Unix con el fin de enviar spam, tal es el caso del malware Linux/Mumblehard como un código malicioso desarrollado en Perl, el cual contaba con dos componentes principales: Un backdor que otorgaba al atacante un acceso remoto, sobre un segundo componente un daemon encargado del envío del spam.

Este informe precisa respecto a las conclusiones de sus analistas que las campañas específicas contra servidores web u otros servicios que están disponibles en la WEB, son objetivos de los criminales que sin lugar a dudas están cobrando fuerza a sus ataques, lo que permite pasar desapercibidos y manejar la persistencia en periodos de tiempo más extensos. Las APTs son armas del futuro, y una característica esencial es que la predicción de sus ataques se torna difícil, las empresas y organizaciones quieren saber si han sido y serán víctimas, y está en ellas proteger y blindar sus mecanismos de seguridad, elevar sus niveles de protección, capacitar a sus empleados y usuarios respecto a minimizar la exposición al riesgo de sus sistemas, como base fundamental del negocio.

- Para el Año 2014, Para Miguel A. Hernández, en su publicación Resumen Ciberdefensa 2014 (Hernandez, 2014), Desde el plano internacional los ataques procedentes desde Rusia se hicieron evidentes durante este año, ataques como Epcic Trula, Snake, Sandworm, Havez y CosmikDuque. Los sucesos de la actividad ciber-ofensiva rusa ha quedado enmarcada en el transcurso de este año en Ucrania, los cuales van de la mano con los incidentes militares como el caso de la anexión de la península de Crimea en una operación realizada por soldados no identificados

pertenecientes a fuerzas especiales rusas. La posterior guerra civil en las provincias del este Ucraniano, entre los leales al gobierno y rebeldes pro-rusos, así como también el reporte Malware Attack Targeting Syrian ISIS Critics (Railton & Hardy, 2014), el cual ofrece una visión al respecto del uso de malware en contra de activistas sirios opuestos al régimen del Estado Islámico, lo cual claramente permite visualizar el empleo de internet por el grupo terrorista ISIS.

En la publicación “Secret Malware in European Union Attack Linked to U.S. and British Intelligence” presentada en el portal Theintercept, Boire, Guarnieri, & Gallagher (2014), refieren la aparición de “Regin” como una operación APT basada en malware altamente sofisticado, la cual entre sus características más destacables ofrecía módulos para atacar la infraestructura interna de operadores GSM, la cual casi total seguridad su origen estaría en una operación del GCHQ británico, posiblemente en colaboración con la NSA, la cual tuvo una atribución concreta a que este malware fue utilizado en el compromiso de Belgacom.

Por otro lado, de acuerdo a la publicación CCN-CERT IA-09/15 (CN-Cert, 2014) el ciberespionaje ha fundamentado sus acciones en el uso de técnicas APT (Advanced Persistent Threat), dirigiéndose contra distintos objetivos que, en el caso de España, se han centrado en determinados departamentos de las administraciones públicas españolas, la industria de la Defensa, aeroespacial, energética, farmacéutica, química, TIC, así como los dispositivos móviles del personal directivo de estos sectores. Conociendo para el año 2104 por parte de CERT Gubernamental Nacional un total de 12.916 incidentes, de los cuales, 132 fueron catalogados como críticos; en los cuales pueden causar degradación de los servicios para un gran número de usuarios, reflejado en una grave violación de la seguridad de la información, que puede afectar a la integridad física de las personas, pérdidas económicas, ocasionar daños irreversibles a los recursos de la organización, e incurrir en delitos.

- Año 2013 En su publicación: Principales incidentes de seguridad que conformaron el campo de amenazas en 2013, kaspersky (2013), refiere a los actores de amenazas avanzadas como responsables de desarrollar operaciones

a gran escala, surgiendo cybermercenarios, grupos de amenazas persistentes avanzadas (APT) que alquilaban sus servicios para focalizar operaciones relámpago. La mayoría de las campañas de ciberespionaje detectadas por los analistas de Kaspersky Lab fueron diseñadas para robar información de agencias gubernamentales e instituciones de investigación, entre ellas: Octubre Rojo, NetTraveler, Icefog y MiniDuke, las cuales en su aplicación, mediante ingeniería social comenzaron "hackeando a los seres humanos", emplearon técnicas spear-phishing para obtener un punto de apoyo inicial en las organizaciones blanco.

La campaña más extendida durante el año fue el espionaje de NetTraveler, que afectó a víctimas de cuarenta países, dejando como precedente a los cibercriminales quienes recolectaron información de dispositivos móviles conectados a las redes de las víctimas, claro indicio de la importancia de los móviles para los hackers. Entre los objetivos de NetTraveler, conocido como "Travnet" o "Netfile", se encontraban activistas tibetanos y uigures, compañías petroleras, centros e institutos de investigación científica, compañías privadas, gobiernos e instituciones gubernamentales, embajadas y contratistas del ejército, para lo cual se infectan a través de ataques de phishing utilizando documentos de oficina que explotan dos vulnerabilidades conocidas públicamente - CVE-2012-0158 y CVE-2010-3333, a pesar de estas vulnerabilidades han sido actualizados por Microsoft, siguen siendo eficaces y se encuentran entre los más explotados en ataques dirigidos.

4.1. Principales características de las APT

Tomando el análisis presentado en su Tesis Doctoral Don Javier Bermejo (2015), quien refiere que aunque los métodos y tecnologías usadas en los ataques pueden variar considerablemente, las características que normalmente posee este tipo de malware y que casi siempre exhiben pueden ser las siguientes según (Curry, et al., 2009):

- Son ataques desarrollados y adaptados a la organización objetivo. Para lo cual los ciberatacantes seleccionan la información y sus objetivos en base a intereses

políticos o tecnológicos y emplean técnicas de reconocimiento e inteligencia para obtener información de los sistemas, aplicaciones y redes de la organización víctima, como paso previo a la explotación de vulnerabilidades encontradas y no corregidas o por el contrario si no se encuentra ninguna aplicar otras desconocidas (zero-day).

- Emplean un perfil Bajo para evitar su detección. Los ciberatacantes mantienen un perfil bajo dentro de los sistemas TIC de las organizaciones en las que se infiltran, se mantienen latentes en búsqueda de condiciones óptimas para un ataque, pudiendo llegar a esperar meses enteros. El comportamiento y las “firmas” de un ataque de este tipo son difíciles de correlar con patrones de ataques conocidos, incluso si la organización dispone de eventos o un sistema SIEM.
- Son desarrolladas por organizaciones con elevada solvencia económica. La aplicación y el desarrollo de ataques APT suelen recurrir o demandar considerables recursos financieros que permiten mantener un ataque durante largos periodos de tiempo. Adicionalmente la evolución de las APTs requiere que sugiere que estas organizaciones dispongan de estructuras organizaciones amplias y de equipos multidisciplinares de ciberatacantes con amplias habilidades y experiencia en lograr el acceso a infraestructuras complejas de sistemas TIC.
- Ofrecen métodos de ataque simultáneos y diversos. Una APT utiliza múltiples vectores de ataque simultáneos. Usan una gran cantidad de métodos y tecnologías para infiltrarse e infectar nodos de un objetivo y con frecuencia utilizan ataques de bajo riesgo para distraer a los administradores y a los analistas de seguridad, evitando que se percaten del ataque. Utilizan sistemas automatizados para atacar múltiples objetivos.
- La aplicación de Ingeniería social. Es muy común utilizar herramientas de ingeniería social y de las redes sociales, y las técnicas que en ellas de

desarrollan, como fuente inequívoca de información, debido a que APT's aprovechan las vulnerabilidades de la naturaleza humana.

- Manejan objetivos claros. Como mencionamos en apartados anteriores, una de las características básicas de las APTs es la de atacar un objetivo en particular, tomar el control de los elementos cruciales en una infraestructura en los diferentes ámbitos de la sociedad.

4.2. Fases de ataque en un APT

La anatomía de un APT es diversa y varía de acuerdo a cada objetivo para el cual ha sido diseñado, la utilización de diversos vectores de ataque y sus combinaciones, hacen que los métodos de detección y defensa de los APTs sean una tarea complicada.

Desde la utilización de la ingeniería social, técnicas de footprinting y los métodos de propagación comunes en los malware con el fin de alcanzar un objetivo, permiten evadir todas las defensas perimetrales y demás mecanismos que puedan existir en un sistema. Y aunque cada ataque difiere uno de otro, la tarea de identificar y unificar su estructura por la diversidad existente, se ha identificado que estos siguen una secuencia estructurada con la aplicación de serie de actividades las cuales se ejecutan de manera progresiva y exitosa. Desde un análisis más detallado, Steve Piper (2013) en su publicación Definitive Guide para la protección contra amenazas de próxima generación, dada a conocer por CyberEdge Group, permiten visualizar desde la aplicación del ciclo de vida del ataque las siguientes fases:

Fase 1: Intrusión inicial aprovechando las vulnerabilidades del sistema

Esta fase tiene por objeto irrumpir en un sistema de una organización previamente seleccionada para infectarlo. La detección de una vulnerabilidad existente podría facilitar el ataque, pero no siempre, ya que los sistemas de seguridad implementados por las organizaciones pueden dificultar considerablemente esta fase.

Los ataques APTs suelen realizarse a través de la web (exploit remoto), o desde archivos adjuntos a un mensaje de correo electrónico específico (exploit local). Estos exploit pueden estar integrados en un objeto web (por ejemplo JavaScript, JPG, archivos en formatos XLS, PDF) que infecta al sistema y permite al atacante ejecutar código como Shell que permiten conectar con los servidores predefinidos, alcanzando así el control total y descargar más malware. En ocasiones el código del exploit del sistema tiene como objeto dañar la memoria o causar un desbordamiento de buffer en el sistema operativo o la aplicación vulnerable, lo que conllevaría a la ejecución de sentencias abriendo paso al ataque.

Fase 2: se instala malware en el sistema afectado

Una vez alcanzada la víctima y estando dentro del sistema se ejecuta el código malicioso para su infección, para ello basta con visitar una página web a través de un link, o simplemente hacer doble clic con el ratón para que el sistema del usuario se infecte con la carga útil del malware.

Aunque no todos los correos electrónicos de phishing selectivo que envía el autor de una amenaza APT contienen datos adjuntos. Muchos de estos contienen hipervínculos que, cuando el usuario hace clic sobre ellos, abren un navegador web (o a veces otra aplicación, como Adobe Reader, Microsoft Word o Microsoft Excel). Cada vínculo se redirige a su vez a una dirección oculta con una clave de codificación en base64. La dirección oculta conduce a un dropsite, un sitio depósito que examina el navegador en busca de vulnerabilidades y devuelve un descargador de troyanos. Al ejecutarse, el descargador transmite una instrucción codificada en base64 a otro dropsite distinto del que se descarga un troyano (malware).

Fase 3: se inicia la conexión de salida

El malware instalado durante la fase anterior a menudo contiene una herramienta de administración remota o RAT, por sus siglas en inglés. En cuanto está configurada y activa, la RAT "llama a casa", para lo cual establece una conexión de salida, con frecuencia un canal cifrado con SSL, entre la computadora infectada y un servidor de comando y control que manejan los autores de la amenaza APT. Estos se toman tantas molestias porque

necesitan establecer comunicaciones salientes que eludan los firewalls tradicionales y de próxima generación, que permiten que el tráfico de la sesión fluya de forma bidireccional si se inicia desde la red de confianza.

En cuanto la RAT logra conectar con el servidor de comando y control, el atacante tiene pleno control sobre el host infectado. Las instrucciones que envía a partir de ese momento se transmiten a la RAT a través de dos vías: o bien el servidor de comando y control conecta con la RAT o viceversa. Esta última es la vía preferida, ya que es mucho menos sospechoso que un host inicie una conexión externa desde dentro de la red.

Fase 4: el atacante se propaga lateralmente

Es muy improbable que el dispositivo informático del usuario final vulnerado en un principio contenga datos estratégicos.

Por lo tanto, el atacante de la APT debe propagarse lateralmente a través de la red en busca de los hosts que manejan los administradores de TI (con objeto de robar credenciales administrativas) y de servidores y bases de datos de gran valor que contengan datos confidenciales: el objetivo último del ataque APT. Así es como funcionaba la campaña denominada Flame, la cual tuvo lugar en el año 2012 y tuvo como objetivo rastrear de forma secreta redes informáticas de Irán y controlar los ordenadores de los funcionarios iraníes, extrayendo de ellos y enviando un flujo constante de información.

El movimiento lateral no implica necesariamente el uso de malware o de herramientas distintas de las que incluye el sistema operativo del host comprometido, como shells de comandos, comandos de NetBIOS, VNC, Windows Terminal Services u otras herramientas similares que utilizan los administradores de red para dar servicio a hosts remotos.

Una vez identificado el blanco final y reunido las credenciales adecuadas de inicio de sesión, el duro trabajo y la determinación del atacante empiezan a dar sus frutos.

Fase 5: se extraen los datos comprometidos

En esta etapa de la intrusión en la red, el responsable de la APT tiene tres obstáculos que vencer. En primer lugar, si transfiere todos los datos seleccionados de una sola vez (los datos suelen cuantificarse en gigabytes), podría activar una alerta de anomalía de flujo (si se utiliza tecnología de análisis del comportamiento de la red (NBA), por ejemplo) por el volumen inusualmente elevado de tráfico iniciado por el servidor o la base de datos atacada.

En segundo lugar, el atacante debe asegurarse de que nadie pueda relacionarle con el host que recibe los datos. Y, por último, si transfiere los datos como texto normal, podría activar una alerta del sistema de prevención de fugas de datos (DLP). Para superar los tres obstáculos los experimentados perpetradores de amenazas APT:

Para vencer el primer obstáculo, el astuto atacante extrae los datos que le interesan del servidor o la base de datos en "bloques", quizá en incrementos de 50-100 megabytes. Una estrategia es agrupar archivos o registros en archivos RAR comprimidos y protegidos con contraseña. Algunos archivos RAR pueden formar parte de secuencias de varios volúmenes, lo que permite al atacante dividir una gran cantidad de datos en volúmenes. Cada archivo RAR tendría una extensión que describiría el número del volumen, como parte1.rar (primer volumen), parte2.rar, parte3.rar, y así sucesivamente.

El segundo obstáculo es un poco más difícil. El atacante necesita extraer los datos lo antes posible, pero no puede arriesgarse a enviarlos a un host cuyo rastro permita localizarle. Para resolver esta dificultad, puede seleccionar un host virtual alojado en un proveedor de servicios en la nube como zona de almacenamiento temporal. De este modo, el host puede destruirse inmediatamente una vez que se han extraído los datos.

El tercer y último obstáculo de esta fase puede solucionarse cifrando los archivos RAR antes de transferirlos (normalmente por FTP) al host provisional. La mayoría de los archivos RAR admiten el fuerte cifrado AES de 128 bits, que es más que suficiente.

Sin embargo, podemos contemplar una fase adicional expuesta en la Tesis Doctoral por Don Javier Bermejo (2015), la cual refiere el Mantenimiento de la persistencia. En esta fase se siembran múltiples vectores de ataque en los equipos comprometidos con el fin de disponer de puntos de acceso adicionales, que servirán en caso de perder el que está en

funcionamiento en ese momento. Algunos de estos exploits pueden a su vez ser señuelos o herramientas que ya han cumplido su propósito y que por tanto podrían ser identificados y eliminados sin afectar el funcionamiento del exploit principal. Por ejemplo, una organización puede tener un sistema afectado fuera de línea para eliminar las amenazas detectadas por conocidos anti-virus. Una vez detectadas las amenazas y eliminadas, el sistema se vuelve a poner en línea.

4.3. Indicadores de Ataque de un APT

Para FireEye, refiere que aunque las APTs son en extremo difíciles de detectar, una de las razones fundamentales por la que las organizaciones no consiguen identificar los ataques APT es que sus dispositivos de seguridad solo están (o están principalmente) configurados para examinar el tráfico que entra en el perímetro (Bennet & Moran, 2013). Existen una serie de signos que indican la posibilidad de estar siendo atacado por una amenaza de este tipo, entre ellas:

- Hallar código para infiltrarse en un sistema en los datos adjuntos a correos electrónicos o descargados de una página web.
- Detectar un incremento de inicios de sesión con privilegios elevados en medio de la noche.
- Detectar conexiones salientes a servidores de comando y control conocidos.
- Hallar troyanos de puerta trasera generalizados en endpoints o en recursos compartidos de archivos de red.
- Observar flujos de datos voluminosos e inesperados desde el interior de la red, ya sea entre servidores, de servidor a cliente, de cliente a servidor o entre redes.
- Descubrir grandes bloques de datos (gigabytes, no de megabytes) en lugares donde no debería haber datos.

- Comunicaciones de red anómalas cifradas con SSL.
- Entradas en el registro de eventos de aplicación de Windows con comandos de activación y desactivación del firewall y el antivirus.

4.4. Defensas ante las APTs

Hemos visto las características propias que permiten conocer las APTs, así como su secuencia de ataque, sin embargo, creemos imperativo conocer desde el punto defensivo cuáles son los métodos o técnicas que contrarrestan estos ataques.

En su Tesis Doctoral, Don Javier Bermejo (2015), expone los siguientes métodos de defensa:

Como sistema de detección de APTs, se puede aplicar la interceptación de las comunicaciones salientes de la Red de la organización con su sistema de mando y control, como medio perceptible de la presencia de un ataque, analizando posible anomalías en el tráfico DNS, HTTP, HTTPS, SMTP e IRC producido por el malware.

La forma más eficaz de la lucha contra el malware tipo es el establecimiento de una estrategia de “Defensa en Profundidad”, consistente en introducir múltiples capas de seguridad que permitan reducir la probabilidad de compromiso en caso de fallo progresivo de cada una de ellas; y en el peor de los casos minimizar el impacto, de forma que:

- Se consiga que la penetración inicial, para obtener acceso, sea muy dificultosa.
- Se controle el tráfico de red para detectar posibles comunicaciones del malware con su sistema de mando y control.

- Se detecten las cuentas comprometidas y la actividad sospechosa al principio del ciberataque, reduciéndose con ello la posibilidad de una escalada de privilegios, en el caso de que una cuenta se vea comprometida.
- Se implanten técnicas de segmentación, de modo que se limite el daño potencial que se puede originar en una cuenta comprometida a exclusivamente a esa máquina (incluso si es privilegiada) y no se propague a las otras.
- Recopilar toda la información posible para una posible investigación forense posterior.

Para ello plantea la aplicación los controles de ciberseguridad que el instituto SANS² ha definido como críticos de manera global, la cual expone a la vez la necesidad de que cada organización establezca sus propias estrategias de seguridad de acuerdo a sus características de negocio, reflejando más o menos controles, para lo cual, como punto de partida puede aplicar la siguiente estrategia básica:

1 Inventory of Authorized and Unauthorized Devices	11 Limitation and Control of Network Ports, Protocols, and Services
2 Inventory of Authorized and Unauthorized Software	12 Controlled Use of Administrative Privileges
3 Secure Configurations for Hardware & Software on Laptops, Workstations, and Servers	13 Boundary Defense
4 Continuous Vulnerability Assessment and Remediation	14 Maintenance, Monitoring, and Analysis of Security Audit Logs
5 Malware Defenses	15 Controlled Access Based on the Need to Know
6 Application Software Security	16 Account Monitoring and Control
7 Wireless Device Control	17 Data Loss Prevention
8 Data Recovery Capability	18 Incident Response Capability
9 Security Skills Assessment and Appropriate Training to Fill Gaps	19 Secure Network Engineering
10 Secure Configurations for Network Devices such as Firewalls, Routers, and Switches	20 Penetration Tests and Red Team Exercises

Figura 6. 20 Critical Security Controls. (blog.segu-info.com.ar, 2012)

² <http://www.sans.org/>

- ✓ Clasificación de activos, datos y flujo (Controles 1 y 2).
- ✓ Formación y concienciación en Seguridad (Control 9).
- ✓ Pruebas regulares de seguridad (Controles 4, 6 y 7).
- ✓ Uso apropiado de Sistemas Críticos (Control 12).
- ✓ Distribución de credenciales de forma segura.
- ✓ Bastionado de sistemas y actualización de parches (Controles 3 y 4).
- ✓ Antivirus / Anti-Spyware (Control 5).
- ✓ Capacidades de Respuesta a Incidentes (Control 18).
- ✓ Auditoría de cuentas de usuario y privilegios (Controles 12 y 15).
- ✓ Monitorización de log (Control 14).
- ✓ Implantación de un sistema anti fuga de datos (DLP) (Control 17).
- ✓ Segmentación de la red de área local en redes virtuales en función del tipo de activos (Control 13).
- ✓ Auditoría de cuentas de usuario (Control 16).
- ✓ Sistemas de detección de anomalías de red (Control 14).
- ✓ Reglas del cortafuegos frontera granulares (filtrado a la salida) (Controles 11, 13 y 19).
- ✓ Formación en análisis de malware (Control 9)
- ✓ Borrado seguro de datos.
- ✓ Monitorización de transacciones anómalas (Control 14)
- ✓ Inspección detallada de paquetes (Control 14)
- ✓ Supervisión del tráfico saliente (Control 14)
- ✓ Medidas de defensa frente ataques DDoS

Y finalmente, presenta como sistema de defensa la implementación de sistemas de alerta temprana que permitan identificar las actividades realizadas por un malware APT, en sus primeras fases de inicio cuando pretenda acceder internamente al sistema. Así como también la utilización de herramientas especializadas en la defensa contra este tipo de malware, como: Splunk, inCircle Suite360 o TRITON.

4.5. Advanced Persistent Threat (APT) POISON IVY

Como observamos en los apartados anteriores las amenazas persistentes avanzadas (APT) a diferencia del malware común diseñado muchas veces para aprovechar una simple oportunidad, manejan una estructura organizada y sistemática en el proceso de ataque.

En este apartado se presenta la Amenaza Persistente Avanzada POISON IVY elegida para desarrollar el piloto experimental, presentando las publicaciones existentes en fuentes abiertas que relacionan este tipo de malware con el fin de comprender su origen, estructura y método de ataque, antes de adentrarnos en la aplicación de la metodología desarrollada por Don Javier Bermejo (2015), con el fin de evaluar las técnicas y procedimientos, resultados que se presentaran más adelante.

Poison IVY³ es una herramienta de administración remota la cual sigue siendo utilizada pese a que lleva algunos años desde su aparición en el año 2008, básicamente maneja una interfaz tipo Windows, ofreciendo numerosas funciones o herramientas fáciles de utilizar, básicas y poderosas a la vez, como medio para administrar como herramienta de remota un sistema (RAT), entre ellas: Capturas de pantallas, capturas de video, registro de pulsaciones en el teclado, robo de contraseñas, administración del sistema, transmisión de tráfico, entre otras.

La amplia disponibilidad, además de sus características y facilidad en su manejo, hacen que se haya convertido en una opción para todo tipo de delincuentes, dejando un registro importante en la historia de los ataques APTs conocidos.

4.5.1. Antecedentes

Tomando el informe proporcionado por FireEye Labs Bennet & Moran (2013), así como otras fuentes, desde la versión 2.3.2 de Poison Ivy lanzada en el año 2008, la utilización de esta herramienta de administración remota (RAT), los atacantes han desarrollado los siguientes campañas de ataque:

³ www.poisonivy-rat.com.

- ✓ admin@338: La cual se encuentra activa desde 2008, esta campaña está diseñada para atacar principalmente al sector de los servicios financieros, aunque también se ha observado actividad en los sectores de telecomunicaciones, gubernamental y defensa. El vector de ataque favorito de esta campaña son los mensajes de correo electrónico de phishing selectivo; mensajes que incluyen contenido de interés para la víctima, incitándola a abrir un archivo adjunto que contiene el código malicioso del servidor Poison IVY.
- ✓ Th3bug: Campaña detectada por primera vez en el año 2009, elige sus objetivos en diversos sectores fundamentalmente la enseñanza superior y los servicios de la salud. Como vector de ataque no aplica el phishing selectivo para la distribución de Poison IVY, sino que compromete sitios web estratégicos para infectar el blanco.
- ✓ MenuPass: Campaña lanzada también en el año 2009, parece tener su origen en China y estar dirigida a contratistas de defensa estadounidenses y de otros países. Utiliza como vector de ataque el phishing selectivo para introducir su carga útil en el blanco seleccionado.
- ✓ RSA: campaña reportada en el año 2011 con el fin de afectar la empresa de seguridad RSA, asociada también a los agresores chinos como uno de los ataques más sofisticados en su momento, utilizando una vulnerabilidad de tipo día cero inyectando Poison IVY como carga útil en el sistema y según los expertos se inició desde el año 2010 afectando a gran cantidad de empresas.
- ✓ Nitro: Campaña reportada también en el año 2011, la cual según GovCERTUK (Govcertuk, 2011) y "Java Zero-Day Used in Targeted Attack Campaign", (symantec.com, 2012). fue diseñada para atacar a fabricantes de productos químicos, organismos públicos, contratistas de defensa y grupos de lucha en favor de los derechos humanos, se utilizó para el ataque una vulnerabilidad de tipo día cero en Java para desplegar Poison IVY.
- ✓ Según Yichong Lin (2013), en este último tiempo, se detectó a Poison IVY utilizado como carga útil de otra vulnerabilidad desconocida en Internet Explorer que

comprometía sitios web estratégicos atacando a los visitantes de un sitio web del gobierno estadounidense y otros de naturaleza variada.

4.5.2. Estructura de ataque de Poison IVY

Desde el análisis presentado por FireEye Labs, Poison IVY permite a los agresores personalizar su herramienta creando su propio servidor en el cual se introduce una especie de código móvil en el equipo comprometido, al que se accede generalmente por técnicas de ingeniería social. Una vez alcanzado el objetivo y ejecutado en Endpoint de la víctima, se conecta a un cliente de poison IVY instalado en la máquina del agresor quien se encarga de los procesos de administración.

Para desarrollar el análisis, la publicación refiere la recaudación de 194 muestras de Poison Ivy utilizadas entre 2008 y 2013 en ataques selectivos, de las cuales se extrajeron 22 contraseñas diferentes y 148 mutex⁴. Se trazó el mapa de las infraestructuras correspondientes de comando y control, que comprendían 147 dominios y 165 direcciones IP. Muestras que fueron analizadas con el fin de comprender las herramientas, tácticas y procedimientos (TTP) de los agresores, examinar sus interconexiones en las campañas y favorecer que los responsables de seguridad protegieran mejor sus redes (Bennet & Moran, 2013).

Puerto TCP utilizado	Número de muestras de PIVY
443	157
80	104
8080	22
8000	12
1863	7

⁴ Mutex: Objetos de Windows que se utilizan para sincronizar procesos.

Figura 7. Puertos TCP variantes de PIVY en ataques mediante APT. (FireEye, 2013 p. 14)

Cada servidor Poison IVY desde donde se envía el malware al objetivo, puede configurarse para conectarse a varios servidores de comando y control a través de cualquier puerto TCP, empleando para sus ataques los puertos destinados al tráfico web, especialmente el 443, puerto TCP utilizado para tráfico web cifrado con SSL. Este puerto es fundamental debido a que: en primer lugar, las defensas del perímetro deben permitir que el tráfico saliente pase a través de este puerto para que los usuarios puedan acceder a sitios web legítimos cifrados con SSL. En segundo, dado que el tráfico del puerto 443 está cifrado, el tráfico igualmente cifrado de Poison IVY puede mezclarse con la actividad normal de la red.

Mutex de proceso de PIVY	Número de muestras de PIVY
)!VoqA.I4	14
K^DJA^#FE	4
KEIVH^#\$\$	3
%1Sjfhtd8	3
2SF#@R@#!	3

Figura 8. Mutex de proceso variante de PIVY asociadas a ataques mediante APT. (FireEye, 2013 p. 14)

El malware suele emplear los mutex para asegurar que en un momento determinado solo se esté ejecutando una instancia del mismo en el sistema infectado, de esta forma, el agresor puede definir el nombre del mutex de proceso de PIVY durante su generación.

Aunque algunos ataques utilizan el mutex predeterminado “)!VoqA.I4”, la mayoría crea un mutex personalizado para cada ataque.

La ejecución del código del servidor poison IVY, puede ejecutarse sobre la víctima de varias formas, de acuerdo a la configuración del agresor, entre ellas, como código de inicialización y mantenimiento, para lo cual se inyecta sobre el proceso **explorer.exe** en ejecución y que

dependiendo la configuración del agresor, puede iniciar un proceso oculto en el navegador web predeterminado en el sistema, para dar paso a una fase de descarga remota del resto de código y datos que la herramienta necesita para desarrollar sus funciones. Este nuevo código se ejecuta en el endpoint de la víctima, almacenando todas las variables globales, procesos de configuración y punteros de funciones en una struct (estructura de datos) tipo C, que se inyecta en los procesos seleccionados, tanto en el código de conexión de red como en el de inicialización y mantenimiento de Poison IVY.

Como efecto secundario de esta característica distintiva, en el código desensamblado se hace referencia a todas las instrucciones CALL y direcciones de variables globales como desplazamientos a un registro. El código inyectado en explorer.exe es peculiar porque, a diferencia de la mayoría del código que se inyecta con malware, este se inyecta función por función, cada una con su propia región de memoria, e inserta los correspondientes punteros de funciones en su struct. Si se activa la opción "persistence" de PIVY, en explorer.exe también se inyecta un subproceso de vigilancia que reinicia automáticamente el proceso del servidor PIVY si el sistema operativo de la víctima lo finaliza inesperadamente. Si está activada, la función de registro de pulsaciones de PIVY también se inyecta en explorer.exe.

4.5.3. Protocolo de Comando y Control (C2) Poison IVY

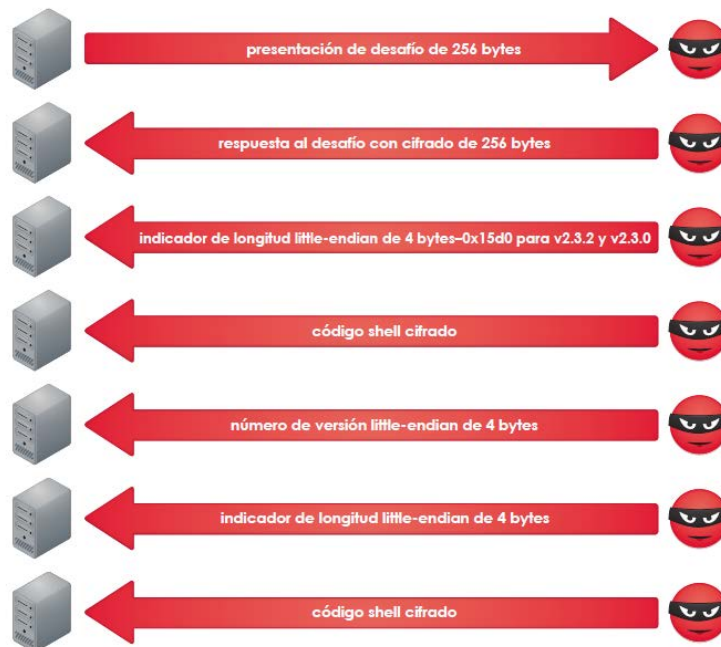


Figura 9. Protocolo de comunicación inicial de PIVY. (FireEye, 2013 p. 8)

Poison IVY maneja su propio protocolo de Red diseñado bajo TCP, tal como lo muestra la figura anterior: Cifra sus comunicaciones con Camelia y una clave de 256 bits⁵, para ello el agresor al generar el servidor Poison IVY proporciona una contraseña, la cual por lo general es “admin”, pudiendo ingresarla en texto normal o hexadecimal-ASCII, la cual se completa hasta alcanzar 32 bytes (256 bits). Esta clave es validada al iniciarse la sesión TCP empleando para ello un algoritmo de challenge-response, el servidor le responde al cliente enviando 256 bytes de datos generados aleatoriamente, para ello gran parte de estos datos se comprimen con el algoritmo LZNT1 de Microsoft mediante la API Rtl CompressBuffer de Windows; datos que son cifrados con la clave generada y los devuelve al servidor para su validación en bloques que contienen el siguiente encabezado:

```
struct PI_chunk_header {
int command_id;
int stream_id;
int padded_chunk_size;
int chunk_size;
int decompressed_chunk_size;
long total_stream_size;
int padding;
};
```

Tabla 1. Estructura de PI_chunk_header

Sobre esta estructura podemos encontrar como elementos:

- ✓ **command_id:** Elemento que identifica con qué función de PIVY está relacionado el bloque de datos.
- ✓ **stream_id:** Elemento que identifica a qué secuencia corresponde este flujo. El protocolo de PIVY permite enviar simultáneamente varias secuencias de datos.
- ✓ **padded_chunk_size:** dado que Camelia es un cifrado en bloques de 16 bytes, se utilizan bytes de relleno en los encabezados y los bloques de datos.

⁵ NTT en <https://info.isl.ntt.co.jp/crypt/eng/camellia/intro.html>

- ✓ **chunk_size:** los bloques se ensamblan en una secuencia de datos que puede ser cualquier cosa, como un archivo transferido, código shell para ejecutar, un archivo de mapa de bits con capturas de pantalla o datos crudos.

- ✓ **decompressed_chunk_size:** si este tamaño es diferente del indicado en chunk_size, el bloque se comprime con LZNT1.

- ✓ **total_stream_size:** este elemento especifica el tamaño total de los datos que se envían a command_id.

- ✓ **padding:** este miembro especifica el relleno con ceros (hasta 32 bytes).

5. Metodología Aplicable al Piloto Experimental

Como fuente inequívoca en el desarrollo del presente trabajo la Tesis Doctoral, Don Javier Bermejo (2015) nos permite introducirnos y desde una visión general conocer cuáles son las técnicas, métodos y herramienta de análisis de malware aplicadas, sobre la clara y expuesta complejidad del análisis de las APTs, debido a su constante evolución y falta de documentación. Creemos interesante presentar de manera resumida algunas definiciones y apartes expuestas en su documento:

- ✓ Métodos y técnicas análisis: consiste en el conjunto de herramientas y procedimientos que permitan obtener, organizar, describir y analizar los datos recogidos con los instrumentos de investigación, con el objetivo de obtener información útil acerca del objeto del estudio. En concreto en lo relativo al análisis de malware desde una revisión expuesta por Theerthagiri en su Tesis Doctoral (Theerthagiri, 2009) identifica las siguientes técnicas:
 - Técnicas de análisis dinámico o de comportamiento: Basada en el análisis binario del malware en ejecución, supervisando y monitorizando su interacción con del entorno de ataque, la cual incluye tanto la maquina afectada, así como también la interacción con sus servidores de mando y control al recibir órdenes, mediante el tráfico de red y comunicaciones. Utiliza herramientas de monitoreado del sistema de archivos, registros en el sistema Windows y del tráfico de red, materializado en la rapidez y precisión en el análisis.
 - Técnicas de análisis dinámico de código: Utiliza como herramienta básica de análisis los llamados “depuradores” que se asocian al código del programa objeto de análisis para tomar control del mismo. Estas herramienta permiten recorrer línea a línea el código ensamblador del malware, detallando cada uno de los registros de la CPU y contenidos de la memoria los cuales se actualizan al tiempo que se ejecuta paso a paso hasta los puntos de interrupción establecidos permitiendo examinar cada suceso.

- Técnicas de análisis estático de código: Mediante la aplicación de técnicas de desensamblado o ingeniería inversa, recoge la mayor cantidad de información posible sobre el binario sin necesidad de ejecutarlo. Utiliza herramientas como desensambladores, descompiladores, análisis de cadenas, descompresores, entre otros. Sin embargo puede requerir de un conocimiento más avanzado respecto a la arquitectura del hardware de ejecución: conjunto de instrucciones en ensamblador y el Formato del ejecutable del sistema operativo.

Dentro de estas técnicas de análisis, se puede contemplar el uso de métodos automatizados de análisis, para lo cual se utilizan sistemas llamados Sandbox que desde un ambiente controlado y supervisado, por lo general desde un entorno de máquina virtual.

- ✓ Herramientas de análisis: Son programas, aplicaciones, script o simplemente instrucciones usadas para permitir a un usuario efectuar estas tareas de un modo más sencillo. De acuerdo a la funcionalidad que proporcionan se pueden clasificar en:
 - Identificación y clasificación de binarios.
 - Motores de antivirus.
 - Análisis de cadenas.
 - Análisis de cambios en máquina host.
 - Análisis de tráfico de red y simulación de servicios de red.
 - Análisis de imágenes de disco.
 - Análisis de memoria.
 - Análisis de ficheros binarios.
 - Análisis dinámico de código.
 - Análisis estático de código.

- ✓ Metodología: hace referencia al conjunto de métodos y técnicas racionales utilizados para alcanzar una serie de objetivos que gobiernan una investigación científica. La aplicación de un proceso sistemático el cual armonice una serie de procedimientos,

el uso de métodos y técnicas de análisis, el uso y aplicación de herramientas respecto específicas en función de cada objetivo, sin lugar a dudas coadyuva al analista en obtener respuestas objetivas respecto al malware bajo estudio.

La Tesis Doctoral (Bermejo, 2015) identifica concretamente como metodología, el análisis de malware e ingeniería inversa. Sobre el particular esta metodología presentada se basa en el documento “Malware Analysis Reverse Engineering (MARE), Methodology & Malware Defense (M.D.) Timeline” (Timeline & Goldman, 2011), la cual permite aplicar desde un proceso estructurado de análisis, la aplicación de una serie de fases, desarrollando en cada una de ella los pasos, objetivos a obtener, técnicas y herramientas a utilizar; fases que son mencionadas a continuación:

- Detección
- Aislamiento y extracción
- Análisis de código
- Análisis de comportamiento

Esta metodología es usada como referencia principal por Don Javier Bermejo (2015) en su Tesis Doctoral, considerándola bastante completa y como la más avanzada entre las ya existentes al momento de desarrollar su investigación, y que le ha permitido como referencia, evaluar las ventajas de la nueva metodología que él propone; metodología como base aplicable en el desarrollo del presente piloto experimental.

5.1. Introducción a la metodología

La Tesis Doctoral Bermejo (2015), justifica en la necesidad de desarrollar una metodología más completa, que garantice las bases necesarias para la realización de un proceso sistemático, flexible (debido a la cantidad de códigos maliciosos existentes y sus tecnologías) y repetible de análisis del malware que proporcione una información completa del mismo, aplicable para contrarrestar el malware que ataque a sistemas Windows. Sin un

enfoque estructurado y detallado respecto al desarrollo de los procesos, la utilización de herramientas y el orden lógico en su aplicación se reflejarían negativamente en el nivel de eficacia del análisis.

La innovación de la metodología propuesta sin lugar a dudas, es la modificación del orden en el desarrollo de las tareas que hacen parte de la aplicación de la fase de análisis de código, para ello parte desde el análisis estático el cual es seguido por un análisis dinámico, contemplando la posibilidad de generar ciclos iterativos derivados del malware y su comportamiento al ser ejecutado, permitiendo desde la fase de análisis de comportamiento volver a la fase de análisis de código. Esta metodología de análisis se centra básicamente en archivos ejecutables y librerías, pudiendo ser aplicable a otros sistemas operativos, como, Linux, Android, siempre y cuando las herramientas diseñadas en estas plataformas ofrezcan capacidades similares a las propuestas en la metodología.

Una visión general de la metodología sobre su estructura, permite descubrir cuatro fases a desarrollarse dentro de un proceso sistemático de aplicación en el análisis del malware, y tal como se mencionó su diseño es flexible, pudiendo ser aplicable a los distintos tipos de códigos existentes, fases relacionadas a continuación:

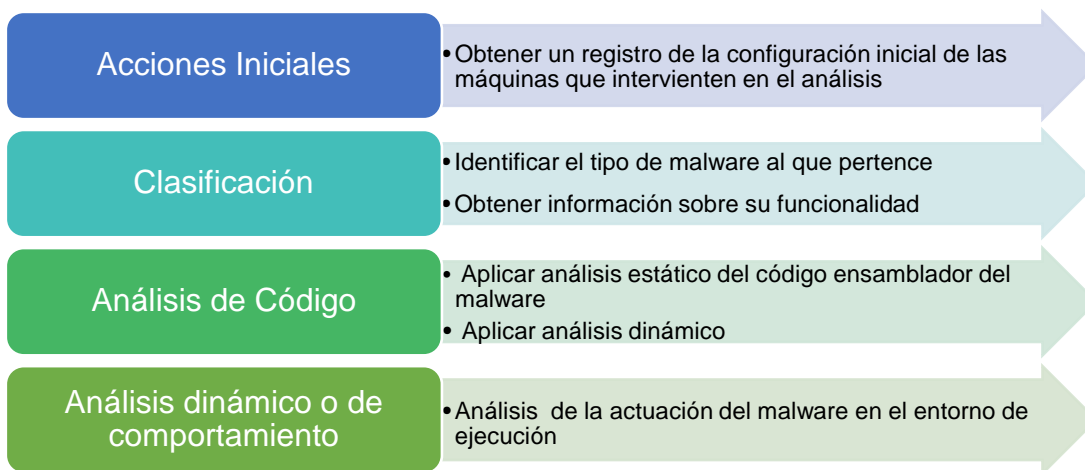


Figura 10. "Sistema de Análisis e Ingeniería Inversa de Código Malicioso" (Bermejo, 2015)

5.2. Descripción de la metodología

Al ser la metodología desarrollada por Don Javier Bermejo la base de aplicación del presente piloto experimental, sobre este apartado, se presentan los aspectos más relevantes de su Tesis Doctoral (Bermejo, 2015).

5.2.1. Acciones Iniciales

Su fin es desarrollar una serie de acciones enfocadas a comprobar la integridad del entorno del análisis, como punto de partida que permita desde un estado inicial obtener los registros de configuración, obteniendo elementos que sirvan de comparación después de ejecutar el análisis global del malware bajo estudio.

De esta forma se parte desde un escenario libre de malware, para ello, deberá garantizarse la ejecución de procedimientos que permitan revertir los procesos adelantados, pudiendo devolver el estado de las máquinas objeto de estudio, al inicio de cada análisis.

Las actividades que se refieren en esta fase son:

- ✓ Realizar una línea de base de la configuración del sistema víctima (foto instantánea) después de instalar todas las herramientas de análisis, con objeto de tener una referencia con la que comparar después de haber realizado procesos de análisis de malware. Se propone la utilización de Sysstracer y WinMD5, y el HASH MD5 archivos ejecutables binarios (sys, dll, ocx, scr, exe, mui, cpl, TLB, etc.), en razón a que muchos de los archivos se modifican durante una ejecución normal del sistema operativo.
- ✓ Si se está trabajado en un escenario virtual, deberá tomarse una instantánea inicial “snapshot”, o bien realizar una imagen si estamos en el entorno físico.
- ✓ En la etapa inicial, al configurar el sistema operativo, desactivar 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: Aplicar las herramientas WinMD5, para comprobar con ellas que se mantiene la integridad del fichero de la línea base de referencia, calculando solo los hashes en archivos ejecutables binarios (sys, dll, ocx, scr, exe, mui, cpl, TLB, etc.) y comprobar que no se ha instalado un malware del tipo rootkit utilizando herramientas como Gmer y rootkitrevealer. Si se descubre la inexistencia de coincidencias de las Línea Base o se detecta algún rootkit instalado, deberá restaurarse el sistema al estado original.
- ✓ Tomar otra instantánea con la herramienta Systracer y compararla con la de referencia, para comprobar si se han realizado cambios en las entradas del registro y/o sistema de ficheros. Si se descubre la inexistencia de coincidencias de las Línea Base o se detecta algún rootkit instalado, deberá restaurarse el sistema al estado original.
- ✓ Mediante la herramienta Vmnetstiffer 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.

5.2.2. Clasificación

Tiene por objeto examinar el archivo ejecutable del malware sin acceder al código malicioso, a fin de obtener información inicial que permitan la aplicación de las siguientes fases. Para ello, esta fase propone la ejecución de las siguientes actividades:

- ✓ Transferencia del malware: Implica trasladar el malware a la máquina virtual o al entorno físico donde se esté desarrollando el análisis.
- ✓ Identificación del malware: Obtener la identificación del malware mediante la aplicación del hash (MD5 o SHA1), utilizando para ello las herramientas md5deep o WinMD5.
- ✓ Clasificación del Malware según su tipo: Actividad que tiene por objeto escanear el malware con distintos filtros antivirus y verificar si ya se encuentra identificado.

Por la versatilidad en las técnicas de análisis de los antivirus, se recomienda ejecutar varios programas de antivirus diferentes sobre el mismo malware, entre ellos: ClamAV, Bitdefender, Antivir, Panda, AVG, F-Prot, o utilizar recursos on-line como: VirusTotal, Anubis, Norman SandBox, Cwsandbox, VirScan.

Una vez identificado, se recomienda clasificar las muestras del malware, o las variantes del mismo, mediante descripciones de malware. Para ello se puede hacer uso de la herramienta YARA, que permite en la descripción, obtener sobre el malware un conjunto de cadenas y una expresión booleana que determina su lógica.

- ✓ Búsqueda de información en fuentes abiertas (OSINT): OSINT por sus siglas Open Source Intelligence, permite obtener datos de distintas fuentes públicas disponibles en Internet o de libre acceso, que permitiría obtener toda la información conocida respecto al malware, en su evolución y clasificación.
- ✓ Búsqueda de cadenas de texto. Una vez aplicado el software antivirus, e identificada la información del mismo existente en fuentes abiertas, se ejecuta un aplicativo de búsqueda de cadenas ANSI y UNICODE sobre ficheros binario o archivos como scripts de Shell, para ello una de las herramientas sugeridas para desarrollar esta tarea es “Strings”. Como resultado adicional, se puede encontrar información detallada y útil sobre protocolos, puertos, comandos de mando y control, archivos, direcciones IP, e incluso información sobre el propio malware.
- ✓ Identificación de técnicas de ofuscación. Es común ver como los fabricantes de malware emplean la combinación de técnicas que permiten ocultar su ejecución y funcionamiento, con el fin de disminuir considerablemente la efectividad de los sistemas de protección como antivirus y detección de intrusos (IDS), entre ellas: empaquetamiento, cifrado, polimorfismo y metamorfismo. Si tras el análisis del código ejecutable del malware se detectan la presencia de cadenas de texto, puede ser indicio de que existe una técnica de ofuscación, ahora si el programa esta empaquetado y comprimido, se puede detectar un pequeño código de inicio

denominado envoltorio que descomprime el archivo empaquetado y carga en memoria el código original, utilizando una herramienta de empaquetado como PEiD que detecta el tipo de compresión para los ejecutables. Otro método para detectar técnicas de ofuscación sobre el empaquetamiento del malware, es editar su hexadecimal con PEBrowse.

- ✓ Formato de estructura del fichero: Analizar el formato y estructura del archivo que contiene el malware puede revelar mucho respecto a su funcionalidad descubriendo información como: la dirección base del archivo, la dirección del punto de entrada y el análisis de tabla de la sección, tipo de aplicación, librerías necesarias y los requisitos de espacio, utilizando herramientas como Dependency Walker y PEBrowse.

5.2.3. Análisis de Código

La aplicación de esta fase implica del análisis de código en formato ensamblador del malware con el fin de comprender a profundidad su funcionamiento, para ello, tras la aplicación de ingeniería inversa sobre el ejecutable se observan las instrucciones del programa en sus condiciones normales e inusuales. Se proponen distintas herramientas recomendadas como son “PE Explorer”, “IDA Pro” y “Ollydbg”.

5.2.4. Análisis Dinámico o de Comportamiento

Esta fase centra en procesos de observación adelantados sobre el sistema víctima una vez se haya ejecutado el malware, con el fin de obtener resultados respecto al comportamiento y cambios que pueden ocurrir. En su Tesis Doctoral Don Javier Bermejo (2015), refiere que si el análisis del comportamiento es profundo y relativamente rápido, es necesario contar con toda la información obtenida en el análisis de código que permite identificar y conocer la forma de actuar del malware. Recomienda que la ejecución de esta fase debe ser progresiva, agregando poco a poco servicios en el entorno de ejecución, ya que si se agregan demasiadas funciones el malware puede realizar muchas acciones nuevas, ampliando la extensión del análisis e incluso perdiendo algunas evidencias. Para ello propone desarrollar:

- ✓ Tareas previas a la ejecución. En este paso se realizarán las tareas necesarias antes de ejecutar el malware.
- ✓ Ejecutar malware. Aquí VMware juega un papel importante, ya que ofrece utilidades de línea de comandos que se pueden utilizar para ejecutar un programa, como el malware, aprovechando los privilegios de cualquier usuario de la máquina a la que se va a transferir. Ahora bien, si el entorno de trabajo es una máquina física, recomienda el uso de PsExec.
- ✓ Proporcionar servicios al malware. Como ya se mencionó 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 (en las herramientas que lo permitan) 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 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 está trabajando con los sistemas físicos, este paso implica la memoria 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 Volatily 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 residente 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 infección de algún rootkits.

6. Escenario de Aplicación

Antes de desarrollar desde un escenario práctico la metodología presentada en el apartado anterior, debemos definir antes el tipo de investigación. Tomando como fuente el documento de S. Coryn "The fundamental characteristics of research" (Coryn, 2006), expuesto por Don Javier Bermejo (2015) en su Tesis Doctoral, el cual realiza una clasificación de los tipos de investigación existentes en la actualidad, nuestro trabajo estaría enmarcado dentro del tipo de "Desarrollo experimental", toda vez que según la definición expuesta, a este grupo corresponden todos los trabajos sistemáticos basados en conocimientos existentes y aprobados, derivados de la investigación y/o la experiencia práctica, dirigidos a la producción de nuevos materiales, productos, dispositivos, procesos, sistemas y servicios, o mejora sobre los publicados.

De esta definición, encontramos que nuestro piloto experimental cumple dos de las características presentadas, ya que está basado en un trabajo sistemático ya existente al relacionar como fuente de conocimiento la Tesis Doctoral y permite desde la experiencia práctica aportando con un caso de estudio adicional, el descubrir los beneficios y aplicabilidad de la metodología desarrollada por Don Javier Bermejo (2015).

6.1. Definición del Entorno o Escenario

El escenario de aplicación debe ser un entorno controlado, que permita desarrollar de modo seguro cada una de las fases expuestas en la metodología, sin afectar entornos de trabajo reales y que a su vez, los resultados de las pruebas aplicadas permitan aproximarse a esa realidad, así como también que el desarrollo del piloto experimental, se encuentre bajo la normatividad legal vigente.

Para alcanzar este fin, nuestro escenario desde su diseño deberá contar con todos los elementos necesarios que permitan simular un entorno real de aplicación, contando con máquinas, aplicativos, canales de comunicación y herramientas que permitan monitorear cada una de los procesos aplicados y obtener así las evidencias respectivas.

Para el diseño del laboratorio, se toma como referencia el diseño expuesto en el documento (Sanabria, 2007) , respecto a la construcción de un laboratorio para análisis de malware mediante la utilización de dos entornos, uno de hardware físico y otro virtual, siendo esta segunda opción la más acorde a nuestro trabajo, debido a la facilidad, funcionalidad y administración de recursos dada la posibilidad de ejecutar en un ordenador las diferentes máquinas del laboratorio en una plataforma virtual.

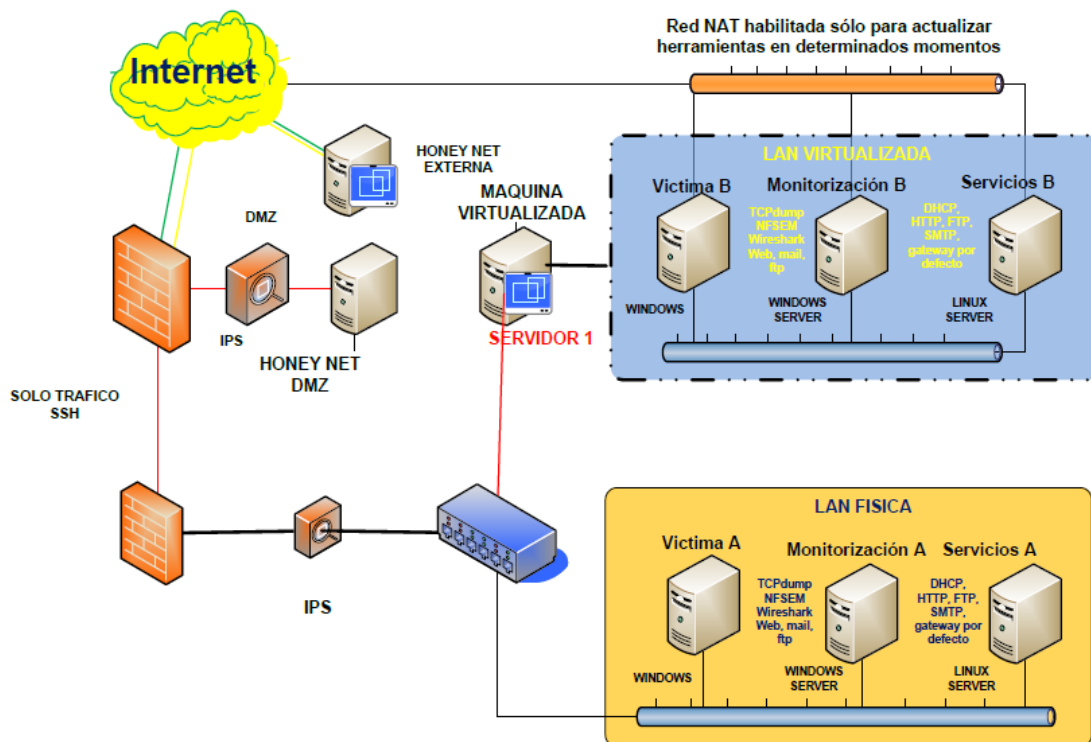


Figura 11. Arquitectura laboratorio de Análisis malware ([1] p. 162).

La aplicación del escenario expuesto en la figura 11, desde el punto de vista presentado por Don Javier Bermejo (2015) en su Tesis Doctoral refiere de los siguientes componentes:

- **Víctima:** Con el fin de crear un escenario o plataforma donde se va a ejecutar el malware, el cual permita desarrollar un proceso de seguimiento y monitorización frente a su comportamiento, que permitiría desarrollar un Análisis dinámico.

- **Monitorización y Servicios Windows:** El cual permita realizar la monitorización del tráfico de la red generado por el malware, proporcione los diferentes servicios de sistemas Windows, y de acuerdo a la metodología la aplicación de las fases de clasificación y análisis estático y dinámico de código.
- **Servicios:** Su objetivo es montar un servidor que permita proporcionar servicios al malware en su interacción con el entorno, entre ellos HTTP, DHCP, Chat, IRC Server, FTP, DNS y SMTP, que simulen su entorno de ejecución.

Con el fin de cumplir con el diseño anterior, se tomó la decisión de seleccionar un entorno virtual basado en la herramienta “VMWare”, en relación a las características que esta herramienta brinda respecto a los recursos tanto físicos como lógicos, su versatilidad y facilidad en la articulación de los sistemas; escenario en el cual se definieron tres subsistemas basados en entornos de plataformas Windows y un tercer subsistema basado en sistemas Linux, los cuales cumplirán los siguientes roles y bajo los siguientes parámetros:

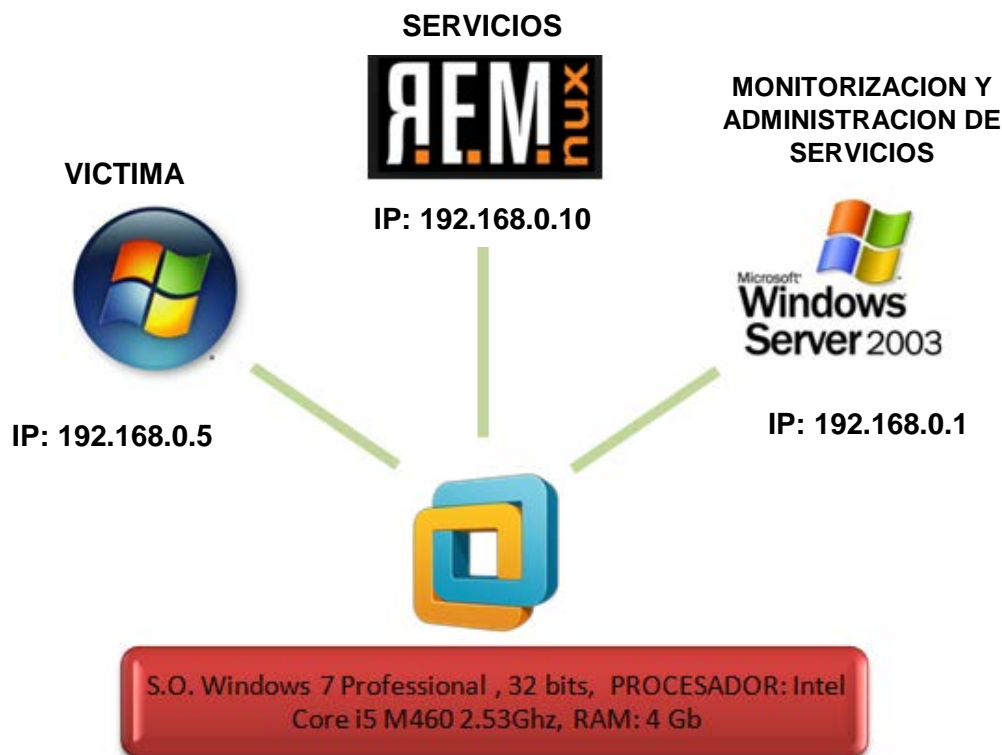


Figura 12. Escenario y Subsistemas del laboratorio.

De esta forma como resultado al montaje de los subsistemas mencionados es:

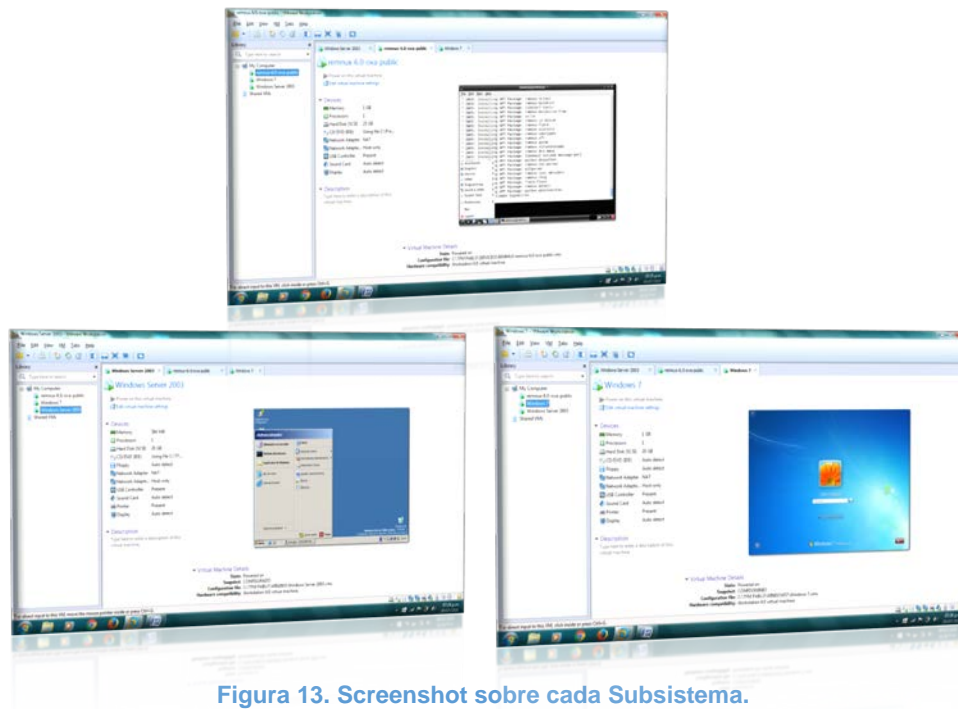


Figura 13. Screenshot sobre cada Subsistema.

6.2. Definición de herramientas específicas del laboratorio

Definidos los componentes del laboratorio, a continuación se presentan las herramientas que desde la sugerencia presentada en la Tesis Doctoral (Bermejo, 2015), permitirán la ejecución de los procesos presentados en la metodología base del presente piloto experimental.

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		
VMMMap		
Systracer		
Resource Hacker		
PEViewer		
HexView		
DiskPulse		
GMER		

Figura 14. Herramientas análisis de malware relacionadas con la máquinas del laboratorio. ([2], 2015, p. 169)

Sobre el listado presentado en la figura anterior respecto al escenario de aplicación particular frente al análisis del malware Poison Ivy, la elección de las herramientas presentadas a continuación, se basa en la buena administración de los recursos y el manejo respecto al tema del licenciamiento tomando las opciones de freeware o “triales”, las cuales fueron probadas y ofrecen todos los resultados necesarios para llevar a cabo.



Figura 15. Definición de Herramientas para Laboratorio.

Una vez construido el entorno virtual, en el cual se instaló cada uno de los sistemas operativos de las diferentes máquinas de laboratorio, se capturó un “snapshot” tomando una

instantánea de cada sistema dejando el respectivo registro inicial como un punto de partida, en caso de producirse eventualidades no programadas y sea necesario restaurar los procesos.

Al respecto de las herramientas expuestas en la figura 14, fueron descargadas previamente desde un ambiente diferente al escenario virtual y fueron exportadas a un dispositivo de almacenamiento óptico digital, previo cálculo de su hash, que garantizará la no modificación de las mismas al ser introducidas e instaladas en cada subsistema virtual; recomendación presentada en la Tesis Doctoral (Bermejo, 2015), en donde adicionalmente se expone la recomendación respecto a la instalación del software de protección de tipo antivirus: antispysware, como, Spy Bot y detección de intrusiones como Zone Alarm, OSSEC, sobre cada máquina incluida en el soporte del escenario virtual.

7. Aplicación de la metodología propuesta al malware Poison IVY.

En este apartado se expone la aplicación práctica de la metodología propuesta por Don Javier Bermejo (2015) en su Tesis Doctoral, con el desarrollo de cada una de las Fases, tomando como caso de estudio el análisis del malware Poison Ivy, tal como se ha expuesto a lo largo del presente documento.

7.1. Fase 1: Acciones Iniciales:

Esta fase tiene por objeto el definir un escenario particular para la aplicación de la metodología, que garantice la integridad de los procesos y su repetición si es necesario sin afectar la cronología y seguimiento de los resultados obtenidos.

Preparación de los Equipos: Tal como se presentó con anterioridad, se desarrolló la instalación de cada uno de los subsistemas y sobre ellos las herramientas seleccionadas para la aplicación de la metodología. Es importante recalcar que en esta etapa fue necesario desactivar las herramientas de actualización particularmente sobre los sistemas Windows, a diferencia de Remnux (el cual una vez instalado se corrió sobre el mismo su actualización), debido a que los parches actuales podrían afectar considerablemente los resultados de la aplicación del malware.

Desarrollo de una línea base: Al ser nuestro ambiente de práctica un escenario virtual creado bajo VMware, utilizando la herramienta “snapshot” sobre cada uno de los subsistemas virtuales, entre ellos: víctima, monitorización y servicios, se crearon apartes que reflejan las etapas de instalación y configuración. De igual forma, utilizando la herramienta Systracer sobre las unidades de almacenamientos principales donde se instalaron las máquinas virtuales, se tomaron imágenes a fin de garantizar su integridad, de las cuales con la herramienta **Md5summ** se calcularon sus respectivos Hash MD5 y posterior comprobación con la herramienta **Winmd5**.

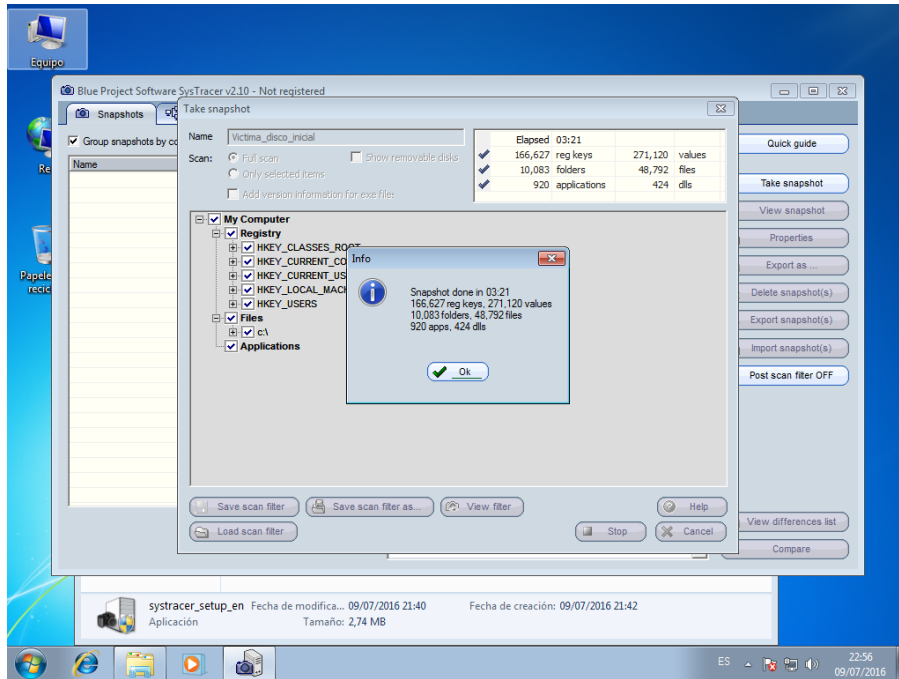


Figura 16. Aplicación de SysTracer y generación de snapshot sobre el sistema víctima.

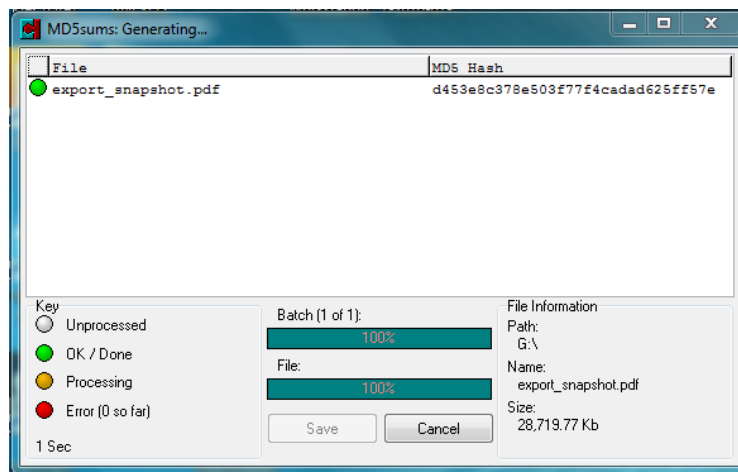


Figura 17. Aplicación de MD5sums, sobre snapshot del sistema víctima.

La aplicación de estos procesos permitirá desarrollar un flujo efectivo de las fases y su control frente a la ejecución del malware sin que este afecte cada momento, teniendo la opción de volver a un estado si es necesario y repetir de la forma más transparente cada actividad.

7.2. Fase 2: Clasificación

Como se precisó anteriormente en el apartado respecto a la descripción de la metodología, esta fase tiene por objeto analizar el malware sin necesidad de ejecutarlo, aplicando siete (7) actividades aplicadas sobre la muestra de Poison Ivy obtenida.

Cabe resaltar que el sitio web oficial⁶ donde por mucho tiempo se expuso la herramienta, sus actualizaciones, soportes y demás documentación, en la actualidad se encuentra fuera de la red, razón por la cual hubo la necesidad de buscar una muestra del malware en otras fuentes de consulta (Desconocido, 2010).

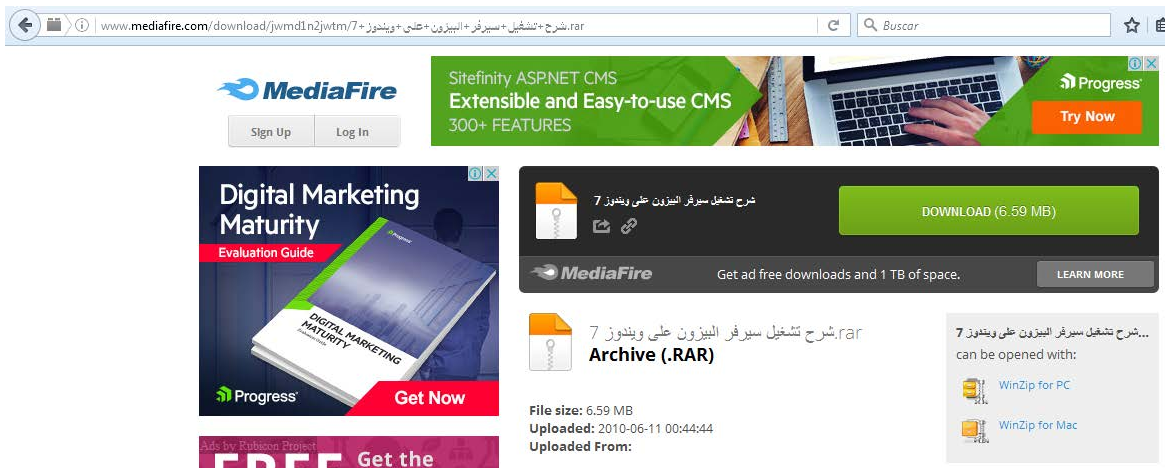


Figura 18. Sitio web MediaFire - fuente de descarga "Poison Ivy".

Como resultado se obtuvo la herramienta Poison Ivy 2.3.0 y un parche específico para sistemas Windows 7, en razón a la construcción de nuestro laboratorio.

7.2.1. Transferencia de Malware

Tomando una unidad de almacenamiento tipo USB, se exporta la muestra del malware y su parche, a los subsistemas que hacen parte del laboratorio, archivos que se encontraban previamente comprimidos.

⁶ www.piosonivy-rat.com

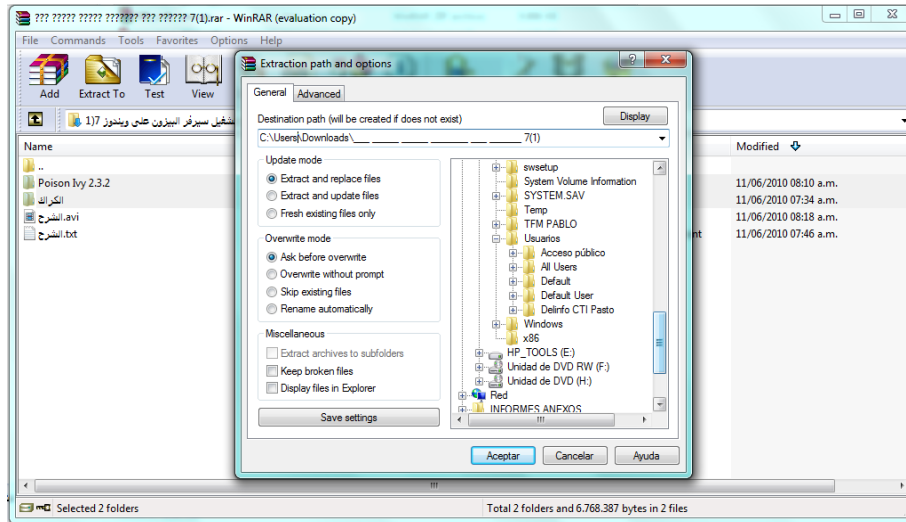


Figura 19. Transferencia del malware a un dispositivo USB.

7.2.2. Identificación del malware

Como resultado dentro del comprimido, encontramos un archivo denominado **Poison Ivy 2.3.2.exe**, al cual mediante la herramienta **Md5sums** se le calculó su **hash md5** obteniendo como resultado: **b4f990cad1d20efab410e98fc7a6c81b**, muestra que refiere un archivo de tipo PE32 ejecutable (GUI) Intel 80386 para MS Windows, con un tamaño de 2141878 bytes, un archivo denominado **Reko24.exe**, el cual su hash md5 refiere: **22577c60a76c77adb433a3406e6a7c56**. Esta información será utilizada posteriormente en el análisis.

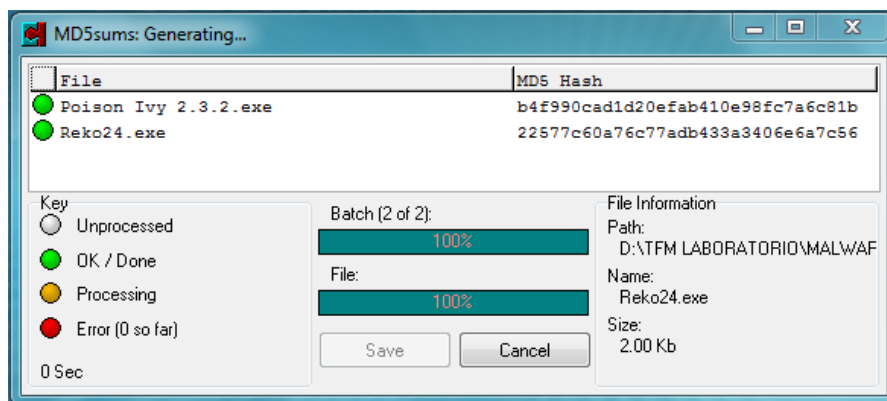


Figura 20. Resultado MD5sums sobre la muestra del malware obtenida.

Para el caso particular de Poison Ivy, tal como se precisó en apartados anteriores, el archivo principal se trata de un kit que funciona en ambiente cliente/servidor, el cual una vez ejecutado y configurado genera un archivo con una serie de parámetros establecidos por el atacante, el cual finalmente será introducido sobre la víctima.

Una vez trasladado sobre nuestro laboratorio, se procedió a su ejecución y configuración utilizando la opción servidor, definiendo una serie de parámetros particulares para nuestro escenario.

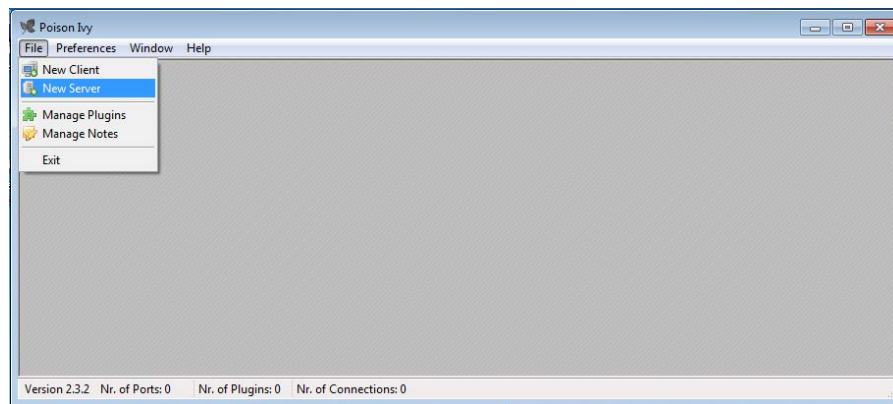
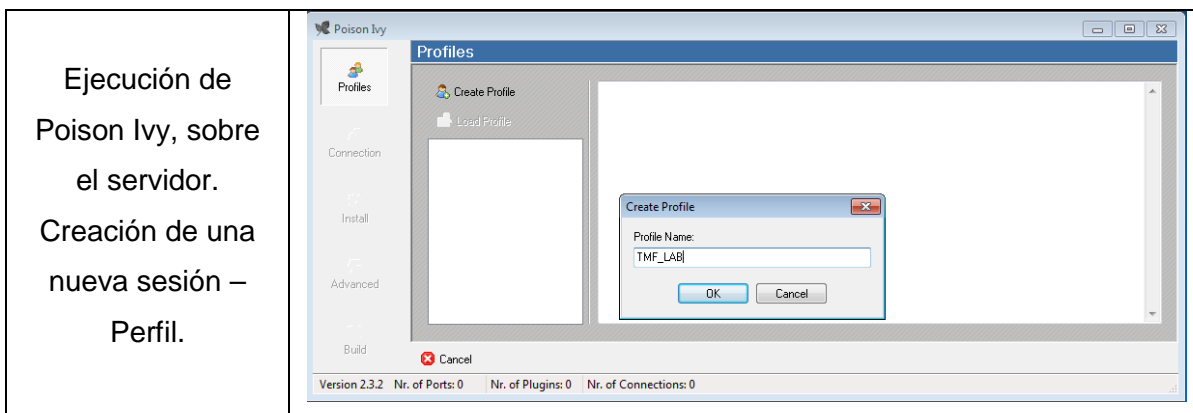


Figura 21. Ejecución Poison Ivy, pantalla principal.

El objeto del presente piloto experimental, NO es documentar la instalación y uso de Poison Ivy, sin embargo se presentan algunas pantallas frente a la configuración y creación del malware, que fueron aplicadas en el desarrollo del presente piloto experimental.



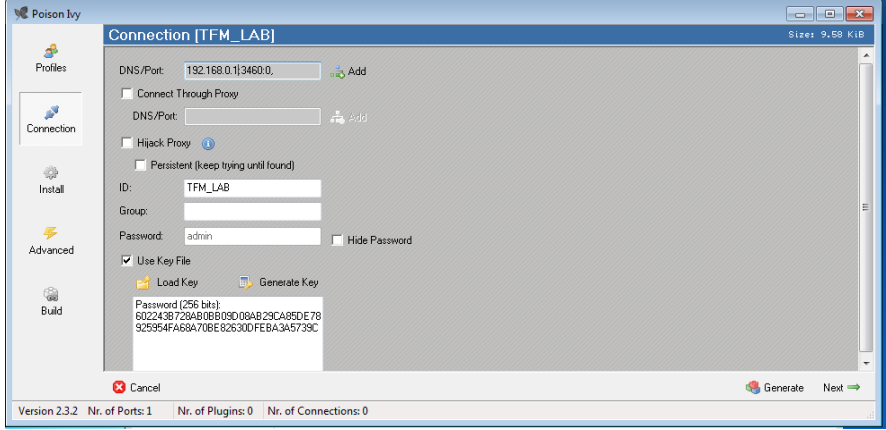
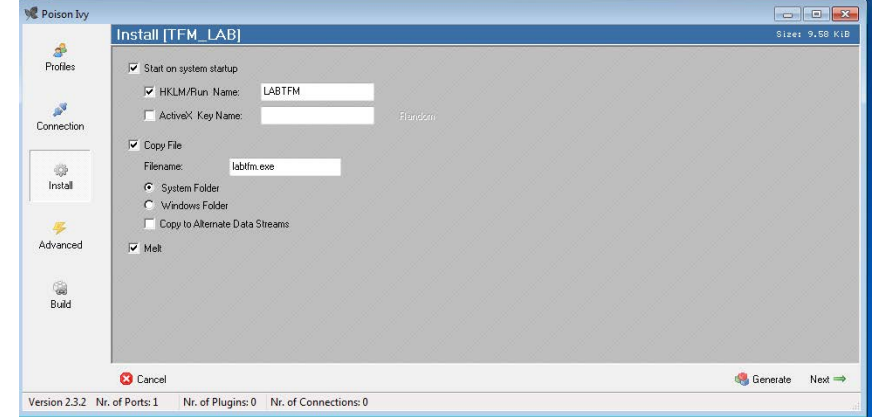
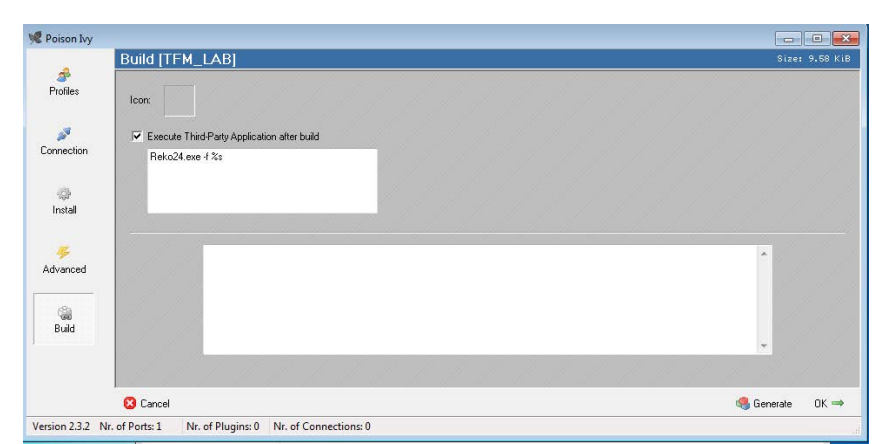
<p>Ejecución de Poison Ivy, sobre el servidor. Definición IP de la máquina servidor, Nombre del ID, Password.</p>	
<p>Ejecución de Poison Ivy, sobre el servidor. Definición del nombre respecto al proceso y el archivo que se ejecutará sobre la víctima.</p>	
<p>Ejecución de Poison Ivy, sobre el servidor. Aplicación del parche "Reko24.exe", para hacer el malware funcional sobre ambiente Windows 7.</p>	

Tabla 2. Poison Ivy - configuración.

Como resultado a la ejecución de “Poison Ivy” y su configuración, se generó un archivo denominado “**Hack Facebook.exe**”, identificado mediante la herramienta **MD5summ**, bajo el **hash md5 : de1cbe2d617a75f9c768f3a1df3e6aa7**.

7.2.3. Comprobación del tipo del malware

Con el fin de analizar los archivos correspondientes a la muestra del malware Poison Ivy, así como también el archivo generado con destino a la víctima, se utilizaron herramientas de detección online y aplicaciones instalables.

- ✓ **Virus Total**⁷ : Como la herramienta de detección más completa, la cual presenta un listado de herramientas antivirus que reconocen el malware, así como los detalles del archivo y su relación con otros elementos e información adicional. En su orden:
 - Sobre la muestra del malware obtenida arrojó los siguientes resultados:

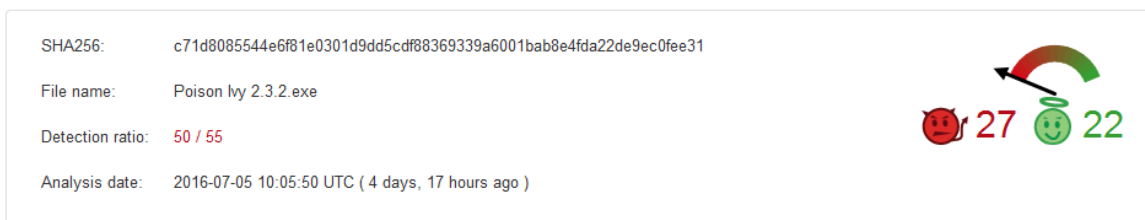


Figura 22. Identificación muestra obtenida del malware archivo “Poison Ivy 2.3.2.exe”, “Virus Total”.

En relación a los resultados obtenidos en el informe presentado por “Virus Total”, en la siguiente figura se presentan las herramientas más importantes de detección, la información completa es relacionada en el Anexo A.

⁷ <https://www.virustotal.com/>

ANTIVIRUS	DETECCIÓN	FECHA REVISIÓN
Ad-Aware	GenPack:Backdoor.Generic.81276	20160705
Avast	Win32:Evo-gen [Susp]	20160705
AVG	Win32/Heur	20160705
Avira (no cloud)	BDC/PoisonIvy.A	20160705
BitDefender	GenPack:Backdoor.Generic.81276	20160705
ESET-NOD32	Win32/RemoteAdmin.PoisonIvy potentially unsafe	20160705
Kaspersky	Backdoor.Win32.Poison.cww	20160705
Kingsoft	Win32.Hack.Poison.2199552.(kcloud)	20160705
McAfee	BackDoor-DIQ	20160705
Microsoft	Backdoor:Win32/Poison.E	20160705
Panda	Application/Poisonivy	20160704
Symantec	Backdoor.ConstructKit	20160630
TrendMicro	BKDR_POISON.BUR	20160705

Tabla 3. Resultado herramientas de detección según “Virus Total” sobre la muestra.

Frente a los resultados obtenidos, podemos encontrar que no todas las herramientas la detectan con el identificador Poison Ivy.

- Sobre el archivo generado “Hack Facebook.exe”, se obtuvieron los siguientes resultados:



SHA256: ff2d15569d8a8eeb93681a543b781578db9093f927659a14d43f01dbd39acba0

File name: Hack Facebook.exe

Detection ratio: 54 / 55

Analysis date: 2016-07-10 21:09:36 UTC (1 minute ago)

Figura 23. Identificación muestra obtenida del malware archivo “Hack Facebook”, “Virus Total”.

Con relación a las herramientas relacionadas por “Virus Total”, a continuación se relacionan los resultados más relevantes, mientras que el resultado en su totalidad es presentado en el Anexo B.

ANTIVIRUS	DETECCIÓN	FECHA REVISIÓN
AVG	Win32/Agent.BB	20160710
Ad-Aware	Generic.PoisonIvy.F2AADF09	20160710
Avast	Win32:Agent-AAGI [Trj]	20160710
Avira (no cloud)	TR/Crypt.XPACK.Gen	20160710
ESET-NOD32	Win32/Poison.NAE	20160710
Kaspersky	Backdoor.Win32.Poison.aec	20160710
McAfee	BackDoor-DSS.gen.a	20160710
Microsoft	Backdoor:Win32/Poison.E	20160710
Panda	Bck/Poison.E	20160710
Symantec	Trojan!gm	20160710
TrendMicro	BKDR_POISON.DS	20160710

Tabla 4. Resultado herramientas de detección según “Virus Total” - “Hack Facebook.exe”

- ✓ En contraste con la aplicación de una herramienta de instalación **Avira**, la cual presenta los siguientes resultados:
 - Sobre la muestra obtenida se obtuvo como resultado: La detección respecto a **BDC/PoisonIvy.A**

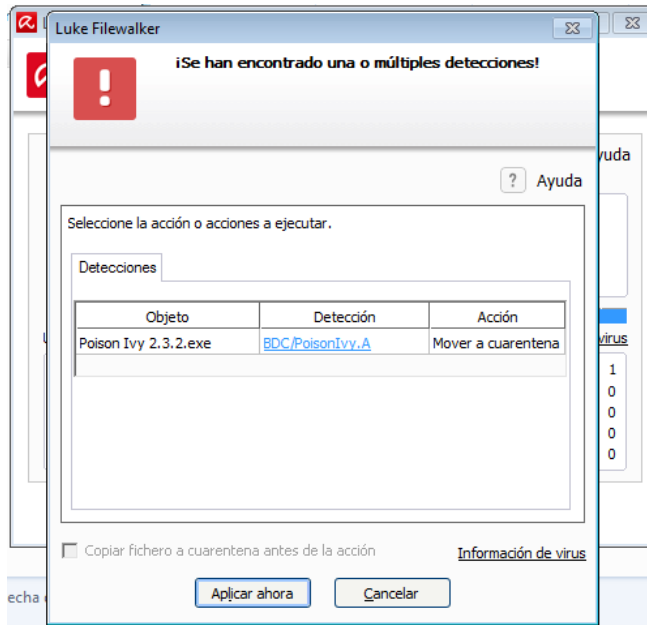


Figura 24. Resultado de Avira, sobre la muestra "Poison Ivy 2.3.2.exe"

Mientras que sobre el archivo "Hack Facebook.exe" generado con destino a la víctima, fue detectado como: **TR/Crypt.XPACK.Gen.**

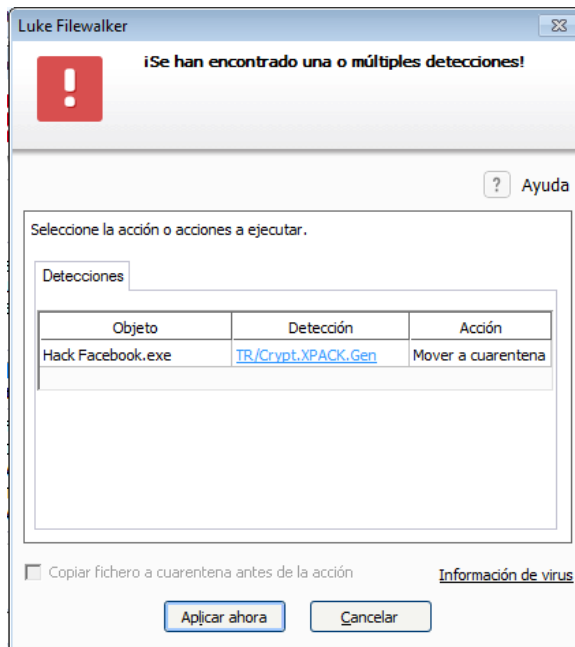


Figura 25. Resultado Avira sobre "Hack Facebook.exe".

7.2.4. Información obtenida por fuentes abiertas

Una vez identificado el malware el cual está relacionado con el APT Poison Ivy, enmarcado para su aplicación sobre la plataforma Windows en donde se desarrollaron los procesos que documentan cada una de las Fases en la aplicación de la metodología, tomando sobre los resultados obtenidos de las fuentes de consulta abierta la publicación Threat Report: Poison Ivy, presentada por Microsoft Malware Protection Center (Saade, Kurc, & Stewart, 2011).

Dicho documento refiere que Poison Ivy se dio a conocer en el segundo semestre del 2006 con el lanzamiento de la versión 2.1.1, la cual no solo era una versión de un malware moderno, si no que su desarrollador proveía de documentación, soporte y facilitaba actualizaciones y nuevas características que eran agregadas al RAT, que permitieron sacar al mercado nuevas versiones.

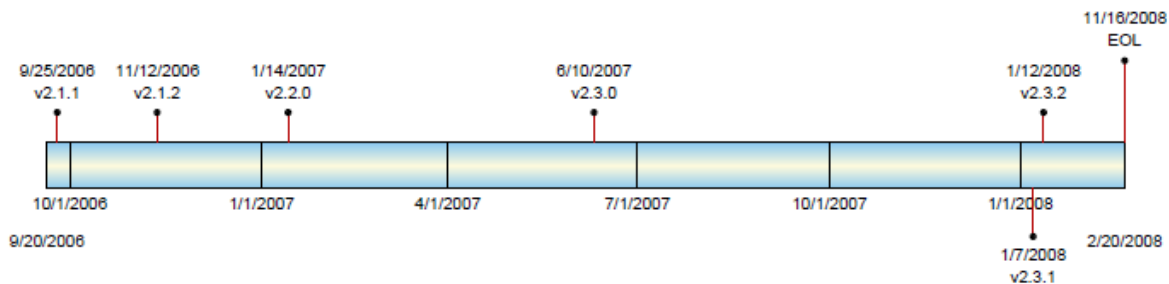


Figura 26. Poison Ivy RAT release timeline, ([23], página 11).

Cronológicamente la figura anterior refiere los siguientes lanzamientos:

- ✓ v2.1.2 presentada el 11/12/2006
- ✓ v2.2.0 presentada el 01/14/2007
- ✓ v2.3.0 presentada el 06/10/2007
- ✓ v2.3.1 presentada el 01/07/2008
- ✓ v2.3.2 presentada el 01/12/2008

Durante el ciclo de vida de Poison Ivy se incluyeron algunas características especiales: La versión 2.3.0 incluyó un plugin con el cual el autor extendió la funcionalidad del malware

creando módulos que se ejecutaban sobre las maquinas comprometidas, entre ellas: escaneo de puertos, escaneo de wifi, y generador de screenshots , la versión 2.2.0 agregó el sistema de cifrado en implementación de 256 bit sobre el algoritmo camellia y finalmente en el año 2008, su autor detuvo el desarrollo de Poison Ivy, aunque pudimos comprobar que aún se encuentran muestras en la red ofrecidas en diferentes sitios y blogs dedicados al tema del Hacking, ofreciendo nuevos plugins que ofrecen su ejecución en plataformas Windows 7 y Windows 8.

De dicho informe es rescatable la clasificación de Poison Ivy dentro de la familia de malware WIN32/Poison, el cual lo define como un Troyano de Acceso Remoto (RAT) que es distribuido y comercializado como un KIT, el cual ha sido detectado como el Hacktool: Win32/poison, así como los binarios que este malware genera clasificándolos como un prefijo Backdoor: Win32/Poison.

Microsoft y sus tecnologías antimalware han encontrado las siguientes firmas que identifican al malware, entre ellas:

- ✓ Backdoor: Win32/Poison
- ✓ Backdoor: Win32/Poison.M
- ✓ Backdoor: Win32/Poison.AQ
- ✓ Backdoor:Win32/Poison.BC
- ✓ HackTool:Win32/Injectxin

En sus diferentes versiones de acuerdo al historial detectado, se determinaron las siguientes firmas:

Version	SHA1
1.0.0	9dcf636d25eff3f080e0d5052ff780c51d736d2e
2.0.0	023da2618f2fde05a8af6f6a26abc1a664fade3d
2.1.0	15a68b4732395b4005c1b6939a64bf0e79538fabj
2.1.1	bce583541cd0d36793965411ee79b796a22c56f9
2.1.2	68a4257381f1dd767698b217fa238f9a116be118
2.1.4	5cab1aa4a122c02661ad10d46ff5f4c6a15cc6d7
2.2.0	e46cbf9f8ebb7da65ff17a678549f874db39ae0b
2.3.0	19862253caacadd621aaa74b78b334c01f4f346c
2.3.2	45419eb6d058766bb4d134bd567bc9ea02ba38b2

Figura 27. SHA1 Hashes for Historic Poison Ivy Builders, ([23], página 8).

Respecto a las vulnerabilidades que el malware explota se encuentran:

- ✓ CVE-2009-4324
- ✓ CVE-2009-3129
- ✓ CVE-2010-2883
- ✓ CVE-2010-1297
- ✓ CVE-2010-3333
- ✓ CVE-2011-2110
- ✓ CVE-2011-0611
- ✓ CVE-2011-0609

Poison Ivy usa una arquitectura cliente/servidor, utiliza la técnica conocida como reverse Shell o "Server" que una vez instala en la máquina infectada, es la encargada de realizar la conexión al panel de control para recibir las órdenes, a diferencia de los malware tradicionales que lo hacían al revés. Se instala sobre la víctima un servidor el cual es ensamblado, generado y posteriormente controlado remotamente desde un cliente desde donde se desarrollan una serie de configuraciones.

La última versión conocida de Poison Ivy es la 2.3.2 la cual fue lanzada en el año 2008, la cual ofrecía entre sus características: comunicación cifrada, navegación remota de archivos, procesos de inyección, manipulación de registros, capturas de pantalla, captura de audio y video, servicios de proxy, password stealing y key logging.

Poison Ivy fue escrito en Delphi y utiliza técnicas de ofuscamiento con variedad de paquetes los cuales a la vez comprimen el código binario con el fin de evadir su detección, típicamente utilizan proveedores que ofrecen IP dinámicas y direcciones intermedias y el troyano que genera puede ser distribuido mediante técnicas de ingeniería social y otros métodos de ataque. Una vez ejecutado sobre la víctima, toma el control total sobre la maquina controlado por un operador y de acuerdo a las necesidades del operador el kit ofrece diferentes tipos de payload. El escenario de payload más común involucra generar un PE binario (ejecutable de Windows), que puede ser ejecutado en equipos que se encuentren

bajo plataformas de la familia Windows y puede ser introducido en cualquiera de los siguientes componentes:

- ✓ PE Binary
- ✓ Shellcode
 - Como un Binario
 - Arreglo en C
 - Arreglo en Delphy
 - Arreglo en Python

Estos escenarios son esencialmente idénticos, las diferencias entre los tipos de shellcode son superficiales: contienen algunos bits, su formato depende del lenguaje que el operador haya seleccionado a momento de crearlo y pueden utilizar características adicionales que su autor ofrecía en la versión de pago, como técnicas de encriptación que podían ocultar su detección por soluciones antimalware.

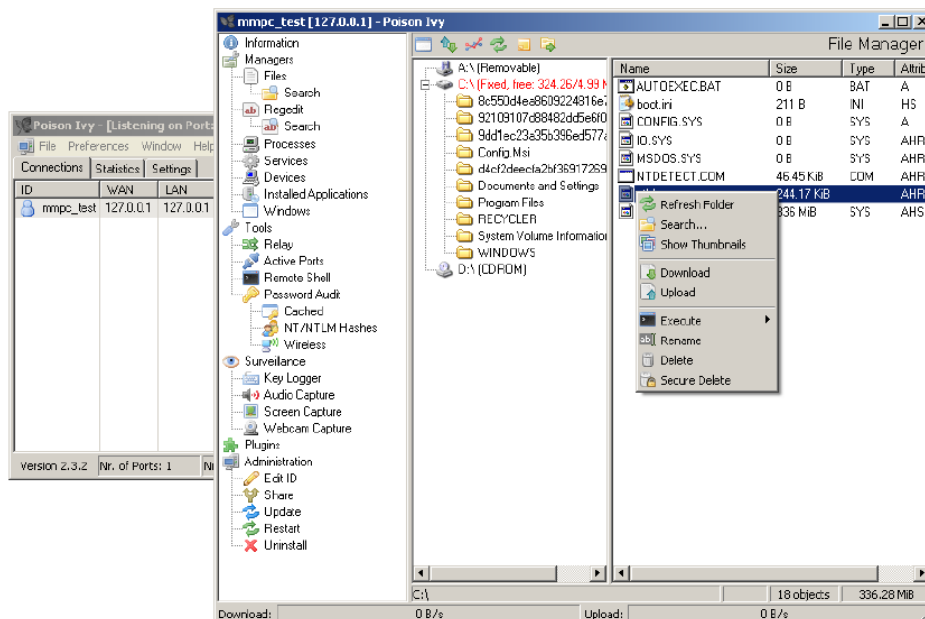


Figura 28. Escenario de Poison Ivy, ([23], página 5)

EL informe presentado por Microsoft refiere tres distintas áreas como configuraciones por defecto que evidencian a Poison Ivy RAT, entre ellas:

- ✓ **Mutex Names:** Característica utilizada para identificar marcas que las aplicaciones pueden registrar. En el caso de Poison Ivy esta característica previene las copias múltiples de servidor desde que se ejecuta y es comprometida su máquina, y que los autores de malware pueden definir como un único string para su chequeo. Por defecto todas las versiones conocidas refieren como mutex por defecto: **“)!VoqA.I4”**
- ✓ **Puerto cliente:** Una característica que como método permite obtener una marca respecto al servicio de comunicación entre la maquina comprometida y su controlador, por lo general se establece como puerto de escucha una conexión bajo TCP en la que el puerto usado por defecto es el **3460**.
- ✓ **Hostnames de IP Dinámica:** Adicionalmente al puerto, los operadores pueden especificar una dirección del servidor, pero en ataques sofisticados se puede hacer uso de un host intermediario. Para ambos casos la dirección IP puede ser cambiada, y tanto en la documentación referida en la página oficial como en los diferentes tutoriales existentes en línea, sugieren el uso de un operador no-ip.org o no-ip.biz.

7.2.5. Búsqueda de cadenas de texto

Continuando con los procedimientos referidos en la Tesis Doctoral (Bermejo, 2015) respecto al objetivo de esta actividad, con el uso o aplicación de las herramientas **Bintex** y **Strings** las cuales permiten ejecutar una análisis y en sus resultados identificar cadenas sospechosas que serán localizarlas mediante la fase del análisis de código y así tener una base objetiva respecto a su comportamiento; dichas cadenas permitirán identificar información particular sobre el malware, como puede ser: palabras particulares relacionadas con el malware que hayan sido descubiertas en otros análisis, directorios, direcciones IP, entre otros elementos.

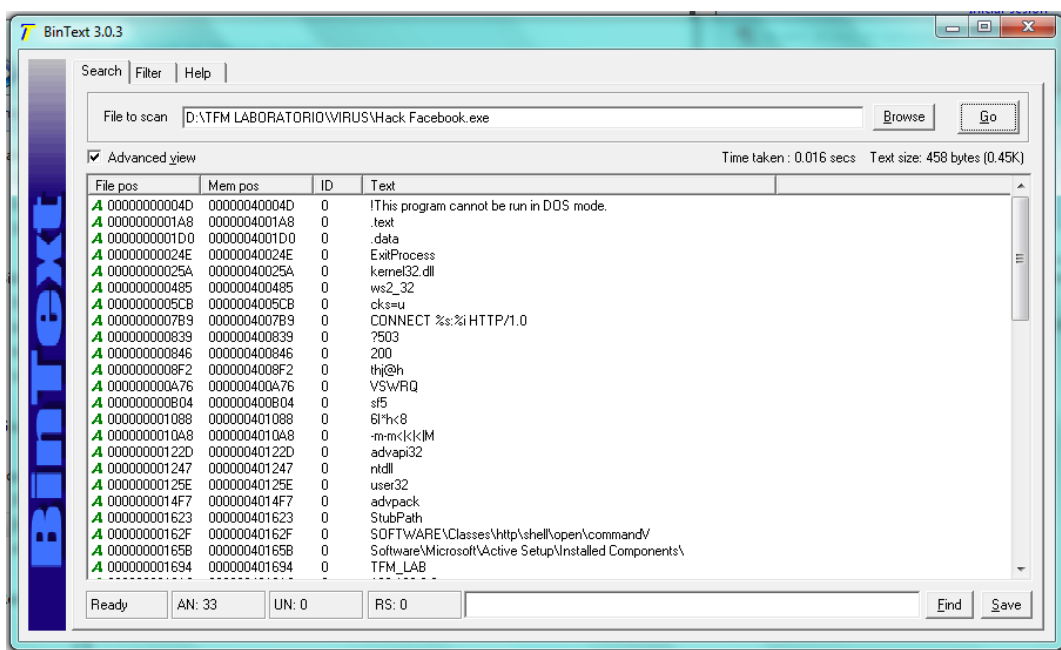


Figura 29. Resultado análisis de Bintex, sobre el malware generado archivo “Hack Facebook.exe”.

Frente al resultado del análisis obtenido con la aplicación de Bintex, el cual se puede consultar en su totalidad en el Anexo C, encontramos información específica que fue ingresada desde el entorno de Poison Ivy en su configuración, al momento de crear el malware (Hack Facebook.exe). En este sentido podemos resaltar:

File pos	Mem pos	ID	Text	Comentario
000000001694	000000401694	0	TFM_LAB	Corresponde a un ID que identifica la conexión generada por el malware.
0000000016A0	0000004016A0	0	192.168.0.1	Corresponde a la IP del Command & Control (C&C).
0000000016EB	0000004016EB	0	LABTFM	Refiere el nombre del proceso de la abreviatura que tiene relación con el alias HKEY_LOCAL_MACHINE del sistema.
0000000016F5	0000004016F5	0)!VoqA.I4-	Hace referencia al mutex.

000000001702	000000401702	0	labtfrm.exe	Nombre del archivo que refiere el proceso.
0000000018CF	0000004018CF	0	explorer.exe	Proceso del sistema operativo al cual se adhiere el malware.

Tabla 5. Extracto resultado del análisis de Bintex sobre el archivo del malware generado.

Continuando con el proceso, sobre el archivo “Hack Facebook.exe” se aplicó un análisis con la herramienta **Strings**, encontrando una serie de cadenas entre las que se encuentran las presentadas en la tabla 5, y adicionalmente:

- ✓ Kernel32.dll : El cual hace referencia al proceso de ejecución frente al llamado de explorer.exe
- ✓ CONNECT %s:%i HTTP/1.0: El cual refiere el protocolo de comunicación persistente utilizado en la conexión con el C&C.

Al respecto los resultados obtenidos se pueden consultar en su totalidad en el Anexo D.

```

C:\windows\system32\cmd.exe
C:\>string>strings.exe -a "Hack Facebook.exe"

Strings v2.53 - Search for ANSI and Unicode strings in binary images.
Copyright (C) 1999-2016 Mark Russinovich
Sysinternals - www.sysinternals.com

!This program cannot be run in DOS mode.
Rich
.text
.data
ExitProcess
kernel32.dll
ws2_32
A)!
~
"p7
cks=u
ttp=
cks=
CONNECT %s:%i HTTP/1.0
QSRW
?503
200
PWW
thjeh
PWW
USWRQ
YZ_I^
f5
YZ_I^
D$0

```

Figura 30. Resultado análisis de Strings, sobre el malware generado archivo “Hack Facebook.exe”.

7.2.6. Identificación de técnicas de ofuscación

Para desarrollar esta actividad la herramienta de aplicación propuesta es **PEiD**, útil en archivos PE⁸, con el fin de identificar técnicas de empaquetamiento, cifrado, polimorfismo y metamorfismo conocidas como técnicas de ofuscación utilizadas por los malware con el fin de protegerse frente posibles detecciones.

Tras la aplicación de la herramienta PEiD sobre el archivo “Hack Facebook.exe”, se obtuvieron los siguientes resultados:

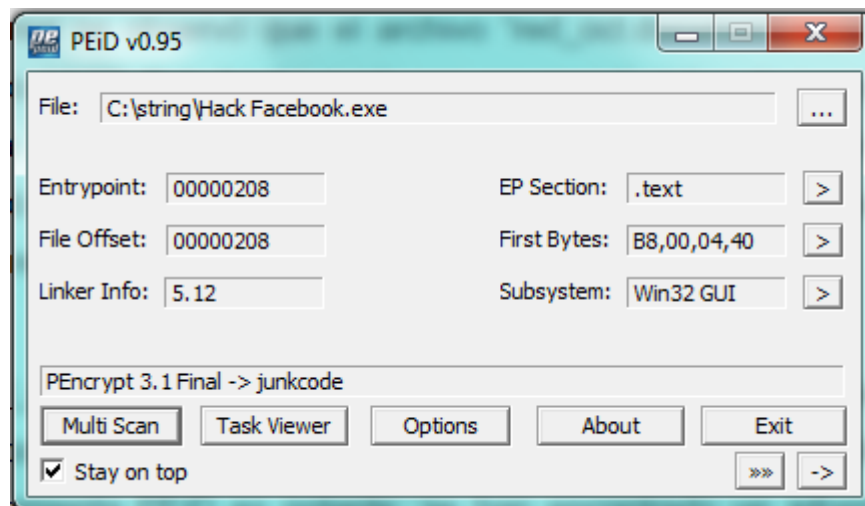


Figura 31. Resultado análisis PEiD, sobre el malware generado archivo “Hack Facebook.exe”.

Tal y como se puede apreciar en la figura anterior, el resultado obtenido sobre el archivo analizado refiere que su formato PE fue generado por la herramienta conocida como “PEncript 3.1 Final⁹ : junkcode”.

⁸ Portable Executable: Son archivos ejecutables y portables diseñados en el ambiente de Windows.

⁹ PEncript. Se trata de una aplicación muy pequeña que tiene como objetivo hacer ejecutables portables (PE) más seguros con un mínimo esfuerzo, así como también introducir un número de capas al dispositivo de cifrado, activar el camuflaje salto al descifrador, depuradores contra el nivel de aplicación, así como el uso de técnicas criptográficas polimórficas

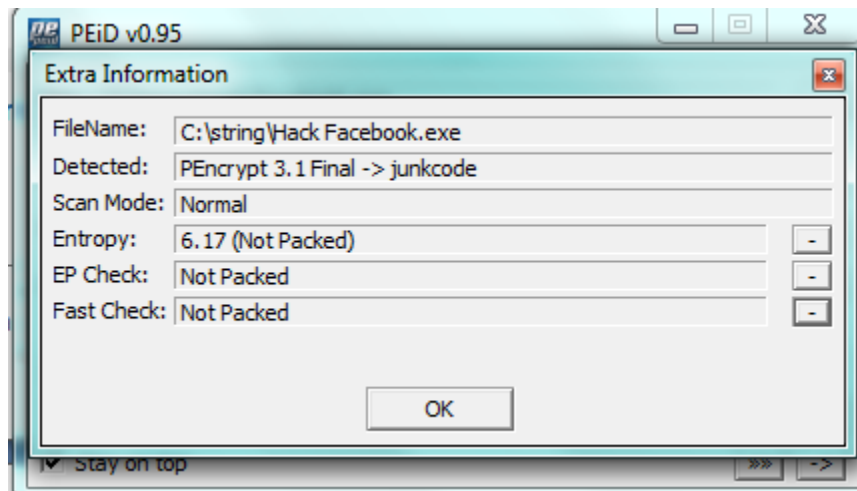


Figura 32. PEiD - “Task Viewer”, compilador y entropía de “Hack Facebook.exe”.

Adicionalmente el resultado anterior permite observar una entropía del archivo correspondiente a “6.17”, que resulta baja para archivos comprimidos.

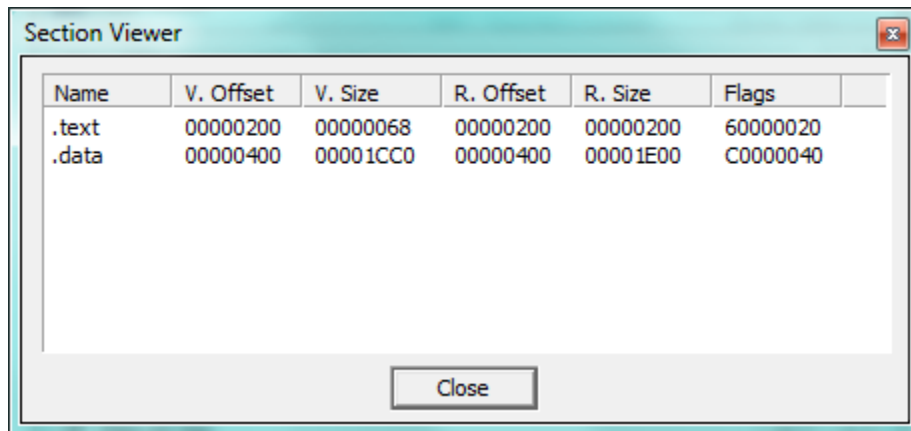


Figura 33. PEiD - Secciones de "Hack Facebook.exe"

La figura anterior refiere solo dos (2) secciones detectadas y analizadas, sobre las cuales tampoco se detectó algún tipo de irregularidad, ya que su estructura refiere un formato típico de las secciones usadas en los binarios de Windows.

El análisis ofrecido por PEiD, nos relaciona el siguiente listado de tareas descubiertas dentro del archivo "Hack Facebook.exe".

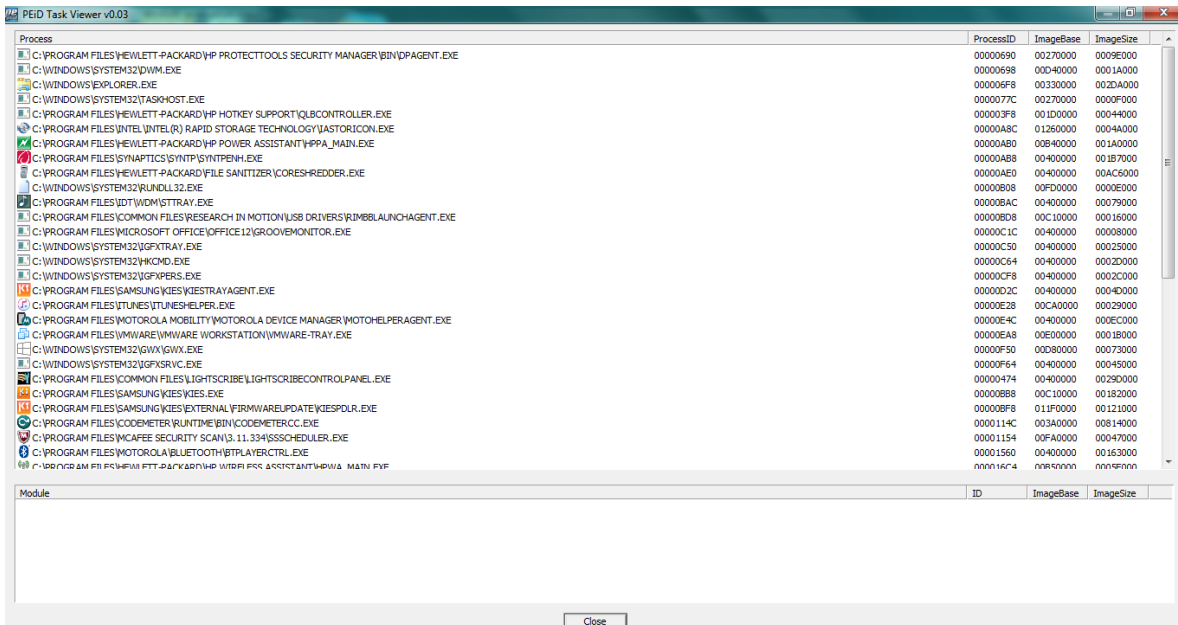


Figura 34. PEiD Task viewer "Hack Facebook.exe"

Adicionalmente navegando sobre las opciones de la herramienta PEiD, se encontró un plugin adicional llamada Kripto ANALizer, el cual en su aplicación ofrece la información respecto al tipo de cifrado “Camelia”, un algoritmo bien reconocido y definido por el RFC 3713 (Matsui, 2004):

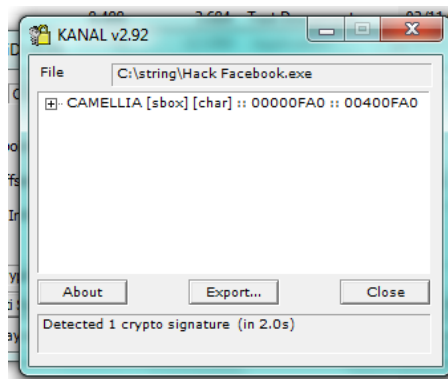


Figura 35. PEiD - Kripto ANALizer "Hack Facebook.exe"

Frente a estos resultados respecto al sistema de cifrado de camelia, Andrzej Dereszowski (2010), refiere que: la primera comunicación que se desarrolla es la autenticación. En primer lugar, se establece una conexión TCP / IP y el servidor le envía 256 bytes aleatorios al

cliente como un desafío de autenticación. Estos bytes se someten entonces a un cálculo aritmético con la contraseña definida en la configuración del cliente, cálculo que es realizado tanto por el servidor como por el cliente.

La contraseña está siempre incrustada en binario del servidor y el cliente responde con 256 bytes que son el resultado de este cálculo. El servidor comprueba la respuesta mediante la comparación de los primeros 64 bytes de la respuesta del cliente con su propio resultado del mismo cálculo. Si el resultado de la operación de comparación tiene éxito, significa que el cliente y el servidor comparten la misma contraseña y la autenticación de respuesta de desafío ha tenido éxito.

Todas las operaciones de cifrado y descifrado posteriores también se realizan utilizando la contraseña como un parámetro para el algoritmo de cifrado. Después de esta fase, el cliente se autentica al servidor, y pueden comenzar a intercambiar datos.

Los resultados obtenidos tras la aplicación de PEiD, no han permitido comprobar la existencia de ofuscación sobre el archivo "Hack Facebook.exe", así que se ejecuta una herramienta llamada "DiE.EXE" (Detect It Easy), la cual identifica la existencia del compilador basado en MASM32. El SDK MASM32¹⁰ está dirigido a programadores experimentados en la escritura de código en versiones de 32 bits de Windows mediante la interfaz API y familiarizados con un poco de programación en ensamblador mnemónico directo, esta característica supone un reto para los programadores principiantes debido a las técnicas avanzadas de programación en ensamblador.

¹⁰ www.masm32.com

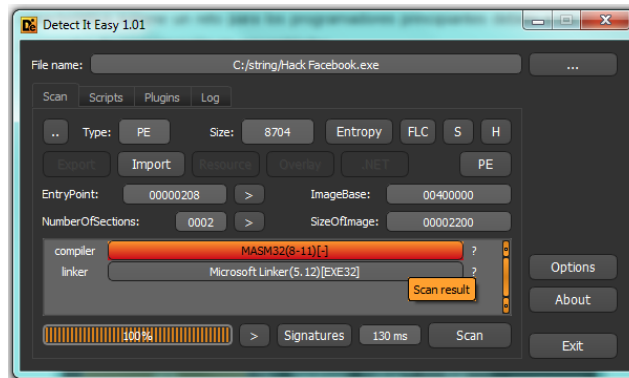


Figura 36. Detect It Easy - Compiler "Hack Facebook.exe".

Navegando un poco sobre las distintas opciones que la herramienta DIE ofrece, encontramos la siguiente información respecto a la existencia de bloques comprimidos en el archivo analizado, a pesar que el análisis con PEiD mostraba el archivo como no empaquetado o comprimido, obteniendo una entropía correspondiente al 75%.

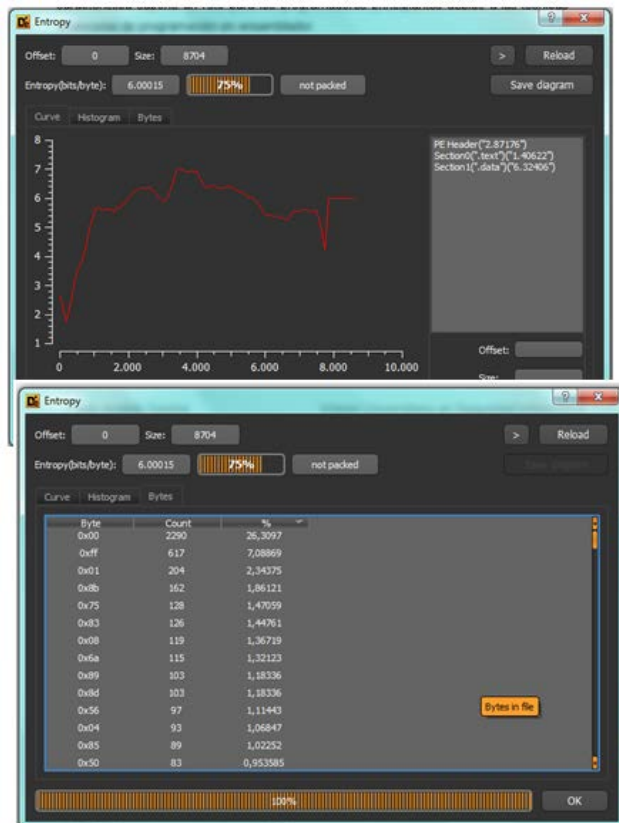


Figura 37. Detect It Easy - Entropy "Hack Facebook.exe".

7.2.7. Formato y estructura del fichero

Con el fin de identificar la información relativa al encabezado “PE”, se presenta la aplicación de la herramienta “PE Explorer” y sus resultados:

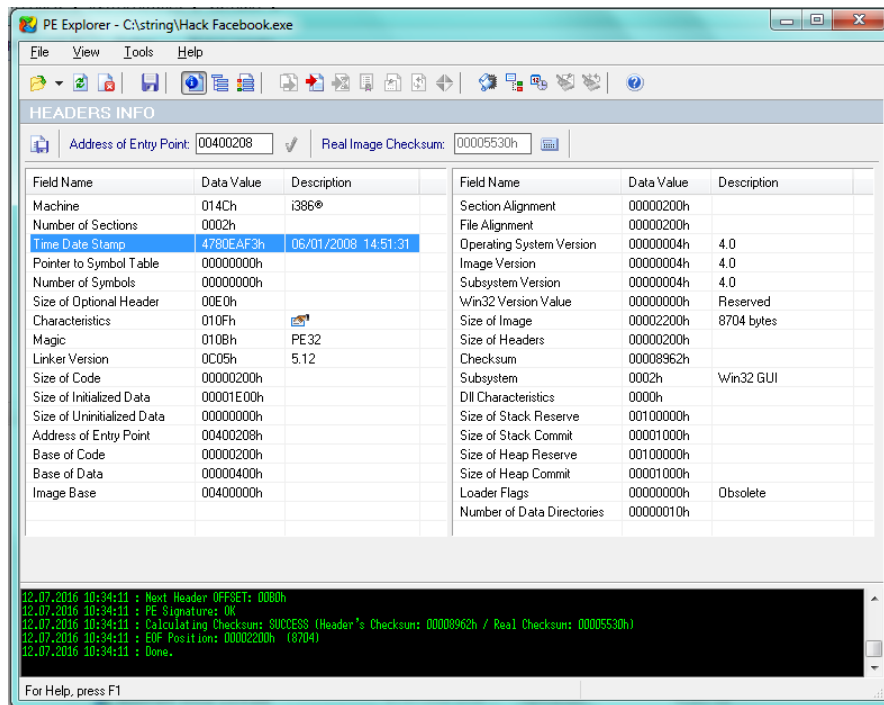


Figura 38. PE Explorer y sus resultados sobre "Hack Facebook.exe"

Según la información presentada en la figura anterior, se puede establecer que la fecha de creación para el archivo “Hack Facebook.exe” corresponde al día 06/01/2008 14:51:31, sin embargo este campo puede ser alterado o modificado, lo que no permite tener una certeza absoluta en su resultado. Ahora, el periodo de tiempo registrado coincide con la última fecha de lanzamiento de Poison Ivy 2.3.2.exe, muestra obtenida para el desarrollo del laboratorio.

Continuando con el análisis se aplica la herramienta “Dependency Walker”, la cual nos permite obtener un listado de los módulos existentes y su relación, aunque sus resultados en su análisis se torna algo complejo por la cantidad de información obtenida, nos permite obtener un panorama que permitirá identificar módulos y sus librerías, con funciones más utilizadas por el malware.

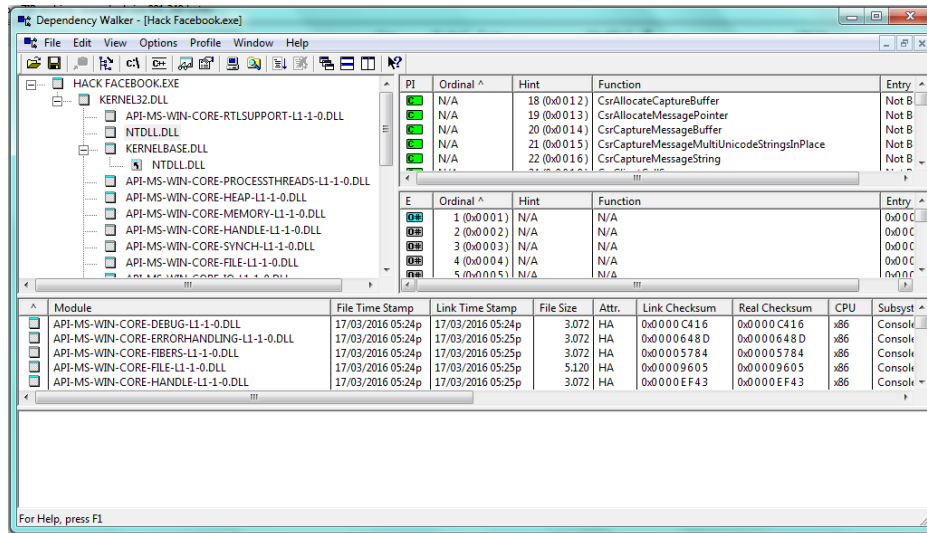


Figura 39. Dependency Walker y sus resultados sobre "Hack Facebook.exe"

Como resultado se obtuvieron las siguientes librerías:

c:\windows\system32\API-MS-WIN-CORE-DEBUG-L1-1-0.DLL

c:\windows\system32\API-MS-WIN-CORE-ERRORHANDLING-L1-1-0.DLL

c:\windows\system32\API-MS-WIN-CORE-FIBERS-L1-1-0.DLL

c:\windows\system32\API-MS-WIN-CORE-FILE-L1-1-0.DLL

c:\windows\system32\API-MS-WIN-CORE-HANDLE-L1-1-0.DLL

c:\windows\system32\API-MS-WIN-CORE-HEAP-L1-1-0.DLL

c:\windows\system32\API-MS-WIN-CORE-IO-L1-1-0.DLL

c:\windows\system32\API-MS-WIN-CORE-LIBRARYLOADER-L1-1-0.DLL

c:\windows\system32\API-MS-WIN-CORE-LOCALIZATION-L1-1-0.DLL

c:\windows\system32\API-MS-WIN-CORE-MEMORY-L1-1-0.DLL

c:\windows\system32\API-MS-WIN-CORE-MISC-L1-1-0.DLL

c:\windows\system32\API-MS-WIN-CORE-NAMEDPIPE-L1-1-0.DLL

c:\windows\system32\API-MS-WIN-CORE-PROCESSENVIRONMENT-L1-1-0.DLL

c:\windows\system32\API-MS-WIN-CORE-PROCESSTHREADS-L1-1-0.DLL

c:\windows\system32\API-MS-WIN-CORE-PROFILE-L1-1-0.DLL

c:\windows\system32\API-MS-WIN-CORE-RTLSUPPORT-L1-1-0.DLL

c:\windows\system32\API-MS-WIN-CORE-STRING-L1-1-0.DLL

c:\windows\system32\API-MS-WIN-CORE-SYNCH-L1-1-0.DLL

c:\windows\system32\API-MS-WIN-CORE-SYSINFO-L1-1-0.DLL

c:\windows\system32\API-MS-WIN-CORE-THREADPOOL-L1-1-0.DLL

c:\windows\system32\API-MS-WIN-CORE-UTIL-L1-1-0.DLL

c:\windows\system32\API-MS-WIN-SECURITY-BASE-L1-1-0.DLL

c:\string\HACK FACEBOOK.EXE

c:\windows\system32\KERNEL32.DLL

c:\windows\system32\KERNELBASE.DLL

c:\windows\system32\NTDLL.DLL

Con la aplicación de la herramienta **PEBrowse**, la cual permite desde un análisis visual detectar información y datos sospechosos, se pudo comprobar la existencia de una dirección IP la cual refiere **192.168.0.1**, la cual fue configurada al momento de generar el archivo "Hack Facebook.exe" y que especifica el equipo desde donde se realizaron las actividades de Control. Esta información es presentada en un árbol que relaciona las principales secciones del archivo PE.

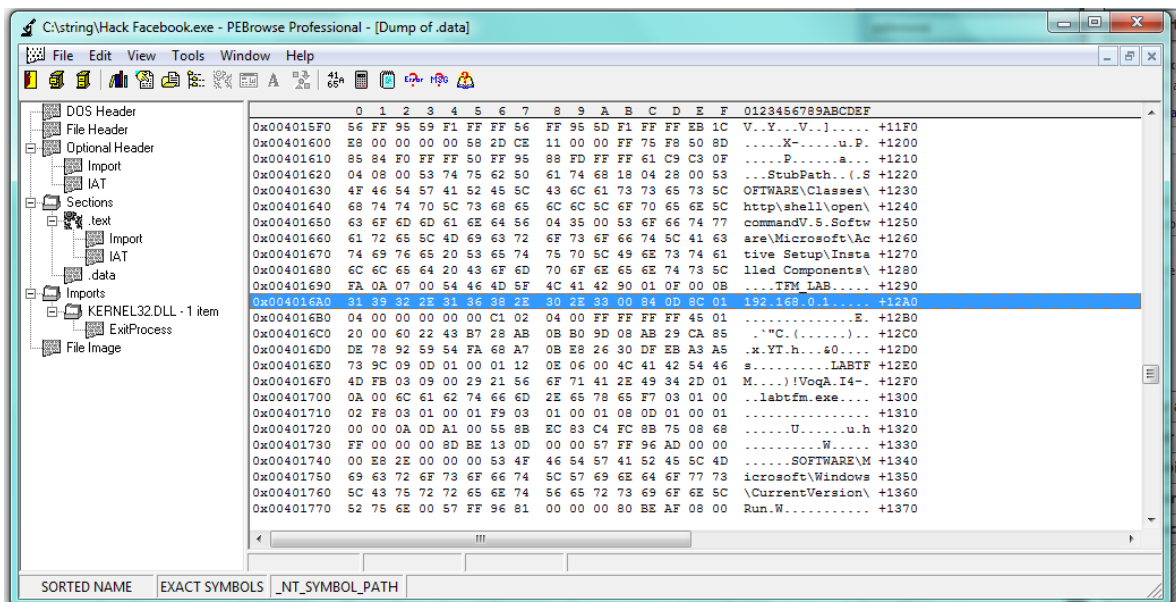


Figura 40. PEBrowse y sus resultados sobre "Hack Facebook.exe".

Se aplicó la herramienta **PEStudio**, la cual desde su análisis permite obtener una serie de resultados globales sobre el archivo. Sobre el particular, la Tesis Doctoral refiere que aunque las librerías o funciones importadas que aparezcan como sospechosas, no significa que el malware realice con ellas acciones ilegítimas, ya que los programas legítimos también podrían utilizarlas, tener el conocimiento del mismo permite más adelante una posible clasificación del código y su análisis dinámico posterior (Bermejo, 2015).

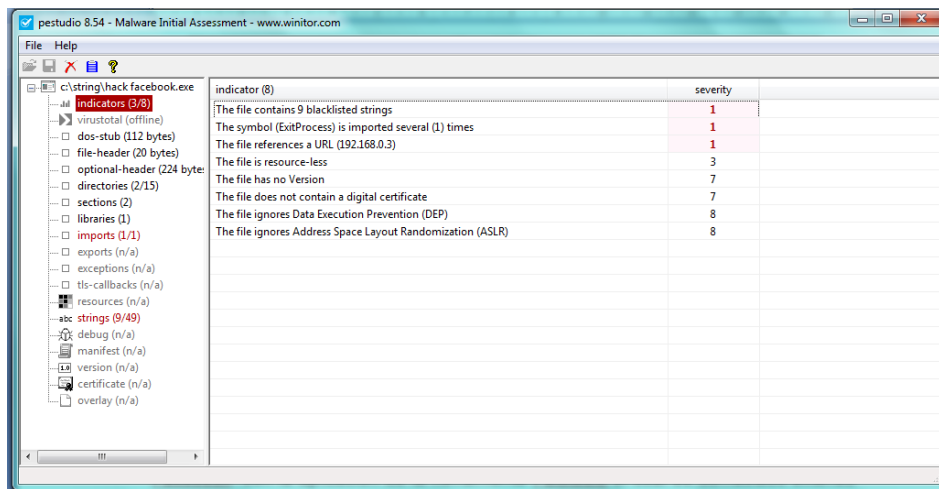


Figura 41. PEStudio y sus resultados sobre "Hack Facebook.exe"

La figura anterior presenta los resultados del análisis de riesgo sobre el archivo "Hack Facebook" con la aplicación de la herramienta PEStudio, sobre la cual se puede apreciar el grado de severidad frente a la clasificación que presenta el aplicativo refiriendo 9 archivos que contienen cadenas marcadas en principio como sospechosas (BlackListed).

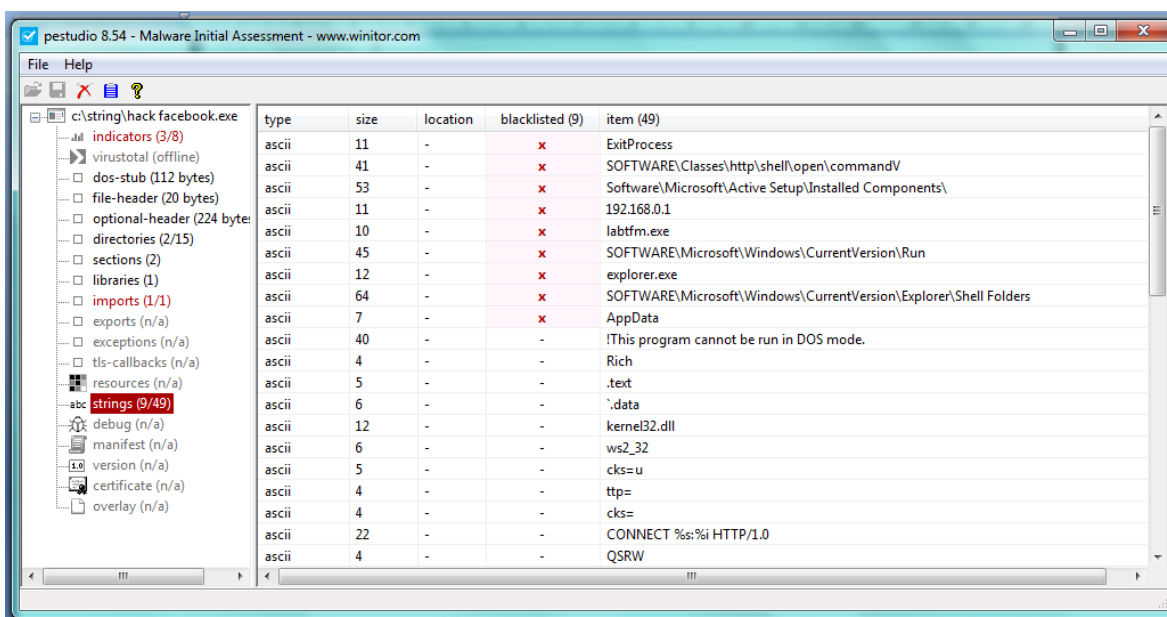


Figura 42. . PEStudio BlackListed DLLs sobre "Hack Facebook.exe"

Ya para este momento se han descubierto en detalle una serie de características sobre el archivo "Hack Facebook.exe", que permiten obtener una base de conocimiento para desarrollar las fases de análisis y comportamiento.

7.3. Fase 3: Análisis de Código

Este apartado tiene por objeto ejecutar el análisis del código del malware, el cual según la metodología desarrollada por Don Javier Bermejo (2015) en su Tesis Doctoral, presenta un cambio en el orden de aplicación, anteponiendo el análisis dinámico o de comportamiento al análisis estático. Sin embargo, en su documento expone como procedimientos estandarizados, la aplicación de tres fases en su orden:

- ✓ Comprobar el funcionamiento del malware de acuerdo a la lectura del código.
- ✓ Desarrollar sobre el mismo un análisis estático utilizando herramientas de desensamblado.
- ✓ Aplicar mediante una herramienta de depuración el análisis dinámico del malware.

Como nuestro objetivo frente al desarrollo del piloto experimental es comprobar la validez de la metodología presentada en la Tesis Doctoral, su aplicación se desarrollará de acuerdo al orden propuesto por Don Javier Bermejo (2015).

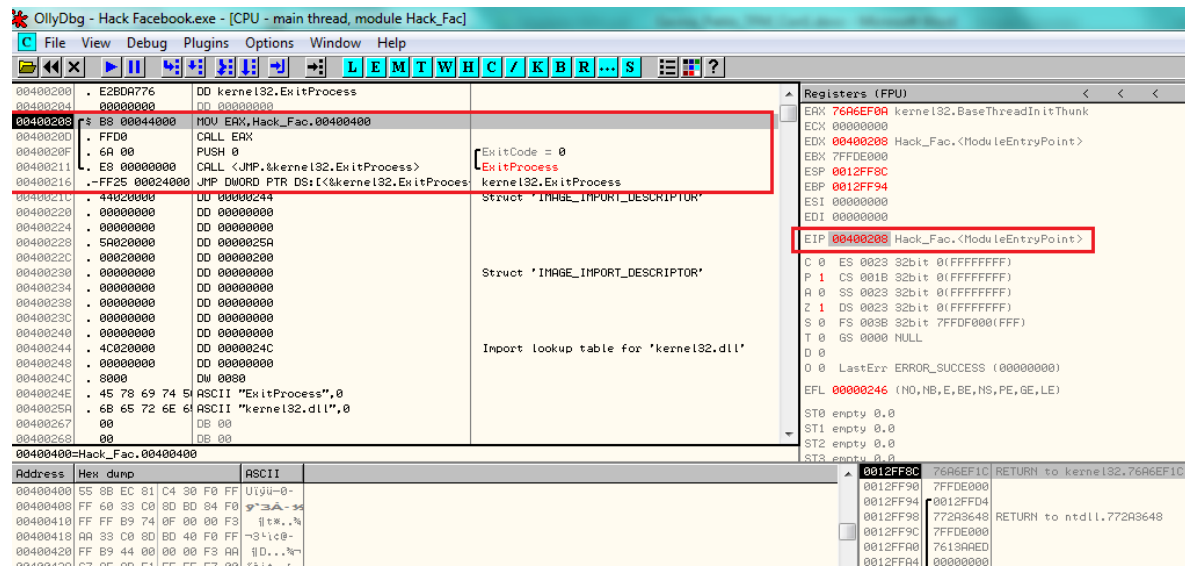


Figura 44. OllyDbg - Llamada a Hack_Face en el proceso explorer.exe

La figura anterior muestra la llamada al proceso Hack_Face, con un parámetro que refiere al proceso explorer.exe.

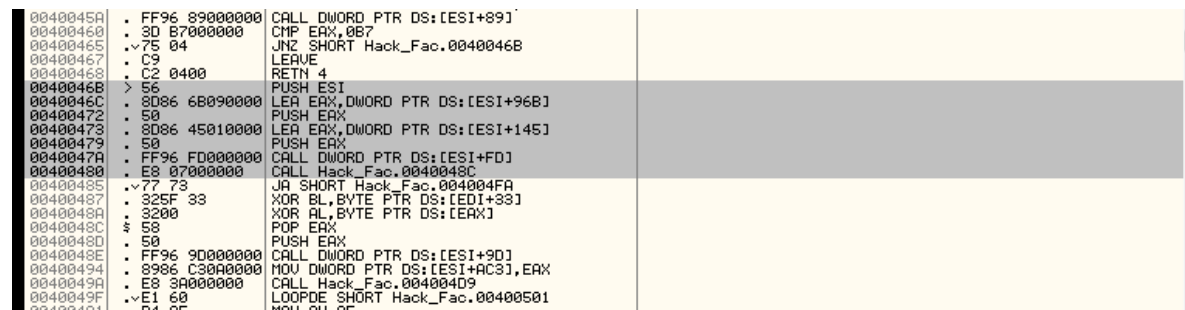


Figura 45. OllyDbg. Datos y funciones.

La figura anterior expone los datos y funciones a los que hace referencia como desplazamientos en la estructura a la que apunta el registro ESI. En la publicación presentada por Andrzej Dereszowski (2010), refiere que esta característica permite observar la forma en que el código llama a sus propias funciones y las funciones de la API. Todo se hace a través de una llamada indirecta `ESI + n`, donde `n` es un desplazamiento a la dirección base almacenada en el registro ESI. Esto hace al código independiente de

posición, el cual está diseñado para funcionar sin importar la dirección de memoria donde ha sido insertado.

Ahora bien, el mecanismo de CALL ESI puede ser utilizado por el servidor para esconderse muy bien en el sistema, y como un mecanismo para frustrar el análisis estático, por lo tanto, algunas herramientas automáticas de análisis de comportamiento no serían capaces de analizarlo porque serían incapaces de reconocer las llamadas a la API correctamente.

Esta característica es observada en una técnica utilizada en Linux denominada “objeto compartido”, el cual utiliza el direccionamiento indirecto a través de la tabla de offset almacenado en la memoria, mientras que en el caso de Poison Ivy los desplazamientos son en relación con el contenido del registro ESI, donde la dirección base se almacena.

```

0012FBEC 7FFDE000
0012FBF0 76E30000 ASCII "PE"
0012FBF4 002D1604 UNICODE "Hack Facebook.exe"
0012FBF8 75770000 kernel32.75770000
0012FBFC 76E30000 ntdll.76E30000
0012FC00 003C0028
0012FC04 002D1718 UNICODE "C:\Windows\SYSTEM32\ntdll.dll"
0012FC08 00000000
0012FC0C 00000000
0012FC10 00000001
0012FC14 0012FD24
0012FC18 00000000
0012FC1C 00560054
0012FC20 002D15D2 UNICODE "E:\TFM LABORATORIO\VIRUS\Hack Facebook.exe"
0012FC24 002D0FD0
0012FC28 02080032
0012FC2C 002D1268 UNICODE "E:\TFM LABORATORIO\VIRUS\"
0012FC30 00560054
0012FC34 002D15D2 UNICODE "E:\TFM LABORATORIO\VIRUS\Hack Facebook.exe"
0012FC38 002D1770 UNICODE "C:\Windows\system32"
0012FC3C 002D0000
0012FC40 00400080 ASCII "PE"
0012FC44 00000042

```

Figura 46. Referencia a la librería ntdll.dll

En el análisis en particular respecto al malware Poison Ivy, el cual presenta un grado alto de complejidad sumado a nuestra poca experiencia y escasa especialización en técnicas de ingeniería inversa, han podido dejar pasar algunos detalles valiosos en sus resultados, como es el caso del método de codificación del código ofuscado, la identificación de su ubicación, entre otros. Sin embargo, hemos podido desde la aplicación de la metodología desarrollada por Don Javier Bermejo (2015) en su Tesis Doctoral, como objetivo y base del presente piloto experimental, obtener el conocimiento necesario respecto a la información que se podría necesitar en las siguientes fases.

7.3.2. Análisis estático del código

Sobre este paso se ejecuta la herramienta “IDA42 Pro” sobre un análisis que permitirá recabar información sobre la estructura del archivo, como la identificación de las posiciones de memoria, la relación entre las funciones y otros datos.

Una vez ejecutada la herramienta sobre el archivo “Hack Facebook.exe”, se puede observar una serie de ventanas las cuales presentan la estructura del archivo y cada uno de los elementos que lo componen.

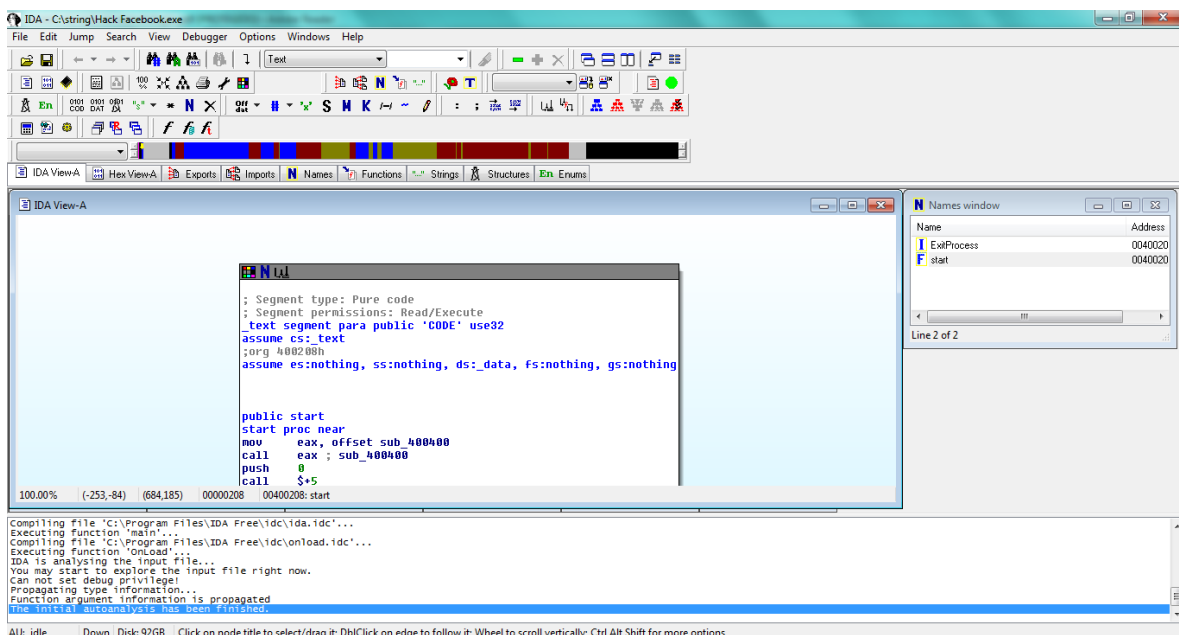


Figura 47. IDA PRO en ejecución sobre "Hack Facebook.exe".

Entre la información desplegada encontramos una ventana que permite obtener las cadenas de caracteres cuyos resultados coinciden con los obtenidos en las otras herramientas (Jale, 2012), así como también de manera gráfica, observar la estructura de las llamadas de funciones del archivo.

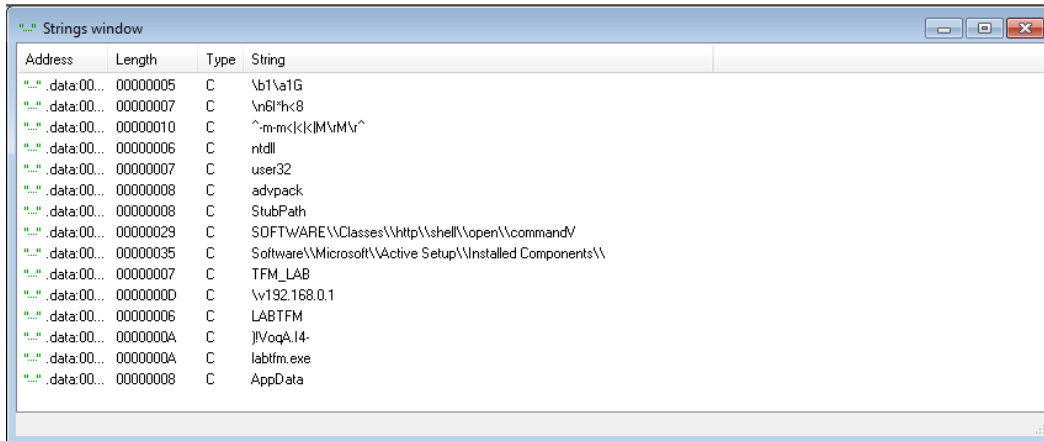


Figura 48. IDA PRO Cadenas de caracteres sobre "Hack Facebook.exe".



Figura 49. Estructura del archivo "Hack Facebook.exe".

En relación con la estructura del archivo, se observa la existencia de código ofuscado, así como también algunos espacios vacíos. Respecto a su composición se pueden diferenciar las siguientes subrutinas.

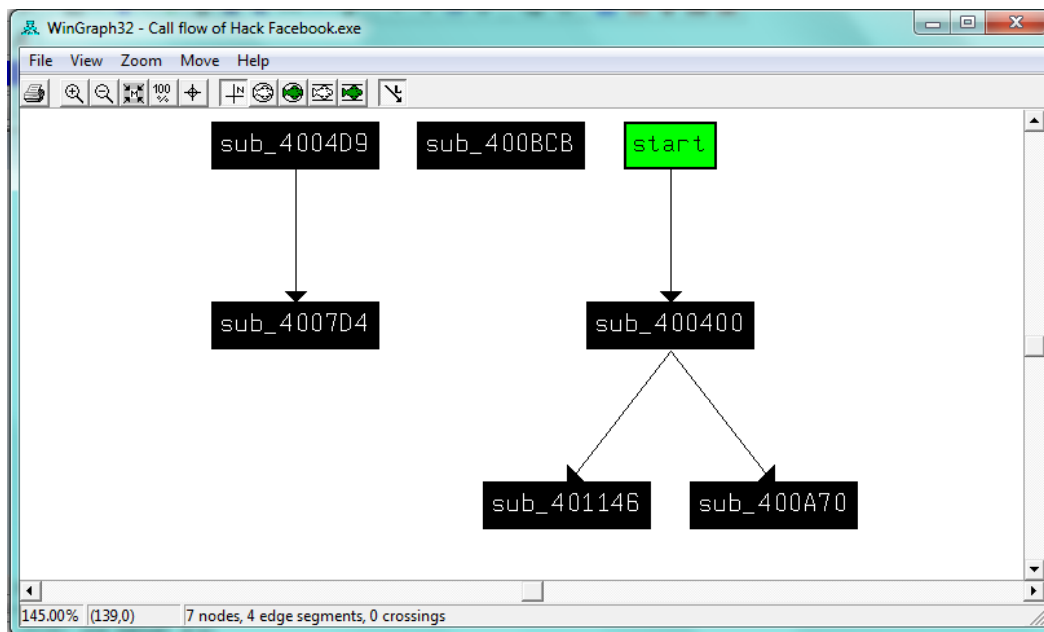
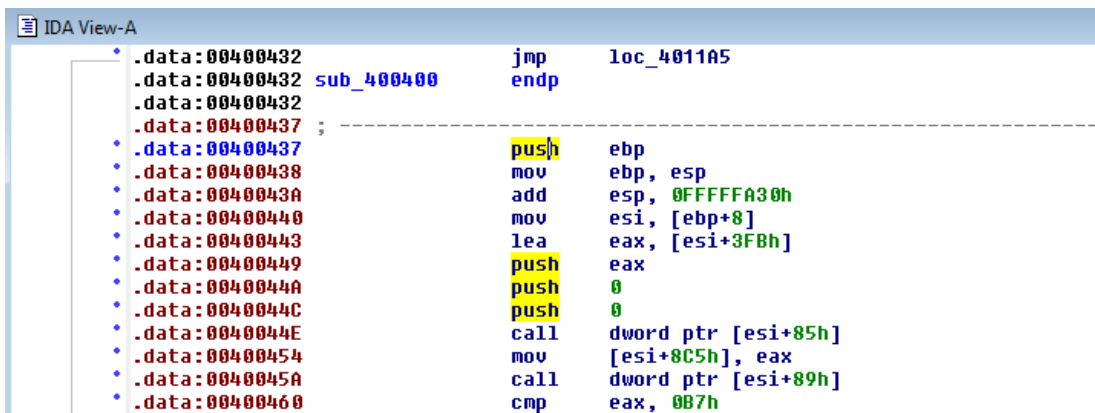


Figura 50. IDA PRO Llamada de funciones sobre "Hack Facebook.exe".

Una vez verificadas diferentes fuentes de consulta abiertas, encontramos una característica técnica respecto a Poison Ivy y es el hecho de que se propaga a través de más de dos docenas de segmentos de memoria pequeña. Esta característica no lo hace totalmente sigiloso (ahora hay más de 20 regiones de memoria sospechosos en vez de uno), pero sí refiere un análisis más desafiante. En lugar de verter un solo segmento de memoria, se precisa que volcar toda la memoria y sus segmentos para luego averiguar cómo están articulados entre sí. La mayoría de los segmentos pueden empezar con instrucciones como EBX PUSH EBP o PUSH. Estas son funciones individuales que Poison Ivy dispersa en toda la memoria de explorer.exe. (Jale, 2012).



```

IDA View-A
. data: 00400432 jmp loc_4011A5
. data: 00400432 sub_400400 endp
. data: 00400432
. data: 00400437 ; -----
. data: 00400437 push ebp
. data: 00400438 mov ebp, esp
. data: 0040043A add esp, 0FFFFFFA30h
. data: 00400440 mov esi, [ebp+8]
. data: 00400443 lea eax, [esi+3FBh]
. data: 00400449 push eax
. data: 0040044A push 0
. data: 0040044C push 0
. data: 0040044E call dword ptr [esi+85h]
. data: 00400454 mov [esi+8C5h], eax
. data: 0040045A call dword ptr [esi+89h]
. data: 00400460 cmp eax, 0B7h

```

Figura 51. IDA PRO - Fragmento subrutina 400400 – “Hack Facebook.exe”.

7.4. Fase 4: Análisis de comportamiento

7.4.1. Ejecución del malware

La ejecución de este tipo de análisis requiere la ejecución del archivo “Hack Facebook.exe” sobre la máquina víctima, para ello, al tratarse de un archivo .exe se encuentra compilado para ser ejecutado. Simplemente con hacer doble click sobre el mismo es suficiente.

Como dato curioso al ejecutar el archivo .exe sobre la máquina, este desaparece inmediatamente y es subsumido por el sistema, no quedando ningún rastro del mismo en el directorio o sitio donde se encontraba inicialmente.

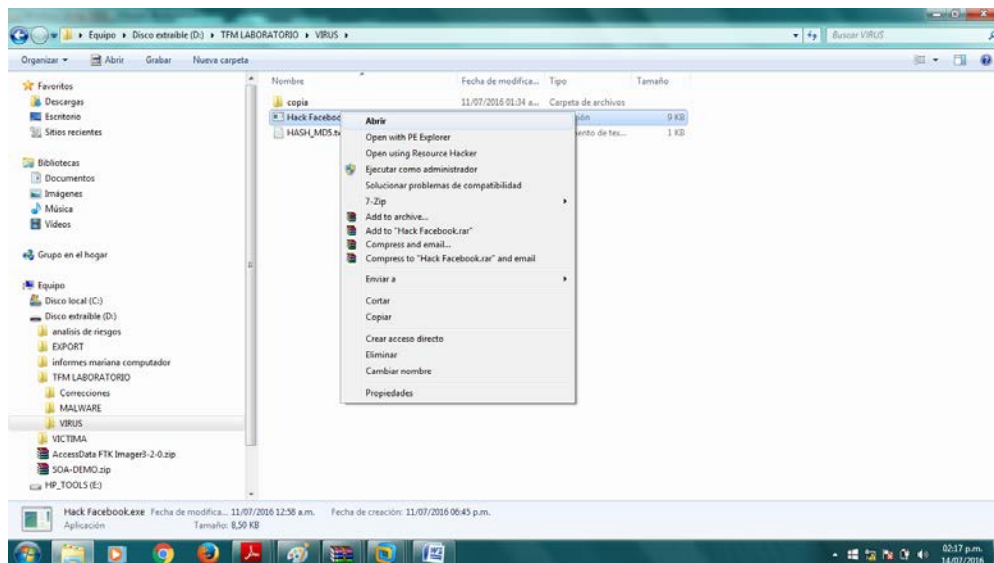


Figura 52. Ejecución del malware "Hack Facebook.exe" sobre la máquina víctima.

Sobre la máquina de control, en cuestión de milésimas de segundo observamos que se activa la conexión con la máquina víctima que para nuestro caso particular se encuentra definida mediante la IP **192.168.0.5**.

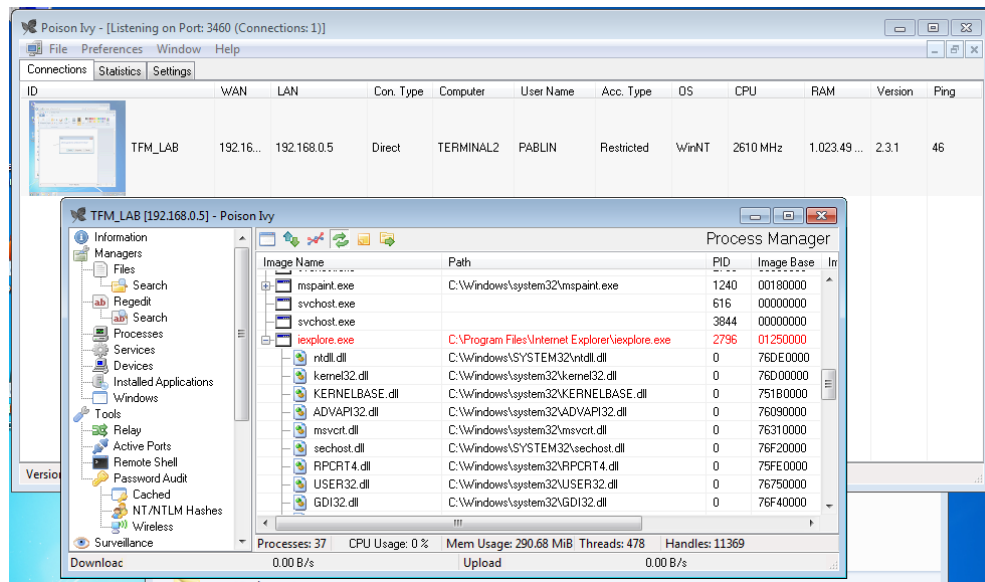


Figura 53. Ejecución Poison Ivy desde la máquina de control.

La figura anterior nos permite visualizar la conexión establecida entre la máquina víctima y el C&C, y cuyo entorno permite obtener un listado de los procesos ejecutados sobre la maquina víctima. Navegando sobre las diferentes opciones de administración que ofrece la herramienta Poison Ivy en su ejecución, se visualizaron adicionalmente un listado de puertos y su estado, resaltando la conectividad sobre el protocolo **TCP**, **IP Local: 192.168.0.5**, **IP Remoto: 192.168.0.1** y **Puerto Remoto: 3460**.

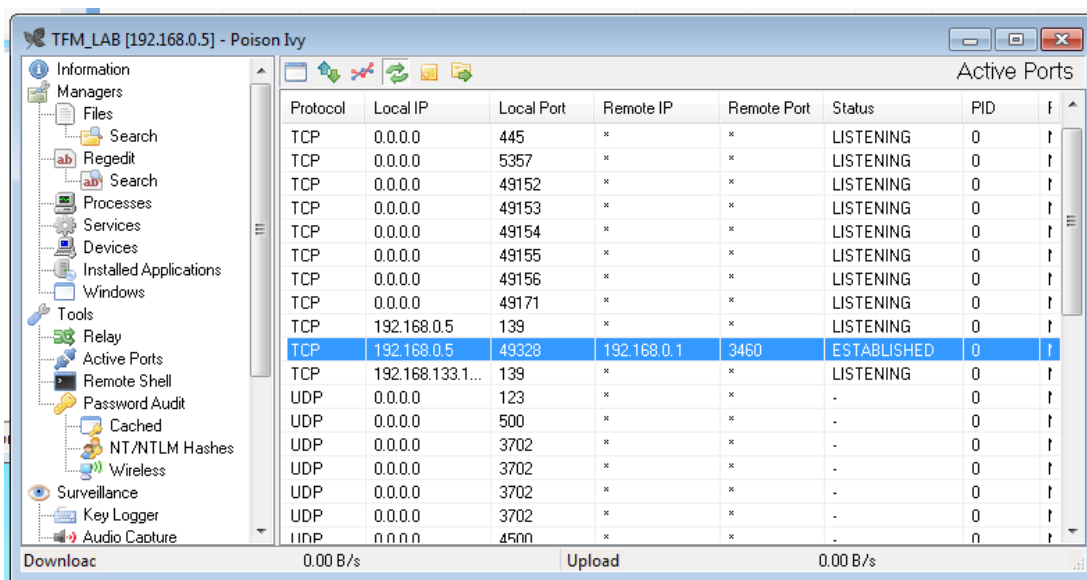


Figura 54. Poison Ivy – Listado de Puertos sobre la víctima.

Una vez ejecutado el malware sobre la máquina, se captura un snapshot mediante la herramienta **Systracer**, con el fin de apreciar el listado de cambios efectuados sobre el sistema.

Name	Version	Modified	Size	Attrib	Info
SysTracer not registered		2016-07-09 20:01:56	1,073,741,824	-HSA-	old
pagefile.sys		2016-07-14 11:41:54	1,073,741,824	-HSA-	new
SysTracer not registered					mod
post_filter.txt		2016-07-14 16:13:36	48	-A-	add
scan_filter.txt		2016-07-14 16:17:38	49	-A-	add
SysTracer.cfg		2016-07-09 22:52:34	1,536	-A-	old
SysTracer.exe		2016-07-09 23:00:37	1,536	-A-	new
SysTracerLog.txt		2016-07-09 22:52:30	2,881,536	-A-	old
SysTracerLog.txt		2016-07-14 16:13:26	2,881,536	-A-	new
SysTracer not registered		2016-07-09 22:56:41	108	-A-	add
SysTracer not registered					mod
snap-0.tmp		2016-07-09 22:53:20	0	-A-	old
snap-001.snap		2016-07-14 16:17:42	0	-A-	new
snap001.snap		2016-07-09 22:56:41	7,069,286	-A-	add
SysTracer not registered					mod
BCD		2016-07-09 20:33:25	28,672	-A-	old
BCD		2016-07-14 11:56:44	28,672	-A-	new
BCD.LOG		2016-07-09 20:33:24	25,600	-HSA-	old
BCD.LOG		2016-07-14 11:56:43	25,600	-HSA-	new
SysTracer not registered					mod
NTUSER.DAT		2016-07-09 22:52:17	262,144	-HSA-	old
NTUSER.DAT		2016-07-14 16:16:56	262,144	-HSA-	new
ntuser.dat.LOG1		2016-07-09 22:52:17	230,400	-HSA-	old
ntuser.dat.LOG1		2016-07-14 16:16:56	230,400	-HSA-	new
SysTracer not registered					mod
index.dat		2016-07-09 22:52:33	32,768	-HSA-	old
index.dat		2016-07-14 16:13:35	32,768	-HSA-	new

Figura 55. Systracer – Listado de Ficheros creados y modificados.

Utilizando las herramientas propias del aplicativo Systracer, se comparó los snapshot obtenidos sobre el sistema de la víctima antes y después de la ejecución del malware; resultados que permiten visualizar mediante un listado los cambios efectuados.

Name	Version	Modified	Size	Attrib	Info
SysTracer not registered					del
Microsoft(R) Uniscribe Unicode s	1.626.7600.1638	2009-07-13 21:16:17	627,200		del
crypt processor	5				del
Microsoft® .NET Framework	2.0.50727.4927	2009-06-10 17:23:23	278,964		del
Microsoft® Windows® Operatin	2001.12.8530.16	2009-07-13 21:15:03	522,240		del
System	385				del
Microsoft® Windows® Operatin	6.1.7600.16385	2009-07-13 21:15:07	36,864		del
System					del
Microsoft® Windows® Operatin	6.1.7600.16385	2009-07-13 21:15:07	78,848		del
System					del
Microsoft® Windows® Operatin	6.1.7600.16385	2009-07-13 21:15:22	304,640		del
System					del
Microsoft® Windows® Operatin	6.1.7600.16385	2009-07-13 21:15:32	118,272		del
System					del
Microsoft® Windows® Operatin	6.1.7600.16385	2009-07-13 21:15:36	26,624		del
System					del
Microsoft® Windows® Operatin	7.0.7600.16385	2009-07-13 21:15:50	690,688		del
System					del
Microsoft® Windows® Operatin	6.1.7600.16385	2009-07-13 21:16:13	45,568		del
System					del
Microsoft® Windows® Operatin	6.1.7600.16385	2009-07-13 21:17:54	242,936		del
System					del
Microsoft® Windows® Operatin	6.1.7600.16385	2009-07-13 21:16:13	92,160		del
System					del
oleaut32.dll	6.1.7600.16385	2009-07-13 21:16:12	571,904		del
Sistema operativo Microsoft® Wi	6.1.7600.16385	2009-07-13 21:14:53	640,000		del
ndows®					del
Sistema operativo Microsoft® Wi	6.1.7600.16385	2009-07-13 21:15:35	857,088		del
ndows®					del
Sistema operativo Microsoft® Wi	6.1.7600.16385	2009-07-13 21:15:35	288,256		del
ndows®					del
Sistema operativo Microsoft® Wi	6.1.7600.16385	2009-07-13 21:15:43	828,928		del
ndows®					del
Sistema operativo Microsoft® Wi	6.1.7600.16385	2009-07-13 21:17:51	1,286,144		del
ndows®					del

Figura 56. Systracer – Listado de Ficheros creados y modificados.

Una vez analizados los resultados, filtrando los archivos que el malware pudo generar al momento de su ejecución, se obtuvo como resultado los siguientes archivos:

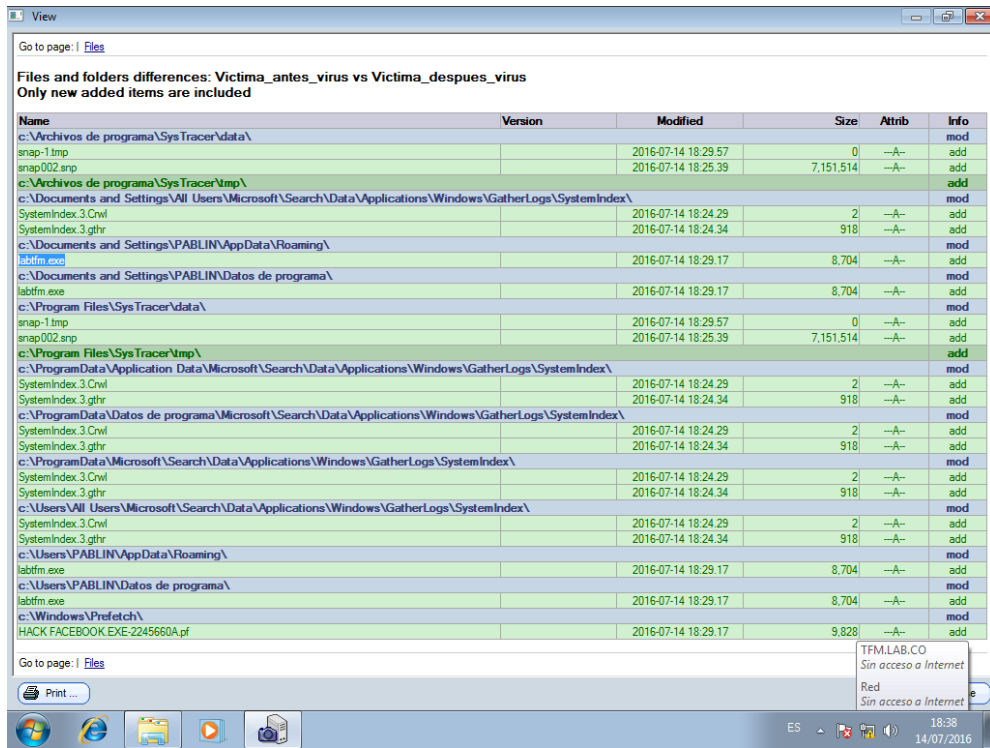


Figura 57. Listado de Ficheros creados y modificados.

Nombre de archivo	Tamaño
Labtfm.exe	8.704
Snap002.snp	7.151.514
HACK FACEBOOK.EXE-2245660A.pf	9.828
SystemIndex.3.gthr	918

Tabla 6. Archivos creados por el malware sobre máquina de la víctima.

De igual forma, sobre la maquina víctima se ejecuta la herramienta “**CaptureBAT**”, instalada junto con el aplicativo “**Winpcap**”. En sus resultados, se pudo apreciar el comportamiento del malware sobre el sistema, observando los archivos que fueron creados, modificados y eliminados, las claves de registro que se modifican, se adicionan o se eliminan, además del tráfico de red generado.

```

C:\Users\Administrador>cd ..
C:\Users>cd..
C:\>cd "Program Files"
C:\Program Files>cd Capture
C:\Program Files\Capture>CaptureBAT.exe -c -n -l reporte.txt
Option: Collecting modified files
Option: Capturing network packets
Option: Logging system events to reporte.txt
Loaded kernel driver: CaptureProcessMonitor
Loaded kernel driver: CaptureRegistryMonitor
Loaded filter driver: CaptureFileMonitor
Creating network dumper
Loading network packet dumper
network adapter found: 192.168.133.143
network adapter found: 192.168.0.5

```

Figura 58. CaptureBAT en ejecución sobre la víctima.

Tal como se observa en la figura anterior, los resultados de la ejecución de CaptureBAT son direccionadas al archivo reporte.txt, el cual en su contenido refiere la siguiente información:

```

20:58.16", "file", "write", "C:\windows\system32\svchost.exe", "C:\windows\System32\winevt\Logs\Microsoft-Windows-CodeIntegrity
20:58.16", "file", "write", "C:\windows\system32\svchost.exe", "C:\windows\System32\winevt\Logs\Microsoft-Windows-CodeIntegrity
20:58.16", "file", "write", "C:\windows\system32\svchost.exe", "C:\windows\System32\winevt\Logs\Microsoft-Windows-CodeIntegrity
20:58.16", "file", "write", "C:\windows\system32\svchost.exe", "C:\windows\System32\winevt\Logs\Microsoft-Windows-CodeIntegrity
20:58.718", "process", "created", "C:\windows\explorer.exe", "E:\TFM LABORATORIO\VIRUS\Hack Facebook.exe"
20:59.186", "process", "terminated", "C:\windows\explorer.exe", "E:\TFM LABORATORIO\VIRUS\Hack Facebook.exe"
20:59.186", "file", "write", "C:\windows\explorer.exe", "C:\windows\System32\labtfm.exe"
20:59.201", "registry", "SetValuekey", "C:\windows\explorer.exe", "HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\LABTFM"
20:59.186", "file", "write", "E:\TFM LABORATORIO\VIRUS\Hack Facebook.exe", "E:\TFM LABORATORIO"
21:0.995", "file", "write", "System", "C:\windows\system32\labtfm.exe"

21:22.25", "file", "write", "C:\windows\system32\svchost.exe", "C:\windows\System32\winevt\Logs\System.evtx"
21:22.25", "file", "write", "C:\windows\system32\svchost.exe", "C:\windows\System32\winevt\Logs\System.evtx"
21:28.31", "file", "write", "C:\windows\system32\svchost.exe", "C:\windows\System32\winevt\Logs\Security.evtx"
21:28.31", "file", "write", "C:\windows\system32\svchost.exe", "C:\windows\System32\winevt\Logs\Security.evtx"
21:28.31", "file", "write", "C:\windows\system32\svchost.exe", "C:\windows\System32\winevt\Logs\Microsoft-Windows-CodeIntegrity
21:44.582", "file", "write", "C:\windows\system32\svchost.exe", "C:\windows\ServiceProfiles\LocalService\AppData\Local\lastalliv
22:44.659", "file", "write", "C:\windows\system32\svchost.exe", "C:\windows\ServiceProfiles\LocalService\AppData\Local\lastalliv
23:44.595", "file", "write", "C:\windows\system32\svchost.exe", "C:\windows\ServiceProfiles\LocalService\AppData\Local\lastalliv
24:44.593", "file", "write", "C:\windows\system32\svchost.exe", "C:\windows\ServiceProfiles\LocalService\AppData\Local\lastalliv
25:11.722", "process", "terminated", "C:\windows\system32\svchost.exe", "3064"

```

Figura 59. Contenido archivo "reporte.txt".

Sobre el resultado podemos observar cómo una vez ejecutado el malware sobre el sistema se crean los siguientes procesos:

- ✓ La ejecución del archivo “Hack Facebook.exe”, pasa a ser atendido por el proceso explorer.exe.

```
"15/7/2016 11:20:58.718", "process", "created", "C:\Windows\explorer.exe", "E:\TFM
LABORATORIO\VIRUS\Hack Facebook.exe"

"15/7/2016 11:20:59.186", "process", "terminated", "C:\Windows\explorer.exe", "E:\TFM
LABORATORIO\VIRUS\Hack Facebook.exe"
```

Tabla 7. Fragmento archivo "reporte.txt".

- ✓ El archivo labtfm.exe (parámetro definido en la creación del malware), pasa a ser accedido por el proceso explorer.exe, sobre el registro de Windows se ejecuta LABTM, y finalmente se escribe sobre el sistema particularmente en el directorio Windows\System32 el archivo labtfm.exe.

```
"15/7/2016
11:20:59.186", "file", "Write", "C:\Windows\explorer.exe", "C:\Windows\System32\labtfm.exe"
"15/7/2016
11:20:59.201", "registry", "SetValueKey", "C:\Windows\explorer.exe", "HKLM\SOFTWARE\Microsoft
\Windows\CurrentVersion\Run\LABTFM"
"15/7/2016 11:20:59.186", "file", "Write", "E:\TFM LABORATORIO\VIRUS\Hack
Facebook.exe", "E:\TFM LABORATORIO"
"15/7/2016 11:21:0.995", "file", "Write", "System", "C:\Windows\System32\labtfm.exe"
```

Tabla 8. Fragmento archivo "reporte.txt" (continuación).

Otra herramienta como lo es “**Disk Pulse**” aplicada sobre nuestro escenario, permite observar y comprobar los resultados respecto a los archivos y procesos generados por el malware.

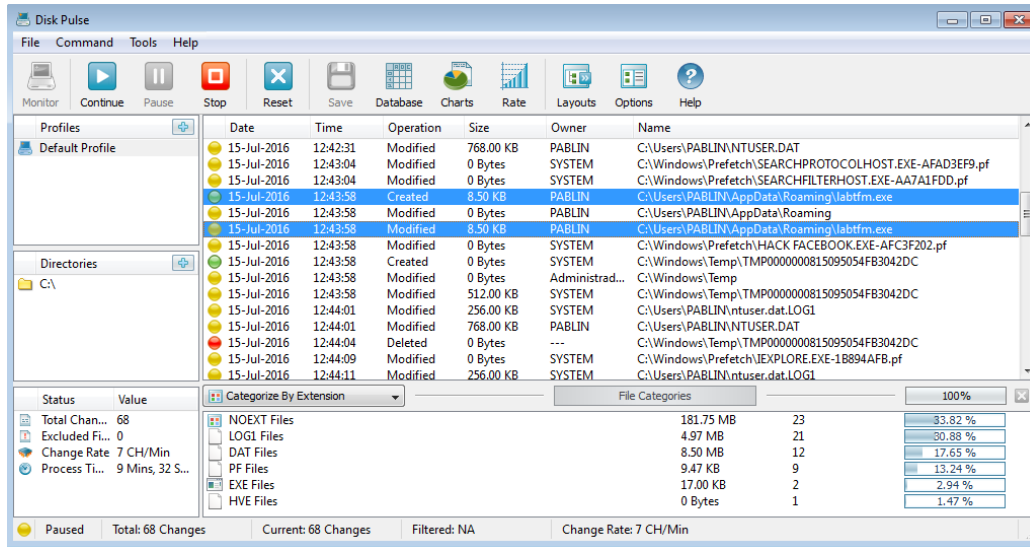


Figura 60. Disk Pulse - Resultados.

Se ejecuta la herramienta **Process Explorer**, la cual permite observar los nuevos procesos que se están ejecutando sobre la máquina víctima y así mismo registrando el sitio donde se encuentran. Estos nuevos procesos pueden modificar el registro para permitir que el malware se cargue en el arranque.

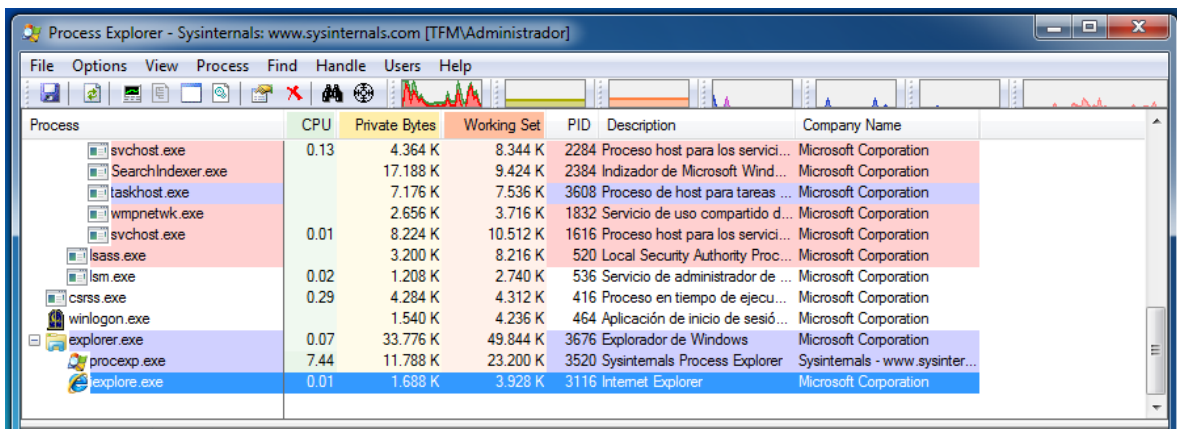


Figura 61. Process Explorer sobre la máquina víctima.

De los resultados obtenidos analizamos cada uno de los procesos con detenimiento, encontramos que existe un proceso demarcado como **PID 3116** llamado **ieexplorer.exe** (el cual hace referencia al Navegador); proceso que no hemos ejecutado y que no lo tenemos abierto en nuestro sistema.

Curiosamente si eliminamos el proceso este automáticamente vuelve a iniciar, algo sospechoso que nos da a pensar que algún proceso está inyectando código sobre explorer.exe cuando se carga en memoria.

Mediante la herramienta “**Process Monitor**” ejecutada sobre la maquina víctima, se puede visualizar la afectación del proceso Explorer.exe una vez ejecutado el malware sobre el sistema.

Time	Process Name	PID	Operation	Path	Result	Detail
15:18:...	SearchIndexer...	2796	File System Control C:		SUCCESS	Control: FSCTL_READ_USN_JOURNAL
15:18:...	SearchIndexer...	2796	File System Control C:		SUCCESS	Control: FSCTL_READ_USN_JOURNAL
15:18:...	svchost.exe	2124	QueryInformation	E:\TFM LABORATORIO\VIRUS\Hack Facebook.exe	SUCCESS	VolumeCreationTime: 0, VolumeSerialNum...
15:18:...	svchost.exe	2124	CloseFile	E:\TFM LABORATORIO\VIRUS\Hack Facebook.exe	SUCCESS	Buffer Overfl... CreationTime: 11/07/2016 18:45:09, Last...
15:18:...	WMIADAP.EXE	944	RegQueryValue	HKLM\SOFTWARE\Microsoft\WBEM\CIMOM\ThrottleDrege	SUCCESS	Type: REG_DWORD, Length: 4, Data: 1
15:18:...	Explorer.EXE	2300	WriteFile	C:\Users\PABLIN\AppData\Roaming\labtfm.exe	SUCCESS	Offset: 0, Length: 8,704, Priority: Normal
15:18:...	Explorer.EXE	2300	CloseFile	C:\Users\PABLIN\AppData\Roaming\labtfm.exe	SUCCESS	
15:18:...	svchost.exe	1124	UDP Send	TERMINAL2.TFM.LAB.CO:59678 -> SERVER.domain	SUCCESS	Length: 90, seqnum: 0, connid: 0
15:18:...	svchost.exe	1124	UDP Send	TERMINAL2.TFM.LAB.CO:59678 -> 192.168.133.2.domain	SUCCESS	Length: 90, seqnum: 0, connid: 0
15:18:...	SearchIndexer...	2796	File System Control C:		SUCCESS	Control: FSCTL_QUERY_USN_JOURNA
15:18:...	SearchIndexer...	2796	File System Control C:		SUCCESS	Control: FSCTL_READ_USN_JOURNAL
15:18:...	svchost.exe	1124	UDP Send	TERMINAL2.TFM.LAB.CO:64518 -> SERVER.domain	SUCCESS	Length: 42, seqnum: 0, connid: 0
15:18:...	svchost.exe	1124	UDP Send	TERMINAL2.TFM.LAB.CO:64518 -> 192.168.133.2.domain	SUCCESS	Length: 42, seqnum: 0, connid: 0
15:18:...	WMIADAP.EXE	944	RegQueryValue	HKLM\SOFTWARE\Microsoft\WBEM\CIMOM\ThrottleDrege	SUCCESS	Type: REG_DWORD, Length: 4, Data: 1
15:18:...	svchost.exe	740	WriteFile	C:\Windows\System32\wininit\Logs\System.evtx	SUCCESS	Offset: 856,064, Length: 512
15:18:...	svchost.exe	740	WriteFile	C:\Windows\System32\wininit\Logs\System.evtx	SUCCESS	Offset: 915,344, Length: 632
15:18:...	WMIADAP.EXE	944	RegQueryValue	HKLM\SOFTWARE\Microsoft\WBEM\CIMOM\ThrottleDrege	SUCCESS	Type: REG_DWORD, Length: 4, Data: 1
15:18:...	WMIADAP.EXE	944	RegQueryValue	HKLM\SOFTWARE\Microsoft\WBEM\CIMOM\ThrottleDrege	SUCCESS	Type: REG_DWORD, Length: 4, Data: 1
15:18:...	WMIADAP.EXE	944	RegQueryValue	HKLM\SOFTWARE\Microsoft\WBEM\CIMOM\ThrottleDrege	SUCCESS	Type: REG_DWORD, Length: 4, Data: 1
15:18:...	Explorer.EXE	2300	CreateFile	E:\TFM LABORATORIO\VIRUS\Hack Facebook.exe	SUCCESS	Desired Access: Read Attributes, Delete...
15:18:...	Explorer.EXE	2300	QueryAttribute T	E:\TFM LABORATORIO\VIRUS\Hack Facebook.exe	INVALID PARAM...	Delete: True
15:18:...	Explorer.EXE	2300	SetDisposition	E:\TFM LABORATORIO\VIRUS\Hack Facebook.exe	SUCCESS	Type: REG_DWORD, Length: 4, Data: 1
15:18:...	WMIADAP.EXE	944	RegQueryValue	HKLM\SOFTWARE\Microsoft\WBEM\CIMOM\ThrottleDrege	SUCCESS	Desired Access: Read Attributes, Synchron...
15:18:...	svchost.exe	2124	CreateFile	E:\TFM LABORATORIO\VIRUS\Hack Facebook.exe	DELETE PENDING	Offset: 0, Length: 4,096, I/O Flags: Non-c...
15:18:...	Explorer.EXE	2300	WriteFile	E:\TFM LABORATORIO\VIRUS	SUCCESS	Type: REG_DWORD, Length: 4, Data: 1
15:18:...	WMIADAP.EXE	944	RegQueryValue	HKLM\SOFTWARE\Microsoft\WBEM\CIMOM\ThrottleDrege	SUCCESS	Type: REG_DWORD, Length: 4, Data: 1
15:18:...	Explorer.EXE	2300	CloseFile	E:\TFM LABORATORIO\VIRUS\Hack Facebook.exe	SUCCESS	
15:18:...	Explorer.EXE	2300	NotifyChangeDi	E:\TFM LABORATORIO\VIRUS	SUCCESS	Filter: FILE_NOTIFY_CHANGE_FILE_NA...
15:18:...	Explorer.EXE	2300	WriteFile	E:\TFM LABORATORIO\VIRUS	SUCCESS	Offset: 0, Length: 4,096, I/O Flag...
15:18:...	Explorer.EXE	2300	WriteFile	E:	SUCCESS	Offset: 1,073,152, Length: 4,096, I/O Flag...
15:18:...	WMIADAP.EXE	944	RegQueryValue	HKLM\SOFTWARE\Microsoft\WBEM\CIMOM\ThrottleDrege	SUCCESS	Type: REG_DWORD, Length: 4, Data: 1
15:18:...	WMIADAP.EXE	944	RegQueryValue	HKLM\SOFTWARE\Microsoft\WBEM\CIMOM\ThrottleDrege	SUCCESS	Type: REG_DWORD, Length: 4, Data: 1
15:18:...	Explorer.EXE	2300	RegOpenKey	HKU\S-1-5-21-3022253727-3793539035-4224281074-1114\Software\Microsoft\Wind...	SUCCESS	Desired Access: All Access
15:18:...	Explorer.EXE	2300	RegSetValue	HKU\S-1-5-21-3022253727-3793539035-4224281074-1114\Software\Microsoft\Wind...	SUCCESS	Type: REG_SZ, Length: 510, Data: C:\Uk...
15:18:...	Explorer.EXE	2300	RegCloseKey	HKU\S-1-5-21-3022253727-3793539035-4224281074-1114\Software\Microsoft\Wind...	SUCCESS	
15:18:...	Explorer.EXE	2300	Thread Create		SUCCESS	Thread ID: 3292
15:18:...	Explorer.EXE	2300	RegOpenKey	HKCR\http\shell\open\command	SUCCESS	Desired Access: Query Value
15:18:...	Explorer.EXE	2300	RegQueryValue	HKCR\http\shell\open\command	SUCCESS	Type: REG_SZ, Length: 112, Data: C:\P...

Figura 62. Process Monitor en ejecución.

Utilizando la herramienta **AutoRuns**, la cual nos permite visualizar los programas configurados para ejecutarse durante el inicio del sistema o el inicio de sesión, y muestra las entradas en el orden en que *Windows* las procesa, nos permite identificar la existencia de un registro en la clave “HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run” con un nombre un tanto sospechoso “LABTFM”, el cual invoca la ejecución del archivo labtfm.exe ubicado en la ruta c:/Windows/system32/.

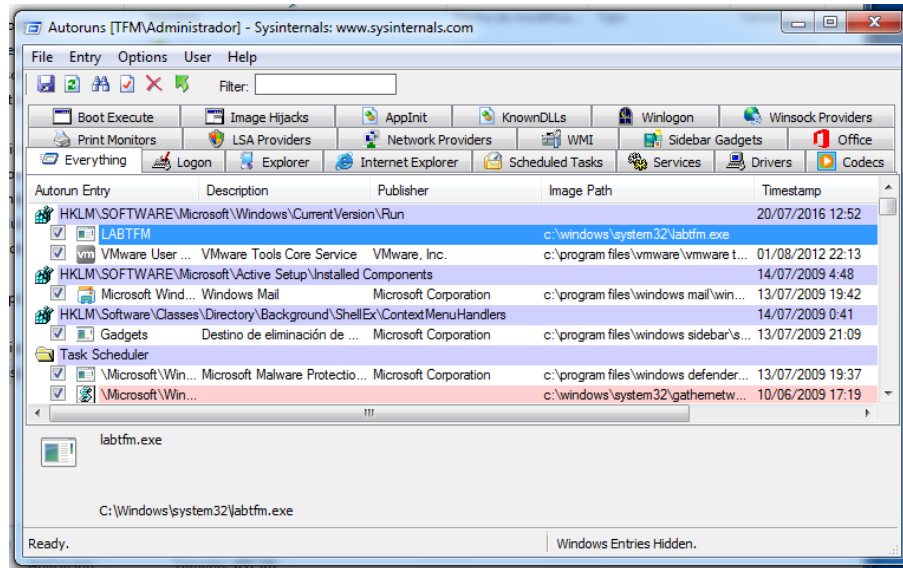
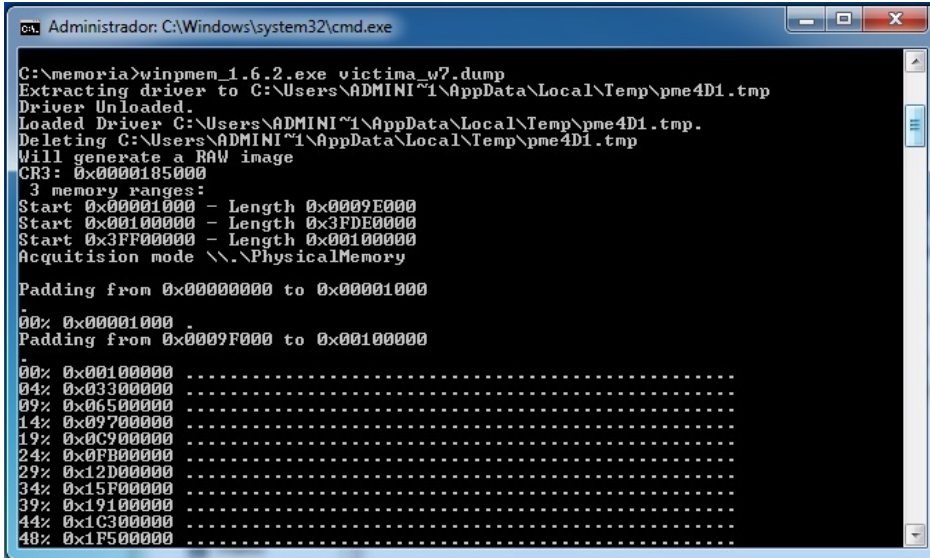


Figura 63. Autoruns sobre la máquina víctima.

7.4.2. Análisis del volcado de Memoria

Con el fin de identificar que otros procesos fueron afectados tras la ejecución del malware sobre el sistema se utilizó la herramienta Volatility, la cual desde el campo forense permite realizar un análisis a un volcado previo de memoria.

Para su desarrollo, un primer paso fue obtener sobre la maquina victima un volcado de la memoria utilizando la herramienta winpmem en su versión 1.6.2, obteniendo como resultado un archivo al que se lo denominó victima_w7.dump con un tamaño de 1 Gb.



```

C:\Windows\system32\cmd.exe
C:\memoria>winpmem_1.6.2.exe victima_w7.dump
Extracting driver to C:\Users\ADMINI~1\AppData\Local\Temp\pme4D1.tmp
Driver Unloaded.
Loaded Driver C:\Users\ADMINI~1\AppData\Local\Temp\pme4D1.tmp.
Deleting C:\Users\ADMINI~1\AppData\Local\Temp\pme4D1.tmp
Will generate a RAW image
CR3: 0x0000185000
3 memory ranges:
Start 0x00001000 - Length 0x0009E000
Start 0x00100000 - Length 0x3FDE0000
Start 0x3FF00000 - Length 0x00100000
Acquisition mode \\.\PhysicalMemory

Padding from 0x00000000 to 0x00001000
.
00% 0x00001000 .
Padding from 0x0009F000 to 0x00100000
.
00% 0x00100000 .....
04% 0x03300000 .....
09% 0x06500000 .....
14% 0x09700000 .....
19% 0x0C900000 .....
24% 0x0FB00000 .....
29% 0x12D00000 .....
34% 0x15F00000 .....
39% 0x19100000 .....
44% 0x1C300000 .....
48% 0x1F500000 .....

```

Figura 64. Volcado de memoria Winpmem.

A continuación el archivo obtenido “victima_w7.exe”, fue analizado con la herramienta Volatility. La conjugación de las diferentes opciones y/o funciones que ofrece, pudo establecer los siguientes resultados:

Como primera medida mediante “imageinfo”, se identificó el sistema respecto al volcado de memoria, obteniendo como resultado Win7SP1x86 y Win7SP0x86.

```

C:\Python27>volatility-2.4.standalone.exe -f victima_w7.dump imageinfo
Volatility Foundation Volatility Framework 2.4
Determining profile based on KDBG search...

Suggested Profile(s) : Win7SP0x86, Win7SP1x86
AS Layer1 : IA32PagedMemoryPae (Kernel AS)
AS Layer2 : FileAddressSpace (C:\Python27\victivirus)
PAE type : PAE
DTB : 0x185000L
KDBG : 0x82938be8L
Number of Processors : 1
Image Type (Service Pack) : 0
KPCR for CPU 0 : 0x82939c00L
KUSER_SHARED_DATA : 0xffdf0000L
Image date and time : 2016-09-19 17:58:46 UTC+0000
Image local date and time : 2016-09-19 13:58:46 -0400

```

Tabla 9. Resultado imageinfo – volatility sobre el volcado de memoria.

- Un primer paso, fue el tratar de establecer mediante la función “connections” los puertos que se encontraban activos y que procesos los estaban utilizando, sin embargo, se pudo establecer y comprobar que volatility no ofrece compatibilidad de análisis con el perfil win7SP71x86.
- Se estableció la lista de procesos que se encontraban activos en la máquina. Al respecto de los hallazgos encontrados anteriormente, se hizo énfasis sobre el proceso explorer.exe, identificado con el PID 2344.

```
C:\Python27>volatility-2.4.standalone.exe --profile=Win7SP1x86 -f victima_w7.dump pslist
Volatility Foundation Volatility Framework 2.4
Offset(V) Name PID PPID Thds Hnds Sess Wow64 Start Exit
-----
....
0x8737dc70 svchost.exe 2140 512 14 369 0 0 2016-09-19 17:56:27 UTC+0000
0x86bf0030 taskhost.exe 2280 512 12 170 1 0 2016-09-19 17:56:35 UTC+0000
0x86c1b4b0 dwm.exe 2332 856 4 69 1 0 2016-09-19 17:56:35 UTC+0000
0x86f29030 explorer.exe 2344 2320 22 778 1 0 2016-09-19 17:56:35 UTC+0000
...
```

Tabla 10. Fragmento resultado pslist – volatility sobre el volcado de memoria.

- Ahondando más sobre los resultados, mediante la función “pstree”, se pudo establecer la relación y dependencia de los procesos ejecutados sobre la víctima, entre ellos vmtoolsd.exe, cmd.exe, winpmem_1.6.2.exe e iexplorer.exe los cuales dependen del proceso explorer.exe PID 2344. La ejecución de iexplorer.exe PID 3972 causa algo de sospecha, toda vez que sobre la máquina hasta el momento de realizar el volcado de memoria no se había ejecutado ningún proceso que invocara este servicio.

```
C:\Python27>volatility-2.4.standalone.exe --profile=Win7SP1x86 -f victima_w7.dump pstree
Volatility Foundation Volatility Framework 2.4
Name Pid PPid Thds Hnds Time
-----
0x86f29030:explorer.exe 2344 2320 22 778 2016-09-19 17:56:35 UTC+0000
. 0x86c0f5f8:vmtoolsd.exe 2564 2344 8 180 2016-09-19 17:56:36 UTC+0000
```

. 0x86c3e030:cmd.exe	840	2344	1	24	2016-09-19 17:58:04 UTC+0000
.. 0x8759cc60:winpmem_1.6.2.	3564	840	1	21	2016-09-19 17:58:40 UTC+0000
. 0x84fe2478:iexplore.exe	3972	2344	3	35	2016-09-19 17:57:12 UTC+0000

Tabla 11. Fragmento resultado pstree - volatility sobre volcado de memoria.

- Mediante la función “malfind”, la cual tiene varios propósitos: se puede utilizar para encontrar el código / DLL ocultos o inyectados en la memoria de modo de usuario, basados en características como la etiqueta de VAD y los permisos de página, así como también se puede utilizar para localizar cualquier secuencia de bytes, expresiones regulares, cadenas ANSI, o cadenas Unicode en modo de usuario o la memoria del núcleo (Google, 2016).

Se pudo identificar alrededor de veinte (20) segmentos de código inyectado sobre la máquina víctima, esta situación tal como se expuso anteriormente hace que el análisis sea mucho más desafiante.

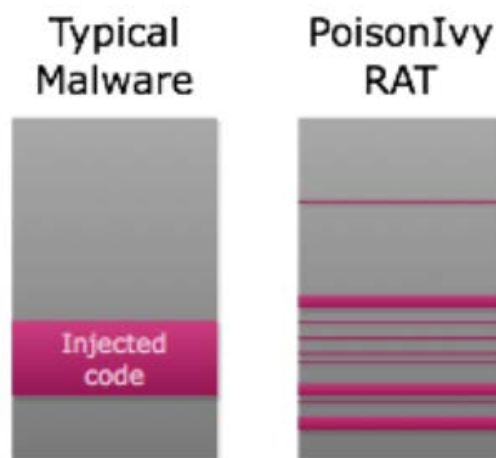


Figura 65. Comparativo inyección de código Poison Ivy Vs otros Malware (Hale, 2016).

Se presentan algunos apartes de los resultados obtenidos con malfind, los cuales pueden ser consultados en su totalidad el anexo E. Estos fragmentos permiten visualizar que la mayoría de los segmentos empiezan con instrucciones como EBX PUSH EBP o PUSH; funciones individuales que Poison Ivy dispersa en toda la memoria de explorer.exe.

```

C:\Python27>volatility-2.4.standalone.exe --profile=Win7SP1x86 -f victima_w7.dump malfind --
dump-dir victima
Volatility Foundation Volatility Framework 2.4
Process: explorer.exe Pid: 2344 Address: 0x1f50000
Vad Tag: VadS Protection: PAGE_EXECUTE_READWRITE
Flags: CommitCharge: 1, MemCommit: 1, PrivateMemory: 1, Protection: 6

0x01f50000 55 8b ec 83 c4 cc 8b 75 08 6a 00 68 80 00 00 00 U.....u.j.h....
0x01f50010 6a 03 6a 00 6a 00 68 00 00 00 80 8d 86 b1 06 00 j.j.j.h.....
0x01f50020 00 50 ff 56 59 50 68 88 b6 b6 fc ff b6 bf 0a 00 .P.VYPh.....
0x01f50030 00 ff b6 e1 00 00 00 ff 96 dd 00 00 00 89 45 e0 .....E.

0x1f50000 55      PUSH EBP
0x1f50001 8bec     MOV EBP, ESP
0x1f50003 83c4cc   ADD ESP, -0x34
0x1f50006 8b7508   MOV ESI, [EBP+0x8]
0x1f50009 6a00     PUSH 0x0
0x1f5000b 6880000000 PUSH DWORD 0x80
0x1f50010 6a03     PUSH 0x3
0x1f50012 6a00     PUSH 0x0
0x1f50014 6a00     PUSH 0x0
0x1f50016 6800000080 PUSH DWORD 0x80000000
0x1f5001b 8d86b1060000 LEA EAX, [ESI+0x6b1]
0x1f50021 50      PUSH EAX
0x1f50022 ff5659   CALL DWORD [ESI+0x59]
0x1f50025 50      PUSH EAX
0x1f50026 6888b6b6fc PUSH DWORD 0xfcb6b688
0x1f5002b ffb6bf0a0000 PUSH DWORD [ESI+0xabf]
0x1f50031 ffb6e1000000 PUSH DWORD [ESI+0xe1]
0x1f50037 ff96dd000000 CALL DWORD [ESI+0xdd]
0x1f5003d 8945e0   MOV [EBP-0x20], EAX

Process: iexplore.exe Pid: 3972 Address: 0x50000
Vad Tag: VadS Protection: PAGE_EXECUTE_READWRITE
Flags: CommitCharge: 1, MemCommit: 1, PrivateMemory: 1, Protection: 6

0x00050000 55 8b ec 81 c4 30 fa ff ff 8b 75 08 8d 86 fb 03 U....0....u.....
0x00050010 00 00 50 6a 00 6a 00 ff 96 85 00 00 00 89 86 c5 ..P.j.....

```



```

0x00050020 08 00 00 ff 96 89 00 00 00 3d b7 00 00 00 75 04 .....=....u.
0x00050030 c9 c2 04 00 56 8d 86 6b 09 00 00 50 8d 86 45 01 ....V..k...P..E.

0x50000 55          PUSH EBP
0x50001 8bec        MOV EBP, ESP
0x50003 81c430faffff  ADD ESP, 0xfffffa30
0x50009 8b7508        MOV ESI, [EBP+0x8]
0x5000c 8d86fb030000    LEA EAX, [ESI+0x3fb]
0x50012 50          PUSH EAX
0x50013 6a00        PUSH 0x0
0x50015 6a00        PUSH 0x0
0x50017 ff9685000000    CALL DWORD [ESI+0x85]
0x5001d 8986c5080000    MOV [ESI+0x8c5], EAX
0x50023 ff9689000000    CALL DWORD [ESI+0x89]
0x50029 3db7000000    CMP EAX, 0xb7
0x5002e 7504        JNZ 0x50034
0x50030 c9          LEAVE
0x50031 c20400      RET 0x4
0x50034 56          PUSH ESI
0x50035 8d866b090000    LEA EAX, [ESI+0x96b]
0x5003b 50          PUSH EAX
0x5003c 8d          DB 0x8d
0x5003d 864501      XCHG [EBP+0x1], AL

```

Process: iexplore.exe Pid: 3972 Address: 0x60000

Vad Tag: VadS Protection: PAGE_EXECUTE_READWRITE

Flags: CommitCharge: 1, MemCommit: 1, PrivateMemory: 1, Protection: 6

```

0x00060000 00 00 3f c9 77 be 48 c9 77 ed 3b c9 77 c8 c4 c9 ..?.w.H.w.;w...
0x00060010 77 df 47 c9 77 ab 2f c9 77 34 32 c9 77 33 71 ca w.G.w./w42.w3q.
0x00060020 77 f4 05 32 76 35 0d 32 76 fd 27 32 76 62 20 2d w..2v5.2v.'2vb.-
0x00060030 76 d4 be f3 75 0d bc f3 75 25 bc f3 75 96 1b f3 v...u...u%..u...

```

```

0x60000 0000      ADD [EAX], AL
0x60002 3f          AAS
0x60003 c9          LEAVE
0x60004 77be       JA 0x5ffc4
0x60006 48          DEC EAX
0x60007 c9          LEAVE
0x60008 77ed       JA 0x5fff7

```

0x6000a 3bc9	CMP ECX, ECX
0x6000c 77c8	JA 0x5ffd6
0x6000e c4	DB 0xc4
0x6000f c9	DB 0xc9
0x60010 77	DB 0x77
0x60011 df	DB 0xdf
0x60012 47	INC EDI
0x60013 c9	LEAVE
0x60014 77ab	JA 0x5ffc1
0x60016 2f	DAS
0x60017 c9	LEAVE
0x60018 7734	JA 0x6004e
0x6001a 32c9	XOR CL, CL
0x6001c 7733	JA 0x60051
0x6001e 71ca	JNO 0x5ffea
0x60020 77f4	JA 0x60016
0x60022 053276350d	ADD EAX, 0xd357632
0x60027 3276fd	XOR DH, [ESI-0x3]
0x6002a 27	DAA
0x6002b 327662	XOR DH, [ESI+0x62]
0x6002e 202d76d4bef3	AND [0xf3bed476], CH
0x60034 750d	JNZ 0x60043
0x60036 bcf37525bc	MOV ESP, 0xbc2575f3
0x6003b f37596	JNZ 0x5ffd4
0x6003e 1bf3	SBB ESI, EBX

Tabla 12. Fragmento resultado maldfind - volatility sobre volcado de memoria.

- Sobre el último segmento obtenidos con maldfind process.0x84fe2478.0x60000.dmp presentado en la tabla anterior, mediante “strings” se visualizó su contenido resaltando la existencia del archivo configurado, así como también la conexión IP con el C &C, el mutex y el archivo del malware ejecutado.

```
C:\Python27\victima>strings process.0x84fe2478.0x60000.dmp
```

```
Strings v2.53 - Search for ANSI and Unicode strings in binary images.
```

```
Copyright (C) 1999-2016 Mark Russinovich
```

Sysinternals - www.sysinternals.com

w42

w3q

2v5

'2vb -v

|3v

(2v

=kvRThv

pev

42vb

iv>khv

1v~

2vw

D1vk

s1v

wd(2v

2vf

1v Q

+1v

p3v

q3v

5vr>1v

labtfm.exe

`"C

192.168.0.1

)!VoqA.I4

StubPath

SOFTWARE\Classes\http\shell\open\command

Software\Microsoft\Active Setup\Installed Components\

C:\virus\Hack Facebook.exe

C:\Users\PABLIN\AppData\Roaming\labtfm.exe

a\Roaming

J;7

r3@5

yz-

RUa,

`"C

}4S

I,Sa

```
U(w  
e9P  
$;r  
:W9  
9nqN  
e9P  
wR8  
INa )  
TFM_LAB  
G]1v  
SOFTWARE\Microsoft\Windows\CurrentVersion\Run  
LABTFM
```

Tabla 13. Resultado strings sobre process.0x84fe2478.0x60000.dmp.

7.5. Resumen del análisis desarrollado

Con el fin verificar la validez de la metodología propuesta en la Tesis Doctoral desarrollada por Don Javier Bermejo (2015), su aplicación se desarrolló frente a un escenario de análisis de un APT basado en la RAT conocida como Poison Ivy; una herramienta cuya fuente original dejó de lado su desarrollo y soporte, sin embargo se pudo establecer que existen aún fuentes que siguen ofreciendo esta herramienta. El objetivo de este análisis, era obtener una perspectiva global respecto a su funcionamiento y mecanismos empleados, su estructura y composición, clasificación, y otra serie de características propias y específicas del malware con la aplicación de la metodología base del presente piloto experimental.

Como resultado al análisis encontramos una herramienta sofisticada, la cual ofrece un nivel complejo de estudio, maneja técnicas de ofuscación, encriptación y una cantidad de módulos y elementos que demandan mucho más tiempo en un análisis profundo, así como también técnicas adicionales como el conocimiento elevado de ingeniería inversa.

Se pudo establecer que ofrece técnicas para mantenerse oculta en el sistema, su entorno es totalmente amigable con el usuario final y actualmente existen contables fuentes de información, que si bien es cierto son pocas, ofrecen suficientes detalles respecto a su instalación, configuración y funcionamiento, entre ellos: manuales de usuario, parches, entre otros. Sin embargo, por el tiempo que lleva en el mercado desde su aparición, hoy en día existen múltiples herramientas genéricas que permiten su detección.

Respecto a la muestra obtenida se pudo obtener los siguientes resultados:

✓ **Identificación**

Kit cliente/Servidor:

Nombre: Poison Ivy 2.3.2

Tipo: EXE, PE32 ejecutable Windows

Tamaño: 2.141.878 bytes

Hash md5: b4f990cad1d20efab410e98fc7a6c81b

Parche para el kit en plataforma Windows 7:

Nombre: Reko24.exe

Tipo: EXE, PE32 ejecutable Windows

Tamaño: 2.048 Bytes

Hash md5: 22577c60a76c77adb433a3406e6a7c56

Troyano generado con el kit Poison Ivy:

Nombre: Hack Facebook.exe

Tipo: EXE, PE32 ejecutable Windows

Tamaño: 8.500 Bytes

Hash md5: de1cbe2d617a75f9c768f3a1df3e6aa7

Detecciones antivirus actuales:

ANTIVIRUS	DETECCION	FECHA REVISION
Ad-Aware	GenPack:Backdoor.Generic.81276	20160705
Avast	Win32:Evo-gen [Susp]	20160705
AVG	Win32/Heur	20160705
Avira (no cloud)	BDC/PoisonIvy.A	20160705
BitDefender	GenPack:Backdoor.Generic.81276	20160705
ESET-NOD32	Win32/RemoteAdmin.PoisonIvy potentially unsafe	20160705
Kaspersky	Backdoor.Win32.Poison.cww	20160705
Kingsoft	Win32.Hack.Poison.2199552.(kcloud)	20160705
McAfee	BackDoor-DIQ	20160705
Microsoft	Backdoor:Win32/Poison.E	20160705
Panda	Application/PoisonIvy	20160704
Symantec	Backdoor.ConstructKit	20160630
TrendMicro	BKDR_POISON.BUR	20160705

✓ Características

A diferencia de otros malware, Poison Ivy utiliza la técnica conocida como reverse Shell o "Server", genera el archivo servidor el cual va a ser instalado en la maquina víctima, y será controlado desde un entorno gráfico cliente previamente instalado en la maquina desde donde se tendrá control de la herramienta. Su proceso de infección empieza desde la ejecución del archivo generado por el kit Poison Ivy, el

cual como dato curioso, desaparece inmediatamente tras su ejecución no dejando rastro de su presencia.

Proceso de infección:

- ✓ Empieza infectando la librería Kernel32.dll, reflejando su ejecución en el proceso Explorer.exe. El archivo labtfm.exe (parámetro definido en la creación del malware), pasa a ser accedido por el proceso explorer.exe, sobre el registro de Windows se ejecuta LABTM, y finalmente se escribe sobre el sistema particularmente en el directorio Windows\System32 el archivo labtfm.exe

Interacción Mando y Control

- ✓ Como se expuso anteriormente, la configuración de Poison Ivy requiere definir una serie de parámetros que serán útiles en el proceso de comunicación, entre ellos: la definición de un puerto, una IP de comunicación, y una contraseña (caso particular puerto 3640, IP 192.168.0.1).
- ✓ La comunicación se encuentra bajo un sistema de cifrado “Camelia”, un algoritmo bien reconocido y definido por el RFC 3713, estableciendo una comunicación TCP / IP con el servidor.
- ✓ Para la comunicación con el sistema de mando y control, Poison Ivy hace uso de los procesos “explorer.exe” e “iexplore.exe”.

Modificaciones de registro

- ✓ La estructura del programa refiere que los datos y la llamada de sus funciones a los que hace referencia, los realiza a través del registro ESI. De esta forma, puede ser utilizado por el servidor para esconderse muy bien en el sistema, y como un mecanismo para frustrar el análisis estático.

- ✓ Encontramos sobre el registro una clave denominada “HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run”, en la cual se hace un llamado a un proceso definido al momento de crear el troyano, que para el caso particular es labtfm.exe.

Técnicas de ofuscación

Inicialmente no se pudo determinar o establecer parámetros reales respecto a las técnicas de ofuscación, empaquetamiento o entropía, ya que la herramienta PEID arrojaba resultados como información basura, archivo no empaquetado y una entropía de 6,17, la cual resultaba muy baja.

Sin embargo, tras la ejecución de una nueva herramienta denominada DiE, permitió sobre el troyano, establecer la existencia del compilador basado en MASM32, la existencia de bloques comprimidos en el archivo analizado y una entropía total del 75%.

Dependencias o librerías

Se pudo establecer la existencia de las siguientes librerías afectadas, ubicadas en la ruta c:\windows\system32: API-MS-WIN-CORE-DEBUG-L1-1-0.DLL , API-MS-WIN-CORE-ERRORHANDLING-L1-1-0.DLL, API-MS-WIN-CORE-FIBERS-L1-1-0.DLL, API-MS-WIN-CORE-FILE-L1-1-0.DLL, API-MS-WIN-CORE-HANDLE-L1-1-0.DLL, API-MS-WIN-CORE-HEAP-L1-1-0.DLL, API-MS-WIN-CORE-IO-L1-1-0.DLL, API-MS-WIN-CORE-LIBRARYLOADER-L1-1-0.DLL, \API-MS-WIN-CORE-LOCALIZATION-L1-1-0.DLL, API-MS-WIN-CORE-MEMORY-L1-1-0.DLL, API-MS-WIN-CORE-MISC-L1-1-0.DLL, API-MS-WIN-CORE-NAMEDPIPE-L1-1-0.DLL, API-MS-WIN-CORE-PROCESSENVIRONMENT-L1-1-0.DLL, API-MS-WIN-CORE-PROCESSTHREADS-L1-1-0.DLL, API-MS-WIN-CORE-PROFILE-L1-1-0.DLL, API-MS-WIN-CORE-RTLSUPPORT-L1-1-0.DLL, API-MS-WIN-CORE-STRING-L1-1-0.DLL, API-MS-WIN-CORE-SYNCH-L1-1-0.DLL, API-MS-WIN-CORE-SYSINFO-L1-1-0.DLL, API-MS-WIN-CORE-THREADPOOL-L1-1-0.DLL, API-MS-WIN-CORE-

UTIL-L1-1-0.DLL, API-MS-WIN-SECURITY-BASE-L1-1-0.DLL, c:\string\HACK
FACEBOOK.EXE, KERNEL32.DLL, \KERNELBASE.DLL, NTDLL.DLL

Recomendaciones ante el incidente

Si bien es cierto hoy en día existen muchos antivirus genéricos que automatizan la y eliminación de Poison Ivy, su eliminación de forma manual puede considerarse como una opción. Para ello, podríamos levantar la máquina infectada con un Live CD y así eliminar los registros y ficheros que fueron detectados y realizan operaciones con el explorer.exe, así como también el fichero o archivo oculto, para nuestro caso particular labtfm.exe.

Conclusiones

El desarrollo del presente piloto experimental, ha conllevado a descubrir en Las APTs un arma sofisticada de ataque, la cual está diseñada para perseguir blancos específicos y utilizan las RAT para mantener la persistencia los ataques sobre la víctima. Para ello aplica cualquier cantidad de métodos que les permite escalar en el sistema, obteniendo cada vez más privilegios para extraer información confidencial. Las RAT que se emplean para desarrollar estos ataques están ampliamente disponibles y documentadas, pudiendo complicar la labor de identificar cuando un ataque hace parte de una campaña APT.

El estudio y análisis de la Amenaza Avanzada Persistente “Poison Ivy”, como muestra seleccionada para el desarrollo del presente piloto experimental, ha permitido conocer características propias y únicas frente a su comportamiento, una vez ha infectado una maquina víctima. Ofrece un nivel de complejidad alto respecto a su análisis, reflejado en la necesidad de emplear recursos especializados desde el campo técnico y humano para comprender en profundidad su funcionamiento. Desde el campo de aplicación, se pudo encontrar, en comparación con otras RAT, que el manejo de Poison IVY es muy sencillo debido a que ofrece una interfaz gráfica de usuario (GUI), la cual facilita la creación de nuevos servidores y el control de los blancos afectados.

Sin duda alguna la aplicación de la metodología utilizada como referencia del presente piloto experimental, el desarrollo de actividades y procedimientos, su aplicación en el orden expuesto, demuestra una validez total frente al tema del análisis de malware permitiendo obtener resultados coherentes y eficaces. Aunque el material cubierto aquí es realmente la punta muy pequeña del iceberg relacionado con el análisis de malware, realmente aporta conocimientos y técnicas necesarios para adentrarnos un poco a este tema bastante extenso y apasionante.

Queda demostrado que la aplicación metódica de las fases contenidas en la metodología aquí desarrollada, ha permitido generar una línea base de aplicación orientando de forma efectiva las diferentes pruebas realizadas. Se resalta la innovación propuesta en la

metodología, de intercambiar o anteponer el análisis dinámico de código antes que el análisis estático que facilita sin duda alguna el estudio del malware y sus resultados.

Finalmente, el paradigma de la innovación tecnológica redundante también en las herramientas empleadas y sugeridas para desarrollar el análisis, es así como en la actualidad muchas de ellas ya no están disponibles en el mercado o han sufrido algunos cambios significativos, lo cual podría en algún momento, dificultar la aplicación de la metodología desarrollada en el piloto experimental.

ANEXOS

Se presenta toda la documentación adicional en referencia de los resultados obtenidos frente al análisis adelantado en el desarrollo del presente piloto experimental

ANEXO A: CLASIFICACION – RESULTADOS VIRUS TOTAL.

Listado de herramientas que detectan el malware



SHA256: c71d8085544e6f81e0301d9dd5cdf88369339a6001bab8e4fda22de9ec0fee31

File name: Poison Ivy 2.3.2.exe

Detection ratio: 50 / 55

Analysis date: 2016-07-05 10:05:50 UTC (4 days, 17 hours ago)

[Analysis](#)
[File detail](#)
[Relationships](#)
[Additional information](#)
[Comments 10+](#)
[Votes](#)

Antivirus	Result	Update
Ad-Aware	GenPack:Backdoor.Generic.81276	20160705
AegisLab	Backdoor.W32.Poison!c	20160705
AhnLab-V3	Trojan/Win32.Poison.N2849045	20160705
Alibaba	✓	20160705
ALYac	GenPack:Backdoor.Generic.81276	20160705
Antiy-AVL	Trojan[Downloader]/Win32.Delf	20160705
Arcabit	GenPack:Backdoor.Generic.D13D7C	20160705
Avast	Win32:Evo-gen [Susp]	20160705
AVG	Win32/Heur	20160705
Avira (no cloud)	BDC/PoisonIvy.A	20160705
AVware	Trojan.Win32.Generic.pak!cobra	20160705
Baidu	Win32.Backdoor.Poison.a	20160705
BitDefender	GenPack:Backdoor.Generic.81276	20160705
Bkav	✓	20160704
CAT-QuickHeal	Backdoor.APT.Poison.S9	20160705
ClamAV	Win.Downloader.24485-1	20160705
CMC	Generic.Win32.b4f990cad1!MD	20160704

Comodo	ApplicUnsaf.Win32.RemoteAdmin.PoisonIvy	20160705
Cyren	W32/Backdoor.KEDZ-1165	20160705
DrWeb	Trojan.PWS.Gamania.32505	20160705
Emsisoft	GenPack.Backdoor.Generic.81276 (B)	20160704
eScan	GenPack.Backdoor.Generic.81276	20160705
ESET-NOD32	Win32/RemoteAdmin.PoisonIvy potentially unsafe	20160705
F-Prot	W32/Backdoor2.HCGS	20160705
F-Secure	GenPack.Backdoor.Generic.81276	20160705
Fortinet	☑	20160705
GData	GenPack.Backdoor.Generic.81276	20160705
Ikarus	Virus.Win32.JunkPoty	20160705
Jiangmin	Backdoor/Agent.bmlv	20160705
K7AntiVirus	Trojan (7000000f1)	20160705
K7GW	Trojan (7000000f1)	20160705
Kaspersky	Backdoor.Win32.Poison.qww	20160705
Kingsoft	Win32.Hack.Poison.2199552.(kcloud)	20160705
Malwarebytes	Backdoor.PoisonIvy	20160705
McAfee	BackDoor-DIQ	20160705
McAfee-GW-Edition	BackDoor-DIQ	20160705
Microsoft	Backdoor.Win32/Poison.E	20160705
NANO-Antivirus	Trojan.Win32.PoisonI.ibcl	20160705
nProtect	Abuse-Worry/W32.RAdmin.2141878	20160705
Panda	Application/Poisonivy	20160704
Qihoo-360	Win32/Backdoor.33b	20160705
Sophos	Mal/EndPk.CI	20160705
SUPERAntiSpyware	☑	20160705
Symantec	Backdoor.ConstructKit	20160630
Tencent	Win32.Backdoor.Generic.Auto	20160705
TheHacker	Backdoor/Poison.qq	20160705

TrendMicro	BKDR_POISON_BUR	20160705
TrendMicro-HouseCall	BKDR_POISON_BUR	20160705
VBA32	BackDoor.Poison	20160705
VIPRE	Trojan.Win32.Generic.pak/coobra	20160705
ViRobot	RemoteApp.PoisonIvy.2141878[h]	20160705
Yandex	Backdoor.PoisonI+BEF1eYjgh4	20160705
Yandex	Backdoor.PoisonI+BEF1eYjgh4	20160705
Zillya	Trojan.Black.Win32.2013	20160705
Zoner		20160705

ANEXO B: CLASIFICACION – RESULTADOS VIRUS TOTAL SOBRE “Hack Facebook.exe”



SHA256: ff2d15569d8a8eeb93681a543b781578db9093f927659a14d43f01dbd39acba0

File name: Hack Facebook.exe

Detection ratio: 54 / 55

Analysis date: 2016-07-10 21:09:38 UTC (1 minute ago)

Analysis File detail Additional information Comments Votes

Antivirus	Result	Update
ALYac	Generic.PoisonIvy.F2AADF09	20160710
AVG	Win32/Agent.BB	20160710
AVware	Backdoor.Win32.Poison.Pg (v)	20160710
Ad-Aware	Generic.PoisonIvy.F2AADF09	20160710
AegisLab	Packer.W32.Katusha.linF	20160710
Yandex	Trojan.DL.CKSPost.Gen	20160709
AhnLab-V3	Trojan/Win32.Poison.R2018	20160710
Antiy-AVL	Trojan[Backdoor]/Win32.Poison	20160710
Arcabit	Generic.PoisonIvy.F2AADF09	20160710
Avast	Win32:Agent-AAGI [Trj]	20160710
Avira (no cloud)	TR/Crypt.XPACK.Gen	20160710
Baidu	Win32.Backdoor.Poison.a	20160708
BitDefender	Generic.PoisonIvy.F2AADF09	20160710
Bkav	W32.OnlineGameXIUB.Trojan	20160708
CAT-QuickHeal	TrojanAPT.PoisonIvy.D3	20160709
CMC	Backdoor.Win32.Poison!O	20160704
ClamAV	Win.Downloader.24568-1	20160710
Comodo	Backdoor.Win32.Poison.NAE	20160710

Cyren	W32/Agent.G.gen!Eldorado	20180710
DrWeb	BackDoor.Poison.688	20180710
ESET-NOD32	Win32/Poison.NAE	20180710
Emsisoft	Generic.Poison!vy.F2AADF09 (B)	20180710
F-Prot	W32/Agent.G.gen!Eldorado	20180710
F-Secure	Backdoor:W32/Poison!vy.GI	20180710
Fortinet	W32/Poison.CWKQ!tr.bdr	20180710
GData	Generic.Poison!vy.F2AADF09	20180710
Ikarus	Backdoor.Win32.Poison	20180710
Jiangmin	Backdoor!Poison!vy.jh	20180710
K7AntiVirus	Backdoor (00199f611)	20180710
K7GW	Backdoor (00199f611)	20180710
Kaspersky	Backdoor.Win32.Poison.aec	20180710
Kingsoft	Win32.Hack.Poison.pg.5844	20180710
Malwarebytes	Backdoor.Poison	20180710
McAfee	BackDoor-DSS.gen.a	20180710
McAfee-GW-Edition	BehavesLike.Win32.Backdoor.xh	20180710
eScan	Generic.Poison!vy.F2AADF09	20180710
Microsoft	Backdoor:Win32/Poison.E	20180710
NANO-Antivirus	Trojan.Win32.Poison.dmikon	20180710
Panda	Bck/Poison.E	20180710
Qihoo-360	Backdoor.Win32.Pl!vy.A	20180710
SUPERAntiSpyware	Trojan.Agent/Gen-Backdoor	20180710
Sophos	Troj/Keylog-JV	20180710
Symantec	Trojan!gm	20180710
Tencent	Backdoor.Win32.Poison.b	20180710

TheHacker	W32/Ivy.gen	20160709
TrendMicro	BKDR_POISON.DS	20160710
TrendMicro-HouseCall	BKDR_POISON.DS	20160710
VBA32	Backdoor.Win32.Hupigon.dguz	20160708
VIPRE	Backdoor.Win32.Poison.Pg (v)	20160710
ViRobot	Backdoor.Win32.Poison.8704.M[h]	20160710
Yandex	Trojan.DL.CKSPost.Gen	20160709
Zillya	Backdoor.Poison.Win32.42544	20160709
Zoner	Trojan.Poison.NAE	20160710
nProtect	Trojan-Downloader/W32.Agent.8192.Z	20160708
Alibaba	✓	20160708

ANEXO C: RESULTADOS Bintex SOBRE “Hack Facebook.exe”

File pos	Mem pos	ID	Text
=====	=====	==	=====
00000000004D	00000040004D	0	!This program cannot be run in DOS mode.
0000000001A8	0000004001A8	0	.text
0000000001D0	0000004001D0	0	.data
00000000024E	00000040024E	0	ExitProcess
00000000025A	00000040025A	0	kernel32.dll
000000000485	000000400485	0	ws2_32
0000000005CB	0000004005CB	0	cks=u
0000000007B9	0000004007B9	0	CONNECT %s:%i HTTP/1.0
000000000839	000000400839	0	?503
000000000846	000000400846	0	200
0000000008F2	0000004008F2	0	thj@h
000000000A76	000000400A76	0	VSWRQ
000000000B04	000000400B04	0	sf5
000000001088	000000401088	0	6l*h<8
0000000010A8	0000004010A8	0	-m-m< < < M
00000000122D	00000040122D	0	advapi32
000000001247	000000401247	0	ntdll
00000000125E	00000040125E	0	user32
0000000014F7	0000004014F7	0	advpack
000000001623	000000401623	0	StubPath
00000000162F	00000040162F	0	SOFTWARE\Classes\http\shell\open\commandV
00000000165B	00000040165B	0	Software\Microsoft\Active Setup\Installed Components\
000000001694	000000401694	0	TFM_LAB

0000000016A0	0000004016A0	0	192.168.0.1
0000000016EB	0000004016EB	0	LABTFM
0000000016F5	0000004016F5	0)!VoqA.I4-
000000001702	000000401702	0	labtfm.exe
000000001746	000000401746	0	
SOFTWARE\Microsoft\Windows\CurrentVersion\Run			
0000000018CF	0000004018CF	0	explorer.exe
000000001993	000000401993	0	QPRRQ
000000001AE8	000000401AE8	0	VSWRQ
000000001D88	000000401D88	0	
SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\Shell Folders			
000000001DE6	000000401DE6	0	AppData
00000000004D	00000040004D	0	!This program cannot be run in DOS mode.
0000000001A8	0000004001A8	0	.text
0000000001D0	0000004001D0	0	.data
00000000024E	00000040024E	0	ExitProcess
00000000025A	00000040025A	0	kernel32.dll
000000000485	000000400485	0	ws2_32
0000000005CB	0000004005CB	0	cks=u
0000000007B9	0000004007B9	0	CONNECT %s:%i HTTP/1.0
000000000839	000000400839	0	?503
000000000846	000000400846	0	200
0000000008F2	0000004008F2	0	thj@h
000000000A76	000000400A76	0	VSWRQ
000000000B04	000000400B04	0	sf5
000000001088	000000401088	0	6l*h<8
0000000010A8	0000004010A8	0	-m-m< < < M
00000000122D	00000040122D	0	advapi32
000000001247	000000401247	0	ntdll
00000000125E	00000040125E	0	user32

```
0000000014F7 0000004014F7 0 advpack
000000001623 000000401623 0 StubPath
00000000162F 00000040162F 0 SOFTWARE\Classes\http\shell\open\commandV
00000000165B 00000040165B 0 Software\Microsoft\Active Setup\Installed
Components\
000000001694 000000401694 0 TFM_LAB
0000000016A0 0000004016A0 0 192.168.0.1
0000000016EB 0000004016EB 0 LABTFM
0000000016F5 0000004016F5 0 )!VoqA.I4-
000000001702 000000401702 0 labtfm.exe
```

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

```
000000001746 000000401746 0
SOFTWARE\Microsoft\Windows\CurrentVersion\Run
0000000018CF 0000004018CF 0 explorer.exe
000000001993 000000401993 0 QPRRQ
000000001AE8 000000401AE8 0 VSWRQ
000000001D88 000000401D88 0
SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\Shell Folders
000000001DE6 000000401DE6 0 AppData
```

ANEXO D: RESULTADOS – STRINGS SOBRE “Hack Facebook.exe”

```
C:\string>strings.exe -a "Hack Facebook.exe"
```

```
Strings v2.53 - Search for ANSI and Unicode strings in binary images.  
Copyright (C) 1999-2016 Mark Russinovich  
Sysinternals - www.sysinternals.com
```

```
!This program cannot be run in DOS mode.
```

```
Rich
```

```
.text
```

```
`.data
```

```
ExitProcess
```

```
kernel32.dll
```

```
ws2_32
```

```
A)|
```

```
-~_
```

```
"p7
```

```
cks=u
```

```
ttp=
```

```
cks=
```

```
CONNECT %s:%i HTTP/1.0
```

```
QSRW
```

```
?503
```

```
200
```

```
PWW
```

```
thj@h
```

```
PWW
```

```
VSWRQ
```

```
YZ_[^
```

```
f5
```

```
YZ_[^
```

```
D$0
```

```
D$0
```

```
D$0
```

```
D$0
```

```
D$0
```

```
D$0
```

```
D$0
```

```
|$,
```

```
D$0
```

```
t$,
```

```
D$0
```

```
t$,
```

```
|$,
```

```
D$4
D$4
D$4
D$4
D$4
D$4
D$4
D$4
D$4
D$4
D$4
G]=
QVIM
4~v
X:a
3sg
6!*h<8
^-m-m<|<|<|M
o/o/
00U
advapi32
ntdll
user32
Jbh
ww!
1+KY
x{w
#%li
}> *K
40j
QQVP
ucj
advpack
hk7
~Pj
<2f
StubPath
SOFTWARE\Classes\http\shell\open\commandV
Software\Microsoft\Active Setup\Installed Components\
TFM_LAB
192.168.0.1
"C
LABTFM
)!VoqA.I4-
labtfm.exe
SOFTWARE\Microsoft\Windows\CurrentVersion\Run
Ph?
```

```
V5h
V)V
explorer.exe
QPRRQ
Wht
j@h
X^_
VSWRQ
W1jD
Wht
YZ_[^
SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\Shell Folders
PWj
AppData
j@h
VQj
ViW
V%X_
VYPh
h N
YPQ
PPPP
```


ANEXO E: RESULTADOS VOLATILITY

```
C:\Python27>volatility-2.4.standalone.exe -f victima_w7.dump kdbgscan
```

```
Volatility Foundation Volatility Framework 2.4
```

```
*****
```

```
Instantiating KDBG using: C:\Python27\victima_w7.dump WinXPSP2x86 (5.1.0 32bit)
```

```
Offset (P)          : 0x2938be8
```

```
KDBG owner tag check    : True
```

```
Profile suggestion (KDBGHeader): Win7SP1x86
```

```
Version64           : 0x2938bc0 (Major: 15, Minor: 7600)
```

```
PsActiveProcessHead   : 0x82950e98
```

```
PsLoadedModuleList    : 0x82958810
```

```
KernelBase           : 0x82810000
```

```
*****
```

```
Instantiating KDBG using: C:\Python27\victima_w7.dump WinXPSP2x86 (5.1.0 32bit)
```

```
Offset (P)          : 0x2938be8
```

```
KDBG owner tag check    : True
```

```
Profile suggestion (KDBGHeader): Win7SP0x86
```

```
Version64           : 0x2938bc0 (Major: 15, Minor: 7600)
```

```
PsActiveProcessHead   : 0x82950e98
```

```
PsLoadedModuleList    : 0x82958810
```

```
KernelBase           : 0x82810000
```

```
C:\Python27>volatility-2.4.standalone.exe --profile=Win7SP1x86 -f victima_w7.dump pslist
```

Volatility Foundation Volatility Framework 2.4

Offset(V)	Name	PID	PPID	Thds	Hnds	Sess	Wow64	Start	Exit
0x84f48ae8	System	4	0	97	668	-----	0	2016-09-19 17:55:22 UTC+0000	
0x8614c020	smss.exe	260	4	2	29	-----	0	2016-09-19 17:55:23 UTC+0000	
0x86d83530	csrss.exe	356	348	8	505	0	0	2016-09-19 17:55:32 UTC+0000	
0x86da4530	wininit.exe	408	348	3	76	0	0	2016-09-19 17:55:33 UTC+0000	
0x86dad530	csrss.exe	416	400	9	215	1	0	2016-09-19 17:55:33 UTC+0000	
0x86ebdd40	winlogon.exe	464	400	4	107	1	0	2016-09-19 17:55:33 UTC+0000	
0x86f8c030	services.exe	512	408	8	222	0	0	2016-09-19 17:55:36 UTC+0000	
0x86fa2030	lsass.exe	520	408	9	800	0	0	2016-09-19 17:55:38 UTC+0000	
0x86fa3030	lsm.exe	528	408	10	141	0	0	2016-09-19 17:55:38 UTC+0000	
0x86f43548	svchost.exe	628	512	11	349	0	0	2016-09-19 17:55:39 UTC+0000	
0x86c2d148	svchost.exe	692	512	7	262	0	0	2016-09-19 17:55:43 UTC+0000	
0x86d9a030	svchost.exe	740	512	23	583	0	0	2016-09-19 17:55:43 UTC+0000	
0x86c28990	svchost.exe	856	512	30	555	0	0	2016-09-19 17:55:43 UTC+0000	
0x86c46d40	svchost.exe	892	512	40	1011	0	0	2016-09-19 17:55:43 UTC+0000	
0x86d57d40	audiodg.exe	972	740	5	122	0	0	2016-09-19 17:55:43 UTC+0000	
0x86d8fd40	svchost.exe	1064	512	23	796	0	0	2016-09-19 17:55:44 UTC+0000	
0x86db5b08	svchost.exe	1164	512	20	407	0	0	2016-09-19 17:55:44 UTC+0000	
0x872b4948	spoolsv.exe	1352	512	14	342	0	0	2016-09-19 17:55:53 UTC+0000	
0x872b2d40	svchost.exe	1424	512	18	311	0	0	2016-09-19 17:56:07 UTC+0000	
0x86def1f8	vmtoolsd.exe	1584	512	11	280	0	0	2016-09-19 17:56:07 UTC+0000	
0x87349b58	TPAutoConnSvc.	1908	512	11	139	0	0	2016-09-19 17:56:11 UTC+0000	
0x85ffd030	svchost.exe	1980	512	5	99	0	0	2016-09-19 17:56:11 UTC+0000	
0x873b4320	dllhost.exe	1300	512	18	197	0	0	2016-09-19 17:56:15 UTC+0000	

0x8740fd40 msdtc.exe	1504	512	15	152	0	0	2016-09-19 17:56:15 UTC+0000
0x87420a00 VSSVC.exe	1328	512	7	116	0	0	2016-09-19 17:56:17 UTC+0000
0x8737dc70 svchost.exe	2140	512	14	369	0	0	2016-09-19 17:56:27 UTC+0000
0x86bf0030 taskhost.exe	2280	512	12	170	1	0	2016-09-19 17:56:35 UTC+0000
0x86c1b4b0 dwm.exe	2332	856	4	69	1	0	2016-09-19 17:56:35 UTC+0000
0x86f29030 explorer.exe	2344	2320	22	778	1	0	2016-09-19 17:56:35 UTC+0000
0x8749b670 TPAutoConnect.	2496	1908	6	123	1	0	2016-09-19 17:56:36 UTC+0000
0x8749b380 conhost.exe	2504	416	1	32	1	0	2016-09-19 17:56:36 UTC+0000
0x86c0f5f8 vmtoolsd.exe	2564	2344	8	180	1	0	2016-09-19 17:56:36 UTC+0000
0x873c4030 SearchIndexer.	2792	512	13	575	0	0	2016-09-19 17:56:43 UTC+0000
0x87546d40 wmpnetwk.exe	2888	512	11	210	0	0	2016-09-19 17:56:43 UTC+0000
0x87543920 svchost.exe	3028	512	22	299	0	0	2016-09-19 17:56:45 UTC+0000
0x875a9398 svchost.exe	3280	512	8	357	0	0	2016-09-19 17:56:46 UTC+0000
0x84fe2478 iexplore.exe	3972	2344	3	35	1	0	2016-09-19 17:57:12 UTC+0000
0x86c3e030 cmd.exe	840	2344	1	24	1	0	2016-09-19 17:58:04 UTC+0000
0x875a6030 conhost.exe	2712	416	3	55	1	0	2016-09-19 17:58:04 UTC+0000
0x87586030 spssvc.exe	2952	512	5	153	0	0	2016-09-19 17:58:11 UTC+0000
0x8759cc60 winpmem_1.6.2.	3564	840	1	21	1	0	2016-09-19 17:58:40 UTC+0000

C:\Python27>volatility-2.4.standalone.exe --profile=Win7SP1x86 -f victima_w7.dump psxview

Volatility Foundation Volatility Framework 2.4

Offset(P)	Name	PID	pslist	psscan	thrdproc	pspcid	csrss	session	deskthrd	ExitTime
0x3d820a00	VSSVC.exe	1328	True	True	True	True	True	True	False	
0x3e18fd40	svchost.exe	1064	True	True	True	True	True	True	True	

0x3e3f0030 taskhost.exe	2280	True	True	True	True	True	True	True
0x3e1b5b08 svchost.exe	1164	True	True	True	True	True	True	True
0x3db7dc70 svchost.exe	2140	True	True	True	True	True	True	False
0x3dbe6810 WmiPrvSE.exe	0	False	True	False	True	True	True	False
0x3dab4948 spoolsv.exe	1352	True	True	True	True	True	True	True
0x3d9a9398 svchost.exe	3280	True	True	True	True	True	True	False
0x3dbb4320 dllhost.exe	1300	True	True	True	True	True	True	False
0x3dfa3030 lsm.exe	528	True	True	True	True	True	True	False
0x3d89b380 conhost.exe	2504	True	True	True	True	True	True	False
0x3d946d40 wmpnetwk.exe	2888	True	True	True	True	True	True	True
0x3effd030 svchost.exe	1980	True	True	True	True	True	True	True
0x3debdd40 winlogon.exe	464	True	True	True	True	True	True	True
0x3e19a030 svchost.exe	740	True	True	True	True	True	True	True
0x3df8c030 services.exe	512	True	True	True	True	True	True	False
0x3df43548 svchost.exe	628	True	True	True	True	True	True	False
0x3d80fd40 msdtc.exe	1504	True	True	True	True	True	True	True
0x3e03e030 cmd.exe	840	True	True	True	True	True	True	True
0x3e046d40 svchost.exe	892	True	True	True	True	True	True	False
0x3d943920 svchost.exe	3028	True	True	True	True	True	True	True
0x3e00f5f8 vmtoolsd.exe	2564	True	True	True	True	True	True	True
0x3e02d148 svchost.exe	692	True	True	True	True	True	True	True
0x3e01b4b0 dwm.exe	2332	True	True	True	True	True	True	False
0x3dfa2030 lsass.exe	520	True	True	True	True	True	True	False
0x3e1a4530 wininit.exe	408	True	True	True	True	True	True	True
0x3ff22478 iexplore.exe	3972	True	True	True	True	True	True	True
0x3dbc4030 SearchIndexer.	2792	True	True	True	True	True	True	False
0x3db49b58 TPAutoConnSvc.	1908	True	True	True	True	True	True	False
0x3d89b670 TPAutoConnect.	2496	True	True	True	True	True	True	False

0x3d986030 sppsvc.exe	2952	True	True	True	True	True	True	True	True
0x3d9a6030 conhost.exe	2712	True	True	True	True	True	True	True	True
0x3d99cc60 winpmem_1.6.2.	3564	True	True	True	True	True	True	True	False
0x3e028990 svchost.exe	856	True	True	True	True	True	True	True	False
0x3df29030 explorer.exe	2344	True	True	True	True	True	True	True	True
0x3e157d40 audiodg.exe	972	True	True	True	True	True	True	True	True
0x3e1ef1f8 vmtoolsd.exe	1584	True	True	True	True	True	True	True	False
0x3dab2d40 svchost.exe	1424	True	True	True	True	True	True	True	True
0x3e183530 csrss.exe	356	True	True	True	True	False	True	True	True
0x3ed4c020 smss.exe	260	True	True	True	True	False	False	False	False
0x3e1ad530 csrss.exe	416	True	True	True	True	False	True	True	True
0x3ff88ae8 System	4	True	True	True	True	False	False	False	False
0x3d647588 conhost.exe 17:54:24 UTC+0000	2676	False	True	False	False	False	False	False	False
0x3d644218 TPAutoConnect. 17:54:24 UTC+0000	2668	False	True	False	False	False	False	False	False

C:\Python27>volatility-2.4.standalone.exe --profile=Win7SP1x86 -f victima_w7.dump pstree

Volatility Foundation Volatility Framework 2.4

Name	Pid	PPid	Thds	Hnds	Time

0x86da4530:wininit.exe	408	348	3	76	2016-09-19 17:55:33 UTC+0000
. 0x86f8c030:services.exe	512	408	8	222	2016-09-19 17:55:36 UTC+0000
.. 0x875a9398:svchost.exe	3280	512	8	357	2016-09-19 17:56:46 UTC+0000
.. 0x86db5b08:svchost.exe	1164	512	20	407	2016-09-19 17:55:44 UTC+0000
.. 0x873c4030:SearchIndexer.	2792	512	13	575	2016-09-19 17:56:43 UTC+0000
.. 0x872b2d40:svchost.exe	1424	512	18	311	2016-09-19 17:56:07 UTC+0000
.. 0x87546d40:wmpnetwk.exe UTC+0000	2888	512	11	210	2016-09-19 17:56:43

.. 0x873b4320:dllhost.exe	1300	512	18	197	2016-09-19 17:56:15 UTC+0000
.. 0x87420a00:VSSVC.exe	1328	512	7	116	2016-09-19 17:56:17 UTC+0000
.. 0x86d8fd40:svchost.exe	1064	512	23	796	2016-09-19 17:55:44 UTC+0000
.. 0x86def1f8:vmtoolsd.exe	1584	512	11	280	2016-09-19 17:56:07 UTC+0000
.. 0x86c2d148:svchost.exe	692	512	7	262	2016-09-19 17:55:43 UTC+0000
.. 0x87349b58:TPAutoConnSvc. UTC+0000	1908	512	11	139	2016-09-19 17:56:11
... 0x8749b670:TPAutoConnect. UTC+0000	2496	1908	6	123	2016-09-19 17:56:36
.. 0x85ffd030:svchost.exe	1980	512	5	99	2016-09-19 17:56:11 UTC+0000
.. 0x872b4948:spoolsv.exe	1352	512	14	342	2016-09-19 17:55:53 UTC+0000
.. 0x87586030:sppsvc.exe	2952	512	5	153	2016-09-19 17:58:11 UTC+0000
.. 0x87543920:svchost.exe	3028	512	22	299	2016-09-19 17:56:45 UTC+0000
.. 0x86c28990:svchost.exe	856	512	30	555	2016-09-19 17:55:43 UTC+0000
... 0x86c1b4b0:dwm.exe	2332	856	4	69	2016-09-19 17:56:35 UTC+0000
.. 0x86d9a030:svchost.exe	740	512	23	583	2016-09-19 17:55:43 UTC+0000
... 0x86d57d40:audiodg.exe	972	740	5	122	2016-09-19 17:55:43 UTC+0000
.. 0x8737dc70:svchost.exe	2140	512	14	369	2016-09-19 17:56:27 UTC+0000
.. 0x8740fd40:msdtc.exe	1504	512	15	152	2016-09-19 17:56:15 UTC+0000
.. 0x86bf0030:taskhost.exe	2280	512	12	170	2016-09-19 17:56:35 UTC+0000
.. 0x86f43548:svchost.exe	628	512	11	349	2016-09-19 17:55:39 UTC+0000
.. 0x86c46d40:svchost.exe	892	512	40	1011	2016-09-19 17:55:43 UTC+0000
. 0x86fa2030:lsass.exe	520	408	9	800	2016-09-19 17:55:38 UTC+0000
. 0x86fa3030:lsm.exe	528	408	10	141	2016-09-19 17:55:38 UTC+0000
0x86d83530:csrss.exe	356	348	8	505	2016-09-19 17:55:32 UTC+0000
0x84f48ae8:System	4	0	97	668	2016-09-19 17:55:22 UTC+0000
. 0x8614c020:smss.exe	260	4	2	29	2016-09-19 17:55:23 UTC+0000
0x86dad530:csrss.exe	416	400	9	215	2016-09-19 17:55:33 UTC+0000

```

. 0x875a6030:conhost.exe          2712  416   3   55 2016-09-19 17:58:04 UTC+0000
. 0x8749b380:conhost.exe          2504  416   1   32 2016-09-19 17:56:36 UTC+0000
0x86ebdd40:winlogon.exe           464   400   4  107 2016-09-19 17:55:33 UTC+0000
0x86f29030:explorer.exe          2344  2320  22  778 2016-09-19 17:56:35 UTC+0000
. 0x86c0f5f8:vmtoolsd.exe         2564  2344   8  180 2016-09-19 17:56:36 UTC+0000
. 0x86c3e030:cmd.exe              840   2344   1   24 2016-09-19 17:58:04 UTC+0000
.. 0x8759cc60:winpmem_1.6.2.      3564   840   1   21 2016-09-19 17:58:40 UTC+0000
. 0x84fe2478:iexplore.exe         3972  2344   3   35 2016-09-19 17:57:12 UTC+0000

```

C:\Python27>volatility-2.4.standalone.exe --profile=Win7SP1x86 -f victima_w7.dump psscan

Volatility Foundation Volatility Framework 2.4

Offset(P)	Name	PID	PPID	PDB	Time created	Time exited
0x00000003d644218	TPAutoConnect.	2668	1788	0x3ed59480	2016-09-19 17:34:26 UTC+0000	2016-09-19 17:54:24 UTC+0000
0x00000003d647588	conhost.exe	2676	420	0x3ed594a0	2016-09-19 17:34:26 UTC+0000	2016-09-19 17:54:24 UTC+0000
0x00000003d80fd40	msdtc.exe	1504	512	0x3ed4d360	2016-09-19 17:56:15 UTC+0000	
0x00000003d820a00	VSSVC.exe	1328	512	0x3ed4d380	2016-09-19 17:56:17 UTC+0000	
0x00000003d89b380	conhost.exe	2504	416	0x3ed4d460	2016-09-19 17:56:36 UTC+0000	
0x00000003d89b670	TPAutoConnect.	2496	1908	0x3ed4d440	2016-09-19 17:56:36 UTC+0000	
0x00000003d943920	svchost.exe	3028	512	0x3ed4d4e0	2016-09-19 17:56:45 UTC+0000	
0x00000003d946d40	wmpnetwk.exe	2888	512	0x3ed4d180	2016-09-19 17:56:43 UTC+0000	
0x00000003d986030	sppsvc.exe	2952	512	0x3ed4d2c0	2016-09-19 17:58:11 UTC+0000	
0x00000003d99cc60	winpmem_1.6.2.	3564	840	0x3ed4d560	2016-09-19 17:58:40 UTC+0000	
0x00000003d9a6030	conhost.exe	2712	416	0x3ed4d540	2016-09-19 17:58:04 UTC+0000	
0x00000003d9a9398	svchost.exe	3280	512	0x3ed4d500	2016-09-19 17:56:46 UTC+0000	
0x00000003dab2d40	svchost.exe	1424	512	0x3ed4d260	2016-09-19 17:56:07 UTC+0000	

0x00000003dab4948 spoolsv.exe	1352	512	0x3ed4d240	2016-09-19 17:55:53 UTC+0000
0x00000003db49b58 TPAutoConnSvc.	1908	512	0x3ed4d300	2016-09-19 17:56:11 UTC+0000
0x00000003db7dc70 svchost.exe	2140	512	0x3ed4d3a0	2016-09-19 17:56:27 UTC+0000
0x00000003dbb4320 dllhost.exe	1300	512	0x3ed4d2e0	2016-09-19 17:56:15 UTC+0000
0x00000003dbc4030 SearchIndexer.	2792	512	0x3ed4d4a0	2016-09-19 17:56:43 UTC+0000
0x00000003dbe6810 WmiPrvSE.exe	0	628	0x3ed4d2a0	
0x00000003debdd40 winlogon.exe	464	400	0x3ed4d0c0	2016-09-19 17:55:33 UTC+0000
0x00000003df29030 explorer.exe	2344	2320	0x3ed4d420	2016-09-19 17:56:35 UTC+0000
0x00000003df43548 svchost.exe	628	512	0x3ed4d120	2016-09-19 17:55:39 UTC+0000
0x00000003df8c030 services.exe	512	408	0x3ed4d080	2016-09-19 17:55:36 UTC+0000
0x00000003dfa2030 lsass.exe	520	408	0x3ed4d0e0	2016-09-19 17:55:38 UTC+0000
0x00000003dfa3030 lsm.exe	528	408	0x3ed4d100	2016-09-19 17:55:38 UTC+0000
0x00000003e00f5f8 vmttoolsd.exe	2564	2344	0x3ed4d480	2016-09-19 17:56:36 UTC+0000
0x00000003e01b4b0 dwm.exe	2332	856	0x3ed4d400	2016-09-19 17:56:35 UTC+0000
0x00000003e028990 svchost.exe	856	512	0x3ed4d1a0	2016-09-19 17:55:43 UTC+0000
0x00000003e02d148 svchost.exe	692	512	0x3ed4d140	2016-09-19 17:55:43 UTC+0000
0x00000003e03e030 cmd.exe	840	2344	0x3ed4d320	2016-09-19 17:58:04 UTC+0000
0x00000003e046d40 svchost.exe	892	512	0x3ed4d1c0	2016-09-19 17:55:43 UTC+0000
0x00000003e157d40 audiodg.exe	972	740	0x3ed4d1e0	2016-09-19 17:55:43 UTC+0000
0x00000003e183530 csrss.exe	356	348	0x3ed4d060	2016-09-19 17:55:32 UTC+0000
0x00000003e18fd40 svchost.exe	1064	512	0x3ed4d200	2016-09-19 17:55:44 UTC+0000
0x00000003e19a030 svchost.exe	740	512	0x3ed4d160	2016-09-19 17:55:43 UTC+0000
0x00000003e1a4530 wininit.exe	408	348	0x3ed4d0a0	2016-09-19 17:55:33 UTC+0000
0x00000003e1ad530 csrss.exe	416	400	0x3ed4d040	2016-09-19 17:55:33 UTC+0000
0x00000003e1b5b08 svchost.exe	1164	512	0x3ed4d220	2016-09-19 17:55:44 UTC+0000
0x00000003e1ef1f8 vmttoolsd.exe	1584	512	0x3ed4d280	2016-09-19 17:56:07 UTC+0000
0x00000003e3f0030 taskhost.exe	2280	512	0x3ed4d3c0	2016-09-19 17:56:35 UTC+0000
0x00000003ed4c020 smss.exe	260	4	0x3ed4d020	2016-09-19 17:55:23 UTC+0000

```

0x00000003effd030 svchost.exe    1980  512 0x3ed4d340 2016-09-19 17:56:11 UTC+0000
0x00000003ff22478 iexplore.exe    3972  2344 0x3ed4d3e0 2016-09-19 17:57:12 UTC+0000
0x00000003ff88ae8 System          4     0 0x00185000 2016-09-19 17:55:22 UTC+0000

```

```

C:\Python27>volatility-2.4.standalone.exe --profile=Win7SP1x86 -f victima_w7.dump malfind --
dump-dir victima

```

Volatility Foundation Volatility Framework 2.4

Process: svchost.exe Pid: 740 Address: 0xce0000

Vad Tag: VadS Protection: PAGE_EXECUTE_READWRITE

Flags: CommitCharge: 2, MemCommit: 1, PrivateMemory: 1, Protection: 6

```

0x00ce0000 b0 00 eb 70 b0 01 eb 6c b0 02 eb 68 b0 03 eb 64 ...p...l...h...d
0x00ce0010 b0 04 eb 60 b0 05 eb 5c b0 06 eb 58 b0 07 eb 54 ...`...\...X...T
0x00ce0020 b0 08 eb 50 b0 09 eb 4c b0 0a eb 48 b0 0b eb 44 ...P...L...H...D
0x00ce0030 b0 0c eb 40 b0 0d eb 3c b0 0e eb 38 b0 0f eb 34 ...@...<...8...4

```

```

0xce0000 b000      MOV AL, 0x0
0xce0002 eb70      JMP 0xce0074
0xce0004 b001      MOV AL, 0x1
0xce0006 eb6c      JMP 0xce0074
0xce0008 b002      MOV AL, 0x2
0xce000a eb68      JMP 0xce0074
0xce000c b003      MOV AL, 0x3
0xce000e eb64      JMP 0xce0074
0xce0010 b004      MOV AL, 0x4
0xce0012 eb60      JMP 0xce0074
0xce0014 b005      MOV AL, 0x5
0xce0016 eb5c      JMP 0xce0074

```

0xce0018 b006	MOV AL, 0x6
0xce001a eb58	JMP 0xce0074
0xce001c b007	MOV AL, 0x7
0xce001e eb54	JMP 0xce0074
0xce0020 b008	MOV AL, 0x8
0xce0022 eb50	JMP 0xce0074
0xce0024 b009	MOV AL, 0x9
0xce0026 eb4c	JMP 0xce0074
0xce0028 b00a	MOV AL, 0xa
0xce002a eb48	JMP 0xce0074
0xce002c b00b	MOV AL, 0xb
0xce002e eb44	JMP 0xce0074
0xce0030 b00c	MOV AL, 0xc
0xce0032 eb40	JMP 0xce0074
0xce0034 b00d	MOV AL, 0xd
0xce0036 eb3c	JMP 0xce0074
0xce0038 b00e	MOV AL, 0xe
0xce003a eb38	JMP 0xce0074
0xce003c b00f	MOV AL, 0xf
0xce003e eb34	JMP 0xce0074

Process: svchost.exe Pid: 1064 Address: 0xab0000

Vad Tag: VadS Protection: PAGE_EXECUTE_READWRITE

Flags: CommitCharge: 2, MemCommit: 1, PrivateMemory: 1, Protection: 6

0x00ab0000 b0 00 eb 70 b0 01 eb 6c b0 02 eb 68 b0 03 eb 64 ...p...l...h...d

0x00ab0010 b0 04 eb 60 b0 05 eb 5c b0 06 eb 58 b0 07 eb 54 ...`...\...X...T

0x00ab0020 b0 08 eb 50 b0 09 eb 4c b0 0a eb 48 b0 0b eb 44 ...P...L...H...D

0x00ab0030 b0 0c eb 40 b0 0d eb 3c b0 0e eb 38 b0 0f eb 34 ...@...<...8...4

0xab0000 b000	MOV AL, 0x0
0xab0002 eb70	JMP 0xab0074
0xab0004 b001	MOV AL, 0x1
0xab0006 eb6c	JMP 0xab0074
0xab0008 b002	MOV AL, 0x2
0xab000a eb68	JMP 0xab0074
0xab000c b003	MOV AL, 0x3
0xab000e eb64	JMP 0xab0074
0xab0010 b004	MOV AL, 0x4
0xab0012 eb60	JMP 0xab0074
0xab0014 b005	MOV AL, 0x5
0xab0016 eb5c	JMP 0xab0074
0xab0018 b006	MOV AL, 0x6
0xab001a eb58	JMP 0xab0074
0xab001c b007	MOV AL, 0x7
0xab001e eb54	JMP 0xab0074
0xab0020 b008	MOV AL, 0x8
0xab0022 eb50	JMP 0xab0074
0xab0024 b009	MOV AL, 0x9
0xab0026 eb4c	JMP 0xab0074
0xab0028 b00a	MOV AL, 0xa
0xab002a eb48	JMP 0xab0074
0xab002c b00b	MOV AL, 0xb
0xab002e eb44	JMP 0xab0074
0xab0030 b00c	MOV AL, 0xc
0xab0032 eb40	JMP 0xab0074

```

0xab0034 b00d    MOV AL, 0xd
0xab0036 eb3c    JMP 0xab0074
0xab0038 b00e    MOV AL, 0xe
0xab003a eb38    JMP 0xab0074
0xab003c b00f    MOV AL, 0xf
0xab003e eb34    JMP 0xab0074

```

Process: svchost.exe Pid: 2140 Address: 0x19d0000

Vad Tag: VadS Protection: PAGE_EXECUTE_READWRITE

Flags: CommitCharge: 224, MemCommit: 1, PrivateMemory: 1, Protection: 6

```

0x019d0000 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
0x019d0010 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
0x019d0020 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
0x019d0030 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....

```

```

0x19d0000 0000    ADD [EAX], AL
0x19d0002 0000    ADD [EAX], AL
0x19d0004 0000    ADD [EAX], AL
0x19d0006 0000    ADD [EAX], AL
0x19d0008 0000    ADD [EAX], AL
0x19d000a 0000    ADD [EAX], AL
0x19d000c 0000    ADD [EAX], AL
0x19d000e 0000    ADD [EAX], AL
0x19d0010 0000    ADD [EAX], AL
0x19d0012 0000    ADD [EAX], AL
0x19d0014 0000    ADD [EAX], AL
0x19d0016 0000    ADD [EAX], AL

```

```
0x19d0018 0000    ADD [EAX], AL
0x19d001a 0000    ADD [EAX], AL
0x19d001c 0000    ADD [EAX], AL
0x19d001e 0000    ADD [EAX], AL
0x19d0020 0000    ADD [EAX], AL
0x19d0022 0000    ADD [EAX], AL
0x19d0024 0000    ADD [EAX], AL
0x19d0026 0000    ADD [EAX], AL
0x19d0028 0000    ADD [EAX], AL
0x19d002a 0000    ADD [EAX], AL
0x19d002c 0000    ADD [EAX], AL
0x19d002e 0000    ADD [EAX], AL
0x19d0030 0000    ADD [EAX], AL
0x19d0032 0000    ADD [EAX], AL
0x19d0034 0000    ADD [EAX], AL
0x19d0036 0000    ADD [EAX], AL
0x19d0038 0000    ADD [EAX], AL
0x19d003a 0000    ADD [EAX], AL
0x19d003c 0000    ADD [EAX], AL
0x19d003e 0000    ADD [EAX], AL
```

Process: svchost.exe Pid: 2140 Address: 0x4ed0000

Vad Tag: VadS Protection: PAGE_EXECUTE_READWRITE

Flags: CommitCharge: 128, MemCommit: 1, PrivateMemory: 1, Protection: 6

```
0x04ed0000 08 00 42 00 00 00 05 8b 45 14 89 c2 8b 45 10  ..B.....E....E.
```

```
0x04ed0010 8b 08 8b 40 04 89 0a 89 42 04 8b 45 14 81 00 88  ...@....B..E....
```

```
0x04ed0020 00 00 00 8d 45 08 89 c2 8b 45 14 8b 08 89 0a 8b  ....E....E.....
```

0x04ed0030 45 14 89 c2 8b 45 08 8b 00 89 02 c7 42 04 00 00 E....E.....B...

```
0x4ed0000 0800      OR [EAX], AL
0x4ed0002 42       INC EDX
0x4ed0003 0000      ADD [EAX], AL
0x4ed0005 0000      ADD [EAX], AL
0x4ed0007 058b451489   ADD EAX, 0x8914458b
0x4ed000c c28b45       RET 0x458b
0x4ed000f 108b088b4004   ADC [EBX+0x4408b08], CL
0x4ed0015 890a       MOV [EDX], ECX
0x4ed0017 894204      MOV [EDX+0x4], EAX
0x4ed001a 8b4514      MOV EAX, [EBP+0x14]
0x4ed001d 810088000000   ADD DWORD [EAX], 0x88
0x4ed0023 8d4508      LEA EAX, [EBP+0x8]
0x4ed0026 89c2       MOV EDX, EAX
0x4ed0028 8b4514      MOV EAX, [EBP+0x14]
0x4ed002b 8b08       MOV ECX, [EAX]
0x4ed002d 890a       MOV [EDX], ECX
0x4ed002f 8b4514      MOV EAX, [EBP+0x14]
0x4ed0032 89c2       MOV EDX, EAX
0x4ed0034 8b4508      MOV EAX, [EBP+0x8]
0x4ed0037 8b00       MOV EAX, [EAX]
0x4ed0039 8902       MOV [EDX], EAX
0x4ed003b c7         DB 0xc7
0x4ed003c 42       INC EDX
0x4ed003d 0400      ADD AL, 0x0
0x4ed003f 00       DB 0x0
```

Process: svchost.exe Pid: 2140 Address: 0x5210000

Vad Tag: VadS Protection: PAGE_EXECUTE_READWRITE

Flags: CommitCharge: 256, MemCommit: 1, PrivateMemory: 1, Protection: 6

0x05210000 09 00 38 00 09 00 01 05 8b 55 18 8b 4d 54 8d 54 ..8.....U..MT.T

0x05210010 0a fc 89 d6 b9 04 00 1a 00 ff 95 54 37 00 00 8bT7...

0x05210020 4d 1c 89 08 83 45 18 fc 8d 45 1c 8b 4d 18 89 08 M....E...E..M...

0x05210030 81 6d 18 98 02 00 00 9f 0f 90 c0 66 89 45 38 8d .m.....f.E8.

0x5210000 0900 OR [EAX], EAX

0x5210002 3800 CMP [EAX], AL

0x5210004 0900 OR [EAX], EAX

0x5210006 01058b55188b ADD [0x8b18558b], EAX

0x521000c 4d DEC EBP

0x521000d 54 PUSH ESP

0x521000e 8d540afc LEA EDX, [EDX+ECX-0x4]

0x5210012 89d6 MOV ESI, EDX

0x5210014 b904001a00 MOV ECX, 0x1a0004

0x5210019 ff9554370000 CALL DWORD [EBP+0x3754]

0x521001f 8b4d1c MOV ECX, [EBP+0x1c]

0x5210022 8908 MOV [EAX], ECX

0x5210024 834518fc ADD DWORD [EBP+0x18], -0x4

0x5210028 8d451c LEA EAX, [EBP+0x1c]

0x521002b 8b4d18 MOV ECX, [EBP+0x18]

0x521002e 8908 MOV [EAX], ECX

0x5210030 816d1898020000 SUB DWORD [EBP+0x18], 0x298

0x5210037 9f LAHF

0x5210038 0f90c0 SETO AL

0x521003b 66894538 MOV [EBP+0x38], AX

0x521003f 8d DB 0x8d

Process: explorer.exe Pid: 2344 Address: 0x1f60000

Vad Tag: VadS Protection: PAGE_EXECUTE_READWRITE

Flags: CommitCharge: 1, MemCommit: 1, PrivateMemory: 1, Protection: 6

0x01f60000 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

0x01f60010 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

0x01f60020 00 f4 05 32 76 35 0d 32 76 fd 27 32 76 62 20 2d ...2v5.2v.'2vb.-

0x01f60030 76 d4 be f3 75 0d bc f3 75 25 bc f3 75 96 1b f3 v...u...u%..u...

0x1f60000 0000 ADD [EAX], AL

0x1f60002 0000 ADD [EAX], AL

0x1f60004 0000 ADD [EAX], AL

0x1f60006 0000 ADD [EAX], AL

0x1f60008 0000 ADD [EAX], AL

0x1f6000a 0000 ADD [EAX], AL

0x1f6000c 0000 ADD [EAX], AL

0x1f6000e 0000 ADD [EAX], AL

0x1f60010 0000 ADD [EAX], AL

0x1f60012 0000 ADD [EAX], AL

0x1f60014 0000 ADD [EAX], AL

0x1f60016 0000 ADD [EAX], AL

0x1f60018 0000 ADD [EAX], AL

0x1f6001a 0000 ADD [EAX], AL

0x1f6001c 0000 ADD [EAX], AL

0x1f6001e 0000 ADD [EAX], AL


```
0x1f60020 00f4      ADD AH, DH
0x1f60022 053276350d  ADD EAX, 0xd357632
0x1f60027 3276fd      XOR DH, [ESI-0x3]
0x1f6002a 27          DAA
0x1f6002b 327662      XOR DH, [ESI+0x62]
0x1f6002e 202d76d4bef3  AND [0xf3bed476], CH
0x1f60034 750d        JNZ 0x1f60043
0x1f60036 bcf37525bc  MOV ESP, 0xbc2575f3
0x1f6003b f37596      JNZ 0x1f5ffd4
0x1f6003e 1bf3        SBB ESI, EBX
```

Process: explorer.exe Pid: 2344 Address: 0x1f50000

Vad Tag: VadS Protection: PAGE_EXECUTE_READWRITE

Flags: CommitCharge: 1, MemCommit: 1, PrivateMemory: 1, Protection: 6

```
0x01f50000 55 8b ec 83 c4 cc 8b 75 08 6a 00 68 80 00 00 00  U.....u.j.h....
0x01f50010 6a 03 6a 00 6a 00 68 00 00 00 80 8d 86 b1 06 00  j.j.j.h.....
0x01f50020 00 50 ff 56 59 50 68 88 b6 b6 fc ff b6 bf 0a 00  .P.VYPh.....
0x01f50030 00 ff b6 e1 00 00 00 ff 96 dd 00 00 00 89 45 e0  ....E.
```

```
0x1f50000 55          PUSH EBP
0x1f50001 8bec        MOV EBP, ESP
0x1f50003 83c4cc      ADD ESP, -0x34
0x1f50006 8b7508      MOV ESI, [EBP+0x8]
0x1f50009 6a00        PUSH 0x0
0x1f5000b 6880000000  PUSH DWORD 0x80
0x1f50010 6a03        PUSH 0x3
0x1f50012 6a00        PUSH 0x0
```

```

0x1f50014 6a00      PUSH 0x0
0x1f50016 6800000080  PUSH DWORD 0x80000000
0x1f5001b 8d86b1060000  LEA EAX, [ESI+0x6b1]
0x1f50021 50          PUSH EAX
0x1f50022 ff5659      CALL DWORD [ESI+0x59]
0x1f50025 50          PUSH EAX
0x1f50026 6888b6b6fc   PUSH DWORD 0xfcb6b688
0x1f5002b ffb6bf0a0000 PUSH DWORD [ESI+0xabf]
0x1f50031 ffb6e1000000 PUSH DWORD [ESI+0xe1]
0x1f50037 ff96dd000000 CALL DWORD [ESI+0xdd]
0x1f5003d 8945e0      MOV [EBP-0x20], EAX

```

Process: explorer.exe Pid: 2344 Address: 0x3590000

Vad Tag: VadS Protection: PAGE_EXECUTE_READWRITE

Flags: CommitCharge: 1, MemCommit: 1, PrivateMemory: 1, Protection: 6

```

0x03590000 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
0x03590010 00 00 59 03 00 00 00 00 00 00 00 00 00 00 00 ..Y.....
0x03590020 10 00 59 03 00 00 00 00 00 00 00 00 00 00 00 ..Y.....
0x03590030 20 00 59 03 00 00 00 00 00 00 00 00 00 00 00 ..Y.....

```

```

0x3590000 0000      ADD [EAX], AL
0x3590002 0000      ADD [EAX], AL
0x3590004 0000      ADD [EAX], AL
0x3590006 0000      ADD [EAX], AL
0x3590008 0000      ADD [EAX], AL
0x359000a 0000      ADD [EAX], AL
0x359000c 0000      ADD [EAX], AL

```

0x359000e 0000	ADD [EAX], AL
0x3590010 0000	ADD [EAX], AL
0x3590012 59	POP ECX
0x3590013 0300	ADD EAX, [EAX]
0x3590015 0000	ADD [EAX], AL
0x3590017 0000	ADD [EAX], AL
0x3590019 0000	ADD [EAX], AL
0x359001b 0000	ADD [EAX], AL
0x359001d 0000	ADD [EAX], AL
0x359001f 0010	ADD [EAX], DL
0x3590021 005903	ADD [ECX+0x3], BL
0x3590024 0000	ADD [EAX], AL
0x3590026 0000	ADD [EAX], AL
0x3590028 0000	ADD [EAX], AL
0x359002a 0000	ADD [EAX], AL
0x359002c 0000	ADD [EAX], AL
0x359002e 0000	ADD [EAX], AL
0x3590030 2000	AND [EAX], AL
0x3590032 59	POP ECX
0x3590033 0300	ADD EAX, [EAX]
0x3590035 0000	ADD [EAX], AL
0x3590037 0000	ADD [EAX], AL
0x3590039 0000	ADD [EAX], AL
0x359003b 0000	ADD [EAX], AL
0x359003d 0000	ADD [EAX], AL
0x359003f 00	DB 0x0

Process: explorer.exe Pid: 2344 Address: 0x46e0000

Vad Tag: VadS Protection: PAGE_EXECUTE_READWRITE

Flags: CommitCharge: 2, MemCommit: 1, PrivateMemory: 1, Protection: 6

0x046e0000 b0 00 eb 70 b0 01 eb 6c b0 02 eb 68 b0 03 eb 64 ...p...l...h...d
0x046e0010 b0 04 eb 60 b0 05 eb 5c b0 06 eb 58 b0 07 eb 54 ...`...\...X...T
0x046e0020 b0 08 eb 50 b0 09 eb 4c b0 0a eb 48 b0 0b eb 44 ...P...L...H...D
0x046e0030 b0 0c eb 40 b0 0d eb 3c b0 0e eb 38 b0 0f eb 34 ...@...<...8...4

0x46e0000 b000	MOV AL, 0x0
0x46e0002 eb70	JMP 0x46e0074
0x46e0004 b001	MOV AL, 0x1
0x46e0006 eb6c	JMP 0x46e0074
0x46e0008 b002	MOV AL, 0x2
0x46e000a eb68	JMP 0x46e0074
0x46e000c b003	MOV AL, 0x3
0x46e000e eb64	JMP 0x46e0074
0x46e0010 b004	MOV AL, 0x4
0x46e0012 eb60	JMP 0x46e0074
0x46e0014 b005	MOV AL, 0x5
0x46e0016 eb5c	JMP 0x46e0074
0x46e0018 b006	MOV AL, 0x6
0x46e001a eb58	JMP 0x46e0074
0x46e001c b007	MOV AL, 0x7
0x46e001e eb54	JMP 0x46e0074
0x46e0020 b008	MOV AL, 0x8
0x46e0022 eb50	JMP 0x46e0074
0x46e0024 b009	MOV AL, 0x9
0x46e0026 eb4c	JMP 0x46e0074

```
0x46e0028 b00a    MOV AL, 0xa
0x46e002a eb48    JMP 0x46e0074
0x46e002c b00b    MOV AL, 0xb
0x46e002e eb44    JMP 0x46e0074
0x46e0030 b00c    MOV AL, 0xc
0x46e0032 eb40    JMP 0x46e0074
0x46e0034 b00d    MOV AL, 0xd
0x46e0036 eb3c    JMP 0x46e0074
0x46e0038 b00e    MOV AL, 0xe
0x46e003a eb38    JMP 0x46e0074
0x46e003c b00f    MOV AL, 0xf
0x46e003e eb34    JMP 0x46e0074
```

Process: wmpnetwk.exe Pid: 2888 Address: 0x2f0000

Vad Tag: VadS Protection: PAGE_EXECUTE_READWRITE

Flags: CommitCharge: 2, MemCommit: 1, PrivateMemory: 1, Protection: 6

```
0x002f0000 b0 00 eb 70 b0 01 eb 6c b0 02 eb 68 b0 03 eb 64 ...p...l...h...d
0x002f0010 b0 04 eb 60 b0 05 eb 5c b0 06 eb 58 b0 07 eb 54 ...`...\...X...T
0x002f0020 b0 08 eb 50 b0 09 eb 4c b0 0a eb 48 b0 0b eb 44 ...P...L...H...D
0x002f0030 b0 0c eb 40 b0 0d eb 3c b0 0e eb 38 b0 0f eb 34 ...@...<...8...4
```

```
0x2f0000 b000    MOV AL, 0x0
0x2f0002 eb70    JMP 0x2f0074
0x2f0004 b001    MOV AL, 0x1
0x2f0006 eb6c    JMP 0x2f0074
0x2f0008 b002    MOV AL, 0x2
0x2f000a eb68    JMP 0x2f0074
```

0x2f000c b003	MOV AL, 0x3
0x2f000e eb64	JMP 0x2f0074
0x2f0010 b004	MOV AL, 0x4
0x2f0012 eb60	JMP 0x2f0074
0x2f0014 b005	MOV AL, 0x5
0x2f0016 eb5c	JMP 0x2f0074
0x2f0018 b006	MOV AL, 0x6
0x2f001a eb58	JMP 0x2f0074
0x2f001c b007	MOV AL, 0x7
0x2f001e eb54	JMP 0x2f0074
0x2f0020 b008	MOV AL, 0x8
0x2f0022 eb50	JMP 0x2f0074
0x2f0024 b009	MOV AL, 0x9
0x2f0026 eb4c	JMP 0x2f0074
0x2f0028 b00a	MOV AL, 0xa
0x2f002a eb48	JMP 0x2f0074
0x2f002c b00b	MOV AL, 0xb
0x2f002e eb44	JMP 0x2f0074
0x2f0030 b00c	MOV AL, 0xc
0x2f0032 eb40	JMP 0x2f0074
0x2f0034 b00d	MOV AL, 0xd
0x2f0036 eb3c	JMP 0x2f0074
0x2f0038 b00e	MOV AL, 0xe
0x2f003a eb38	JMP 0x2f0074
0x2f003c b00f	MOV AL, 0xf
0x2f003e eb34	JMP 0x2f0074

Process: iexplore.exe Pid: 3972 Address: 0x60000

Vad Tag: VadS Protection: PAGE_EXECUTE_READWRITE

Flags: CommitCharge: 1, MemCommit: 1, PrivateMemory: 1, Protection: 6

0x00060000 00 00 3f c9 77 be 48 c9 77 ed 3b c9 77 c8 c4 c9 ..?.w.H.w.;.w...

0x00060010 77 df 47 c9 77 ab 2f c9 77 34 32 c9 77 33 71 ca w.G.w./..w42.w3q.

0x00060020 77 f4 05 32 76 35 0d 32 76 fd 27 32 76 62 20 2d w..2v5.2v.'2vb.-

0x00060030 76 d4 be f3 75 0d bc f3 75 25 bc f3 75 96 1b f3 v...u...u%..u...

0x60000 0000 ADD [EAX], AL

0x60002 3f AAS

0x60003 c9 LEAVE

0x60004 77be JA 0x5ffc4

0x60006 48 DEC EAX

0x60007 c9 LEAVE

0x60008 77ed JA 0x5fff7

0x6000a 3bc9 CMP ECX, ECX

0x6000c 77c8 JA 0x5ffd6

0x6000e c4 DB 0xc4

0x6000f c9 DB 0xc9

0x60010 77 DB 0x77

0x60011 df DB 0xdf

0x60012 47 INC EDI

0x60013 c9 LEAVE

0x60014 77ab JA 0x5ffc1

0x60016 2f DAS

0x60017 c9 LEAVE

0x60018 7734 JA 0x6004e

0x6001a 32c9 XOR CL, CL

```
0x6001c 7733      JA 0x60051
0x6001e 71ca      JNO 0x5ffea
0x60020 77f4      JA 0x60016
0x60022 053276350d  ADD EAX, 0xd357632
0x60027 3276fd      XOR DH, [ESI-0x3]
0x6002a 27          DAA
0x6002b 327662      XOR DH, [ESI+0x62]
0x6002e 202d76d4bef3  AND [0xf3bed476], CH
0x60034 750d      JNZ 0x60043
0x60036 bcf37525bc  MOV ESP, 0xbc2575f3
0x6003b f37596      JNZ 0x5ffd4
0x6003e 1bf3      SBB ESI, EBX
```

Process: iexplore.exe Pid: 3972 Address: 0x50000

Vad Tag: VadS Protection: PAGE_EXECUTE_READWRITE

Flags: CommitCharge: 1, MemCommit: 1, PrivateMemory: 1, Protection: 6

```
0x00050000 55 8b ec 81 c4 30 fa ff ff 8b 75 08 8d 86 fb 03  U....0....u....
0x00050010 00 00 50 6a 00 6a 00 ff 96 85 00 00 00 89 86 c5  ..Pj.j.....
0x00050020 08 00 00 ff 96 89 00 00 00 3d b7 00 00 00 75 04  .....=....u.
0x00050030 c9 c2 04 00 56 8d 86 6b 09 00 00 50 8d 86 45 01  ....V..k...P..E.
```

```
0x50000 55          PUSH EBP
0x50001 8bec        MOV EBP, ESP
0x50003 81c430faffff  ADD ESP, 0xfffffa30
0x50009 8b7508      MOV ESI, [EBP+0x8]
0x5000c 8d86fb030000  LEA EAX, [ESI+0x3fb]
0x50012 50          PUSH EAX
```



```
0x50013 6a00    PUSH 0x0
0x50015 6a00    PUSH 0x0
0x50017 ff9685000000  CALL DWORD [ESI+0x85]
0x5001d 8986c5080000  MOV [ESI+0x8c5], EAX
0x50023 ff9689000000  CALL DWORD [ESI+0x89]
0x50029 3db7000000    CMP EAX, 0xb7
0x5002e 7504         JNZ 0x50034
0x50030 c9          LEAVE
0x50031 c20400     RET 0x4
0x50034 56         PUSH ESI
0x50035 8d866b090000  LEA EAX, [ESI+0x96b]
0x5003b 50         PUSH EAX
0x5003c 8d         DB 0x8d
0x5003d 864501     XCHG [EBP+0x1], AL
```

Referencias y Bibliografía

Adrien, C., & Robinson, D. a. (21 de noviembre de 2103). *Conix Security*. Obtenido de Poison Ivy RAT: Configuration & Communications: <http://blog.conixsecurity.fr/wp-content/uploads/2013/10/Poison-Ivy-RAT-conf-comms.pdf>

Bennet, J. T., & Moran, N. y. (2013). *FireEye*. Recuperado el 2016 de abril de 5, de Poison Ivy: Evaluación de daños: https://www.fireeye.de/content/dam/fireeye-www/regional/mx_ES/current-threats/pdfs/rpt-poison-ivy.pdf

Bermejo, J. (12 de junio de 2015). *Desarrollo de un sistema de análisis e ingeniería inversa de código malicioso*.

Boire, M. M., Guarnieri, C., & Gallagher, R. (24 de noviembre de 2014). *Theintercep*. Recuperado el 7 de mayo de 2016, de Secret Malware in European Union Attack Linked to U.S. and British Intelligence: <https://theintercept.com/2014/11/24/secret-regin-malware-belgacom-nsa-gchq/>

CN-Cert. (diciembre de 2014). *DEFENSA FRENTE A LAS CIBERAMENAZAS*. Recuperado el 8 de mayo de 2016, de CIBERAMENAZAS 2014 TENDENCIAS 2015: <https://www.ccn-cert.cni.es/publico/dmpublidocuments/IE-Ciberamenazas2014-Tendencias-2015.pdf>

Coryn, C. L. (2006). *The fundamental characteristics of research. Journal of MultiDisciplinary*.

Curry, S., Hartman, B., Hunter, D. P., Martin, D., Moreau, D. R., Oprea, A., & otros., y. (2009). Intelligent Security Operations for Advanced Persistent Threats. *RSA Security Brief*.

Dereszowski, A. (15 de Maezo de 2010). *SIGNAL 11*. Obtenido de Targeted attacks: From being a victim to counter attacking.: https://docs.google.com/viewer?url=http://www.signal11.eu/en/research/articles/targeted_2010.pdf

Desconocido. (11 de Junio de 2010). *MediaFire*. Obtenido de نوزيبل رفر يس لي غشت حرش
ع ل ع زودنيو 7(1).rar:
<http://www.mediafire.com/download/jwmd1n2jwtdm/%D8%B4%D8%B1%D8%AD+%D8%A>

A%D8%B4%D8%BA%D9%8A%D9%84+%D8%B3%D9%8A%D8%B1%D9%81%D8%B1
+%D8%A7%D9%84%D8%A8%D9%8A%D8%B2%D9%88%D9%86+%D8%B9%D9%84%
D9%89+%D9%88%D9%8A%D9%86%D8%AF%D9%88%D8%B2+7.rar

Durán, J. (15 de Mayo de 2010). *Desarrollo de un laboratorio para el análisis automatizado de códigos maliciosos*. Obtenido de <http://www.ptolomeo.unam.mx:8080/xmlui/handle/132.248.52.100/916>

Eset. (2015). Recuperado el 26 de abril de 2016, de Definición de virus, códigos maliciosos y ataques remotos.: http://soporte.eset-la.com/kb186/?&page=content&id=SOLN186&querysource=external_es&locale=es_ES

Eset. (2016). *Eset*. Recuperado el 3 de mayo de 2016, de Tendencias 2016 Insecurity Everywhere: http://www.welivesecurity.com/wp-content/uploads/2016/01/Tendencias_2016_insecurity_everywhere_eset.pdf

Eset. (26 de 04 de 2016). *Guía de respuesta a una infección por malware*. Obtenido de eset - ENJOY SAFER TECHNOLOGY: http://www.welivesecurity.com/wp-content/uploads/2015/11/Guia_respuesta_infeccion_malware_ESET.pdf

GitHub, I. (s.f.). *GitHub, Inc*. Recuperado el 3 de mayo de 2016, de Kbandla/APTnotes: <https://github.com/kbandla/APTnotes>

Google. (12 de 9 de 2016). *volatility - CommandReference.wiki*. Obtenido de Image Identification: <https://code.google.com/archive/p/volatility/wikis/CommandReference.wiki#malfind>

Govcertuk. (octubre de 2011). *Team Computer Emergency Response*. Recuperado el 10 de mayo de 2016, de "Targeted Email Attack Alert" : <https://www.cesg.gov.uk/articles/govcertuk>

Hale, M. (16 de Sep de 2016). *Volatility Labs*. Obtenido de Reverse Engineering Poison Ivy's Injected Code Fragments, : <http://volatility-labs.blogspot.com.co/2012/10/reverse-engineering-poison-ivy.html>.

Hernandez, M. (diciembre de 2014). *aeropago21.org*. Recuperado el 2016 de mayo de 5, de Resumen de ciberdefensa 2014: <http://www.aeropago21.org/2014/12/resumen-ciberdefensa-2014.html>

Institute, S. (s.f.). "20 Critical Security Controls". Obtenido de <http://blog.segu-info.com.ar/2012/11/nueva-version-de-20-critical-security.html>

Institute, S. (s.f.). 20 Critical Security Controls. Obtenido de <http://blog.segu-info.com.ar/2012/11/nueva-version-de-20-critical-security.html>

Jale, M. (16 de octubre de 2012). *Volatility Labs*. Obtenido de Reverse Engineering Poison Ivy's Injected Code Fragments: <http://volatility-labs.blogspot.com.co/2012/10/reverse-engineering-poison-ivys.html>

kaspersky Lab. (20 de Abril de 2016). *kaspersky Lab*. Obtenido de ¿Qué es un código malicioso?: <http://latam.kaspersky.com/mx/internet-security-center/definitions/malicious-code>

kaspersky.com. (20 de mayo de 2013). *Comunicados de prensa*. Recuperado el 2016 de mayo de 9, de Principales incidentes de seguridad que conformaron el campo de amenazas en 2013: <http://latam.kaspersky.com/mx/sobre-kaspersky/centro-de-prensa/comunicados-de-prensa/2013/principales-incidentes-de-seguridad-que-confo>

López, D. (21 de abril de 2016). *Grupo Control Seguridad*. Obtenido de Grupo Control Seguridad: <https://www.grupocontrol.com/evolucion-de-la-seguridad-informatica>

Marshall, A. (20 de Abril de 2016). *scienceblog.com*. Obtenido de Scientists demonstrate first contagious airborne WiFi virus. : <http://scienceblog.com/70678/scientists-demonstrate-first-contagious-airborne-wifi-virus/>

Matsui, M. &. (Abril de 2004). *The Internet Engineering Task Force (IETF®)*. Obtenido de RFC 3713 A Description of the Camellia Encryption Algorithm: <https://www.ietf.org/rfc/rfc3713.txt>

Norman, G. (28 de Marzo de 2016). *findmysoft.com*. Obtenido de Detect It Easy – Free and Portable Package Identifier: <http://detect-it-easy.findmysoft.com/>

Panda Labs. (28 de 04 de 2016). *INFORME ANUAL PANDALABS 2015*. Obtenido de <http://www.pandasecurity.com/spain/mediacenter/src/uploads/2014/07/Pandalabs-2015-anual-ES.pdf>

Piper, S. (2013). CyberEdge Group. *Defenitive Guide para la protección contra amenazas de próxima generación Defenitive Guide para la protección c.* [https://www.fireeye.com/content/dam/fireeye-www/regional/mx_ES/solutions/pdfs/eb-definitive-guide-next-gen-threat-protection.pdf.

Railton, J. S., & Hardy, S. . (18 de diciembre de 2014). *citizenlab.org*. Recuperado el 6 de mayo de 2016, de Malware Attack Targeting Syrian ISIS Critics: <https://citizenlab.org/2014/12/malware-attack-targeting-syrian-isis-critics/>

Saade, T., Kurc, D., & Stewart, H. (October de 2011). *Thear Report: Poison Ivy*. Obtenido de Microsoft Malware Protection Center: www.microsoft.com/en-us/download/details.aspx?id=27871

Sanabria, A. (2007). *Malware Analysis: Environment Design and Artitecture*. SANS Institute.

Segu.info. (2009). Recuperado el 26 de abril de 2016, de Adware / Spyware: <http://www.segu-info.com.ar/malware/spyware.htm>

symantec.com. (30 de agosto de 2012). *Security Response*. Recuperado el 12 de mayo de 2016, de Java zero day used targeted attack campaign: <http://www.symantec.com/connect/blogs/java-zero-day-used-targeted-attack-campaign>

Theerthagiri, D. (23 de noviembre de 2009). *Reversing Malware: A detection intelligence with in-depth security analysis*. Recuperado el 12 de mayo de 2016, de <http://www.diva-portal.org/smash/get/diva2:284604/FULLTEXT01.pdf>

Timeline, M. &, & Goldman, J. E. (2011). *Malware Analysis Reverse Engineering (MARE)*. Recuperado el 15 de mayo de 2016

Yichong, L. (3 de mayo de 2013). *"IE Zero Day is Used in DoL Watering Hole Attack"*. Recuperado el 12 de mayo de 2016, de <http://www.fireeye.com/blog/technical/yber-exploits/2013/05/ie-zero-day-is-used-in-dol-watering-hole-attack.html>.

Webgrafía

Chevalier, A., & Delaugerre, R. (2016). *Arbor Networks*. New Poison Ivy Activity Targeting Myanmar, Asian Countries. <https://www.arbornetworks.com/blog/asert/recent-poison-iv/>

Jones, J. (2013). *Conix Security*. Poison Ivy RAT: Configuration & Communications. <http://blog.conixsecurity.fr/wp-content/uploads/2013/10/Poison-Ivy-RAT-conf-comms.pdf>

Kovah, V. (12 de 04 de 2016). *Malware Dynamic Analysis*. Obtenido de <http://opensecuritytraining.info/MalwareDynamicAnalysis.html>

Parikh, P. (2012). *QUALYS BLOG: Risks of Vulnerabilities in Example Scripts Bundled with Software*. Obtenido de <https://blog.qualys.com/securitylabs/page/8>

Prince, B. (2013). *FireEye Unveils New Research, Analysis Tools for Poison Ivy RAT*. Obtenido de <http://www.securityweek.com/fireeye-unveils-new-research-analysis-tools-poison-ivy-rat>

RSA FraudAction Research Labs. (2 de 05 de 2016). *Anatomy of an attack*. Obtenido de <http://www.trendmicro.com/cloud-content/us/pdfs/security-intelligence/white-papers/wp-detecting-apt-activity-with-network-traffic-analysis.pdf>

Trend Micro. (28 de 04 de 2016). *Research Paper 2012: Detecting APT Activity with Network Traffic Analysis*. Obtenido de <http://www.trendmicro.com/cloud-content/us/pdfs/security-intelligence/white-papers/wp-detecting-apt-activity-with-network-traffic-analysis.pdf>