Como Realizar Análisis Dinámico y Estático a una Muestra de Malware

Gustavo Adolfo Quicasaque Mesa
Corporación Universidad Piloto de Colombia
Programa de Ingeniería de Sistemas
Bogotá, Colombia
gusquica@hotmail.com

Abstract. Most of important attacks are performed to steal confidential information such a phishing attacks, MITM attacks and so on. However, those attacks have a complex process in order to be distributed to the final users, some of them are distributed by social engineering, but other ones are distributed by malware. Malware´ analysis covers a lot of knowledge areas such connection analysis and low level programming. Currently, the analysis of a sample malware has no a specific process or a specific tools due to most of them were programmed differently. Taking in to account, it is important to have expertise to recognize different types of patterns to identify the behavior of a sample of malware.

Resumen. La mayoría de los ataques son configurados para robar información confidencial como los son los ataques de phishing, MITM y otros más. Sin embargo, estos ataques tienen un proceso complejo cuando son distribuidos al usuario final, alguno de ellos son distribuidos por medio de ingeniería social y otros son distribuidos por medio de malware. El análisis de malware cubre un amplio rango de áreas conocimiento como análisis de conexiones y programación de bajo nivel. Actualmente, el análisis de una muestra de malware no tiene un proceso específico o una herramienta especifica debido a que la mayoría del malware fueron programados sin seguir ningún tipo de patrón. Tomando en cuenta lo anterior, es importante tener la experticia para reconocer diferentes tipos de patrones que permitan identificar el comportamiento de una muestra de malware.

Keywords: malware, static analysis, dynamic analysis, attack patterns, programming language.

Palabras Clave: malware, análisis estático, análisis dinámico, patrones de ataque, lenguaje de programación.

I. INTRODUCCIÓN

Los Malware son piezas de código ejecutables que tienen como finalidad dañar un computador o un sistema de información sin el consentimiento del propietario. El malware se puede clasificar de acuerdo al tipo de daño que desea causar dentro del sistema afectado.

La problemática del malware se puede estudiar desde dos grandes puntos de vista que a la vez son complementarios. El primero, hace referencia a la complejidad de desarrollo y detección y el segundo hace referencia al tipo de entidades al cual está enfocado el ataque.

La complejidad de desarrollo de un archivo de malware está atada de forma directa con el constante crecimiento y evolución de las herramientas anti-malware.

Se puede decir que este tipo de herramientas son una respuesta reactiva a los nuevos tipos de ataques generados. Cada vez que sale un nuevo ataque los diferentes fabricantes lanzan una ¨actualización de la base de datos¨ tiempo después que el ataque denominado como ¨día cero¨ ya fue lanzado. En ese lapso de tiempo, el ataque es donde genera la mayor cantidad de daño y por ende la mayor pérdida de información. Muchos de los archivos de malware diseñados hoy en día tienen como atributos principales los siguientes factores, los cuales los hacen mucho más difíciles de detectar y analizar:

- El ataque es desarrollado para sistemas operativos específicos, si el ataque detecta otro sistema operativo distinto para el cual fue creado, sencillamente no se ejecuta.
- El ataque es capaz de detectar si su ejecución se realiza sobre un entorno virtualizado o un entorno real, al detectar el primer escenario, el ataque no se ejecuta.
- Los malware iniciales, creaban nuevos procesos de máquina y llaves de registro al momento de su ejecución, lo cual los hacia fácilmente detectables. Actualmente, los nuevos malware tienen la capacidad de infectar un proceso legitimo del sistema (ejemplo: Firefox o Skype) para realizar la ejecución del código sin que muchas veces la máquina o los motores antivirus lo detecten.
- En los malware iniciales, los datos capturados eran enviados de forma clara y mayormente por el protocolo FTP a un servidor remoto. Ahora, Los datos capturados por parte del malware son enviados de forma cifrada y a distintos puertos que manejan distinto tipo de protocolos.

El tipo de entidades en las cuales se encuentra focalizado el malware han variado de forma considerable. Al principio, el malware estaba enfocado en atacar cualquier tipo de usuarios. Ahora, la mayor cantidad de malware es desarrollada con el fin de atacar entidades del sector financiero ya sea como abuso de marca o directamente para extraer información confidencial que permita continuar con el ciclo del fraude.

Tomando como punto de partida lo anterior, el análisis de las muestras de malware se ha vuelto más complejo y requiere de un procedimiento analítico y

cauteloso que permita develar las intenciones del malware a analizar.

II. TIPOS DE ANÁLISIS DE MALWARE

Varios autores sugieren dos tipos de análisis para tratar de dilucidar el comportamiento de las muestras de malware. Dichos análisis están discriminados de la siguiente manera:

- Análisis Dinámico: En el análisis dinámico, la muestra de malware es ejecutada dentro de un entorno controlado virtualizado. Al momento de la ejecución, se observan parámetros importantes del malware como registros creados y conexiones realizadas por la muestra de malware.
- Análisis Estático: En el análisis estático, la muestra analizada a profundidad del código assembler, allí se observa la ejecución del malware a nivel de memoria. Adicional a ello, se observa que DLL´s fueron usadas, en que lenguaje de alto nivel fue escrita la muestra y si se encuentra empaquetada o no.

A. Análisis Dinámico

El análisis dinámico, normalmente se encuentra subdivido en 5 categorías en las cuales se busca colectar información de acuerdo a la ejecución de la muestra de malware, en cada categoría de ellas, se busca recopilar información específica que provean indicios de la ejecución del malware.

En la primera categoría denominada ¨procesos¨ se busca que nuevos procesos fueron creados con la ejecución del malware. Una herramienta muy útil para dicho análisis se denomina ¨Process Explorer¨. La cual indica en tiempo real que procesos fueron creados o modificados de acuerdo a cierto archivo específico, que para este caso, será la muestra de malware a analizar (*ver Fig. 1*).

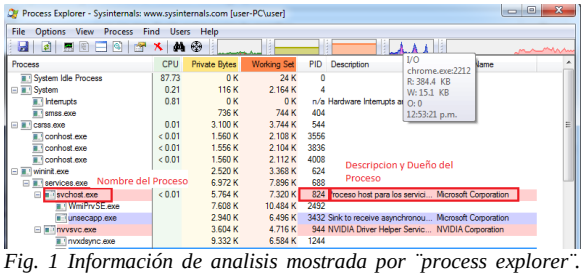

Fig. 1 Información de analisis mostrada por ¨process explorer¨. Fuente: El Autor.

Otra herramienta útil para observar dicho comportamiento es ¨Process Monitor¨ la cual muestra de una forma más detallada la relación directa de un proceso en ejecución con las llaves de registro y el sistema de carpetas y archivos que maneja la máquina. Dicha relación también es mostrada en tiempo real (*ver Fig. 2*).

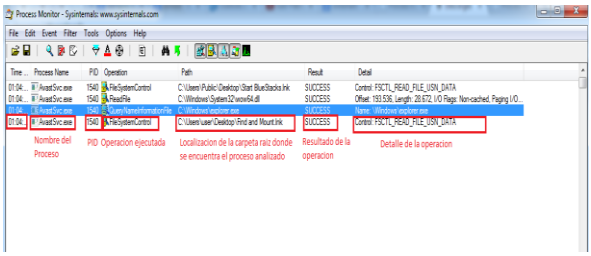

Fig. 2 Información de analisis mostrada por ¨process monitor¨. Fuente: El Autor.

La segunda categoría denominada ¨registros¨ se busca que llaves de registro fueron creadas o modificadas. Con este análisis, es posible verificar si una muestra de malware está realizando las siguientes rutinas en la ejecución del sistema operativo dentro de la máquina:

- Creación de copias del archivo de malware
- Ejecución persistente al momento de arrancar el sistema operativo.
- Ejecución de procesos sin autorización del usuario.
- Modificación en el modo de ejecución de un proceso.

Una herramienta bastante útil para observar dicho comportamiento se denomina ¨Regshot¨ la cual permite realizar comparaciones de las llaves de registro pre y post ejecución de la muestra de malware (*ver Fig. 3*). La herramienta arroja información relacionada a las llaves de registro discriminadas de la siguiente manera:

- Llaves de registro creadas al momento de la ejecución.
- Llaves de registro eliminadas al momento de la ejecución.
- Llaves de registro modificadas al momento de la ejecución.

La tercera categoría denominada ¨cadenas de texto¨ busca patrones que permitan inferir el comportamiento de la muestra de malware. Cualquier proceso ejecutado dentro del sistema operativo genera cadenas de texto de comunicación hacia otros procesos, muchas veces esta información no se encuentra codificada y es fácilmente visible.

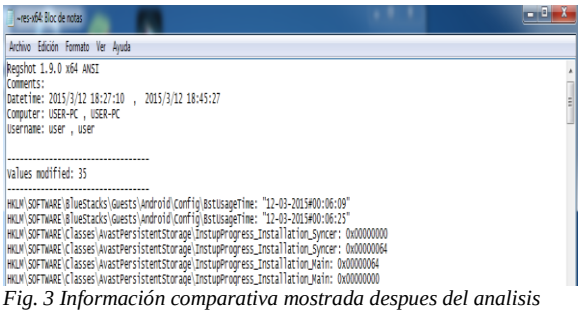

Fig. 3 Información comparativa mostrada despues del analisis realizado por la herramienta ¨regshot¨. Fuente: El Autor.

En dichas cadenas, es posible encontrar información que hace referencia a datos específicos como:

- Direcciones IP.
- URL´s.
- Datos de Autenticación.

- Nombre de Procesos de Máquina.

Una herramienta útil para realizar visualización y análisis de cadenas de texto es ¨BinText¨ la cual permite observar de una manera ordenada las cadenas de texto asociadas a un proceso especifico (*ver Fig. 4*).

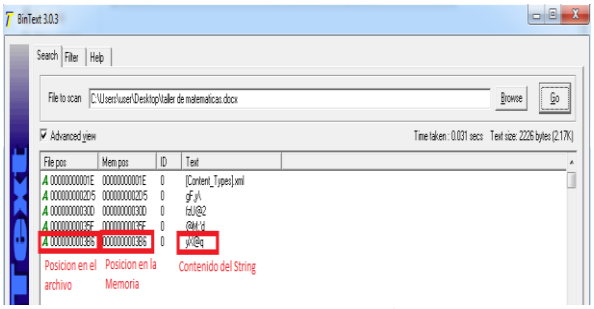

Fig. 4 Strings discriminados por tipo de codificacion y texto mostrados por la herramienta ¨Bintext¨. Fuente: El Autor.

En la cuarta categoría denominada ¨puertos y conexiones¨, se busca que direcciones IP y que puertos son usados para establecer conexión por parte del malware, en esta categoría normalmente es posible visualizar la siguiente información:

- Tipo de información enviada (si se encuentra cifrada o no).
- Puertos Usados.
- Método de envió de la información (GET o POST).
- Dominios Comprometidos.

Para realizar un análisis más asertivo, es conveniente crear una pequeña red entre dos máquinas virtuales. En una de ellas se debe encontrar instalado el sistema operativo Windows y en la otra instalado Linux. En la máquina donde se encuentra instalado Windows es donde se realiza la ejecución de la muestra de malware en la otra máquina se puede realizar el análisis de las conexiones con cualquiera de las herramientas descritas a continuación:

- Wireshark: Es un analizador de protocolos que es usado para diagnosticar y solucionar problemas de red.
- NetCat: Es una herramienta que por medio comandos permite abrir puertos TCP/UDP en un host. Después de ello, NetCat queda en modo escucha. NetCat normalmente entra en modo de escucha con el siguiente comando:

  nc – lvv –p 80.

  Dónde:-l es en modo escucha, vv indica que la escucha es en modo promiscuo y –p indica el puerto que se quiere escuchar, que en el caso de demostración es 80.
- Servidor IRC: IRC se basa en redes de máquinas. En estas redes, al menos una de las máquinas trabaja como servidor, y sus funciones son recoger todos los mensajes enviados por los usuarios y reenviarlos. Por tanto, los mensajes no son enviados directamente del emisor al destinatario, sino que pasan por el servidor. Esta filosofía de funcionamiento permite al administrador del servidor ejercer un cierto control: por ejemplo, puede impedir el acceso a determinados usuarios, ampliar las posibilidades de conversación comunicando varios servidores o limitar las posibilidades de los usuarios. Además, hace que todos los usuarios se conecten, en principio, en igualdad de condiciones, excepto el administrador.

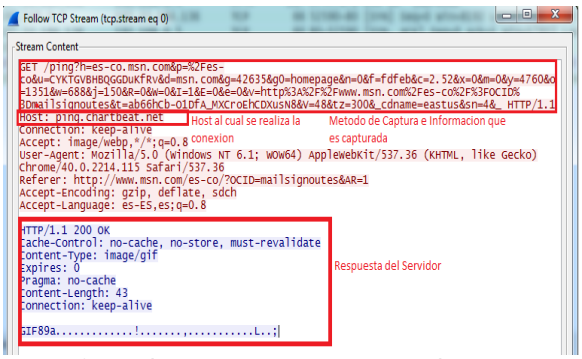

Fig. 5 Información obtenida despues de la ejecución de la herramienta ¨Wireshark¨. Fuente: El Autor.

Mientras que se realiza la ejecución controlada de la muestra de malware en la máquina que tiene instalado el sistema operativo Windows, en la máquina que tiene instalado el sistema operativo Linux se ejecutan simultáneamente las herramientas previamente mencionadas al mismo tiempo. Wireshark realiza la captura de paquetes TCP y UDP mostrando información como puerto, servidor de conexión y método de captura de la información. Netcat muestra que puertos están en modo escucha de acuerdo al protocolo seleccionado. En este punto, netcat permite mostrar puertos que en muchos casos parecen indetectables o con comportamiento normal para wireshark (*ver Fig. 5*).

Muchas veces las peticiones del malware o la ejecucuion total del mismo queda incompleta debido a que muchas solicitudes son realizadas a un servidor DNS que no existe (dentro de la red de ejecución controlada) o necesitan de un comando o una instrucción adicional para continuar la ejecución (lo que muchas veces hace indetectable la ejecución del malware para el usuario final). Es alli donde estra el servidor IRC. Normalmente, las muestra de malware realizan peticion de conexión a un servidor IRC por medio del puerto 194. Cuando la máquina victima realiza la conexión al servidor de control, recibe las instrucciones necesarias y acordes a la ejecución del malware (desde robo de información hasta control total para formar parte de una botnet). Al tener un entorno IRC funcional, es posible interactuar con la muestra de malware con el fin de saber que tipo de información esta capturando de la máquina infectada. Muchas veces, se necesita información del analisis estatico a fin de obtener credenciales de acceso a servidores remotos donde la información es enviada.

Después de realizar el análisis dinámico, es posible deducir el comportamiento básico que posee un archivo de malware, sin embargo, dicho análisis no es suficiente debido a que se hace necesario conocer el origen de la información enviada, que proceso la genera y si es posible, la llave de cifrado que se usó para él envió de la información, es allí donde entra el análisis estático.

B. Análisis Estático

En el análisis estático de una muestra de malware es importante tener en cuenta 4 conceptos que son de gran utilidad al momento de realizar dicho análisis.

- Lenguaje de programación de alto nivel: Es un lenguaje que se caracteriza por expresar los algoritmos de una manera adecuada a la capacidad cognitiva humana, en lugar de la capacidad ejecutora de las máquinas.
- Lenguaje de programación de bajo nivel: Es un lenguaje en el que sus instrucciones ejercen un control directo sobre el hardware y están condicionados por la estructura física de la máquina que lo soporta. El uso de la palabra *bajo* en su definición no implica que el lenguaje sea inferior a un lenguaje de alto nivel, si no que se refiere a la reducida abstracción entre el lenguaje y el hardware.
- Stack: Es la forma en la que la CPU realiza el ordenamiento de datos para su posterior ejecución. Dicho orden de ejecución está definido en una estructura LIFO (Last In First Out).
- Tiempos de ejecución a nivel de procesamiento: Este concepto hace referencia al orden dentro de la memoria RAM de los procesos almacenados asociados al tiempo de ejecución de acuerdo al precepto que el proceso padre brinda.

Para llevar acabo dicho análisis, se cuenta con dos herramientas que son independientes pero que deben ser ejecutadas al mismo tiempo con el fin de correlacionar los eventos que arroja cada una de ellas entre sí.

La primera de ellas es Ollydbg es un depurador de código que funciona para sistemas operativos de Windows a 32 bits. Hace análisis de Trazas de registros, reconoce procedimientos, llamadas a las API, switches, tablas, constantes y strings, así como localiza rutinas de archivos objeto y de bibliotecas poniendo especial énfasis en el análisis de código binario.

La segunda de ellas se conoce como IDA Pro el cual es un desensamblador utilizado para realizar ingeniera inversa. Dicho programa realiza el análisis usando referencias cruzadas entre las distintas secciones de código analizadas.

Este análisis es dispendioso pero muy útil ya que nos arroja información precisa y confiable de lo que realiza la muestra analizada. La información mostrada por las dos herramientas es la siguiente:

- Registros: Son las ¨variables¨ usadas por la ejecución de la muestra analizada.
- Tipo de variable, la cual puede ser tipo .text o .data, la diferencia entre las dos es que la primera es una variable estática y la segunda es una variable de ejecución.
- Las operaciones que son ejecutadas con los registros (están escritas en código assembler).

Lo que se busca al momento de utilizar las anteriores herramientas de forma conjunta es (*ver Fig. 6*):

- IDA Pro brinda que comando era ejecutado en cierta posición de memoria en cierto tiempo. Con esta información, se puede realizar la búsqueda en OllyDbg de la información que tenía la posición de memoria de interés (previamente obtenida por IDA Pro). Este procedimiento es muy útil en búsqueda de contraseñas o llaves de cifrado.
- Información de sub-procesos infectados o rutinas recurrentes.

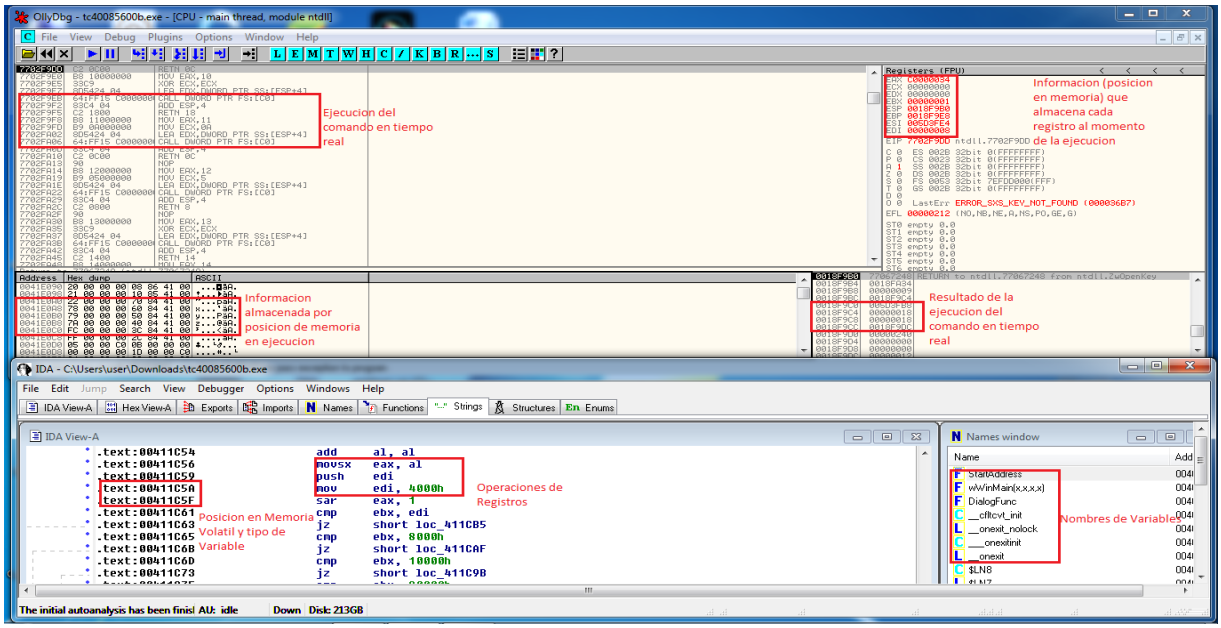

Fig. 6 Información que debe ser analizada al momento de realizar la ejecución de ¨IDA Pro¨ y ¨Ollydbg. Fuente: El Autor.

III. CONCLUSIONES

Como conclusiones del análisis realizado a los métodos comúnmente usados para la disección de muestras de malware, se pueden tomar las siguientes:

- No existe una metodología estándar para realizar análisis de malware, esto es debido a que no todas las muestras de malware están elaboradas de la misma manera ni tienen la misma finalidad de ataque.

- La metodología de análisis estático y análisis dinámico no son excluyentes entre sí, es decir cada metodología brinda información valiosa y complementaria para llevar a cabo el análisis de la muestra de malware.

- Los conocimientos para realizar análisis de malware tienen una variedad importante que pueden ir desde conocimiento de redes hasta el conocimiento de lenguajes de programación de bajo nivel.

REFERENCIAS

[1] R. Gómez. (2013) *Análisis de Malware* [online]. Disponible en: http://cryptomex.org/SlidesForensia/ForensiaMalware.pdf.

[2] P. Ramos. (2011) *Herramientas para análisis dinámico de malware* [online]. Disponible en: http://www.welivesecurity.com/la-es/2011/12/22/herramientas-analisis-dinamico-malware/.

[3] M. Russinovich. (2011) *Process Monitor v2.94* [online]. Disponible en: https://technet.microsoft.com/es-co/sysinternals/bb896645.aspx .

[4] M. Russinovich. (2007) *Process Explorer v11.02* [online]. Disponible en: https://technet.microsoft.com/es-co/sysinternals/bb896653.aspx .

[5] *¿Qué es el IRC* [online]. Disponible en: http://es.tldp.org/COMO-INSFLUG/COMOs/Servidor-IRC-Como/Servidor-IRC-Como-2.html .

[6] P. Ramos. (2014) *Las Bases del Análisis estático de Malware: Las bases del desensamblado* [online]. Disponible en: http://www.welivesecurity.com/la-es/2014/01/14/bases-analisis-estatico-malware-bases-desensamblado/ .