ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL
CENTRO DE INVESTIGACIÓN CIENTÍFICA Y TECNOLÓGICA

Análisis de Archivos de Origen Desconocido

Gabriela Rivera [1]; Dayana Vera [2]
Facultad de Ingeniería en Electricidad y Computación
Escuela Superior Politécnica del Litoral
Campus Gustavo Galindo Velasco, Km. 30.5 vía Perimetral
Apartado 09-01-5863. Guayaquil-Ecuador
gdrivera@espol.edu.ec[1]; dayevera@espol.edu.ec[2]
Febrero del 2012 – Febrero del 2013
Guayaquil-Ecuador

Director de Tesis Ing. Karina Astudillo, mail karina.astudillo@elixircorp.biz

Resumen

La presente tesis consiste en el exhaustivo análisis y obtención de información sobre archivos de origen desconocido para su revisión. Así encontrar información vital que permita descubrir pautas, acciones y procesos realizados al ejecutar los mismos, empleando métodos que distorsionen en lo menos posible los datos, con el objetivo de reconstruir todos los eventos posibles. Se tiene como indicio que varios sistemas de Windows de una organización fueron comprometidos recientemente, durante la respuesta al incidente inicial se obtuvieron imágenes forenses, y el archivo sak.exe fue encontrado en varios sistemas, el objetivo es determinar todo lo que pueda acerca de este ejecutable. Se parte en la metodología utilizada en el análisis estático y dinámico y la reingeniería inversa. Se presenta un informe que refleja el análisis forense realizado, esta asignación de datos se encuentra relacionada con el uso de aplicaciones para análisis forense en la recolección, comparación, análisis y evaluación de datos procedentes de cualquier medio informático logrando resultados de las posibles causas y objetivo de la ejecución de dicho archivo en sistemas Windows y la red de la organización en mención.

Palabras Claves: *Análisis, Ejecutable, Forense, Estático, Dinámico*

Abstract

The present work consists of the analysis and obtention of comprehensive information about unknown files for review, in order to .find vital information that allows the discovery of patterns, actions and processes undertaken to implement them, using methods that distort as little as possible the data, with the goal of reconstructing all possible events. It is known that several Windows systems were compromised recently, during the initial incident response forensic images were obtained, and the file sak.exe was found on several systems, the goal is to determine all we can about this executable. The methodology used the static and dynamic analysis and reverse engineering. A report reflecting the forensic analysis, this mapping data is related to the use of forensic analysis applications in the collection, collation, analysis and evaluation of data from any computer means achieving results possible causes and objective execution of that file in Windows systems and the network of the organization in question.

Keywords: *Analysis, Executable, Forensics, Static, Dynamic*

1. Introducción

En la actualidad hay varios indicios de constante reportes de vulnerabilidades en sistemas de información, estos ofrecen un escenario perfecto para que se cultiven tendencias relacionadas con intrusos informáticos.

En los últimos años el área de la ciencia forense es la que más ha evolucionado debido a que los últimos años los incidentes de seguridad han incrementado, los ataques que se realizan son diferentes por lo tanto constantemente se tiene que actualizar las técnicas que se utilizan.

Se tiene como indicio que varios sistemas de Windows de una organización fueron comprometidos recientemente. Equipos en respuesta a estos incidentes tomaron las medidas necesarias para responder y proteger la red.

Se presenta un informe que refleja el análisis forense realizado, esta asignación de datos se encuentra relacionada con el uso de aplicaciones para análisis forense en la recolección, comparación, análisis y evaluación de datos procedentes de cualquier medio informático logrando resultados de las posibles causas y objetivo de la ejecución de dicho archivo en sistemas Windows y la red de la organización en mención.

2. Metodología

A falta del código fuente, nos enfrentamos a un conjunto muy limitado de opciones para descubrir exactamente cómo el malware se comporta. Se parte en la metodología utilizada en el análisis estático y dinámico y la ingeniería inversa.

2.1 Análisis Estático

El análisis estático se realiza sin ejecutar el código malicioso ni desensamblar su código, con este tipo de análisis se puede determinar el tipo de archivo que se está examinando revisar las cadenas de caracteres ASCII y Unicode contenidos en el archivo.

2.2 Análisis Dinámico

El análisis dinámico se lleva a cabo cuando se ejecuta el código malicioso e interpreta su interacción con el sistema operativo. Con este tipo de análisis se puede ejecutar el programa para interceptar las llamadas al sistema, además de interactuar con el registro y poder realizar la monitorización de red.

2.3 Ingeniería Inversa

La ingeniería inversa es desarmar un objeto para ver cómo funciona con el fin de duplicar o aumentar el objeto, se utiliza para descubrir vulnerabilidades en los archivos binarios y para identificar el contenido malicioso en un programa como un virus.

2.3.1 Propósito de la Ingeniería Inversa

El propósito de las herramientas de la ingeniería inversa es a menudo para facilitar la comprensión de los programas cuando el código fuente no está disponible. Por lo general se utiliza en los siguientes casos:

- Análisis de malware
- Análisis del software de código cerrado para vulnerabilidades
- Análisis del software de código cerrado para la interoperabilidad
- Visualización de instrucciones de programa durante la depuración

3. Desarrollo del proyecto

Se nos dio un archivo comprimido lo primero que realizamos para poder realizar el análisis fue descomprimirlo, en el cual se encontró los siguiente archivos.

```
root@daya-laptop:/home/archivo# unrar e WINDOWS.rar

UNRAR 3.90 beta 2 freeware      Copyright (c) 1993-2009 Alexander Roshal


Extracting from WINDOWS.rar

Extracting  HELLO.ASM                                                 OK
Extracting  HELLO.C                                                   OK
Extracting  HELLO.EXE                                                 OK
Extracting  SAK.EXE                                                   OK
Extracting  SAK_QL~1.EXE                                              OK
Extracting  SAK_UN~1.EXE                                              OK
Extracting  SAK_UN~2.EXE                                              OK
All OK
```

Figura 1: Archivos Obtenidos

3.1 Análisis de los archivos Hello.exe

3.1.1 Compilar en C

El primer archivo a analizar es el hello.exe el cual está escrito en C lo abriremos con el programa Visual Studio.
Luego compilamos y ejecutamos el arhivo.
Se compilara el archivo como en nuestro caso estamos utilizando el programa Visual C de Microsoft usaremos el siguiente comando:

>cl hello.c
hello.c
Microsoft (R) Incremental Linker Version 10.00.30319.01

/out: hello.exe
hello.obj
Una vez compilado el código fuente se genera un archivo llamado archivo objeto o programa objeto que es luego enlazado, para generar el archivo ejecutable.
Al ejecutarlo se producirá la salida deseada en una ventana de consola.
> hello.exe
HelloWorld!

3.2 Análisis Estático

3.2.1 Hashing de ficheros con MD5DEEP

Ahora vamos a realizar un Hashing con todos los archivos que tenemos este es un valor prácticamente único para cada archivo.

```
# md5deep64 –l –z
  101 cde0ae9578275011fd4037f6cb095cfe hello.c
  964 79852d0750f1ca0e15c9d711422ecdb3 hello.asm
  631 51d150d96f9675dbf69b99cf5c71c194 hello.obj
44544 928d8c2c52b7a99fb49b1a1d9fbb474hello.exe
```

3.2.2 Uso del comando File

Este comando nos permite reconocer el formato del archivo.

```
# file hello.exe
HELLO.EXE: PE32 executable for MS Windows (console) Intel 80386 32-bit
HELLO.ASM: ASCII text, with CRLF line terminators
HELLO.C: ASCII text, with CRLF line terminators
HELLO.OBJ: ASCII text, with CRLF line terminators
```

El comando nos indica que el archivo hello.exe es un ejecutable para Intel con arquitecturax86 con entorno Windows y de 32bit. En los otros archivos la única información que nos muestra es que se trata de un código de caracteres con salto de líneas.

3.2.3 Editor Hexadecimal

El visualizador hexadecimal nos permitirá examinar el archivo entero.

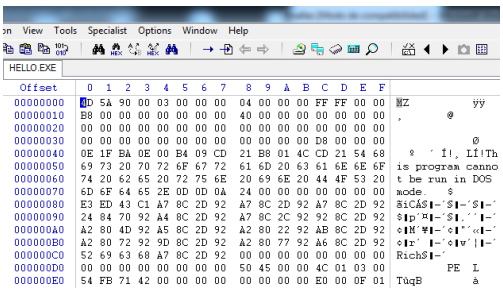

Figura 2: Abriendo el archivo con el WinHex

El primer campo de esta estructura, en el desplazamiento 0000, hay dos caracteres: "MZ", que indican que se trata de un archivo ejecutable .EXE.
Si se trata de un archivo con un programa W32, este campo apunta a dos caracteres: "PE" (Portable Ejecutable), el formato elegido por M$ para los archivos con programas W32.

3.2.4 Desensamblador IDA

La herramienta IDA que es un desensamblador empleado para ingeniería inversa.

La pantalla de texto presenta el listado completo del desmontaje de un programa y nos proporciona el único medio para visualizar las regiones de datos de un binario.

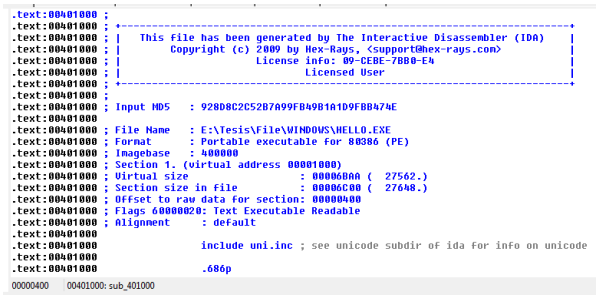

Figura 3: Abriendo el archivo en IDA

3.3 Análisis Dinámico

3.3.1 Usando StraceNT

Nos ayuda a rastrear el uso de las llamadas del sistema por un proceso ejecutado, es esencialmente una intervención telefónica entre un programa y el sistema operativo. Nos va a mostrar información sobre el acceso a archivos, acceso a redes, acceso de memoria.

3.4 Análisis de los archivos Sak.exe

Se procede al análisis del archivo SAK.EXE, será descompreso usando un ambiente LINUX CAINE.
Generamos un MD5 Hash de todos los archivos en la carpeta windows y crear con esta información una archivo de texto y enviarlo al directorio.

Para revisar las cadenas de caracteres imprimibles que contengan los ficheros, útil para visualizar ficheros que no sean, o que no se sepa que son de texto plano, ejecutaremos el comando STRING para cada uno de los ficheros SAK.

3.4.1 Archivos a analizar

stringsak.txt
Observamos las siguientes salidas del comando donde solo nos certifica que es un archivo ejecutable, hace referencia a KERNEL32-dll y LOADLIBRARY GETPROCADDRESS.

stringsak_ol.txt
Aquí observamos más funciones con la excepción, es una variación de SAK.EXE con código y una leyenda de Desempaquetado con ProcDump 32.

stringsak_un1.txt
El mismo código que el archivo SAK_OL~1.EXE pero sin la leyenda de desempaquetado.

stringsak_un2.txt
Observamos código adicional al desempaquetado. Ejecutamos el comando file *
Entendiendo que son archivos ejecutables Windows procederemos a realizar un análisis dentro de su entorno en una PC con Windows 7.

3.5 Análisis Estático

Una vez que se intenta descomprimir con el antivirus AVAST se presentan los siguientes mensajes de alerta contra los archivos.

Figura 4: Análisis con el programa Avast

3.5.1 Análisis Virus Total SAK.EXE

Figura 5. Análisis con Virus Total

Después de analizar archivo sak.exe en los enlaces web se obtiene que SAK.EXE:

- Es ejecutable lee y modifica los valores del registro. También crea y controla las claves de registro.
- Carga las Librerías: ntdll.dll y kernel32.dll

ntdll.dll es un módulo que contiene funciones de sistema del NT.

- Es un proceso del sistema necesario para que su sistema de funcione correctamente. No debe eliminarse.

kernel32.dll es el Microsoft Windows Kernel más importante.

Las funciones que tratan la mayor parte de funciones de las ventanas se conectan a este DLL del núcleo de cierta manera.

En su mayoría los programas antivirus lo reconocen como:

- Un ejecutable empaquetado con FSG.
- Una modificación de NETCAT.
- Un backdoor clásico para controlar el ordenador infectado.

Descargarnos y ejecutamos el programa Stud_PE, con estos datos creamos una tabla comparativa de los ficheros analizados.

Analizaremos si es un programa empaquetado con Fast Small good.
Verificamos la firma y encontramos que SAK.EXE se encuentra empaquetado bajo FGS 1.0.

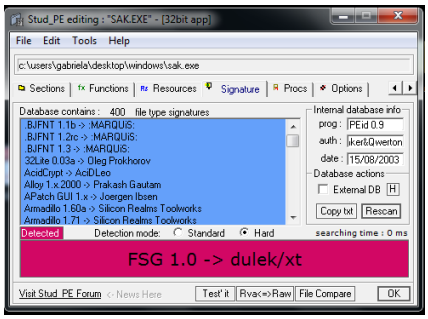

Figura 6: Análisis con el programa FSG

Con RDG Packer Detector podemos obtener datos más precisos del empaquetado de SAK.EXE

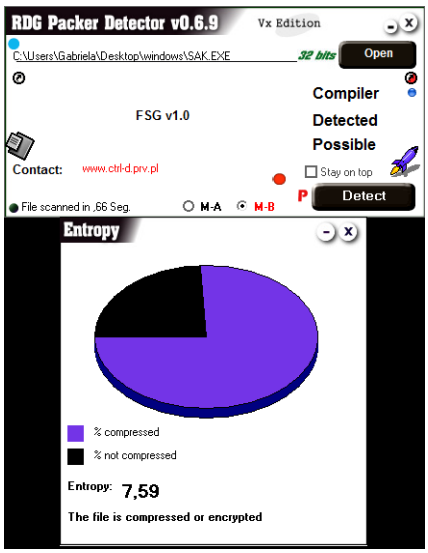

Figura 7: Análisis con RDG Packer Detector

Se utiliza el programa UNFGS para desempaquetar el fichero SAK.EXE

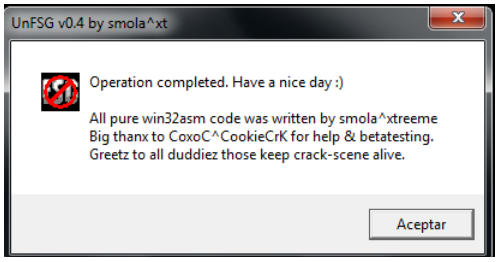

Figura 8: Desempaquetado del archivo

Los archivos creados UNFGS.EXE y SAK_UN~1.EXE tienen el mismo tamaño 87040.
El archivo SAK.EXE está desempaquetado de FSG, con la utilidad BinText, revisamos la estructura de los archivos SAK.EXE, UNFSGSAK.EXE y SAK_UN~1.EXE.

3.5.2 Análisis del archivo con IDA PRO

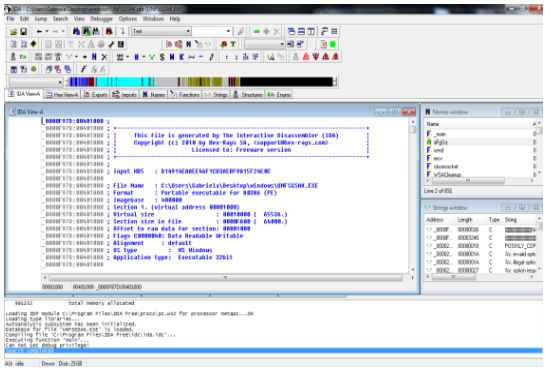

Figura 9: Análisis con el programa IDA PRO

Con las opciones de buscar, encontramos la línea donde pide una contraseña:

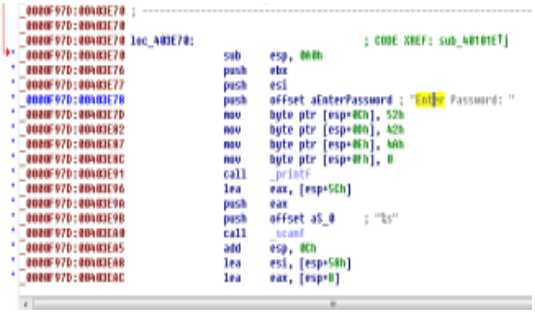

Figura 10: Password encontrado con el programa IDA

Encontramos el texto seleccionado, que es el valor que se ingresa para acceder a este ejecutable.

3.5.3 Expresiones del lenguaje ensamblador

El registro de stack, esp, es básicamente un registro que apunta a una ubicación arbitraria en memoria llamada " stack". Stack es sólo una sección muy grande de memoria temporal en donde los datos pueden ser almacenados y recuperados.

Cuando se llama a una función, un poco de espacio del stack se asigna a la función, y cuando una función devuelve el stack debe estar en el mismo estado en que comenzó.

En esta función conserva los valores para compararlos, dentro del Código ASCII:
52h R
42h B
4Ah J
0

Esta cadena de Valores vendría a ser la que se solicita al ejecutar el archivo.

3.6 Análisis Dinámico

Se ejecuta el archivo SAK.EXE y se ingresa la cadena de valores encontrada en el análisis estático.
RBJ

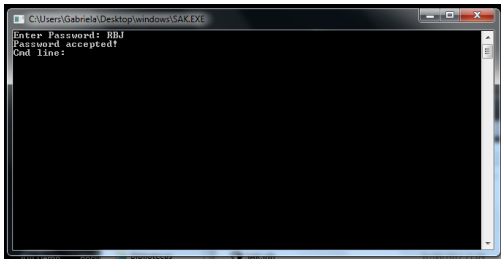

Figura 11: Ejecución del password encontrado

Si ejecuta el archivo UNFSGSAK.EXE, no se ejecuta debido a que no es un archivo ejecutable. Con la aplicación LORDPE podemos ejecutar este archivo como el fichero en análisis sin que este empaquetado.

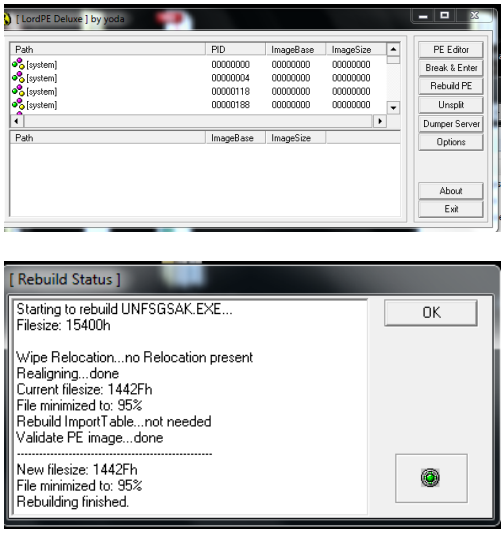

Figura 12: Análisis con el programa Lordpe

En un principio gracias a la información generada por los enlaces Web que analizaron los archivos, entendimos que puede ser:

- Una modificación de NETCAT.
- Un backdoor clásico para controlar el ordenador infectado

Si ejecutamos las Aplicaciones se puede ingresar comandos disponibles en NETC

4. Conclusiones

Después de haber realizado el análisis respectivo hemos determinado que el archivo SAK.EXE es un archivo ejecutable y los archivos SAK_OL~1.EXE, SAK_UN~1.EXE, SAK_UN~2.EXE son una variación de este, además al aplicar las herramientas necesarias rectificamos que se reconoce a los cuatro archivos

como un virus caballo de troya Win32: Trojan-gen.

Después de haber analizado el archivo **hello.exe** con los dos tipos de análisis, el estático y el dinámico empleando las herramientas necesarias, se pudo determinar que el archivo no presenta ninguna amenaza para los sistemas, sólo contiene una función de usuario principal para poder mostrar el mensaje "HelloWorld!" y luego se detiene.

Al termino del proceso investigativo hemos logrado conocer que existe una gran variedad de herramientas de libre distribución y propietarias que están consolidadas en el medio para realizar este tipo de procedimientos, las mismas que mantienen los preceptos forenses y otorgan resultados altamente confiables.

5. Recomendaciones

Utilizar herramientas de análisis adecuadas puede ayudar a la organización a prevenir futuros ataques, determinar el grado de compromiso, y determinar el número y tipo de intrusos

Disponer de una correcta gestión de parches y actualizaciones de su hardware y software, ya que gran parte de los ataques se basan en explotar un número reducido de vulnerabilidades en sistemas y aplicaciones.

6. Referencias

[1] Casey, E. (2004). *Digital Evidencie and Computer Crime.* Academy Press.

[2] HeavenTools Software. (2000). Obtenido de http://www.heaventools.com/overview.htm

[3] Mandia, K., Prosise, C., & Pepe, M. (2003). Incident Reponse & Computer Forensics, Second Edition. Estados Unidos.

[4] Carrera, E., & Elser, D. (2004). *OllyDbg Plugins*. Obtenido de http://www.openrce.org/downloads/details/108/OllyDump

[5] Eagle, C. (2011). *The IDA Pro Book.*Canada.

[6] Contreras, F. (Abril de 2009). Monografias.com. Obtenido de http://www.monografias.com/trabajos74/herramientas-computacion-forense-control-digital/herramientas-computacion-forense-control-digital.shtml

[7] Mignolo, a. (2009). Análisis básico de ejecutables. Obtenido de http://seguinfo.wordpress.com/2009/01/13/analisis-basico-de-ejecutables/