Práctica 2 Análisis básico de malware

PARTE 3

Lectura recomendada:

- Practical Malware Analysis. The Hands-On Guide to Dissecting Malicious Software. Autor: Michael Sikorski, Andrew Honig
    - Chapter 1: Basic Static Techniques
    - Chapter 2: Malware Analysis in Virtual Machines
    - Chapter 3: Basic Dynamic Analysis

Herramientas utilizadas:

- Microsoft Sysinternals, Regshot, Fakenet, CFF Explorer, etc.
- Otras herramientas:https://github.com/rshipp/awesome-malware-analysis

---

Definiciones: malware / análisis de malware

El software malicioso, o malware, juega un rol importante en la mayoría de los incidentes de seguridad. Cualquier software que haga algo que dañe al usuario, sistema, o red, puede ser considerado malware, incluyendo virus, troyanos, gusanos, rootkits, scareware y spyware.
Si bien las diversas formas de malware hacen todo tipo de cosas diferentes tenemos un conjunto de herramientas y técnicas a nuestra disposición para poder analizarlo efectivamente.
El análisis de malware es el arte de diseccionar el programa malicioso para entender que hace, como funciona, como identificarlo, como neutralizarlo, y como eliminarlo.

Con millones de programas maliciosos pululando por internet, y continuamente apareciendo nuevos, el análisis de malware es fundamental para cualquier persona que responda a incidentes de seguridad.

Objetivo del análisis de malware / Indicadores de compromiso

El propósito del análisis de malware es proporcionar la información que necesitamos para responder a una intrusión en nuestra red. El análisis de malware permite determinar que hace el binario determinado. Además de entender qué hace un malware, uno de los objetivos del análisis de malware es identificar indicadores de compromiso.

Los IOCs (**Indicators Of Compromise**) pueden ser usados para ayudar a detectar dicho malware en otras PC a partir de archivos asociados o comportamiento de red observado. Los IOC se utilizan para realizar firmas de antivirus, reglas de IDS y cualquier herramienta que se quiera utilizar para determinar si una PC está infectada.

Existen de dos tipos de IOC: basados en host y basados en red:

- Basados en el equipo host (host-based signatures), buscan cambios que pudo haber realizado el malware en el equipo infectado:
    - Archivos creados o modificados: PATH, HASH
    - Cambios en la registry.
    - Nuevos servicios creados.
    - Nuevas tareas programadas.

- De red (network signatures), usadas para detectar presencia de código malicioso al monitorear el tráfico de red:
    - URLs accedidas: Se está intentando descargar un archivo determinado
    - Direcciones IPs:
        - Se intenta realizar una conexión con un servidor determinado (por ejemplo C&C server)
        - Se intenta exfiltrar información

Técnicas de análisis de malware

Análisis estático básico

El primer enfoque para analizar una muestra de malware es mediante un análisis estático. El análisis estático, que consiste en examinar el archivo malicioso sin ejecutarlo. Se busca conseguir información sobre su funcionalidad analizando su contenido y la estructura del archivo.

A este tipo de técnica de análisis, se la considera seguro porque no es necesario ejecutar el código y por ende no representa una amenaza para la máquina donde se lo analiza.

La información útil con la que podemos dar en esta etapa:

- Detecciones de motores de antivirus existentes
- Estructura de los archivos:
    - Análisis de las secciones del ejecutable
    - Librerías y funciones dentro de las mismas importadas
    - Funciones exportadas
    - Recursos contenidos
    - Uso de empaquetador o packers
- Strings observados

Usar un antivirus para confirmar que el archivo es malicioso

Un buen primer paso para realizar este tipo de análisis es analizar el archivo sospechoso con distintos antivirus. La razón de esto es que alguno de los motores de antivirus pudo haberlo identificado y nos pueda dar información valiosa sobre la amenaza.
Los antivirus se basan principalmente en una base de datos de piezas identificables de código sospechoso (firmas de archivo), así como en el análisis de patrones y comportamiento (heurísticas) para detectar los archivos maliciosos.
Sin embargo, los antivirus no son infalibles. El problema para quien está realizando el análisis, es que los creadores del malware pueden haber modificado sutilmente el código del malware con el objeto evadir las comprobaciones de los antivirus al poseer firmas que no coinciden con las que pueden llegar a estar reconocidas por los antivirus. Puede suceder también, que el malware aún no haya sido catalogado por los antivirus y por esta razón no esté en las bases de datos. Finalmente las comprobaciones heurísticas pueden ser bypasseadas por programas con comportamientos nuevos que no hayan sido registrados previamente.
Como los distintos antivirus usan diferentes firmas y heurísticas, es útil verificar el archivo sospechoso utilizando diversos antivirus.

Estructura de los archivos ejecutables

Los archivos ejecutables, tanto de windows como Linux, tienen un formato y estructura definida, lo cual puede revelar información importante. Entre los formatos de binarios y librerías más conocidos, podemos mencionar:

- PE (Portable Executable)[1]: Formato de archivo para ejecutables (.exe), código objeto y DLLs usado en versiones de 32 y 64 bits del sistema operativo Windows. https://github.com/corkami/pics/blob/master/binary/pe101/pe101.pdf

[1] https://docs.microsoft.com/en-us/windows/desktop/debug/pe-format

- ELF (Executable and Linkable Format)[2]: Formato de archivo para ejecutables, código objeto, bibliotecas compartidas, y volcados de núcleo. Usado en Linux, Unix, Mac. https://github.com/corkami/pics/blob/master/binary/elf101/elf101.pdf

Los archivos binarios, tanto PE como ELF, están compuestos por una gran colección de headers y secciones.

- Los headers tienen un significado particular y almacenan un valor particular. Por ejemplo: Fecha de creación, última vez modificado, número de secciones, etc.
- Las secciones son colecciones lógicas de código o datos; tienen permisos (lectura, escritura, ejecución) y tienen un nombre. En un PE, las secciones más comunes son:
    - .text : Contiene el código máquina, las instrucciones del proceso que se van a ejecutar en la CPU**.**
    - .rdata: Contiene datos de solo lectura que son accesibles globalmente por el proceso.
    - .data: Contiene constantes las cuales son accesibles desde cualquier parte del programa.
    - .idata: A veces presente, almacena información sobre funciones importadas de otras librerías, si no está presente esta información se carga en la sección .rdata.
    - .edata: A veces, almacena información de funciones exportadas. Si no está presente esta información se carga en la sección .rdata.
    - .rsrc: Almacena distintos tipos de recursos utilizados por el ejecutable.

Mirando la estructura del binario, podremos conseguir información valiosa como:

- Anomalías en las secciones
- Información valiosa en la sección de recursos .rsrc
- Qué librerías externas carga para su funcionamiento.
- Que funciones de tales librerías son utilizadas.
- Que funciones exporta, al estilo de las DLLs.

Esta información es valiosa porque dependiendo de las librerías y funciones que use, nos podemos dar una idea de que hace, o al menos si el binario es malicioso.
El anexo A del libro **Practical Malware Analysis** dá información sobre las funciones consideradas importantes.

La sección de recursos puede dar información importante como menús, iconos, o incluso podría alojar un ejecutable por ejemplo.

[2] http://man7.org/linux/man-pages/man5/elf.5.html

Por último, se puede determinar si el binario está empaquetado[3]. Los escritores de malware frecuentemente empaquetan u ofuscan sus archivos para hacerlos más difíciles de detectar o analizar. Al ofuscar un programa se intenta ocultar su funcionamiento. Los programas empaquetados son un subconjunto de programas ofuscados en los cuales el programa malicioso es comprimido y no puede ser analizado fácilmente.

En caso de detectar que el binario está empaquetado, si es posible desempaquetarlo, deberíamos volver a realizar las acciones anteriores:

- Detecciones de motores de antivirus existentes
- Estructura de los archivos:
    - Análisis de las secciones del ejecutable
    - Librerías y funciones dentro de las mismas importadas
    - Funciones exportadas

Observar Strings en el binario

Adicionalmente, se puede analizar el archivo malicioso buscando strings sospechosos. En los strings se pueden identificar indicadores de compromiso como:

- Direcciones IP
- URLs
- Cadenas de caracteres

Herramientas para análisis de PE

- Antivirus: https://www.virustotal.com
- Para analizar la estructura de los archivos:
    - Para Windows: CFF Explorer / Pestudio / ResourceHacker / strings

---

Análisis dinámico básico

Cuando ya no podemos extraer más información del archivo, sea porque está fuertemente ofuscado, empaquetado o porque hemos agotado las técnicas de análisis estático disponible, podemos proceder a analizarlo dinámicamente. Con análisis dinámico, se hace referencia a monitorear el malware mientras el mismo es ejecutado. El objetivo es comprobar el estado del sistema cuando ya ha terminado la ejecución y detectar las diferencias respecto al estado previo a la ejecución.

[3] https://en.wikipedia.org/wiki/Executable_compression

Este tipo de análisis nos permite conocer la real funcionalidad del malware. Por ejemplo, la existencia de un import de una función no quiere decir que efectivamente esa función se vaya a llamar y ejecutar.

Para realizar este tipo de análisis, como vamos a estar ejecutando programas maliciosos, para evitar que se dañe el equipo y/o se infecten otras máquinas, debemos hacerlo en un ambiente seguro y controlado.

Es factible utilizar una máquina virtual y tomar las precauciones necesarias a la hora de configurar la red que la misma no pueda conectarse a otras redes para evitar que el programa infecte otras máquinas, o pase a formar parte de un nodo en un ataque de DDOS, o envíe spam, etc. Por lo general se configura en modo Host-Only creando una red privada entre el host y la vm donde el host simula los servicios de red que el malware intenta acceder. En este escenario el host es el GW de la VM y puede ser útil analizar el tráfico que nos envía la VM.

Un concepto útil a la hora de utilizar máquinas virtuales para el análisis, es el de realizar capturas utilizando snapshots de la VM. Los snapshots nos permiten salvar el estado completo del sistema para poder más tarde volver a ese mismo punto. Se debe considerar que al querer ejecutar un malware por segunda vez, no se vean todas las acciones que suceden la primera vez, por lo que el uso de los snapshots se vuelven realmente útiles.

Una buena práctica es instalar el SO, todas las herramientas de análisis que vayamos a usar, configurar la red y tomar el snapshot. A partir de allí usaremos ese snapshot como base y cuando terminemos de realizar el análisis de una muestra de malware, podremos volver al estado inicial y limpio, de manera fácil.

Herramientas útiles durante el análisis dinámico

Comparar snapshots del Registro con Regshot

Para utilizar regshot a la hora de analizar malware, tomamos un primer snapshot de la registry antes de ejecutar el malware. Luego, ejecutamos el malware y esperamos a que termine de hacer cualquier cambio en el sistema. Por último, tomamos un segundo snapshot de la registry. La herramienta comparará las dos capturas y nos mostrará rápidamente un log con las diferencias.

Monitorear con Process Monitor

Process Monitor[4] o procmon es una herramienta de Windows para monitorear actividad en el registro, el sistema de archivos, la red y los procesos.

[4] https://docs.microsoft.com/en-us/sysinternals/downloads/procmon

- Registro: Podremos ver si el malware se instala a sí mismo en el registro[5].
- Sistema de archivos: Podremos ver todos los archivos que el malware crea o los archivos de configuración que usa.
- Procesos: Podremos ver si el malware inició procesos adicionales.
- Red: Podremos identificar los puertos donde el malware está escuchando.

Procmon tiene la capacidad de capturar toda esa actividad en el periodo de tiempo que le indiquemos. Al utilizar este programa nos damos una idea más certera de que hace el programa que estamos analizando.

Analizar los procesos con Process Explorer

Process Explorer, es un poderoso administrador de tareas que debemos tener en cuenta a la hora de realizar el análisis dinámico. Provee información valiosa sobre los procesos que están corriendo actualmente en el sistema y los muestra en una estructura de árbol distinguiendo las relaciones entre procesos padres e hijos.
Podremos:

- Notar si el malware dispara procesos nuevos.
- Verificar archivos de Windows firmados digitalmente, para chequear si el malware ha reemplazado algún archivo con contenido malicioso.
- Comparar strings entre el proceso en ejecución y el archivo en disco.
- etc.

Detección rápida con servicios online.

Utilizar servicios como https://hybrid-analysis.com o https://any.run donde podemos enviar nuestra muestra de malware para que la ejecuten en una VM y nos generen un reporte rápido del comportamiento del malware.

Simular conexión a internet con Fakenet

Frecuentemente el malware intenta conectarse a internet. Con la herramienta Fakenet, podemos simular una conexión falsa y obtener indicadores de actividad de red sin estar necesariamente conectados a Internet. Estos indicadores pueden incluir nombres de dominio, direcciones ip, etc.
En caso de necesitar dar Internet al malware debemos estar muy seguros que el mismo no pasa por nuestra red.

---

[5] https://www.symantec.com/connect/articles/most-common-registry-key-check-while-dealing-virus-issue

Ejercicio 01 (estatico.exe)

Analice estáticamente la muestra de “malware” provista y responda (puede ayudarse de herramientas como CFF Explorer, strings, virustotal, hybrid-analysis):

1. Busque indicios de IOC que puedan ser usados para identificar este malware en otras máquinas o redes infectadas.
2. Mediante qué mecanismo parece que intenta persistir en el sistema?

Ejercicio 02 (calc2.exe)

Analice la muestra de “malware” provista y responda:

1. Según su comportamiento, ¿qué tipo de malware es?
2. ¿Que programa ejecuta al inicio (nombre del ejecutable)?
3. ¿A que uri accede?
4. ¿Cómo nombra al archivo que descarga?
5. ¿En qué directorio lo descarga? (Path Absoluto)
6. ¿A través de qué mecanismo logra persistencia?

Ejercicio 03 (malware1.exe)

Analice la muestra de “malware” provista y responda:

1. ¿A que uri accede?
2. ¿Cómo nombra al archivo que descarga?
3. ¿En qué directorio lo descarga?
4. ¿Qué clave/s del registro crea/modifica?
5. ¿Qué guid utiliza?
6. Específicamente como se nombra a esta técnica de persistencia? Pista: “The art of persistence…”

Ejercicio 04 (malware2.exe)

Analice la muestra de “malware” provista y responda:

1. Además de moverse se renombra, ¿Cuál es el nuevo nombre?
2. ¿Con qué clave del registro logra la persistencia?
3. Ingrese IP y puerto a la que el malware intenta conectarse (sin éxito puesto que el puerto está cerrado). Ingrese IP:puerto
4. Luego de fallar la conexión, el malware se conecta exitosamente a otro lugar. Ingrese el dominio al cual se conecta.
5. Ingrese la URL completa de conexión.
6. En la conexión exitosa se exfiltra el listado de un directorio(ls). Ingrese el path absoluto de dicho directorio.
7. ¿Qué respuesta recibió del servidor?

Ejercicio 05 (ofuscado.vbs)

Encontramos un pendrive que adentro contiene un archivo sospechoso. Analícelo.

1. ¿Qué técnica de ofuscación utiliza?
2. ¿Qué actividad realiza?

Malware en macros de Office

Las macros son una forma de automatizar tareas comunes en Microsoft Office y pueden aumentar la productividad. También tienen la capacidad de interpretar scripts programados en visual basic y acceder a funciones del sistema operativo. Todas estas características pueden ser aprovechadas por determinados malwares de macros para infectar los dispositivos de sus víctimas.

El malware de macro se esconde en archivos de Microsoft Office y es entregado a la víctima como archivos adjuntos de correo electrónico o dentro de archivos ZIP. Estos archivos usan nombres que están destinados a atraer o asustar a las personas para que los abran. A menudo parecen facturas, recibos o documentos legales entre otros.

Este tipo de malware era bastante común hace varios años porque las macros se ejecutaban automáticamente cada vez que se abría un documento office. En las versiones recientes de Microsoft Office, las macros están deshabilitadas de forma predeterminada. Ahora, los autores

de malware deben convencer a los usuarios de que activen las macros para que su malware pueda ejecutarse.

Analisis de macros:

- Se pueden utilizar tools como **oledump**[6] y **oletools**[7], para extraer el código de las macro en un archivo de office para su posterior análisis.
- También podemos utilizar herramientas de análisis estático/dinámico online como **hybrid-analysis.com** y **any.run** para ejecutar la muestra en un entorno controlado y detectar distintos indicadores de compromiso.
- En un entorno controlado se pueden habilitar las macros y utilizar en office el depurador de visual basic para realizar el análisis del script en forma dinámica y seguir su ejecución paso por paso, observando estructuras de control, valor de variables, etc.

Ejercicio 06 (reto.docm)

Un empleado ha abierto un documento de word que le llegó por mail, luego se detectó que es SPAM. Analícelo.

Ejercicio 07 (bitcoin_price_predictor.xls)

Un conocido Youtuber nos pasó un Excel para maximizar las ganancias haciendo trading!!!
Al menos espero no perder nada...

Ejercicio 08

¿Podrás poner a prueba lo que aprendiste hasta ahora en “Introducción a la Ciberseguridad” para dar un poco de luz a este desafío extraído de la Eko?

- Reto Eko - Parte 1
- Reto Eko - Parte 2

---

[6] https://github.com/DidierStevens/DidierStevensSuite/blob/master/oledump.py
[7] https://github.com/decalage2/oletools