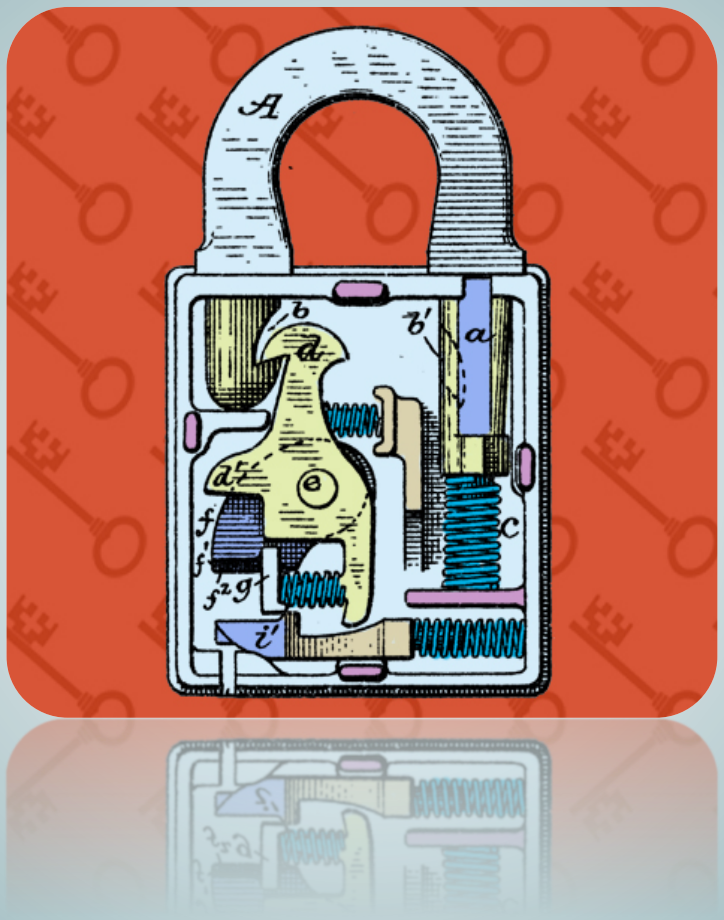
Garantía y Seguridad en Sistemas y Redes

Práctica 3. Ingeniería Inversa de Malware



Esteban Stafford

Departamento de Ingeniería
Informática y Electrónica

Práctica 3

Ingeniería inversa de malware

1. Introducción

2. Objetivos

Los objetivos fundamentales que persigue esta práctica son los siguientes:

- Introducir algunas técnicas de ingeniería inversa.
- Explorar algún ejemplo de malware.

3. Conceptos básicos

3.1. Ingeniería Inversa

En el sector de la informática se entiende ingeniería inversa como el proceso de entender el funcionamiento de un programa sin tener acceso a su código fuente. En lineas generales esto se puede hacer de dos maneras. Por un lado está el análisis estático, en donde el programa no se ejecuta y lo que se hace es investigar directamente el archivo binario. Y por otro el análisis dinámico, que implica la ejecución del programa.

3.2. Formato PE (Portable Executable)

Es el formato que Windows utiliza para sus ejecutables, librerias y otros objetos, tanto para 32 como para 64 bits. El formato encapsula toda la información que necesita el sistema operativo para cargar en memoria y ejecutar el código de los programas. Incluyendo, ademas de las secciones de código y datos, tablas de referencias a otras librerias, listas de APIs.

3.3. Compactadores de PE (PE Packers)

Resulta que el formato PE puede ser ineficiente en espacio y por ello existen compactadores de PE. Incialmente estos programas tomaban un binario en formato PE, reorganizando las diferentes secciones y eliminando partes vacias, conseguía reducir el tamaño del binario final. Estos compactadores han evolucionado con el tiempo utilizando técnicas más agresivas como la compresión, ofuscación o incluso la encriptación. Así lo que incialmente servia para reducir el tamaño de los ejecutables y librerias dinámicas, ahora es usado con los siguiente fines.

- Limitar el robo de código. Protección de la propiedad intelectual de los creadores de los algoritmos.

- Ocultar cadenas como mensajes, nombres de archivos o URLs.
- Limitar la modificación del código. En particular en lo referente a protecciones anti-copia.
- Reducir el tamaño del binario. Sigue siendo un efecto interesante ya que mejora los tiempos de descarga.

Es conveniente enfatizar que los packers son ampliamente usados, tanto en malware, como en software legítimo y comercial.

4. Desarrollo

En esta práctica se va a analizar el funcionamiento de un malware. Inicalmente se hará un análisis estático con las herramientas disponibles en la distribución RemNux, que fué concebida para el análisis de malware. La clave del usuario `remnux` es `malware`.

4.1. Captura de una imagen de Windows

Durante la práctica usaremos como víctima una distribución de WindowsXP sin protección. Es posible que ésta deje de funcionar correctamente debido al malware que vamos a analizar. Por ello es fundamental tener un mecanismo para tomar una imagen de la partición de Windows y poderla restaurar. Con este primer comando se guarda el contenido de la partición primera del disco en un archivo. Es recomendable ponerle la fecha al archivo, o la hora, si se van a manejar varias imágenes.

```
ntfsclone -s -o /extra/image_20151022.img /dev/sda1
```

Para restaurar la imagen, se usa el siguiente comando.

```
ntfsclone -r -O /dev/sda1 /extra/image_20151022.img
```

4.1.1. Malware

En esta práctica analizaremos dos muestras de malware. Se pueden descargar en las direcciones siguientes:

- `http://172.31.16.10/bootios2.zip`
- `http://172.31.16.10/credito.zip`

Ambos vienen empaquetados en archivos `zip` cifrados con la clave `infected`.

4.1.2. Herramientas de análisis Remnux

Antes de copiar el malware a la partición de Windows vamos a hacer unos análisis con las herramientas de RemNux.

4.1.3. `strings`

Esta herramienta es estándar de los sistemas UNIX. Su proposito es mostrar las cadenas de texto presentes en un fichero binario. Es de gran utilidad para obtener pistas sobre el contenido y/o proposito de un binario. Por ejemplo, nos puede mostrar mensajes de texto que se presentarán al usuario, rutas de archivos de configuración, URLs que se van a visitar o identificadores de variables del registro.

¿Qué puedes deducir de la inspección de las cadenas del malware? ¿Utiliza el executable algún packer?

4.1.4. `trid`

Este programa sirve para identificar la estructura interna de un ejecutable. Principalmente indica qué packer se esta usando.

¿Coincide el análisis de `trid` con lo observado con `strings`?

4.1.5. `bytehist`

Muestra un histograma de los bytes de un fichero, que puede dar pistas sobre su contenido.

4.1.6. Copiar malware a windows

Una vez agotadas las posibilidades de análisis estático desde RemNux. Se puede pasar a un análisis dinámico en Windows. Para copiar el ejecutable a la partición de Windows, se puede usar los siguientes comandos.

```
mount -t ntfs /dev/sda1 /mnt
cp <NOMBRE>.EXE /mnt/Documents And Settings/admin/Escritorio
umount /mnt
```

En windows se hará uso de una serie de herramientas de análisis que están disponibles en `http://172.31.16.10/tools.tar.gz`.

4.1.7. strings

Es un programa equivalente al `strings` de UNIX.

4.1.8. AspackDie

Es un desempaquetador del formato Aspack.

4.1.9. ByteHist

Es otro programa para visualizar estadísticas de los bytes de un fichero.

4.1.10. Regshot

Regshot is an open-source (LGPL) registry compare utility that allows you to quickly take a snapshot of your registry and then compare it with a second one - done after doing system changes or installing a new software product.

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

4.1.11. Process Monitor

Process Monitor is an advanced monitoring tool for Windows that shows real-time file system, Registry and process/thread activity.

http://technet.microsoft.com/en-gb/sysinternals/bb896645.aspx

4.1.12. Capture BAT

Capture BAT is a behavioral analysis tool of applications for the Win32 operating system family. Capture BAT is able to monitor the state of a system during the execution of applications and processing of documents, which provides an analyst with insights on how the software operates even if no source code is available. Capture BAT monitors state changes on a low kernel level and can easily be used across various Win32 operating system versions and configurations.

https://www.honeynet.org/node/315

4.1.13. Autoruns

Para ver qué se ejecuta en inicio y posibles modificaciones de los procesos iniciales.

http://technet.microsoft.com/es-es/sysinternals/bb963902

4.1.14. Malwarebytes

Una vez termina el escaneo genera un log de lo que ha encontrado

http://es.malwarebytes.org/

4.1.15. Combofix

Lo mismo, cuando finaliza abre un notepad con lo encontrado/limpiado. Este siempre lo ejecuto en modo seguro con red.

http://www.bleepingcomputer.com/download/combofix/

4.1.16. Process Explorer

Permite ver en tiempo real lo que se está ejecutando, desde donde se llama, e incluso monitorizar el registro.

http://technet.microsoft.com/es-es/sysinternals/bb896653

4.1.17. OllyDbg

OllyDbg is a 32-bit assembler level analysing debugger for Microsoft Windows. Emphasis on binary code analysis makes it particularly useful in cases where source is unavailable.

http://www.ollydbg.de/

- Search all referenced strings.