Análisis del Malware Tema 2

Sandboxes

La idea de las sandbox es poder analizar dinámicamente el malware en un entorno aislado. Estas Herramientas registran toda la actividad del malware y generan un reporte. Para esto también se puede usar un KVM (Kernel Virtual Machine)que nos permite arrancar una máquina virtual para analizar automáticamente la actividad de un malware, generar un reporte y recuperar la máquina a una snapshot anterior.

Cuckoo Sandbox

Inicia varias máquinas virtuales conectadas a una red virtual para analizar el comportamiento de un malware. Lo primero que se debe hacer tras instalar Cuckoo es configurar la aplicación, para ello vamos a la carpeta conf:

- cuckoo.conf: contiene la config de la VM, permite elegir entre KVM y virtualbox
- virtualbox.conf: configuración que usará el virtualbox como la plataforma y la ip
- kvm.conf: lo mismo que virtualbox.conf pero para kvm
- reporting.conf: Para definir el formato del reporte

Para inicializar cuckoo primero se debe instalar el agente en la máquina virtual (agent/agent.py). Finalmente se debe realizar una snapshot con el agente arrancado, de forma que cuckoo pueda restaurar la máquina virtual tras cada análisis. Para iniciar cuckoo se debe ejecutar el script cuckoo.py en el host. Tras eso envías el archivo que se quiere analizar con submit.py, que se encuentra en la carpeta utils. Se pueden seleccionar los formatos con el flag -package.

Los resultados del análisis se guardan en la carpeta storage/analysis:

- files: Archivos creados cuando se ejecutó el malware
- logs: Actividad del malware como llamadas a librerías del sistema dll
- reports: reporte con el formati anteriormente mencionado

Cuckoo tiene algunas limitaciones ya que al volverse muy popular algunos malwares traen contramedidas como sistemas anti análisis. Lo que hacen estos sistemas es:

- Revisar la resolución de pantalla
- Revisar el número de núcleos disponibles
- Revisar el historial del navegador (Suele estar vacío en las máquinas virtuales)

Debido a esto cuckoo está en constante evolución, al igual que el malware.

REMnux

Kit de herramientras para hacer ingeniería inversa y revisar software malicioso. Provee de una colección de herramientras creadas por la comunidad que se pueden usar para el análisis del

malware. También ofrece imágenes de docker con herramientas de análisis de malware populares.

FlareVM Sandbox

Colección de scripts que permiten montar y mantener un ambiente de ingeniería inversa en una máquina virtual. Depende de 2 principales tecnologías:

- Chocolatey: Sistema de gestión de paquetes para windows donde un paquete es un archivo zip que contiene un script de instalación de PowerShell que descarga y configura herramientas específicas.
- Boxstarter: Usa los paquetes de chocolatey para automatizar la instalación de software y crear ambientes repetibles automatizados.

Contramedidas del Malware

El malware puede tener numerosas medidas contra la ingeniería inversa:

- Ofuscación
- Ocultar información de configuración
- Encriptación de la comunicación de red
- Codificación de datos.

Ofuscación

Transformar el programa dejando las funcionalidades intactas para dificultar el entendimiento de su funcionamiento. Normalmente la ofuscación se usa para:

- Proteger propiedad intelectual: para evitar el plagio
- Hacer los antivirus menos efectivos: evita la detección de malware por firma digital
- Ralentizar el análisis de malware: Para mantener el código del malware más tiempo en funcionamiento.

Codificación de datos

Cifrado simple

Hay muchos cifrados simples que aplican operaciones de bit sobre los registros de datos:

- ADD y SUB: Añade o elimina caracteres del set de caracteres
- ROL y ROR: Rota hacia la derecha o la izquierda los bots de ciertos sets de caracteres
- ROT: Intecambia 13 veces cada caracter del alfabeto
- XOR

Algoritmos standar de ciptografía

La forma más fácil de descubrir el algoritmo es usando uno estándar:

- Mirar los strings e imports para detectar el cifrado de APIs
- Detectar el uso de constantes mágicas de encriptado.
- Analizar la entropía de un archivo para detectar partes que han sido comprimidas o cifradas.

Custom Encoding

Sistemas de codificación caseros:

- Combinar varias capas de multiples metodos de codificados
- Algoritmos completamente customizados creados por el autor del malware.

Reverse Encoding

Procedimiento estándar para encontrar que codificado se ha utilizado y deducir como descifrarlo:

1. Trazar la ejecución del programa buscando funciones de codificado y decodificado
2. Deducir cuando y como se usan estas funciones.
3. Usar el malware contra sí mismo: reprogramar las funcioanes, usarlas como son en el malware…

Usando el malware contra sí mismo

Arrancar el malware en un debugger y establecer breakpointas antes y despues del codificado o decodificado. Tiene sus problemas:

- Puede que el malware no desencripte la información que nos interesa
- No se puede deducir como hacer que el malware ejecute la función de desencriptado

Empaquetado del malware

Los diseñadores de malware suelen añadir un componente llamado run-time packer que son aplicaciones que comprimen el código como otras aplicaciones típicas como Zip o RAR. Cuando estas aplicaciones son descomprimidas con el empaquetador, la aplicación será descomprimida en la memoria de sistema en vez de en el sistema de archivos. La ventaja de esto es que el código de un malware es más pequeño y menos detectable, lo que dificulta su detección por parte de antivirus. Otra medida que se puede tomar es encriptar el malware con el empaquetador, de forma que ni los antivirus puedan desempaquetar el código.

Procedimiento

1. Descomprimir el código empaquetado

2. cargar el código empaquetado en memoria
3. Resolver las importaciones del ejecutable original
4. Transferir la ejecución al OEP (Original entry point.)

Normalmente se realizan tareas anti análisis en cada paso.