Práctica Final

Ejercicio de desarrollo (A elegir este o reglas Yara)

En este ejercicio se va a pedir desarrollar un malware de tipo Ransomware con las siguientes características:

- Escrito en Python 3
- El vector de entrada debe ser un PDF con JavaScript (preferiblemente) o un documento ofimático con macros. Este vector deberá bajar el ransomware de un servidor (público o privado).
- Opcionalmente, es recomendable crear un proxy para que el vector de entrada no contacte directamente con el servidor.
- El ransomware debe usar encriptación híbrida y encriptar todo aquel fichero que se encuentre en la ruta "C:\Users\<username>\Desktop\keepcoding". Los ficheros encriptados deben ser de tipo PDF, WORD o TXT.
- En ransomware, antes de encriptar, debe subir los ficheros que encuentre al servidor.

La entrega se realizará en un ZIP llamado *"<nombre_del_alumno_completo> - practica final.zip*" con contraseña "infected" donde existirán los siguientes directorios:

- Codigo: Directorio con todo el código necesario, así como los ejecutables ya creados.
- Memoria.pdf: Memoria donde se detalle paso a paso todo el proceso de creación del ransomware en detalle, así como pruebas del funcionamiento.

NOTA: El ejercicio es idéntico a la práctica 1.

Ejercicio de reglas Yara (A elegir este o el de desarrollo)

Este ejercicio tratará de realizar un detector de malware estático basado en reglas Yara que sea capaz de detectar malware en un porcentaje aceptable. Para ello, se hará uso de repositorios de reglas Yara de código abierto como el visto en clase de GitHub. Estos repositorios deberán compilarse para hacer match contra ficheros malware.

La entrega constará de un ZIP <nombre_alumno>_yara.zip:

- Una memoria PDF (memoria.pdf) donde se explique con detalle todos los pasos realizados.
- Un ZIP con los repositorios de reglas Yara.
- Un ZIP con todos los malware probados. Este ZIP debe llevar contraseña y debe ser “infected”.
- El compilado de las reglas Yara, que debe ser un único fichero.

Todo lo que debe hacerse en esta práctica se ha visto durante la clase 2 y en especial durante el desarrollo de la práctica 5, donde se hizo un ejemplo de integración en Python con las reglas Yara de CapeV2.

En caso de tener dificultad con Python, se puede elegir cualquier otro idioma de programación, siempre detallando todos los detalles en la memoria.

Ejercicio de análisis malware (obligatorio)

Este ejercicio tratará de analizar 2 malware dados. Una vez analizados, debe entregarse una memoria independiente por malware cuyo nombre sea *"<hash_sha256>.pdf"* con los siguientes apartados mínimos:

- Datos generales
- Análisis estático
- Análisis dinámico
- Herramientas online
- Comportamiento
- Mitigación
- Recomendaciones

Se podrán añadir todos los apartados o subapartados que se consideren, sin ningún tipo de restricción. En cada uno de los apartados, debe detallarse todos y cada uno de los pasos dados así como toda la información obtenida.

Por último, debe entregarse un PDF conjunto para ambas muestras llamado "*ticketing.pdf*" donde se detalle la información que se ha introducido en las siguientes plataformas:

- FAME
- VIPER
- MISP

Es decir, se deben crear tickets en las dos primeras plataformas que se documenten y crear un evento en MISP para compartir la información obtenida.

Todo se entregará en un ZIP que contenga los tres PDF y se llame "*<nombre_completo_del_alumno > - analisis.zip*".