



Taller de Análisis de Malware



Hack by Security

Taller de Análisis de Malware



Taller de Análisis de Malware

■ Índice

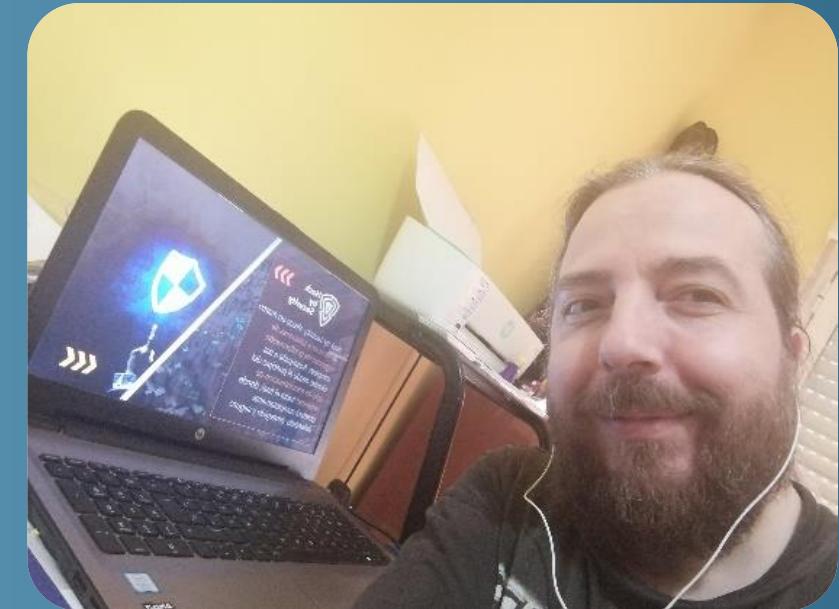
- Ping
- Análisis de malware
- Conceptos generales
- Flujo de trabajo
- IOCs
- Tipos de pruebas
 - Análisis estático
 - Análisis dinámico
 - Análisis de memoria
- Ejemplo de análisis de un malware estático
- Ejemplo de análisis de un malware en memoria
- Detectando IOCs con yara
- Ejemplo de análisis de un malware online





```
$ping RafaelGarcía  
ping: RafaelGarcía: Nombre o servicio desconocido
```

- Rafael García
- COO/CTO - CoFounder of Hack by Security
-  @Gwalrock
-  <https://mypublicinbox.com/gwalrock>





Taller de Análisis de Malware – Conceptos generales

- El malware (del inglés malicious software) es un término utilizado para describir cualquier tipo de software malintencionado diseñado para infiltrarse o dañar un sistema.
- El malware se distribuye comúnmente a través de correos electrónicos no solicitados, sitios web maliciosos, descargas de software, mensajes de texto y otros medios.





Taller de Análisis de Malware – Conceptos generales

■ Tipos de malware más comunes:

- **Virus:** Es un programa malicioso que se adjunta a un archivo legítimo y se propaga cuando se ejecuta el archivo infectado. Los virus pueden causar una variedad de daños, como la eliminación de archivos, el robo de información o el bloqueo del sistema.
- **Gusanos:** Son programas maliciosos que se propagan a través de redes y sistemas informáticos. Los gusanos pueden causar una sobrecarga en la red y ralentizar el rendimiento del sistema.
- **Troyanos:** Son programas maliciosos que se disfrazan como software legítimo para engañar al usuario para que lo descargue e instale. Los troyanos pueden permitir que un atacante acceda remotamente al sistema y lo controle.
- **Spyware:** Es un tipo de software malicioso diseñado para recopilar información del usuario sin su conocimiento o consentimiento. El spyware puede recopilar contraseñas, información financiera y otros datos sensibles.
- **Adware:** Es un software que muestra anuncios no deseados en el sistema infectado. El adware puede causar un rendimiento lento del sistema y, en algunos casos, robar información del usuario.
- **Ransomware:** Es un tipo de malware que cifra los archivos del usuario y exige un rescate para desbloquearlos. El ransomware puede causar una interrupción significativa en la productividad y causar pérdidas financieras.
- **Rootkits:** Son programas maliciosos que ocultan su presencia en el sistema infectado. Los rootkits pueden permitir a los atacantes controlar el sistema y recopilar información sin ser detectados.





Taller de Análisis de Malware – Conceptos generales

■ Ejemplos reales:

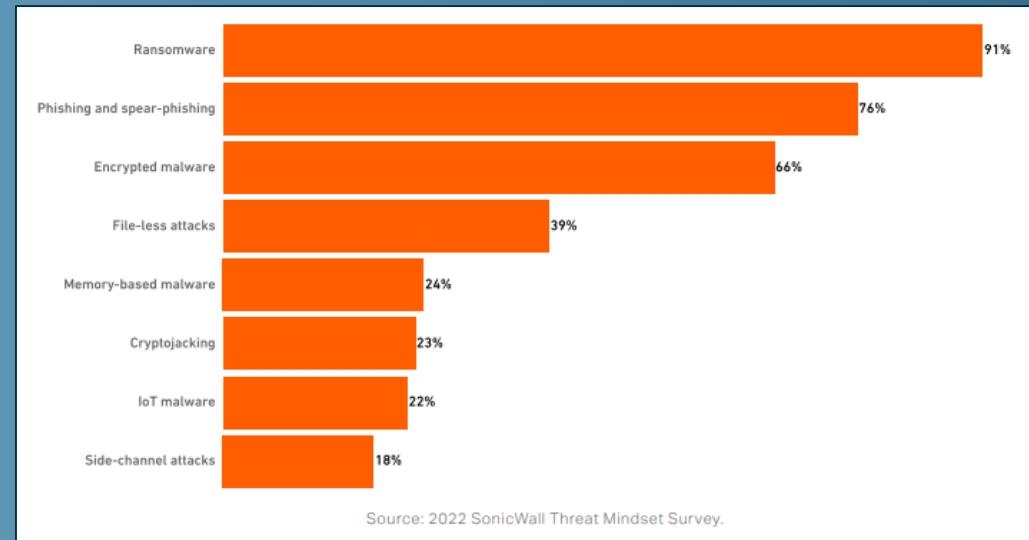
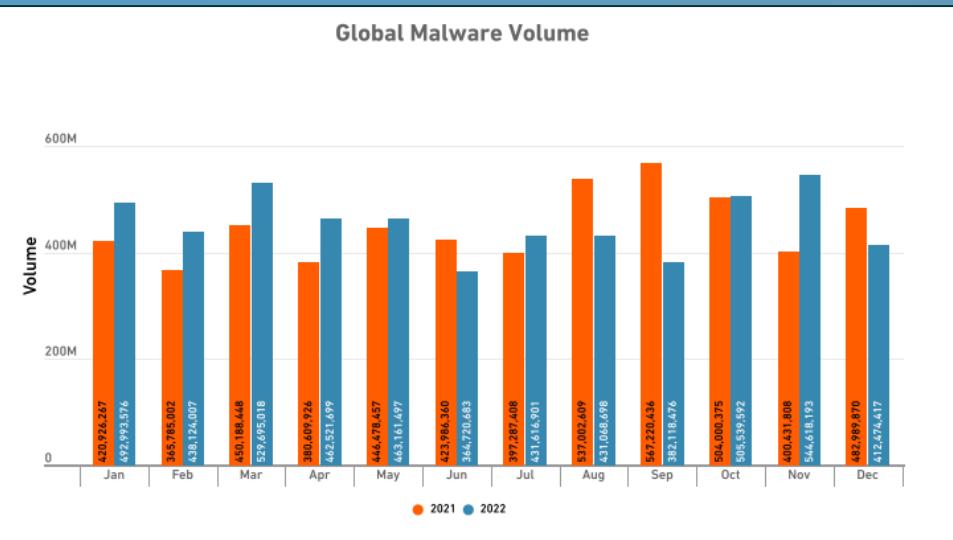
- **WannaCry:** En 2017, este ransomware se propagó rápidamente por todo el mundo y afectó a más de 200,000 sistemas en 150 países. Causó daños estimados en miles de millones de dólares y afectó a empresas, hospitales y organizaciones gubernamentales.
- **Stuxnet:** Este gusano informático fue descubierto en 2010 y se utilizó para atacar el programa nuclear de Irán. Fue el primer malware conocido diseñado para dañar físicamente equipos industriales y causó daños significativos a las centrifugadoras utilizadas en la producción de uranio.
- **Mirai:** Este malware de botnet se descubrió en 2016 y se utilizó para lanzar ataques de denegación de servicio distribuido (DDoS) a escala masiva. Se utilizó para atacar a proveedores de servicios de internet, sitios web y otros objetivos, y causó interrupciones significativas en la red.
- **NotPetya:** Este ransomware se propagó en 2017 y afectó a empresas y organizaciones en todo el mundo. Se cree que se originó en Ucrania y causó daños estimados en miles de millones de dólares.





Taller de Análisis de Malware – Conceptos generales

- Algunos datos:



- Fuente: <https://www.sonicwall.com/medialibrary/en/white-paper/2023-cyber-threat-report.pdf>





Taller de Análisis de Malware – flujo de trabajo

- **Aislar el malware:** lo primero que debes hacer es aislar el malware para que no infecte otros sistemas. Puedes hacer esto en un entorno controlado como una máquina virtual o una sandbox.
- **Identificar el tipo de malware:** identifica el tipo de malware que estás tratando de analizar. Puede ser un virus, gusano, troyano, ransomware, etc.
- **Analizar el comportamiento:** ejecuta el malware en un entorno controlado y observa su comportamiento. Puedes utilizar herramientas de monitoreo de sistemas para esto.
- **Analizar el código:** si tienes acceso al código fuente del malware, analízalo para comprender su funcionamiento interno. Si no tienes acceso al código fuente, puedes utilizar herramientas de ingeniería inversa como descompiladores y depuradores para examinar el código compilado.





Taller de Análisis de Malware – flujo de trabajo

- **Analizar el tráfico de red:** si el malware se comunica con servidores remotos, captura y analiza el tráfico de red para identificar la dirección IP del servidor remoto y el tipo de datos que se están intercambiando.
- **Identificar las vulnerabilidades explotadas:** identifica las vulnerabilidades que el malware explota para infectar sistemas. Esto te ayudará a fortalecer la seguridad de tu sistema.
- **Comprobar las firmas de virus:** utiliza herramientas antivirus para comprobar si el malware es conocido y si existen actualizaciones para eliminarlo.
- **Documentar los hallazgos:** registra y documenta tus hallazgos durante todo el proceso de análisis.





Taller de Análisis de Malware – IOCs

- Los IOCs (Indicators of Compromise) son evidencias o señales que sugieren que un sistema ha sido comprometido por malware.
- Algunos ejemplos pueden ser:
 - **Archivos maliciosos:** estos pueden ser archivos ejecutables, documentos de Office, archivos PDF, etc. Algunos de los IOCs asociados con archivos maliciosos incluyen nombres de archivo sospechosos, rutas de archivo inusuales, firmas de hash maliciosas, entre otros.
 - **Registros del sistema:** los registros del sistema, como los registros de eventos de Windows o los registros de Apache, pueden proporcionar IOCs. Algunos de estos IOCs incluyen registros de autenticación sospechosos, registros de errores de red y registros de actividad de usuario inusual.
 - **Direcciones IP:** las direcciones IP que se han utilizado para llevar a cabo ataques de malware también pueden servir como IOCs. Las direcciones IP pueden estar asociadas con servidores de comando y control, servidores de correo basura, sitios web maliciosos, entre otros.
 - **Dominios:** los nombres de dominio que se utilizan para alojar sitios web maliciosos o como parte de ataques de phishing también pueden servir como IOCs.
 - **Patrones de tráfico de red:** los patrones de tráfico de red inusuales, como una gran cantidad de tráfico dirigido a una dirección IP específica, también pueden servir como IOCs.





Taller de Análisis de Malware – Tipos de pruebas

- Para el análisis de malware básicamente se usan tres técnicas:
 - Análisis estático
 - Análisis dinámico
 - Análisis de memoria





Taller de Análisis de Malware – análisis estático

- El análisis estático de malware es un proceso en el que se examina el código del malware sin ejecutarlo.
- Se examina el código fuente del malware o el archivo binario ejecutable para determinar las funciones que realiza el malware, las bibliotecas o dependencias que utiliza, la forma en que se comunica con servidores remotos y cómo se evade la detección por parte de los antivirus y otras herramientas de seguridad.





Taller de Análisis de Malware – análisis estático

■ Herramientas:

- Para el análisis estático podemos utilizar múltiples herramientas:
 - Ghidra
 - Es una herramienta gratuita y de código abierto, permite descompilar y desensamblar código binario, así como realizar análisis de malware.
 - PEiD
 - Es una herramienta especializada en análisis de binarios de Windows. Permite identificar el tipo de archivo, el compilador utilizado, las bibliotecas y dependencias que utiliza el malware, y realizar análisis de firmas
 - Radare2
 - Es una herramienta de análisis de binarios de código abierto que permite analizar y desensamblar código binario.





Taller de Análisis de Malware – análisis estático

- Herramientas:

- Pero no debemos olvidarnos de herramientas o comandos muy útiles como:
 - File
 - xxd
 - strings





Taller de Análisis de Malware – análisis dinámico

- El análisis dinámico de malware es un proceso en el que se ejecuta el malware en un entorno controlado y monitoreado para observar su comportamiento y funcionalidad en tiempo real.
- El objetivo de este análisis es entender cómo funciona el malware y qué acciones realiza durante la ejecución, lo que puede ayudar a identificar su propósito y cómo mitigar su impacto.





Taller de Análisis de Malware – análisis dinámico

- Durante el análisis dinámico, el malware se ejecuta en una máquina virtual o un sandbox, que proporciona un entorno controlado y aislado del sistema operativo y otros recursos del equipo anfitrión.
- Se puede monitorear el comportamiento del malware mientras se ejecuta, registrar las acciones que realiza, capturar el tráfico de red, entre otros datos.





Taller de Análisis de Malware – análisis dinámico

■ Herramientas:

- Cuckoo Sandbox
 - Una herramienta de código abierto que ejecuta el malware en un entorno virtualizado y monitorea su comportamiento para detectar y analizar cualquier actividad malintencionada.
- Wireshark
 - Un analizador de protocolos de red que captura y analiza el tráfico de red
- Procmon o process explorer
 - Una herramienta de monitoreo de procesos que registra la actividad del sistema en tiempo real
- RegShot
 - Una herramienta que realiza capturas de registro antes y después de la ejecución del malware para detectar cambios en el registro de Windows





Taller de Análisis de Malware – análisis de memoria

- El análisis de memoria de malware es una técnica utilizada para examinar el contenido de la memoria de un sistema infectado con malware. La memoria del sistema contiene información crítica sobre el estado actual del sistema, incluyendo procesos en ejecución, servicios y controladores de dispositivos.
- El análisis de memoria de malware se utiliza para identificar y extraer información relevante del malware que se encuentra en la memoria, incluyendo sus procesos y recursos en uso, así como otros datos que pueden ser útiles para la investigación y análisis del malware.





Taller de Análisis de Malware – análisis de memoria

■ Cosas a mirar:

- **Procesos en ejecución:** El malware a menudo se ejecuta como un proceso en segundo plano, por lo que los investigadores pueden identificar procesos sospechosos o desconocidos en la memoria del sistema.
- **Conexiones de red:** El malware puede establecer conexiones de red para enviar o recibir información. Estas conexiones se pueden identificar a través de la memoria del sistema.
- **Archivos sospechosos:** El malware puede crear archivos o ejecutables adicionales en el sistema. Estos archivos pueden identificarse en la memoria del sistema.
- **Inyección de código:** Algunos tipos de malware utilizan técnicas de inyección de código para ocultar su presencia en el sistema. El análisis de memoria puede revelar estos códigos injectados.





Taller de Análisis de Malware – análisis de memoria

- Para realizar un análisis de memoria, normalmente pasamos por 2 pasos principales:
 - Adquisición de memoria
 - Análisis de memoria
- La adquisición de memoria es la operación de volcar cuidadosamente todo el contenido de la RAM y almacenarlo en un dispositivo de almacenamiento.
- Despues de adquirir un volcado de memoria, es hora de analizarlo.





Taller de Análisis de Malware – análisis de memoria

■ Herramientas:

- Parte forense (obtención de la memoria)
 - FTK Imager
 - The Linux Memory extractor
- Parte de análisis
 - Volatility





Taller de Análisis de Malware

**ATENCIÓN DE AQUÍ EN ADELANTE
SI NO ESTÁS SEGURO DE ALGO, NO LO HAGAS.
TRABAJA EN ENTORNOS CONTROLADOS.
UTILIZA SANDBOXES.
NO TE CONTAGIES DE TUS PROPIOS ANÁLISIS.**





Taller de Análisis de Malware - laboratorios

- Cuando realices análisis de malware, asegúrate de que realizas las pruebas en un entorno dedicado y aislado.
- Siempre es una mala idea probar y analizar malware en sistemas de producción.
- La técnica más básica es desplegar algunas máquinas virtuales aisladas (Linux y Windows) o puedes desplegar sandboxes:
 - <https://cuckoosandbox.org>
 - <https://www.mandiant.com/resources/blog/flare-vm-the-windows-malware>





Taller de Análisis de Malware - laboratorios

- En tu viaje de aprendizaje del análisis de malware, es esencial adquirir algunas muestras de malware para que puedas empezar a practicar.
- Aquí tienes unas muestras:
 - <https://github.com/endgameinc/ember>
 - <https://zeltser.com/malware-sample-sources/>
 - <http://www.tekdefense.com/downloads/malware-samples/>
 - <http://syrianmalware.com/>





Malware

Análisis estático



Hack by Security

Taller de Análisis de Malware



Taller de Análisis de Malware – análisis estático

- Podemos localizar multitud de muestras de malware en el laboratorio, dentro del directorio:
 - /home/lab/taller/theZoo/malware/Binaries
 - Los zip de las muestras están protegidos con contraseña “infected” utilízalos con responsabilidad y mucho cuidado.

```
lab@malware:~/taller/theZoo/malware/Binaries$ ls
All.ElectroRAT
Android.PegasusB
Android.Skygofree
Android.Spy.49_iBanking_Feb2014
Android.VikingHorde
AndroRat_6Dec2013
AntiExe.A
Artemis
Backdoor.MSIL.Tyupkin
BAT.Drop
Ransomware.Hive
Ransomware.Jigsaw
Ransomware.Locky
Ransomware.Mamba
Ransomware.Matsnu
Ransomware.Petrwrap
Ransomware.Petya
Ransomware.Radamant
Ransomware.RedBoot
Ransomware.Rex
Win32.APT32.Windshield
Win32.Avatar
Win32.BigBang
Win32.Boaxxe.BB
Win32.Cainxpii
Win32.Caphaw.Shylock
Win32.Carberp
Win32.Cridex
Win32.Cutwail
Win32.DarkTequila
```





Taller de Análisis de Malware – análisis estático

- Las herramientas utilizadas para este análisis serán:
 - Comandos:
 - file, xxd
 - strings
 - Herramientas
 - PEV
 - Pehash
 - Peframe
 - ...





Taller de Análisis de Malware – análisis estático

■ Trabajando con el malware

- Dentro del directorio taller/estatico encontramos el malware.
- La descomprimimos:
 - unzip Folio-854500047700.zip

```
lab@malware:~/taller/estatico$ ls -la
total 3108
drwxrwxr-x 2 lab lab    4096 mar 17 13:22 .
drwxrwxr-x 4 lab lab    4096 mar 17 13:16 ..
-rw-rw-r-- 1 lab lab 3172614 mar 17 13:22 Folio-854500047700.zip
lab@malware:~/taller/estatico$ unzip Folio-854500047700.zip
Archive: Folio-854500047700.zip
[Folio-854500047700.zip] Folio-854500047700.bin password:
      inflating: Folio-854500047700.bin
lab@malware:~/taller/estatico$
```





Taller de Análisis de Malware – análisis estático

- **Definición de hash**

- Un hash es una función criptográfica que toma una entrada (como una cadena de texto o un archivo) y produce una cadena de longitud fija que representa de manera única la entrada original.
- ¿Porqué es importante en el análisis de malware la obtención del hash?





Taller de Análisis de Malware – análisis estático

- Obtención del hash

- md5sum Folio-854500047700.bin
- Sha1sum Folio-854500047700.bin

```
lab@malware:~/taller/estatico$ md5sum Folio-854500047700.bin  
ec9eb9f2c9f5000ae6e4e2a3e4fd1daf  Folio-854500047700.bin  
lab@malware:~/taller/estatico$ sha1sum Folio-854500047700.bin  
78b5b016f5a0671db11858b2b8e683b783a44ee2  Folio-854500047700.bin  
lab@malware:~/taller/estatico$ █
```





Taller de Análisis de Malware – análisis estático

- Primeros pasos
 - Identificación de la arquitectura
 - Lo podemos hacer utilizando el comando xxd
 - xxd -g 1 Folio-854500047700.bin | more

```
lab@malware:~/taller/estatico$ xxd -g 1 Folio-854500047700.bin | more
00000000: 4d 5a 50 00 02 00 00 00 04 00 0f 00 ff ff 00 00 MZP.....@.....
00000010: b8 00 00 00 00 00 00 40 00 1a 00 00 00 00 00 00 .....!..L...
00000020: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....This program mus
00000030: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 01 00 .....t be run under W
00000040: ba 10 00 0e 1f b4 09 cd 21 b8 01 4c cd 21 90 90 .....in32..$7.....
00000050: 54 68 69 73 20 70 72 6f 67 72 61 6d 20 6d 75 73 .....@.....
00000060: 74 20 62 65 20 72 75 6e 20 75 6e 64 65 72 20 57 .....@.....
00000070: 69 6e 33 32 0d 0a 24 37 00 00 00 00 00 00 00 00 .....@.....
00000080: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....@.....
00000090: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
```

- O





Taller de Análisis de Malware – análisis estático

- Primeros pasos
 - Obtención de la arquitectura
 - También podemos utilizar el comando file
 - file Folio-854500047700.bin

```
lab@malware:~/taller/estatico$ file Folio-854500047700.bin
Folio-854500047700.bin: PE32 executable (GUI) Intel 80386, for MS Windows
```





Taller de Análisis de Malware – análisis estático

■ Herramienta PEV

- PEV (Portable Executable Viewer) es una herramienta de análisis de binarios que se utiliza para analizar archivos ejecutables de Windows y bibliotecas compartidas (DLLs).
- Proporciona información detallada sobre los archivos ejecutables, incluyendo la estructura de los encabezados PE (Portable Executable), las secciones y los recursos, la información de importación y exportación, así como la información de depuración y los datos de carga de la imagen.





Taller de Análisis de Malware – análisis estático

■ Pehash

- Nos permite obtener el hash del fichero con el que trabajamos con distintos funciones.
- pehash Folio-854500047700.bin

```
lab@malware:~/taller/estatico$ pehash Folio-854500047700.bin
file
  filepath:          Folio-854500047700.bin
  md5:              ec9eb9f2c9f5000ae6e4e2a3e4fd1daf
  sha1:              78b5b016f5a0671db11858b2b8e683b783a44ee2
  sha256:             06a39e32a13239675c197a284f06c3b933aa776ae71253791912bcb5164aa1b2
  ssdeep:            98304:QTEBexx5q1sXQc9NsFzd2aNpS/zoJcXGc5kXINYxNH:gtXQlzrpS/zfkX
  imphash:           b872d9baae5aa631a07d1d013982a7cb
```





Taller de Análisis de Malware – análisis estático

■ ssdeep

- Nos permite comparar ficheros
 - cp Folio-854500047700.bin otro.exe
 - ssdeep -pb *

```
lab@malware:~/taller/estatico$ cp Folio-854500047700.bin otro.exe
lab@malware:~/taller/estatico$ ssdeep -pb *
Folio-854500047700.bin matches otro.exe (100)

otro.exe matches Folio-854500047700.bin (100)
```





Taller de Análisis de Malware – análisis estático

- **Obtención de cadenas de texto**
 - Para obtener cadenas de texto del binario podemos utilizar el comando strings
 - `strings Folio-854500047700.bin | more`

```
lab@malware:~/taller/estatico$ strings Folio-854500047700.bin | more
This program must be run under Win32
.text
`.iTEXT
`.data
.bss
.idata
.didata
.edata
@.tls
.rdata
```





Taller de Análisis de Malware – análisis estático

■ Obtención de cadenas de texto

- Por defecto strings nos muestra la salida en ASCII, si queremos ver otras codificaciones se lo debemos indicar.
- Para extraer las cadenas Unicode le pasamos como parámetro el tipo de codificación al comando strings, en este caso **-el** donde **e:** es para establecer el tipo de codificación y **l:** es para decirle que usaremos codificación de 16-bits.
 - `strings -el Folio-854500047700.bin | more`

```
lab@malware:~/taller/estatico$ strings -el Folio-854500047700.bin | more
kernel32.dll
kernel32.dll
kernel32.dll
Software\Embarcadero\Locales
```





Taller de Análisis de Malware – análisis estático

- Obtención de cadenas de texto

- Dado que la salida de strings puede ser extensa, es recomendable filtrarla, algunos ejemplos:
 - Para obtener ejecutables
 - `strings -el Folio-854500047700.bin | grep -E ".exe"`
 - Para obtener librerías
 - `strings -el Folio-854500047700.bin | grep -E ".dll"`





Taller de Análisis de Malware – análisis estático

■ Identificación de packers

- Un packer es un programa que se utiliza para comprimir y cifrar el contenido de un archivo ejecutable con el fin de ocultar su verdadero propósito y dificultar su análisis y detección por parte de los antivirus y otras herramientas de seguridad informática.
- Identificar el packer utilizado en el malware es importante para poder aplicar las técnicas de desempaquetamiento adecuadas, analizar el malware de manera efectiva, detectarlo y atribuirlo a su origen, ya que algunos grupos utilizan packers propios.





Taller de Análisis de Malware – análisis estático

- Identificación de packers

- pepack Folio-854500047700.bin -d db_packers.txt

```
lab@malware:~/taller/estatico$ pepack Folio-854500047700.bin -d db_packers.txt
packer: BobSoft Mini Delphi -> BoB / BobSoft
```

- BobSoft Mini fue diseñado originalmente para ser utilizado en software comercial para proteger contra la ingeniería inversa y la piratería, pero su uso también se ha extendido al malware.





Taller de Análisis de Malware – análisis estático

- Propiedades del fichero
 - Entropía
 - Medida de la cantidad de información y aleatoriedad que se encuentra en el archivo
 - Fecha de compilado
 - Fecha de creación
 - Funciones para evitar ser desensamblado
 - Técnicas para dificultar la tarea de los analistas de malware
 - Ofuscación, código dinámico, encriptación, ...





Taller de Análisis de Malware – análisis estático

- Propiedades del fichero

- pscan -v Folio-854500047700.bin

```
lab@malware:~/taller/estatico$ pscan -v Folio-854500047700.bin
file entropy:                                6.415937 (normal)
fpu anti-disassembly:                         yes
imagebase:                                    normal - 0x400000
entrypoint:                                   normal - va: 0x52cfdc - raw: 0x52b9dc
DOS stub:                                     suspicious - raw: 0x40
TLS callback function:                        0x963000
TLS directory:                               found - 1 function(s)
timestamp:                                    normal - Thu, 16 Jan 2020 13:16:13 UTC
section count:                               11 (high)
sections
    section
        .text:                                 normal
```





Taller de Análisis de Malware – análisis estático

■ Mecanismos de protección

- El malware también puede hacer uso de mecanismos de protección para su propia defensa, debemos conocer si están activados o no para aprovechárnos y poder combatirlo.
- Algunos de estos mecanismos son:
 - ASLR
 - Se utiliza para ejecutar el código en zonas de memoria aleatoria
 - DEP
 - Se utiliza para evitar que se ejecute en zonas de memoria que no deba
 - SEH
 - Se utiliza para evitar que se explote el manejo de excepciones
 - Stack cookies
 - Se utiliza para prevenir desbordamientos del buffer





Taller de Análisis de Malware – análisis estático

- Mecanismos de protección
 - pesec Folio-854500047700.bin

```
lab@malware:~/taller/estatico$ pesec Folio-854500047700.bin
ASLR: no
DEP/NX: no
SEH: yes
Stack cookies (EXPERIMENTAL): yes
```





Taller de Análisis de Malware – análisis estático

- **Cabeceras PE**

- Describen la estructura de un archivo ejecutable y proporcionan información sobre cómo se debe cargar y ejecutar dicho archivo.

- **Entre la información que proporcionan las cabeceras PE se incluyen:**

- La arquitectura del procesador.
- La dirección en la que comienza el código ejecutable.
- La dirección en la que comienza la tabla de importación, que lista las funciones y datos utilizados por el archivo ejecutable que se encuentran en otros archivos.
- La dirección en la que comienza la tabla de exportación, que lista las funciones y datos que se pueden utilizar desde fuera del archivo ejecutable.
- La dirección en la que comienza la tabla de recursos, que contiene información como iconos, diálogos y cadenas de texto que se utilizan en el archivo ejecutable.
- La versión del sistema operativo Windows para la que se diseñó el archivo ejecutable.

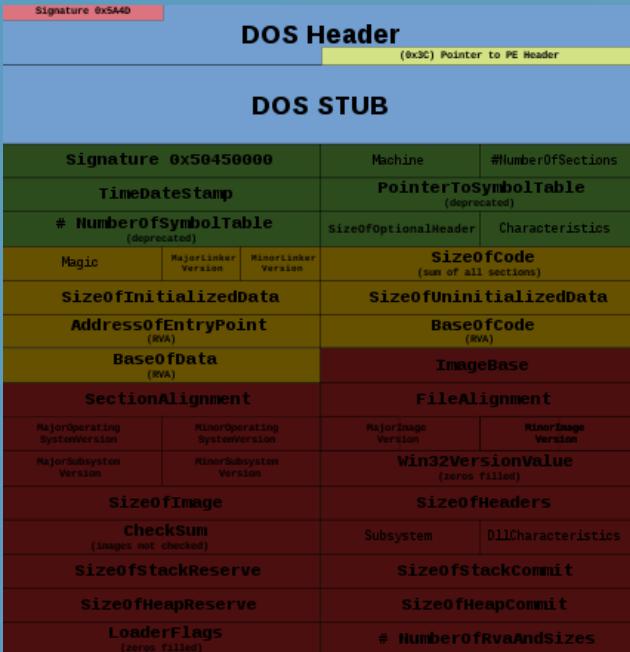




Taller de Análisis de Malware – análisis estático

■ Estructura

- https://upload.wikimedia.org/wikipedia/commons/1/1b/Portable_Executable_32_bit_Structure_in_SVG_fixed.svg



ExportTable (RVA)	SizeOfExportTable
ImportTable (RVA)	SizeOfImportTable
ResourceTable (RVA)	SizeOfResourceTable
ExceptionTable (RVA)	SizeOfExceptionTable
CertificateTable (RVA)	SizeOfCertificateTable
BaseRelocationTable (RVA)	SizeOfBaseRelocationTable
Debug (RVA)	SizeofDebug
ArchitectureData (RVA)	SizeofArchitectureData
GlobalPtr (RVA)	00 00 00 00
TLS Table (RVA)	SizeofTLS Table
LoadConfigTable (RVA)	SizeofLoadConfigTable
Bound Import (RVA)	SizeofBound Import
Import Address Table (RVA)	SizeofImport Address Table
Delay Import Descriptor (RVA)	SizeofDelay Import Descriptor
CLRRuntimeHeader (RVA)	SizeofCLRRuntimeHeader
00 00 00 00	00 00 00 00
Name	
VirtualSize	VirtualAddress (RVA)
SizeOfRawData	PointerToRawData
PointerToRelocations	PointerToLinenumbers
NumberOfRelocations	NumberOfLinenumbers
Characteristics	





Taller de Análisis de Malware – análisis estático

■ Herramienta peframe

- PEframe es una herramienta de análisis de malware diseñada para analizar archivos ejecutables de Windows en formato Portable Executable (PE). La herramienta se utiliza para examinar la estructura de archivos PE y buscar indicadores de compromiso (IOCs) y otras características de malware.
- Características:
 - Análisis del encabezado PE
 - Tablas de importación y exportación
 - Firma digital
 - Packers
 - ...





Taller de Análisis de Malware – análisis estático

■ Herramienta peframe

- peframe Folio-854500047700.bin
- Cabecera

```
File Information (time: 0:00:27.833763)
-----
filename      Folio-854500047700.bin
filetype      PE32 executable (GUI) Intel 80386, for MS Windows
filesize      8449536
hash sha256   06a39e32a13239675c197a284f06c3b933aa776ae71253791912bcb5164aa1b2
virustotal    /
imagebase     0x400000
entrypoint    0x52cfdc
imphash       b872d9baae5aa631a07d1d013982a7cb
datetime      2020-01-16 13:16:13
dll          False
directories   import, export, tls, resources, relocations
sections      .itext, .bss, .idata, .didata, .edata, .tls, .rdata, .rsrc, .text
features      mutex, antidbg, packer, crypto
```





Taller de Análisis de Malware – análisis estático

- Herramienta peframe
 - comportamiento

```
Behavior
-----
anti dbg
inject thread
network tcp listen
network dropper
network ssl
escalate priv
screenshot
keylogger
win mutex
win registry
win token
win files operation
win hook
```





Taller de Análisis de Malware – análisis estático

- Herramienta peframe
 - Funciones criptográficas

```
-.-.-.-.  
Crypto  
-.-.-.-  
Big Numbers1  
CRC32 poly Constant  
CRC32 table  
MD5 Constants  
OpenSSL DSA  
BASE64 table  
Delphi FormShow  
Delphi CompareCall
```





Taller de Análisis de Malware – análisis estático

- Herramienta peframe
 - Librerías utilizadas

Import function	
winmm.dll	1
wininet.dll	9
winspool.drv	5
comdlg32.dll	3
comctl32.dll	36
shell32.dll	8
URLMON.DLL	1
user32.dll	216
version.dll	3
olepro32.dll	1
oleaut32.dll	18
advapi32.dll	21
netapi32.dll	2
msvcrt.dll	2
kernel32.dll	153
SHFolder.dll	1
ole32.dll	16
gdi32.dll	123





Taller de Análisis de Malware – análisis estático

- Herramienta peframe
 - Direcciones IP

```
Ip Address
-----
5.4.3.3
4.4.4.4
7.6.5.4
6.5.5.4
6.5.4.3
9.9.9.9
6.5.5.5
5.3.3.3
5.4.3.1
7.6.5.3
4.3.3.3
6.5.5.3
5.5.4.4
8.7.7.6
4.3.2.1
```





Taller de Análisis de Malware – análisis estático

- Herramienta peframe

- URLs

```
Url
-----
http://ns.adobe.com/xap/1.0/
http://www.mozilla.org/editor/midasdemo/securityprefs.html
http://ns.adobe.com/xap/1.0/mm/
http://purl.org/dc/elements/1.1/
http://go.microsoft.com/fwlink/?linkid=94243.
http://fontawesome.io
http://www.w3.org/1999/02/22-rdf-syntax-ns#
http://ns.adobe.com/xap/1.0/sType/ResourceRef#
http://
http://digitalbush.com/projects/masked-input-plugin/#license
http://schemas.microsoft.com/SMI/2016/WindowsSettings
http://schemas.microsoft.com/SMI/2005/WindowsSettings
http://fontawesome.io/license/
```





Taller de Análisis de Malware – análisis estático

■ Conclusiones

- El archivo es un malware diseñado para Windows
- Arquitectura de 32 bits
- Usa el packer BobSoft Mini
- Tiene mecanismos de defensa
 - SEH
 - Stack cookies
- Altera las claves registro de Windows
- Tiene capacidades de keylogger





Malware cridex

Análisis de memoria



Hack by Security

Taller de Análisis de Malware



Taller de Análisis de Malware – cridex

- Cridex era un gusano que se propagaba a través de discos extraíbles. El gusano evolucionó a lo largo de los años hasta convertirse en un malware bancario completo.
- Las versiones posteriores del malware pueden realizar las siguientes acciones:
 - Inyecciones web
 - Capturas de pantalla y clics (imágenes de páginas web cuando el usuario hace clic en el mouse)
 - Bloquea el acceso a ciertos sitios de Internet
 - Redirige al usuario de una URL a otra Gusano.





Taller de Análisis de Malware – herramientas

- Las herramientas para este análisis de memoria serán:

- Volatility
- Bulk_extractor
- Wireshark
- Comandos
 - strings, grep, mactime





Taller de Análisis de Malware – volatility

- Volatility es un framework forense de memoria de código abierto para la respuesta a incidentes y el análisis de malware. Está escrito en Python y es compatible con Microsoft Windows, Mac OS X y Linux.





Taller de Análisis de Malware – volatility

- Instalación de volatility3

```
(kali㉿kali)-[~/tools]
$ git clone https://github.com/volatilityfoundation/volatility3.git
Cloning into 'volatility3' ...
remote: Enumerating objects: 30266, done.
remote: Counting objects: 100% (1110/1110), done.
remote: Compressing objects: 100% (598/598), done.
```

```
(kali㉿kali)-[~/tools/volatility3]
$ python3 setup.py build
running build
running build_py
creating build
creating build/lib
creating build/lib/volatility3
```





Taller de Análisis de Malware – volatility

- Instalación **volatility3** resumen:

- git clone <https://github.com/volatilityfoundation/volatility3.git>
- cd volatility3
- sudo python3 setup.py build
- sudo python3 setup.py install





Taller de Análisis de Malware – cheat sheet

- **Identificación de arquitectura**

- `python vol.py -f /home/kali/lab/cridex.vmem windows.info`

- **Trabajando con procesos**

- Listado de procesos

- `python vol.py -f /home/kali/lab/cridex.vmem windows.pslist`

- Escaneo de procesos

- `python vol.py -f /home/kali/lab/cridex.vmem windows.psscan`

- Árbol de procesos

- `python vol.py -f /home/kali/lab/cridex.vmem windows.pstree`





Taller de Análisis de Malware – cheat sheet

- **Trabajando con librerías (DLLs)**
 - Listado de dlls
 - `python vol.py -f /home/lab/taller/cridex.vmem windows.dlllist`
- **Trabajando con comandos**
 - `python vol.py -f /home/lab/taller/cridex.vmem windows.cmdline`
 - `python vol.py -f /home/lab/taller/cridex.vmem windows.sessions`
- **Obteniendo el SID**
 - `python vol.py -f /home/lab/taller/cridex.vmem windows.getsids`





Taller de Análisis de Malware – análisis de memoria

■ Primeros pasos

- Identificación de la plataforma
 - volatility -f cridex.vmem imageinfo

```
lab@malware:~/taller/memoria$ volatility -f cridex.vmem imageinfo
Volatility Foundation Volatility Framework 2.6
INFO    : volatility.debug : Determining profile based on KDBG search...
          Suggested Profile(s) : WinXPSP2x86, WinXPSP3x86 (Instantiated with WinXPSP2x86)
                                AS Layer1 : IA32PagedMemoryPae (Kernel AS)
                                AS Layer2 : FileAddressSpace (/home/lab/taller/memoria/cridex.vmem)
                                PAE type : PAE
                                DTB : 0x2fe000L
                                KDBG : 0x80545ae0L
          Number of Processors : 1
          Image Type (Service Pack) : 3
                                KPCR for CPU 0 : 0xffdff000L
                                KUSER_SHARED_DATA : 0xfffff0000L
          Image date and time : 2012-07-22 02:45:08 UTC+0000
          Image local date and time : 2012-07-21 22:45:08 -0400
```





Taller de Análisis de Malware – análisis de memoria

- Identificación de procesos

- volatility -f cridex.vmem --profile=WinXPSP3x86 pslist

```
lab@malware:~/taller/memoria$ volatility -f cridex.vmem --profile=WinXPSP3x86 pslist
Volatility Foundation Volatility Framework 2.6
Offset(V)  Name          PID  PPID  Thds  Hnds  Sess  Wow64 Start          Exit
-----  -----
0x823c89c8 System        4     0    53    240  -----  0
0x822f1020 smss.exe     368    4     3    19  -----  0 2012-07-22 02:42:31 UTC+0000
0x822a0598 csrss.exe    584   368     9    326  0      0 2012-07-22 02:42:32 UTC+0000
0x82298700 winlogon.exe 608   368    23    519  0      0 2012-07-22 02:42:32 UTC+0000
```





Taller de Análisis de Malware – análisis de memoria

- Árbol de procesos

- volatility -f cridex.vmem --profile=WinXPSP3x86 pstree

```
lab@malware:~/taller/memoria$ volatility -f cridex.vmem --profile=WinXPSP3x86 pstree
Volatility Foundation Volatility Framework 2.6
```

Name	Pid	PPid	Thds	Hnds	Time	
0x823c89c8:System	4	0	53	240	1970-01-01 00:00:00	UTC+0000
. 0x822f1020:smss.exe	368	4	3	19	2012-07-22 02:42:31	UTC+0000
.. 0x82298700:winlogon.exe	608	368	23	519	2012-07-22 02:42:32	UTC+0000
... 0x81e2ab28:services.exe	652	608	16	243	2012-07-22 02:42:32	UTC+0000
.... 0x821dfda0:svchost.exe	1056	652	5	60	2012-07-22 02:42:33	UTC+0000
..... 0x81eb17b8:spoolsv.exe	1512	652	14	113	2012-07-22 02:42:36	UTC+0000
..... 0x81e29ab8:svchost.exe	908	652	9	226	2012-07-22 02:42:33	UTC+0000
..... 0x823001d0:svchost.exe	1004	652	64	1118	2012-07-22 02:42:33	UTC+0000
..... 0x8205bda0:wuauclt.exe	1588	1004	5	132	2012-07-22 02:44:01	UTC+0000
..... 0x821fcda0:wuauclt.exe	1136	1004	8	173	2012-07-22 02:43:46	UTC+0000
.... 0x82311360:svchost.exe	824	652	20	194	2012-07-22 02:42:33	UTC+0000
.... 0x820e8da0:alg.exe	788	652	7	104	2012-07-22 02:43:01	UTC+0000





Taller de Análisis de Malware – análisis de memoria

- Conexiones externas
 - volatility -f cridex.vmem --profile=WinXPSP3x86 connscan

```
lab@malware:~/taller/memoria$ volatility -f cridex.vmem --profile=WinXPSP3x86 connscan
Volatility Foundation Volatility Framework 2.6
Offset(P) Local Address           Remote Address          Pid
-----  -----
0x02087620 172.16.112.128:1038  41.168.5.140:8080    1484
0x023a8008 172.16.112.128:1037  125.19.103.198:8080 1484
```





Taller de Análisis de Malware – análisis de memoria

■ Análisis de las conexiones externas

- Verificaremos si alguna de ellas es maliciosa
 - 41.168.5.140:8080
 - 125.19.103.198:8080
- Podemos usar el servicio <https://mxtoolbox.com/>

The screenshot shows the MXToolbox interface with the IP address 41.168.5.140 entered into a search field. Below the search field, the text "blacklist:41.168.5.140" is displayed in bold black font, followed by a green button labeled "Monitor This". A message at the bottom states "Checking 41.168.5.140 against 82 known blacklists" and "Listed 0 times with 4 timeouts".

The screenshot shows the MXToolbox interface with the IP address 125.19.103.198 entered into a search field. Below the search field, the text "blacklist:125.19.103.198" is displayed in bold black font, followed by a green button labeled "Monitor This". A red warning message at the bottom states "We notice you are on a blacklist." with a "Click here" link. A message at the bottom states "Checking 125.19.103.198 against 82 known blacklists..." and "Listed 3 times with 2 timeouts".





Taller de Análisis de Malware – análisis de memoria

■ Extracción del tráfico

■ Bulk_extractor

- alerts.txt: Contiene alertas de eventos de interés.
- ether_histogram.txt: Contiene un histograma de direcciones MAC (Media Access Control)
- ether.txt: Contiene una lista de direcciones MAC.
- ip_histogram.txt: Contiene un histograma de direcciones IP.
- ip.txt: Contiene una lista de direcciones IP.
- packets.pcap: Contiene información detallada sobre el tráfico de red capturado durante el análisis.
- report.xml: Es un informe en formato xml de la herramienta.





Taller de Análisis de Malware – análisis de memoria

- Extracción del tráfico
 - bulk_extractor -E net -o pcap cridex.vmem

```
lab@malware:~/taller/memoria$ sudo bulk_extractor -E net -o pcap cridex.vmem
bulk_extractor version: 1.5.5
Hostname: malware
Input file: cridex.vmem
Output directory: pcap
Disk Size: 536870912
Threads: 1
Attempt to open cridex.vmem
11:12:23 Offset 67MB (12.50%) Done in 0:00:30 at 11:12:53
11:12:28 Offset 150MB (28.12%) Done in 0:00:24 at 11:12:52
11:12:33 Offset 234MB (43.75%) Done in 0:00:19 at 11:12:52
11:12:36 Offset 318MB (59.38%) Done in 0:00:12 at 11:12:48
11:12:42 Offset 402MB (75.00%) Done in 0:00:07 at 11:12:49
11:12:45 Offset 486MB (90.62%) Done in 0:00:02 at 11:12:47
All data are read; waiting for threads to finish...
```





Taller de Análisis de Malware – análisis de memoria

- Análisis del tráfico de red
 - Bulk_extractor nos ha generado un serie de ficheros

```
lab@malware:~/taller/memoria/pcap$ ls -la
total 160
drwxr-xr-x 2 root root 4096 mar 18 11:12 .
drwxrwxr-x 3 lab lab 4096 mar 18 11:12 ..
-rw-r--r-- 1 root root 0 mar 18 11:12 alerts.txt
-rw-r--r-- 1 root root 215 mar 18 11:12 ether_histogram.txt
-rw-r--r-- 1 root root 11087 mar 18 11:12 ether.txt
-rw-r--r-- 1 root root 296 mar 18 11:12 ip_histogram.txt
-rw-r--r-- 1 root root 13493 mar 18 11:12 ip.txt
-rw-r--r-- 1 root root 103629 mar 18 11:12 packets.pcap
-rw-r--r-- 1 root root 9315 mar 18 11:12 report.xml
```





Taller de Análisis de Malware – análisis de memoria

- Análisis del tráfico de red
 - Detección de nuevas IPs de destino:

```
lab@malware:~/taller/memoria/pcap$ cat ip_histogram.txt
# BANNER FILE NOT PROVIDED (-b option)
# BULK_EXTRACTOR-Version: 1.5.5 ($Rev: 10844 $)
# Feature-Recorder: ip
# Filename: cridex.vmem
# Histogram-File-Version: 1.1
n=135    172.16.112.128
n=110    41.168.5.140
n=8      172.16.112.255
n=6      172.16.112.2
n=3      125.19.103.198
n=3      190.81.107.70
n=1      211.44.250.173
```





Taller de Análisis de Malware – análisis de memoria

- Análisis de las nuevas IPs
 - Ambas están en listas negras

We notice you are on a blacklist. [Click here](#)

Checking **190.81.107.70** against **82** known blacklists...
Listed **2** times with **2** timeouts

We notice you are on a blacklist.

Checking **211.44.250.173** against **82** known blacklists...
Listed **2** times with **2** timeouts





Taller de Análisis de Malware – análisis de memoria

- Análisis del tráfico de red
 - Mediante wireshark
 - Filtramos por IP: ip.addr == 41.168.5.140

ip.addr == 41.168.5.140						
No.	Time	Source	Destination	Protocol	Length	Info
61	0.000000	172.16.112.128	41.168.5.140	TCP	265	1038 → 8080 [PSH, ACK] Seq=1 Ack=2
62	0.000000	172.16.112.128	41.168.5.140	HTTP	283	POST /zb/v_01_a/in/ HTTP/1.1
63	0.000000	172.16.112.128	41.168.5.140	TCP	54	1038 → 8080 [ACK] Seq=441 Ack=2
64	0.000000	172.16.112.128	41.168.5.140	TCP	54	1038 → 8080 [ACK] Seq=441 Ack=4
65	0.000000	41.168.5.140	172.16.112.128	HTTP	1502	Continuation
66	0.000000	41.168.5.140	172.16.112.128	HTTP	1502	Continuation
67	0.000000	41.168.5.140	172.16.112.128	HTTP	1502	Continuation
68	0.000000	41.168.5.140	172.16.112.128	HTTP	1502	Continuation
69	0.000000	41.168.5.140	172.16.112.128	HTTP	1502	Continuation
70	0.000000	41.168.5.140	172.16.112.128	HTTP	1502	Continuation
71	0.000000	41.168.5.140	172.16.112.128	HTTP	1502	Continuation
72	0.000000	41.168.5.140	172.16.112.128	HTTP	1502	Continuation
73	0.000000	41.168.5.140	172.16.112.128	HTTP	1502	Continuation





Taller de Análisis de Malware – análisis de memoria

- Búsqueda de malware

- volatility -f cridex.vmem --profile=WinXPSP3x86 malfind -D malfind

```
lab@malware:~/taller/memoria$ volatility -f cridex.vmem --profile WinXPSP3x86 malfind -D malfind
Volatility Foundation Volatility Framework 2.6
Process: csrss.exe Pid: 584 Address: 0x7f6f0000
Vad Tag: Vad Protection: PAGE_EXECUTE_READWRITE
Flags: Protection: 6

0x7f6f0000  c8 00 00 00 91 01 00 00 ff ee ff ee 08 70 00 00 .....p..
0x7f6f0010  08 00 00 00 00 fe 00 00 00 00 10 00 00 20 00 00 .....
0x7f6f0020  00 02 00 00 00 20 00 00 8d 01 00 00 ff ef fd 7f .....
0x7f6f0030  03 00 08 06 00 00 00 00 00 00 00 00 00 00 00 00 .....
```





Taller de Análisis de Malware – análisis de memoria

- Búsqueda de malware
 - Dentro del directorio creado podemos observar los procesos maliciosos:

```
lab@malware:~/taller/memoria/malfind$ ls -la
total 1440
drwxrwxr-x 2 lab lab    4096 mar 18 11:27 .
drwxrwxr-x 4 lab lab    4096 mar 18 11:26 ..
-rw-rw-r-- 1 lab lab  135168 mar 18 11:27 process.0x81e7bda0.0x3d0000.dmp
-rw-rw-r-- 1 lab lab  135168 mar 18 11:27 process.0x821dea70.0x1460000.dmp
-rw-rw-r-- 1 lab lab   16384 mar 18 11:27 process.0x82298700.0x13410000.dmp
-rw-rw-r-- 1 lab lab   16384 mar 18 11:27 process.0x82298700.0x4c540000.dmp
-rw-rw-r-- 1 lab lab   16384 mar 18 11:27 process.0x82298700.0x4dc40000.dmp
-rw-rw-r-- 1 lab lab   16384 mar 18 11:27 process.0x82298700.0x4ee0000.dmp
-rw-rw-r-- 1 lab lab   16384 mar 18 11:27 process.0x82298700.0x554c0000.dmp
-rw-rw-r-- 1 lab lab   16384 mar 18 11:27 process.0x82298700.0x5de10000.dmp
-rw-rw-r-- 1 lab lab   16384 mar 18 11:27 process.0x82298700.0x6a230000.dmp
-rw-rw-r-- 1 lab lab   16384 mar 18 11:27 process.0x82298700.0x73f40000.dmp
-rw-rw-r-- 1 lab lab   16384 mar 18 11:27 process.0x82298700.0xf9e0000.dmp
-rw-rw-r-- 1 lab lab 1048576 mar 18 11:27 process.0x822a0598.0x7f6f0000.dmp
```





Taller de Análisis de Malware – análisis de memoria

- Búsqueda de malware

- Obteniendo como resultado sospechoso los procesos:
 - csrrs.exe
 - winlogon.exe
 - explorer.exe
 - reader_sl.exe





Taller de Análisis de Malware – análisis de memoria

- Obtención de información de los procesos maliciosos
 - Utilizamos el comando file

```
lab@malware:~/taller/memoria/malfind$ file *
process.0x81e7bda0.0x3d0000.dmp: PE32 executable (GUI) Intel 80386, for MS Windows
process.0x821dea70.0x1460000.dmp: PE32 executable (GUI) Intel 80386, for MS Windows
process.0x82298700.0x13410000.dmp: data
process.0x82298700.0x4c540000.dmp: data
process.0x82298700.0x4dc40000.dmp: data
process.0x82298700.0x4ee0000.dmp: data
process.0x82298700.0x554c0000.dmp: data
process.0x82298700.0x5de10000.dmp: data
process.0x82298700.0x6a230000.dmp: data
process.0x82298700.0x73f40000.dmp: data
process.0x82298700.0xf9e0000.dmp: data
process.0x822a0598.0x7f6f0000.dmp: data
```





Taller de Análisis de Malware – análisis de memoria

- Verificación de los ejecutables localizados
 - Utilizamos el comando md5sum

```
lab@malware:~/taller/memoria/malfind$ md5sum process.0x81e7bda0.0x3d0000.dmp
fb367e7c360735a58ac80fe625d9bf5a  process.0x81e7bda0.0x3d0000.dmp
lab@malware:~/taller/memoria/malfind$ md5sum process.0x821dea70.0x1460000.dmp
16a6b5e927845866d8a57eb8b7cd718e  process.0x821dea70.0x1460000.dmp
```

- Obteniendo:
 - fb367e7c360735a58ac80fe625d9bf5a
 - 16a6b5e927845866d8a57eb8b7cd718e





Taller de Análisis de Malware – análisis de memoria

- Verificación de los ejecutables localizados
 - Podemos analizar los ejecutables en virustotal.com

SHA256: cbe5f4af18753839d7e47ee41e6a6c1a1d03e806a77ba7a585ac7b7cad92450		
Name: process.0x81e7bda0.0x3d0000.dmp		
Detection ratio: 55/73		
Security vendor	Result	Update
Lionic	malicious	20230312
Elastic	malicious	20230302
Cynet	malicious	20230312
ALYac	malicious	20230312
Cylance	malicious	20230302
Zillya	malicious	20230310

SHA256: e00a1143fea8568f5bcbe2793c6b87032ba57f2fdd122266ea799658169d36b2		
Name: -		
Detection ratio: 50/62 (Analyzing...)		
Security vendor	Result	Update
Lionic	malicious	20230312
Elastic	malicious	20230302
Cynet	malicious	20230312
ALYac	malicious	20230312
Zillya	malicious	20230310
Sangfor	malicious	20230309
K7AntiVirus	malicious	20230310





Taller de Análisis de Malware – análisis de memoria

- ¿A qué ejecutables corresponden?
 - Si miramos las posiciones de memoria

```
Process: reader_sl.exe Pid: 1640 Address: 0x3d0000
Vad Tag: VadS Protection: PAGE_EXECUTE_READWRITE
Flags: CommitCharge: 33, MemCommit: 1, PrivateMemory: 1, Protection: 6
```

```
Process: explorer.exe Pid: 1484 Address: 0x1460000
Vad Tag: VadS Protection: PAGE_EXECUTE_READWRITE
Flags: CommitCharge: 33, MemCommit: 1, PrivateMemory: 1, Protection: 6
```

- Ya tenemos los procesos maliciosos
 - reader_sl.exe cuyo PID es 1640
 - explorer.exe cuyo PID es 1484





Taller de Análisis de Malware – análisis de memoria

■ Análisis de procesos maliciosos

- Vemos que objetos y ficheros están utilizando los procesos, para ello utilizamos el parámetro handles y limitaremos la salida a una serie de objetos:
 - **Key:** se refiere a un objeto del registro de Windows, que almacena información de configuración y preferencias del sistema y de las aplicaciones.
 - **Mutant:** se refiere a un objeto de sincronización del sistema que se utiliza para garantizar que solo un proceso tenga acceso a un recurso compartido en un momento dado.
 - **File:** se refiere a un objeto de archivo, que puede estar abierto por uno o varios procesos en el sistema.
 - **Event:** se refiere a un objeto de sincronización del sistema que se utiliza para señalar que se ha producido un evento en el sistema.





Taller de Análisis de Malware – análisis de memoria

- Análisis de procesos maliciosos

- volatility -f cridex.vmem --profile=WinXPSP3x86 handles -p1640 -t Key,Mutant,File,Event

```
lab@malware:~/taller/memoria$ volatility -f cridex.vmem --profile=WinXPSP3x86 handles -p1640 -t Key,Mutant,File,Event
Volatility Foundation Volatility Framework 2.6
Offset(V)      Pid      Handle      Access Type      Details
-----
0x82211678    1640      0xc      0x100020 File      \Device\HarddiskVolume1\Documents and Settings\Robert
0x82210208    1640      0x10     0x100020 File      \Device\HarddiskVolume1\WINDOWS\WinSxS\x86_Microsoft.VC80.CRT_1fc8b
e18e3b_8.0.50727.762_x-ww_6b128700
0x82319610    1640      0x1c     0x21f0003 Event
0xe1c042d0    1640      0x34     0x20f003f Key      MACHINE
0xe1835648    1640      0x40     0x20f003f Key      USER\S-1-5-21-789336058-261478967-1417001333-1003
0x820d2f28    1640      0x44     0x100020 File      \Device\HarddiskVolume1\WINDOWS\WinSxS\x86_Microsoft.Windows.Common
rolls_6595b64144ccf1df_6.0.2600.5512_x-ww_35d4ce83
0x81de10c8    1640      0x50     0x1f0003 Event
0x821dd728    1640      0x58     0x1f0003 Event
0x82196418    1640      0x5c     0x1f0003 Event
0x820022e0    1640      0x60     0x1f0003 Event
0x82002a18    1640      0x64     0x1f0003 Event
0x821dc270    1640      0x6c     0x100001 File      \Device\KsecDD
0xe1c5cfb8    1640      0x70     0x10 Key      USER\S-1-5-21-789336058-261478967-1417001333-1003\SOFTWARE\MICROSO
\8149A9A8
0x81de1e68    1640      0x78     0x1f0003 Event
0x822fdb00    1640      0x88     0x1f0001 Mutant
0x822d0d98    1640      0x8c     0x1f0003 Event
0xe154db20    1640      0x90     0x10 Key      XMM000000668
\90BBCFAD
0x81e9d708    1640      0x98     0x1f0001 Mutant
0x81e1d3c0    1640      0x9c     0x1f0003 Event      XME000000668
                                                               XMR8149A9A8
```





Taller de Análisis de Malware – análisis de memoria

- Análisis de procesos maliciosos

- volatility -f cridex.vmem --profile=WinXPSP3x86 handles -p1484 -t Key,Mutant,File,Event

```
lab@malware:~/taller/memoria$ volatility -f cridex.vmem --profile=WinXPSP3x86 handles -p1484 -t Key,Mutant,File,Event
Volatility Foundation Volatility Framework 2.6
Offset(V)      Pid      Handle      Access Type          Details
-----
0x81e7bd00    1484      0xc        0x100020 File           \Device\HarddiskVolume1\Documents and Settings\Robert
0x81e77150    1484      0x20       0x1f0001 Mutant        SHIMLIB_LOG_MUTEX
0xe1bac260    1484      0x28       0x20f003f Key          MACHINE
0x8227acb8    1484      0x2c       0x21f0003 Event        \Device\KsecDD
0x82219ad0    1484      0x3c       0x100001 File           USER\S-1-5-21-789336058-261478967-1417001333-1003
0xe18a2d00    1484      0x44       0x20f003f Key          \Device\HarddiskVolume1\WINDOWS\WinSxS\x86_Microsoft.Windows.Com
rolls_6595b64144ccf1df_6.0.2600.5512_x-ww_35d4ce83
0x8226a0d8    1484      0x4c       0x100000 Event         crypt32LogoffEvent
```





Taller de Análisis de Malware – análisis de memoria

- #### ■ Análisis de procesos maliciosos

- Ambos procesos se comunican

0x8205d460	1484	0x630	0x1f0003	Event	
0x822cf490	1484	0x634	0x1f0003	Event	XME000005CC
0xe1881d68	1484	0x638	0xf003f	Key	MACHINE\SYSTEM\CONTROLSET001\SERV
LOG5					
0x81e9d708	1484	0x63c	0x1f0001	Mutant	XMR8149A9A8
0x822403c8	1484	0x640	0x1f0001	Mutant	XMQ8149A9A8
0x82278190	1484	0x644	0x1f0003	Event	B8149A9A8
0x8216ee30	1484	0x648	0x1f0001	Mutant	XMS8149A9A8
0x8223ded8	1484	0x64c	0x1f0003	Event	XMF8149A9A8

- Se está modificando el registro

0x821dc270	1640	0x6c	0x100001 File	\Device\KsecDD
0xe1c5cfb8	1640	0x70	0x10 Key	USER\S-1-5-21-789336058-261478967-1417001333-1003\SOFTWARE\MICROSOFT
\8149A9A8				
0x81de1e68	1640	0x78	0x1f0003 Event	
0x822fdb00	1640	0x88	0x1f0001 Mutant	XMM00000668
0x822d0d98	1640	0x8c	0x1f0003 Event	XME00000668
0xe154db20	1640	0x90	0x10 Key	USER\S-1-5-21-789336058-261478967-1417001333-1003\SOFTWARE\MICROSOFT
\9DBBCFAD				
0x81e9d708	1640	0x98	0x1f0001 Mutant	XMR8149A9A8
0x81e1d3c0	1640	0x9c	0x1f0003 Event	





Taller de Análisis de Malware – análisis de memoria

■ Persistencia

- Para ver si un programa se inicia automáticamente, podemos mirar la clave de registro "Microsoft\Windows\CurrentVersion\Run"
- La clave de registro "Microsoft\Windows\CurrentVersion\Run" es una ubicación en el registro de Windows que se utiliza para almacenar una lista de programas que se ejecutan automáticamente cada vez que se inicia el sistema operativo. Estos programas se inician automáticamente en segundo plano, sin la intervención del usuario.





Taller de Análisis de Malware – análisis de memoria

■ Persistencia

- volatility -f cridex.vmem --profile=WinXPSP3x86 printkey -K "Microsoft\Windows\CurrentVersion\Run"

```
lab@malware:~/taller/memoria$ volatility -f cridex.vmem --profile=WinXPSP3x86 printkey -K "Microsoft\Windows\CurrentVersion\Run"
Volatility Foundation Volatility Framework 2.6
Legend: (S) = Stable   (V) = Volatile

-----
Registry: \Device\HarddiskVolume1\WINDOWS\system32\config\software
Key name: Run (S)
Last updated: 2012-02-18 20:09:37 UTC+0000

Subkeys:
(S) OptionalComponents

Values:
REG_SZ      Adobe Reader Speed Launcher : (S) "C:\Program Files\Adobe\Reader 9.0\Reader\Reader_sl.exe"
```





Taller de Análisis de Malware – análisis de memoria

- Volcado de memoria de procesos maliciosos

- volatility -f cridex.vmem --profile=WinXPSP3x86 memdump -p1484,1640 -D memdumps/

```
lab@malware:~/taller/memoria$ volatility -f cridex.vmem --profile=WinXPSP3x86 memdump -p1484,1640 -D memdumps
Volatility Foundation Volatility Framework 2.6
*****
Writing explorer.exe [ 1484] to 1484.dmp
*****
Writing reader_sl.exe [ 1640] to 1640.dmp
```





Taller de Análisis de Malware – análisis de memoria

■ Análisis del volcado de memoria

- Podemos utilizar el comando strings para obtener información, de esta manera obtendremos las cadenas de texto
 - strings 1484.dmp | more
 - Se ven IPs, mails, comunicaciones (se han visto anteriormente con wireshark)
 - Si queremos ver las conexiones, podríamos hacer un grep
 - strings 1484.dmp | grep http://
 - strings 1640 .dmp | more
 - Se ven conexiones bancarias
 - Ips,..





Taller de Análisis de Malware – análisis de memoria

■ Generación de línea temporal

- Para crear una línea temporal de lo sucedido, usaremos los plugins de volatility:
 - timeliner: se utiliza para crear una línea de tiempo de actividad del sistema operativo. La línea de tiempo muestra una lista de eventos relevantes que se han registrado en el sistema, junto con la fecha y hora en que ocurrieron.
 - Mftparser: se utiliza para analizar el archivo de tabla de archivos maestros (MFT) de un sistema de archivos NTFS. El MFT es una base de datos que contiene información sobre cada archivo y carpeta en un volumen NTFS.
 - Shellbags: se utiliza para analizar la información de "Shellbags" de un usuario. Los Shellbags son registros en el registro de Windows que contienen información sobre la forma en que se mostraron las carpetas en el Explorador de Windows, como el tamaño y la ubicación de la ventana, la posición de la barra de desplazamiento y el orden de visualización de los archivos.





Taller de Análisis de Malware – análisis de memoria

■ Generación de línea temporal

- Timeliner
 - `volatility -f cridex.vmem --profile=WinXPSP3x86 timeliner --output-file=timeline/timeliner.txt --output=body`

- Mftparser
 - `volatility -f cridex.vmem --profile=WinXPSP3x86 mftparser --output-file=timeline/mftparser.txt --output=body`

- Shellbags
 - `volatility -f cridex.vmem --profile=WinXPSP3x86 shellbags --output-file=timeline/shellbags.txt --output=body`





Taller de Análisis de Malware – análisis de memoria

- Generación de línea temporal

```
lab@malware:~/taller/memoria$ volatility -f cridex.vmem --profile=WinXPSP3x86 timeliner --output-file=timeline/timeliner.txt --output=body
Volatility Foundation Volatility Framework 2.6
Outputting to: timeline/timeliner.txt
WARNING : volatility.debug      : No ShimCache data found
lab@malware:~/taller/memoria$ volatility -f cridex.vmem --profile=WinXPSP3x86 mftparser --output-file=timeline/mftparser.txt --output=body
Volatility Foundation Volatility Framework 2.6
Outputting to: timeline/mftparser.txt
Scanning for MFT entries and building directory, this can take a while
lab@malware:~/taller/memoria$ volatility -f cridex.vmem --profile=WinXPSP3x86 shellbags --output-file=timeline/shellbags.txt --output=body
Volatility Foundation Volatility Framework 2.6
Scanning for registries...
Gathering shellbag items and building path tree...
Outputting to: timeline/shellbags.txt
```





Taller de Análisis de Malware – análisis de memoria

- Generación de línea temporal
 - Unimos los tres ficheros

```
lab@malware:~/taller/memoria/timeline$ ls -la
total 1092
drwxrwxr-x 2 lab lab    4096 mar 18 16:58 .
drwxrwxr-x 6 lab lab    4096 mar 18 16:56 ..
-rw-rw-r-- 1 lab lab  635192 mar 18 16:58 mftparser.txt
-rw-rw-r-- 1 lab lab   1553 mar 18 16:58 shellbags.txt
-rw-rw-r-- 1 lab lab 466379 mar 18 16:57 timeliner.txt
lab@malware:~/taller/memoria/timeline$ cat timeliner.txt > timelinecompleto.txt
lab@malware:~/taller/memoria/timeline$ cat mftparser.txt >> timelinecompleto.txt
lab@malware:~/taller/memoria/timeline$ cat shellbags.txt >> timelinecompleto.txt
```





Taller de Análisis de Malware – análisis de memoria

- Análisis de la línea temporal

- Con mactime hacemos el filtrado:

- `mactime -b timelinecompleto.txt -d -z UTC-0400 | grep reader_sl`

```
lab@malware:~/taller/memoria/timeline$ mactime -b timelinecompleto.txt -d -z UTC-0400 | grep reader_sl
Xxx Xxx 00 0000 00:00:00,0,m...,-----,0,0,0,"[PROCESS] reader_sl.exe PID: 1640/PPID: 1484/POffset: 0x0207bda0"
Xxx Xxx 00 0000 00:00:00,0,m...,-----,0,0,0,"[THREAD] reader_sl.exe PID: 1640/TID: 1332"
Xxx Xxx 00 0000 00:00:00,0,m...,-----,0,0,0,"[THREAD] reader_sl.exe PID: 1640/TID: 1448"
Xxx Xxx 00 0000 00:00:00,0,m...,-----,0,0,0,"[THREAD] reader_sl.exe PID: 1640/TID: 1600"
Xxx Xxx 00 0000 00:00:00,0,m...,-----,0,0,0,"[THREAD] reader_sl.exe PID: 1640/TID: 1644"
Xxx Xxx 00 0000 00:00:00,0,m...,-----,0,0,0,"[THREAD] reader_sl.exe PID: 1640/TID: 1648"
Sun Dec 03 2006 06:50:32,0,macb,-----,0,0,0,"[PE DEBUG] MSVCR80.dll Process: reader_sl.exe/PID: 1640/PPID: 1484/
207bda0/DLL Base: 0x78130000"
```

```
lab@malware:~/taller/memoria/timeline$ mactime -b timelinecompleto.txt -d -z UTC-0400 | grep explorer
Xxx Xxx 00 0000 00:00:00,0,m...,-----,0,0,0,"[PROCESS] explorer.exe PID: 1484/PPID: 1464/POffset: 0x023dea70"
Xxx Xxx 00 0000 00:00:00,0,m...,-----,0,0,0,"[THREAD] explorer.exe PID: 1484/TID: 1444"
Xxx Xxx 00 0000 00:00:00,0,m...,-----,0,0,0,"[THREAD] explorer.exe PID: 1484/TID: 1452"
Xxx Xxx 00 0000 00:00:00,0,m...,-----,0,0,0,"[THREAD] explorer.exe PID: 1484/TID: 1456"
Xxx Xxx 00 0000 00:00:00,0,m...,-----,0,0,0,"[THREAD] explorer.exe PID: 1484/TID: 1472"
Xxx Xxx 00 0000 00:00:00,0,m...,-----,0,0,0,"[THREAD] explorer.exe PID: 1484/TID: 1476"
Xxx Xxx 00 0000 00:00:00,0,m...,-----,0,0,0,"[THREAD] explorer.exe PID: 1484/TID: 1488"
Xxx Xxx 00 0000 00:00:00,0,m...,-----,0,0,0,"[THREAD] explorer.exe PID: 1484/TID: 1540"
Xxx Xxx 00 0000 00:00:00,0,m...,-----,0,0,0,"[THREAD] explorer.exe PID: 1484/TID: 1548"
Xxx Xxx 00 0000 00:00:00,0,m...,-----,0,0,0,"[THREAD] explorer.exe PID: 1484/TID: 1552"
```





Taller de Análisis de Malware – análisis de memoria

■ Conclusiones

- El archivo es un malware diseñado para Windows
- Arquitectura de 32 bits
- Hace conexiones externas con múltiples IP, algunas en listas negras
 - 41.168.5.140:8080
 - 125.19.103.198:8080
- Distinguimos un patrón de conexión /zb/v_01_a/in
- Existen 4 procesos sospechosos
- Existen 2 ejecutables maliciosos reader y explorer
- Se acceder a las claves de registro
- A través del ejecutable reader se consigue la persistencia





Malware

IOCs



Hack by Security

Taller de Análisis de Malware



Taller de Análisis de Malware – detectando IOCs

- Para detectar IOCs vamos a utilizar YARA
- Yara es una herramienta de detección de malware que se utiliza para identificar patrones y firmas específicas en archivos y procesos en base a unas reglas.

```
lab@malware:~/taller/memoria$ yara --help
YARA 3.7.1, the pattern matching swiss army knife.
Usage: yara [OPTION]... [NAMESPACE:]RULES_FILE... FILE | DIR | PID

Mandatory arguments to long options are mandatory for short options too.

-t, --tag=TAG          print only rules tagged as TAG
-i, --identifier=IDENTIFIER  print only rules named IDENTIFIER
-c, --count            print only number of matches
-n, --negate           print only not satisfied rules (negate)
-D, --print-module-data  print module data
-g, --print-tags        print tags
```





Taller de Análisis de Malware – detectando IOCs

- Sabemos de nuestro análisis que hay dos IPs que están en una lista negra
 - 41.168.5.140
 - 125.19.103.198
- Crearemos una regla de yara para localizar dichas IPs en un fichero
- Posteriormente ejecutaremos yara para ver el resultado





Taller de Análisis de Malware – detectando IOCs

- Yara rule
 - Creamos un fichero con extensión .yar

```
rule detectar_cridex
{
    strings:
        $ip1="41.168.5.140"
        $ip2="125.19.103.198"
    condition:
        any of ($ip1,$ip2)
}
```





Taller de Análisis de Malware – detectando IOCs

- Ejecutamos la regla creada

- yara [regla] [fichero]
- yara yara_cridex.yar cridex.vmem

```
lab@malware:~/taller/memoria$ yara yara_cridex.yar cridex.vmem  
detectar_cridex cridex.vmem
```

- Detecta que alguna de las IPs está en el fichero, por lo tanto ya tenemos una posible regla para detectar cridex en un fichero.





Malware

Análisis online



Hack by Security

Taller de Análisis de Malware



Taller de Análisis de Malware – análisis online

- Para la realización del análisis online podemos utilizar diferentes servicios online:
 - **VirusTotal:** es un servicio gratuito que permite analizar archivos sospechosos con más de 70 motores antivirus y herramientas de análisis de malware. Además, cuenta con una opción de análisis dinámico llamada "Sandbox" que permite ejecutar el archivo en un entorno virtualizado para observar su comportamiento.
 - **Hybrid Analysis:** es un servicio gratuito que ofrece análisis de malware tanto estático como dinámico. Utiliza una técnica de sandboxing en la nube para ejecutar el archivo sospechoso y obtener información sobre su comportamiento.
 - **Joe Sandbox:** es un servicio en línea que ofrece análisis de malware mediante técnicas de sandboxing en la nube. Cuenta con varias opciones de análisis dinámico, incluyendo una versión gratuita limitada y varias versiones de pago con características adicionales.





Taller de Análisis de Malware – análisis online

- Análisis de los ejecutables de cridex,(volatility malfind)
 - Hybrid análisis

Analysis Overview

Submission name: reader_sLexe.207bda0.0x003d0000-0x003f0fff.dmp ⓘ
Size: 132KB
Type: peexe executable ⓘ
Mime: application/x-dosexec
SHA256: cbe5f4af18753839d7e47ee41e6a6c1a1d03e806a77ba7a585ac7b7cad92450 ⓘ
Operating System: Windows ⓘ
Last Anti-Virus Scan: 03/05/2023 13:42:38 (UTC)
Last Sandbox Report: 10/20/2022 14:36:39 (UTC)

Request Report Deletion

malicious

Threat Score: 100/100
AV Detection: 88%
Labeled as: Trojan.Agent

Link Twitter E-Mail

Anti-Virus Results

CrowdStrike Falcon: 100% Static Analysis and ML ⓘ Last Update: 03/05/2023 13:42:38 (UTC)

MetaDefender: 77% Multi Scan Analysis Last Update: 03/05/2023 13:42:38 (UTC)

VirusTotal: 86% Multi Scan Analysis Last Update: 03/05/2023 13:42:38 (UTC)

Refresh Required

⌚ Risk Assessment

Remote Access	Reads terminal service related keys (often RDP related)
Spyware	Found browser information locations related strings
	Hooks API calls
Persistence	Installs hooks/patches the running process
Fingerprint	Reads the windows installation language
Evasive	Input file contains API references not part of its Import Address Table (IAT)



Hack by Security

Taller de Análisis de Malware



Taller de Análisis de Malware – análisis online

- Análisis de los ejecutables de cridex,(volatility malfind)
 - Hybrid análisis

Submission name: process.Ox821dea70.Ox1460000.dmp ⓘ

Size: 132KiB

Type: peexe executable ⓘ

Mime: application/x-dosexec

SHA256: e00a1143fea8568f5bcbe2793c6b87032ba57f2fd122266ea799658169d36b2 ⓘ

Operating System: Windows ⓘ

Last Anti-Virus Scan: 02/16/2023 16:24:27 (UTC)

Last Sandbox Report: 03/02/2023 18:07:37 (UTC)

Threat Score: 100/100
AV Detection: 86%
Labeled as: WORM_CRIDEX.SADU

malicious

Link Twitter E-Mail

Anti-Virus Results

CrowdStrike Falcon
100%
Static Analysis and ML ⓘ
Last Update: 02/16/2023 16:24:27 (UTC)
View Details: ⓘ Visit Vendor: ⓘ

MetaDefender
81%
Multi Scan Analysis
Last Update: 02/16/2023 16:24:27 (UTC)
View Details: ⓘ Visit Vendor: ⓘ

VirusTotal
78%
Multi Scan Analysis
Last Update: 02/16/2023 16:24:27 (UTC)
View Details: ⓘ Visit Vendor: ⓘ

⚠ Refresh Required

Risk Assessment

Remote Access	Reads terminal service related keys (often RDP related)
Spyware	Found browser information locations related strings
Fingerprint	Reads the windows installation language
Evasive	Input file contains API references not part of its Import Address Table (IAT)





Taller de Análisis de Malware – análisis online

- Análisis de los ejecutables de Folio
 - Hybrid análisis

Submission name: muestra.bin [i](#)
Size: 8.1MiB
Type: [peexe](#) [executable](#) [i](#)
Mime: application/x-dosexec
SHA256: [06a39e32a13239675c197a284f06c3b933aa776ae71253791912bcb5164aa1b2](#) [e](#)
Operating System: Windows
Last Anti-Virus Scan: 03/18/2023 16:32:06 (UTC)
Last Sandbox Report: 09/13/2021 10:10:12 (UTC)

malicious

Threat Score: 100/100
AV Detection: 71%
Labeled as: Trojan.Generic

[Link](#) [Twitter](#) [E-Mail](#)

Anti-Virus Results ✓ Up-to-date

Antivirus	Result	Details
CrowdStrike Falcon	100%	Static Analysis and ML i Last Update: 03/18/2023 16:32:06 (UTC) View Details N/A Visit Vendor e
MetaDefender	44%	Multi Scan Analysis Last Update: 03/18/2023 16:32:06 (UTC) View Details e Visit Vendor e
VirusTotal	68%	Multi Scan Analysis Last Update: 03/18/2023 16:32:06 (UTC) View Details e Visit Vendor e

Risk Assessment	
Remote Access	Reads terminal service related keys (often RDP related)
Spyware	Found a string that may be used as part of an injection method
Fingerprint	Queries kernel debugger information Queries sensitive IE security settings Queries the internet cache settings (often used to hide footprints in index.dat or internet cache) Reads the active computer name Reads the cryptographic machine GUID
Spreading	Opens the MountPointManager (often used to detect additional infection locations)
Network Behavior	Contacts 1 domain and 1 host. View all details



Hack by Security

Taller de Análisis de Malware



Qué hemos aprendido

- Hemos aprendido qué es un malware y sus diferentes tipos.
- Los distintos tipos de análisis de malware, estático, dinámico y en memoria.
- Que debemos estar seguros de lo que hacemos.
- A utilizar algunas herramientas de análisis.





Hack by Security

Taller de análisis de malware

info@hackbysecurity.com

www.hackbysecurity.com

academy.hackbysecurity.com



— Hack by Security —